Behavioural Modeling and Linearization of RF Power Amplifier using Artificial Neural Networks

by

Farouk Mkadem

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Farouk Mkadem

Abstract

Power Amplifiers (PAs) are the key building blocks of the emerging wireless radios systems. They dominate the power consumption and sources of distortion, especially when driven with modulated signals. Several approaches have been devised to characterize the nonlinearity of a PA. Among these approaches, dynamic amplitude (AM/AM) and phase (AM/PM) distortion characteristics are widely used to characterize the PA nonlinearity and its effects on the output signal in power, frequency or time domains, when driven with realistic modulated signals. The inherent nonlinear behaviour of PAs generally yield output signals with an unacceptable quality, an undesirable level of out-of-band emission, high Error Vector Magnitudes (EVMs) and low Adjacent Channel Power Ratios (ACPRs), which usually fail to meet the established performance standards.

Traditionally, PAs are forced to operate deeply in their back-off region, far from their power capacity, in order to pass the mandatory spectrum mask (ACPR requirement) and to achieve acceptable EVM. Despite its simplicity, this solution is increasingly discarded, as it leads to cost and power inefficient radios. Alternatively, several linearization techniques, such as feedback, feed-forward and predistortion, have been devised to tackle PA nonlinearity and, consequently, improve the achievable the linearity versus power efficiency trade-off.

Among these linearization techniques, the Digital Pre-Distortion (DPD) technique consists of incorporating an extra nonlinear function before the PA, in order to preprocess the input signal to the PA, so that the overall cascaded systems behave linearly. The overall linearity of the cascaded system (DPD plus PA) relies primarily on the ability of the DPD function to produce nonlinearities that are equal in magnitude and out-of-phase to those generated by the PA. Hence, a good understanding and accurate modeling of PA distortions is a crucial step in the construction of an adequate DPD function.

This thesis explores DPD through techniques based on Artificial Neural Networks (ANNs). The choice of ANN as a modeling tool was motivated by its proven strength in modeling dynamic nonlinear systems. This thesis starts by providing a summary of the PA nonlinearity problem background, as well as an overview of the most well-known linearization techniques, with a special focus on DPD techniques. The thesis then discusses ANN structures and the learning parameters. Finally, a novel Two Hidden Layers ANN (2HLANN) model is suggested to predict the dynamic nonlinear behaviour of wideband PAs. An extensive validation of the 2HLANN model demonstrates its excellent modeling accuracy and linearization capability.

Acknowledgements

I want to express my gratitude to the following people for their help over the period of my Master of Applied Sciences Degree. Their continuous support and concern have made the achievement of this work possible.

First and foremost, I would like to acknowledge my supervisor, Dr. Slim Boumaiza, whose guidance and patience guided my work and assisted me in writing this thesis. He ensured my academic development while emphasizing the importance of collaborative work with fellow students.

Special thanks go to Dr. Fakhri Karray for his continued advice and assistance in all levels of my Thesis.

I would also like to thank all Emerging Radio System Group for their continued involvement and help. They were my friend and my family in Canada, and I enjoyed collaborating with them.

I would also like to express my gratitude to all my professors in University of Waterloo and in the Tunisia Polytechnic School for every bit of inspiration and insight they caused.

Last but not least, I don't find suitable words to express my gratitude, love and respect for those to whom I am offering this work; my family. They were, are and will be always my inspiration and my strength for any accomplishment.

To my family.

A happy family is but an earlier heaven.
John Bowring

Contents

List of Tables

List of Figures

Nomenclature

1dB	1dB Compression Point
2HLANN	Two Hidden Layers Artificial Neural Network
ACLR	Adjacent Channel Leakage Ratio
ACPR	Adjacent Channel Power Ratio
ADPA	Asymmetrical Doherty Power Amplifier
ADS	Advanced Design System
AF	Activation Function
AM/AM	amplitude dependent amplitude distortion
AM/PM	amplitude dependent phase distortion
ANN	Artificial Neural Network
BER	Bit Error Rate
BPLA	Back Propagation Learning Algorithm
CW	Continuous Wave
dB	Decibels
DC	Direct Current
DE	Drain Efficiency
DPD	Digital Pre-Distortion
DUT	Device Under Test

EPA	Error Power Amplifier
EVM	Error Vector Magnitude
FFNN	Feed Forward Neural Network
FIR	Finite Impulse Response
GPIB	General Purpose Interface Bus
GSM	Global System for Mobile Communication
IM3	third order frequencies
IMD	InterModulation Distortion
IMR	InterModulation distortion Ratio
IP2	Second Order Intercept Point
IP3	Third-Order Intercept Point
ISI	InterSymbol Interference
LSE	Least Square Error
LTE	Long Term Evolution
LTM	Long-Term Memory Effects
LUT	Look-Up Table
ME	Memory Effect
MLP	Multi Layers Perceptron
MP	Memory Polynomial
NMSE	Normalized Mean Square Error
NMT	Nordic Mobile Telephone
NPR	Noise Power Ratio
OFDM	Orthogonal Frequency-Division Multiplexing
PA	Power Amplifier

PAE	Power Added Efficiency
PAPR	Peak-to-Average Power Ratio
PEP	Peak Envelop Power
PSD	Power Spectral Density
RF	Radio Frequency
RMS	Root Mean Square
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
RVTDNN	Real Valued Time Delay Neural Network
SDPA	Symmetrical Doherty Power Amplifier
SSG	Small Signal Gain
STM	Short-Term Memory
TDMA	Time Division Multiple Access
VSA	Vector Software Analyzer
W	Watts
WCDMA	Wideband Code Division Multiple Access

Chapter 1

Introduction

State of the Art and Background

In the past decade, mobile and telecommunication services have seen enormous development with a significant shift from simple voice-only communication to multimedia type applications (Internet, video, TV streaming, etc.). In addition, consumer requirements increased the demand for low cost, low consumption and signal integrity equipments. Consequently, wireless design systems have become more challenging demanding wider bandwidths and stringent operational and technical requirements to meet these consumer requirements. This result in a non-primitive requirement for the hardware.

Indeed, the design of these systems has become more sensitive to linear and nonlinear distortions exhibited by the radio chain. Radio Frequency (RF) Power Amplifiers (PAs) remain the key building blocks of these emerging wireless radios systems, because they dominate the overall performance of linearity and efficiency; therefore, significant attention should be focused on these RF front ends. In fact, PAs dominate the radio systems' power consumption and are the main sources of distortions, especially when driven with modulated signals. These distortions are manifested both in-band and out-of-band of the signal bandwidth. Consequently, they engender undesirable degradation of the Bit Error Rate (BER) and spectral regrowth, creating adjacent channel interference and violating the emission requirements mandated by regulatory bodies. Thus, a lot of interest in the literature have been sought for PA linearity and efficiency.

Two areas of improvement in PA linearity and efficiency have been sought. In the first one, advanced power amplification techniques, such as Doherty PAs [26] and envelope-tracking approaches [32], were devised for the enhancement of the power efficiency in the back-off region. In the second area, a number of linearization techniques, such as feed-forward, feedback and Digital Pre-Distortion (DPD) methods, have been suggested to extend the linear range of the PA by reducing the necessary power back-off and meeting the linearity requirements. Despite their development challenges, when properly combined, the two previously mentioned areas of improvement allow for tremendous advances in power efficiency and linearity technologies. In fact, without linearization techniques, the PA would have to be backed off to operate within the linear portion of its operating curve.

In addition, newer transmission formats, such as Wideband Code Division Multiple Access (WCDMA), Long Term Evolution (LTE) and Orthogonal Frequency-Division Multiplexing (OFDM), have large variations in their signal envelopes, i.e. high Peak-to-Average Power Ratio (PAPR). Thus, the PA should be backed off very far from its saturation point, which results in a very low average efficiency, typically less than 10%, and more than 90% of the Direct Current (DC) power turns into heat. Therefore, improving PA efficiency is needed to reduce the cost of electricity consumption and the cost of cooling systems; and, to improve the PA efficiency without compromising its linearity, PA linearization techniques is essential.

Among all the linearization techniques, DPD is one of the most cost-effective techniques, due to its excellent linearization capability and relatively limited extra power consumption. The principle of DPD is based on intentionally introducing a nonlinear function in the base-band part of the transmitter to generate additional distortions that are complementary to those introduced by the PA, thus leading to a linear cascade. With a digital predistorter, the PA can operate up to its saturation point, while maintaining excellent linearity. Nevertheless, successful deployment of a digital predistorter requires the proper choice of the predistortion function and accurate modeling of the PA behaviour.

In literature, nonlinear PA behavioural models can be classified in two categories: memoryless nonlinear models and memory nonlinear capable models. For memoryless nonlinearity, the amplitude and phase distortions of the PA output are typically defined as a function of the amplitude of its current input signal strength. This instantaneous nonlinearity is characterized by the amplitude dependent amplitude distortion (AM/AM) and amplitude dependent phase distortion (AM/PM) responses of the PA.

Different PA model schemes have been suggested in the past to deal with static nonlinearity, namely the Saleh model [59], the memoryless polynomial model [64] and the Look-Up Table (LUT) model [7]. On the other hand, when the signal bandwidth gets wider, such as with WCDMA, LTE or OFDM, the PA begins to exhibit significant Memory Effects (MEs); and, memoryless models and predistorters are no longer sufficient for good modeling or linearization capabilities. MEs are generally attributed to the biasing and matching network frequency-dependent behaviours and require more comprehensive behavioural modeling schemes [68]. Therefore, the current output of the PA is no longer a function of only the current input, but also of past input and output values. In other terms, the PA becomes a dynamic nonlinear system; and, PA models and predistorters should be memory capable.

Volterra series [58, 68, 73, 74] is a comprehensive nonlinear modeling scheme that can account for MEs. Despite its comprehensiveness, the Volterra model has always been criticized for its prohibitive complexity and restricted applicability to high nonlinear PAs [68]. A number of Volterra series derivations and approximations have been developed to reduce the implementation burden, namely the Memory Polynomial (MP) model [31] and the Hammerstein-Wiener model [17, 41, 47, 48, 72]. Alternatively, the excellent capability of Artificial Neural Networks (ANN) to accurately approximate dynamic nonlinear functions [11, 14, 19, 20, 28] has been successfully utilized in the modeling of RF and microwave devices/circuits [1, 5, 9, 12, 23, 40, 45, 46, 49, 51, 62, 63, 70, 71].

Another way to classify PA behavioural models and/or DPD linearizers is the selection of the model structure. One class of these models is empirical, also called black-box models, where the model structure and parameters are done without any a priori knowledge about the device operation. In fact, the identification of the model is totally based on measurement data and no a priori information about the system being measured is used for the structure selection or for the identification. Although this model type demonstrated good performance in the literature, there is no justification about the optimality of the model.

Conversely, physical models are based on circuit-level structure and device operation. These models are powerful, because their structure is influenced by the device itself, which make them very accurate. However, they usually end up as very complicated models that are very difficult to identify [67].

In this thesis, the objective is the combination of the physical knowledge of PA operation with the simplicity of black-box models to develop an optimized model that accurately models the PA response and, consequently, leads to robust and effective linearizers.

Objectives and Outline

The objective of this thesis is the development of a behaviour model to predict the response of RF PAs with MEs and/or construct the corresponding DPD. Unlike previous works, the proposed model structure is developed using a priori knowledge about the PA.

This thesis focuses on three areas:

- Survey of previous techniques used to address the PA nonlinearity problem.
- A comprehensive analysis of ANNs.
- A novel behaviour model with memory, based on ANNs, is discussed and validated.

This thesis is organized as follows:

- Chapter 1 serves as an introduction to the problem tackled in this thesis.
- In Chapter 2, the PA nonlinearity problem is discussed, and essential characteristics are defined.
- Chapter 3 starts with an overview of linearization techniques with a special focus on DPD. Afterwards, various DPD linearizers, namely memoryless models and memory capable models, are examined.
- Chapter 4 is dedicated to a comprehensive study of ANNs, namely structures and learning parameters.
- In Chapter 5, the novel Two Hidden Layers Artificial Neural Network (2HLANN) [45, 46, 49] model is proposed to predict the dynamic nonlinear behaviour of wideband PAs. An extensive validation of the 2HLANN model is conducted to demonstrate its modeling and linearization capabilities.
- Finally, a summary of the work is presented along with a discussion of the major contributions of the proposed model and potential future research directions.

Chapter 2

RF Power Amplifier Characteristics

Power Amplifiers (PAs) are the most challenging building blocks of modern wireless communication radio systems (handset and base stations). They dominate by large the energy consumption of theses radios and consequently dominate their power efficiency. In addition, PAs are also responsible for the main nonlinear effects in the transmitter chain.

The first generation of mobile radio systems, e.g. Nordic Mobile Telephone (NMT), employed continuous envelope modulation that did not impose major linearity requirements. Consequently, highly nonlinear and efficient PAs were utilized. The second generation of mobile radio systems, e.g. Global System for Mobile Communication (GSM), involved Time Division Multiple Access (TDMA) and constant envelope signals, but the change of power level from a time slot to another yielded some constraints on linearity. The third generation of mobile radios systems, e.g. Wideband Code Division Multiple Access (WCDMA), enabled higher data throughputs. However, due to the limited spectrum availability, efforts were made to transmit the maximum amount of data within the smallest bandwidth possible. Thus, sophisticated modulation techniques were introduced, yielding wide bandwidth and dynamic range signals and resulting in stringent requirements on the PA linearity.

It is possible to achieve a highly linear PA; unfortunately, it comes at the cost of the efficiency, which is very undesirable. This solution is widely criticized, because the efficiency of the PA is more critical and has a much greater influence on the overall efficiency of the radio, compared to a second generation radio, as the operation time of the PA is much higher due to continuous transmission. On the other hand, the design of a highly efficient PA results in poor linearity and vice versa. Since inefficiency translates into loss of dollars in utility costs (electricity and cooling), space, racks, heat sinks, fans, etc., designers of PAs have no choice, but to design highly efficient PA, which will be followed later by some linearization techniques to meet linearity requirements.

In the remainder of this chapter, PA efficiency and linearity are explained with figures of merit, in order to serve as an introduction to the next chapter.

2.1 RF Power Amplifier Power Efficiency

PA efficiency can be evaluated using two figure of merits:

- The Drain Efficiency (DE), η, of a PA characterizes the ratio of the output RF power to the DC supplied power, expressed as follows:

$$\eta = \frac{P_{out}}{P_{DC}} \tag{2.1}$$

  where P_{out} and P_{DC} designate the RF power at the output of the PA and the supplied DC power, respectively.

- The Power Added Efficiency (PAE), given in Equation 2.2, takes into account the input power of the PA to provide a more complete figure of merit.

$$PAE = \frac{P_{out} - P_{in}}{P_{DC}} = \eta(1 - \frac{1}{G}) \tag{2.2}$$

  where P_{in} is the RF input power feed to the PA and G is the gain.

Figure 2.1 illustrates the typical DE and output power curves versus the input power of the PA. It is clear from the figure that the efficiency decreases rapidly, as the input power back-off increases. However, at the high input power region, where the efficiency is high, the output power drifts from the ideal linear one, as a consequence of the PA nonlinearity. Thus, improving the efficiency of a PA comes at the expense of its linearity. In the next subsections, the origin and effects of nonlinearity are discussed in more detail.

2.2 RF Power Amplifier Nonlinearity

2.2.1 One Tone Nonlinear Characterization

PA nonlinearity can be characterized under Continuous Wave (CW) stimulus.

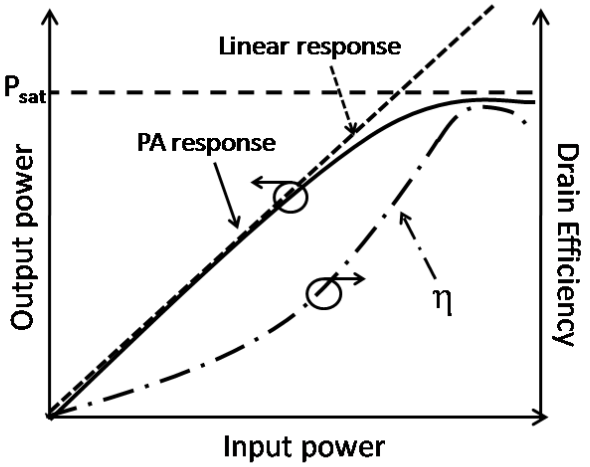

Figure 2.1: Power Amplifier Efficiency versus Linearity

Amplitude Distortion

An ideal PA amplifies the input signal with a constant factor:

$$V_{output}(t) = G_1 V_{input}(t) \tag{2.3}$$

where V_{output}, V_{input}, and G_1 are the output voltage, input voltage and gain of the PA, respectively. The output signal characteristic of the PA is identical to that of the input with the only difference in the scaling of the amplitude.

A simple amplitude distortion can be expressed when inserting a second-order term, G_2, to the transfer function of the PA.

$$V_{output}(t) = G_1 V_{input}(t) + G_2 V^2_{input}(t) \tag{2.4}$$

Figure 2.2 shows the effect of the PA nonlinearity in the time and frequency domains for a CW, where G_1 and G_2 were set to 10 and 3, respectively. According to Figure 2.2, a second-order harmonic appeared as a consequence of the introduction of the second-order term in the PA transfer function.

An inspection of the amplitude of the second harmonic component shows that it increases with respect to the square of the input signal and the constant G_2. However, the amplitude of the fundamental frequency component varies only with respect to the input signal and

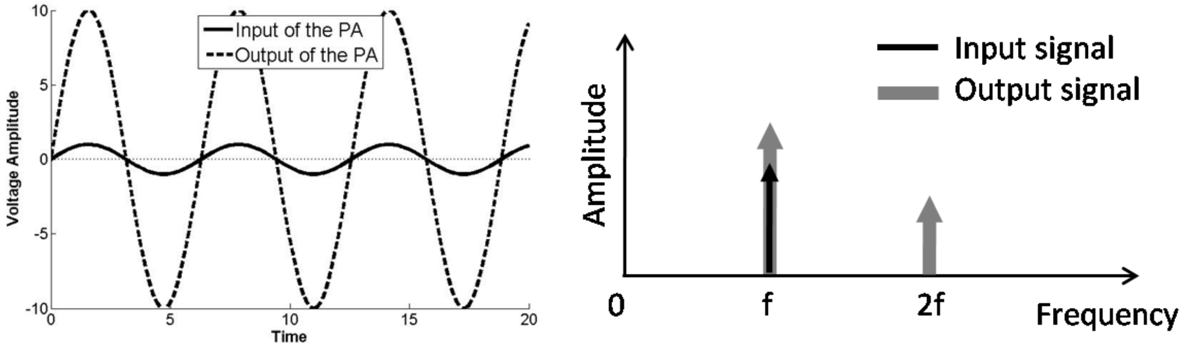

Figure 2.2: Effect on a Sinusoid in the Time Domain and Frequency Domain of a Second-Order Nonlinear Power Amplifier

the constant G_1. Therefore the second harmonic increases at a higher speed than the fundamental frequency.

The desired output, fundamental component and second harmonic become equal in amplitude at a point that is known as Second-Order Intercept Point (IP2) and is illustrated in Figure 2.3. IP2 describes the degree of nonlinearity of a second-order nonlinear PA. Note that the second-order distortion did not affect the signal close to the input signal carrier frequency.

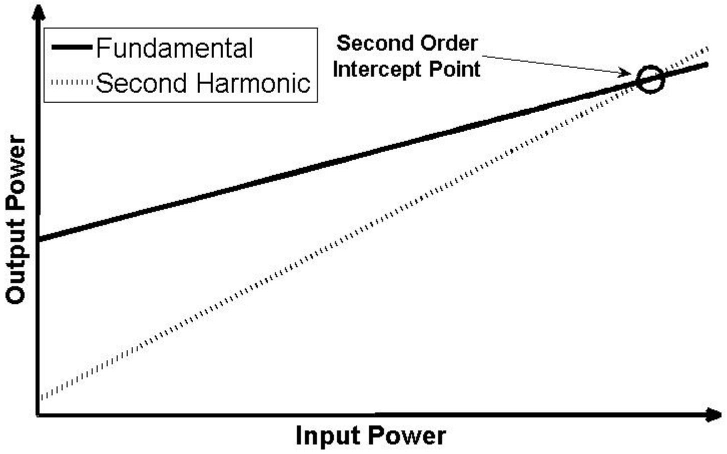

Figure 2.3: Illustration of the Second-Order Intercept Point of a Nonlinear Power Amplifier

Another type of distortion in nonlinear PAs occurs when a third-order term is added to the transfer function of the PA, as illustrated in Equation 2.5:

$$V_{output}(t) = G_1 V_{input}(t) + G_3 V_{input}^3(t) \tag{2.5}$$

Figure 2.4 shows the effect on the time and frequency domain of a CW, where G_1 and G_3 were set to 10 and -3, respectively. Note that here the signal shape is now symmetrical around the horizontal axis, which was not the case in Figure 2.2; however, the output signal peak of the sinusoidal signal is distorted, unlike the second-order distortion.

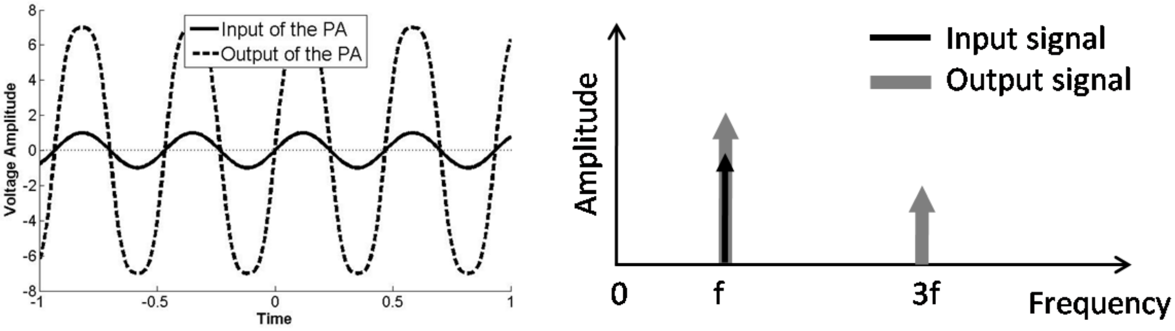

Figure 2.4: Effect on a Sinusoid in the Time Domain and Frequency Domain of a Third-Order Nonlinear Power Amplifier

A more detailed explanation about the signal shape and the third harmonic that can be observe in Figure 2.4 is given by rewriting the transfer function in Equation 2.5 as follows:

$$\begin{aligned} V_{output}(t) &= G_1 V_{input}(t) + G_3 V_{input}^3(t) & (2.6)\\ &= 10A\sin(wt) - 3(A\sin(wt))^3 & (2.7)\\ &= 10A\sin(wt) - \frac{9}{4}A^3\sin(wt) + \frac{3}{4}\sin(3wt) & (2.8)\end{aligned}$$

where the input signal is set to $A\sin(wt)$, A and w represent the amplitude and the frequency of the input signal, respectively.

The first term in Equation 2.8 characterizes the linear amplification of the fundamental frequency, and the third term points out the emergence of the third harmonic in the frequency domain, as observed in Figure 2.4. On the other hand, the distortion in the signal shape of the output signal is explained by the middle term, which occurs at the same frequency as the fundamental frequency. Note that this second term is proportional to the cube of the input signal amplitude and, therefore, can significantly affect the output level at the fundamental frequency, i.e. the desired signal.

Two interesting points can be deduced from the third-order distortion, as illustrated in Figure 2.5:

- the 1dB Compression Point (1dB) refers to the output level where the deviation from the linear response drops by 1dB. 1dB characterizes the degree of nonlinearity of the PA.
- The Third-Order Intercept Point (IP3), which is similar in definition to IP2, refers to the theoretical point where the linear response of the PA and the third order become equal in amplitude. IP3 describes the degree of distortion of the fundamental signal of a third-order nonlinear PA.

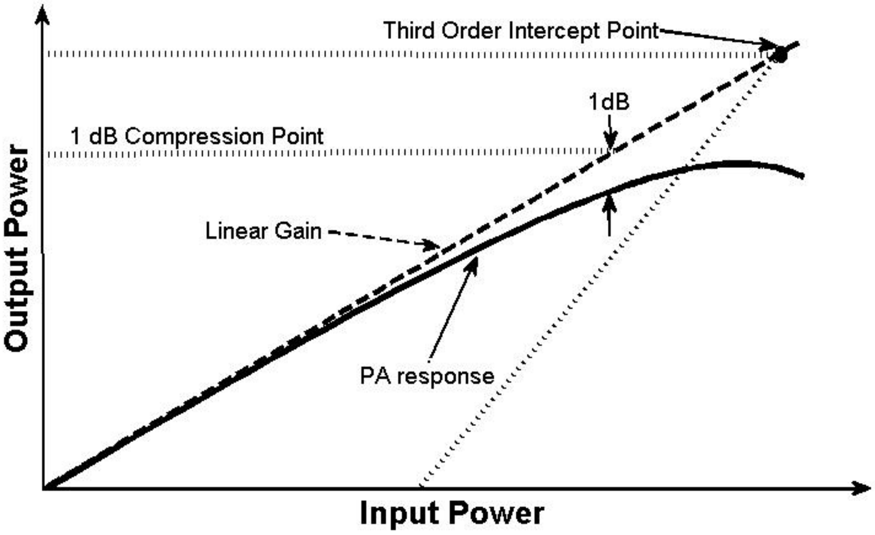

Figure 2.5: Illustration of the Third-Order Intercept Point and the 1dB Compression Point of a Nonlinear Power Amplifier

The amplitude distortion is usually expressed as a change of the output amplitude power level as a function of the input amplitude power level: this is referred to as the AM/AM characteristic:

$$r_{out}(t) = g\left(r_{in}(t)\right) \tag{2.9}$$

where r_{in} and r_{out} are the input and output amplitude signals of the PA, respectively, and $g(.)$ represents a nonlinear function. Figure 2.6 illustrates a typical AM/AM, depending on the input amplitude power of a typical nonlinear PA. The nonlinear PA exhibits linear output amplitude variation at back-off power; however, at high input power, the output amplitude component drifts from the linear response and results in a nonlinear amplitude change.

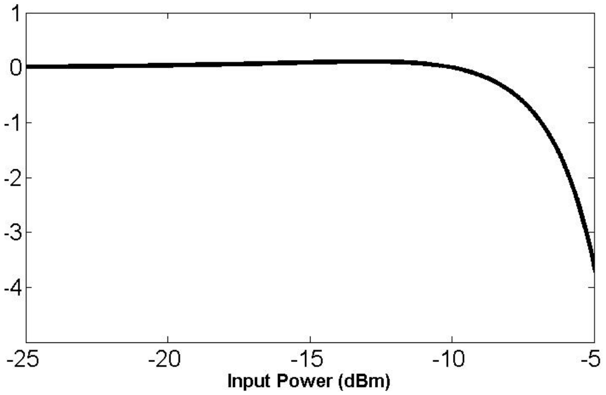

Figure 2.6: AM/AM Distortion in a Nonlinear Power Amplifier

Phase Distortion

The distortion of the output amplitude in a nonlinear PA, i.e. AM/AM, is usually accompanied by phase distortion, i.e. AM/PM. The AM/PM corresponds to a non-constant phase shift of the RF signal at the output of the PA when the envelope signal at the input of the PA changes. In other words, AM/PM is expressed as a change of the phase component of the output signal according to the envelope of the input signal, as expressed in Equation 2.10

$$\phi_{out} = \phi_{in} + f\left(r(t)\right) \tag{2.10}$$

where ϕ_{in}, ϕ_{out} and $r(t)$ are the input and output phases and the input amplitude signals of the PA, respectively, and $f(.)$ represents a nonlinear function.

Figure 2.7 illustrates a typical AM/PM, depending on the input amplitude power of a typical nonlinear PA. Similar to the case of the AM/AM, the PA exhibits constant phase variation at back-off power; however, at high input power, the phase component drifts from the linear response and results in a nonlinear phase change.

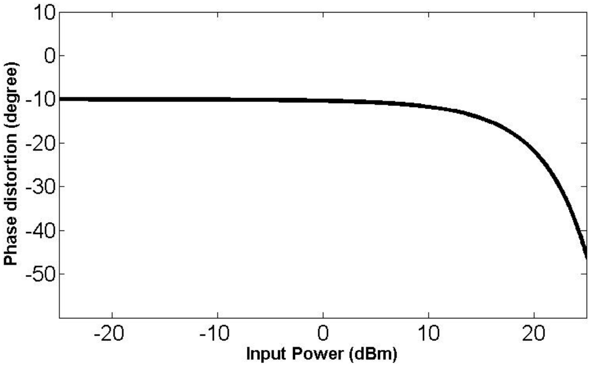

Figure 2.7: AM/PM Distortion in a nonlinear Power Amplifier

2.2.2 Two-Tone Test

In the previous subsection, the nonlinear characterization of the PA was conducted using a CW. A more sophisticated characterization can be done using a two-tone test. The two-tone test provides a good illustration of the amplitude and phase distortion discussed above. The test consists of a simulation of a nonlinear PA excited with two sinusoids at different frequencies, f_1 and f_2. Note that the two-tone test assumes that the frequency separation $|f_2 - f_1|$ is negligible compared to f_1 and f_2. The two-tone test representation allows for a good understanding of the in-band, out-of-band, intermodulation, amplitude and phase distortions; however, it is not sufficient for a complete modeling and explanation of the source and effects of the nonlinear behaviour of modulated signals.

An example of a two-tone test is the representation of PA by a third-degree polynomial, as shown in Equation 2.11.

$$y = a_1 x + a_2 x^2 + a_3 x^3 \tag{2.11}$$

where x and y represent the input and output of the PA, respectively; a_1 describes the linear Small Signal Gain (SSG); and, a_2 and a_3 are the quadratic and cubic nonlinearities coefficients, respectively. For the purpose of simplicity, the coefficients in Equation 2.11 are considered real valued; and, the PA is considered memoryless with no phase distortion. The more general case of complex coefficients is discussed when introducing the memory polynomial model in the next chapter.

Let consider that the two tones used for the test are:

$$v_1(t) = A\sin(w_1 t) \tag{2.12}$$

$$v_2(t) = A\sin(w_2 t) \tag{2.13}$$

where A is the amplitude of the two tones. The resulting output of the PA represented by the transfer function in Equation 2.11 is as follows:

$$\begin{aligned}
y(t) &= a_1\Big(A\sin(w_1 t) + A\sin(w_2 t)\Big) \\
&\quad +a_2\Big(A\sin(w_1 t) + A\sin(w_2 t)\Big)^2 \\
&\quad +a_3\Big(A\sin(w_1 t) + A\sin(w_2 t)\Big)^3 \qquad (2.14)\\
\Rightarrow y(t) &= a_2A^2\bigg(1+\cos\Big((w_1-w_2)\,t\Big)\bigg) \\
&\quad +a_1A\Big(\sin(w_1 t)+\sin(w_2 t)\Big) \\
&\quad +\frac{9}{4}a_3A^3\Big(\sin(w_1 t)+\sin(w_2 t)\Big) \\
&\quad +\frac{3}{4}a_3A^3\bigg(\sin\Big((2w_1-w_2)\,t\Big)+\sin\Big((2w_2-w_1)\,t\Big)\bigg) \\
&\quad -\frac{1}{2}a_2A^2\Big(\cos(2w_1 t)-\cos(2w_2 t)\Big) \\
&\quad -a_2A^2\cos\Big((w_1+w_2)\,t\Big) \\
&\quad -\frac{1}{4}a_3A^3\Big(\sin(3w_1)-\sin(3w_2)\Big) \\
&\quad -\frac{3}{4}a_3A^3\bigg(\sin\Big((2w_1+w_2)\,t\Big)-\sin\Big((2w_1+w_2)\,t\Big)\bigg) \qquad (2.15)
\end{aligned}$$

From Equation 2.15 and as illustrated in the frequency domain in Figure 2.8, one can conclude that the first linear term in Equation 2.11, i.e. a_1x, amplifies the fundamental tones. On the other hand, the quadratic term, a_2x^2, downconverts the RF signal to the DC band and generates the second harmonic band. However, the cubic term, a_3x^3, generates intermodulation, known as third-order frequency terms (IM3) around the fundamental frequency and is also responsible for the third harmonic band. Note that the second and third harmonic bands can be easily filtered out by a passband filter around the fundamental frequency; therefore, their significance is minor.

On the other hand, when examining the second- and third-order distortions, it comes to mind that, if only the quadratic term generates significant signals around the carrier frequency, and the cubic term does not, this fact can be generalized for all odd and even terms [44]. This is true, because the PA is assumed to have a narrow fractional bandwidth compared to the carrier frequency; therefore, all even-order term are harmonics and, consequently, do not interfere with the fundamental frequency. In fact, when applying a two-tone test to a nonlinear PA, a number of harmonics and intermodulation products are generated; and, these are all given by the general expression of Equation 2.16 [68]:

$$f_{im} = mf_1 \pm nf_2 \tag{2.16}$$

where f_{im} are the new generated frequencies, m and n are positive integers (including zero) and $m + n$ is equal to the order of the PA nonlinearity.

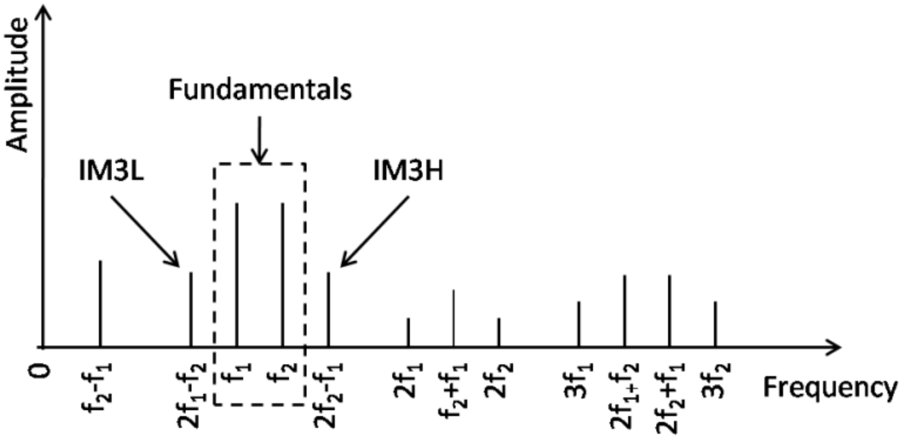

Figure 2.8: Spectral Regrowth of a Two-Tone Signal

For example, in the case of third-order nonlinearity, the new generated frequencies are:

$$\begin{array}{llllllllllll} f_{im1} & = & 0 & f_{im2} & = & f_2 - f_1 & & & & & & \\ f_{im3} & = & 2w_1 - w_2 & f_{im4} & = & 2w_2 - w_1 & & & & & & \\ f_{im5} & = & 2w_1 & f_{im6} & = & 2w_2 & f_{im7} & = & w_1 + w_2 & & & \\ f_{im8} & = & 3w_1 & f_{im9} & = & 3w_2 & f_{im10} & = & 2w_1 + w_2 & f_{im11} & = & w_1 + 2w_2 \end{array} \tag{2.17}$$

InterModulation Distortion (IMD) products are generated by the nonlinearity of the PA through a mixture of the input signal frequency components. These intermodulations are very critical, unlike harmonics, because they cannot be easily filtered out. For example, for a nonlinear PA with a transfer function as in Equation 2.18 with a two-tone input as in Equations 2.12 and 2.13, results in different IMDs around the carrier frequency, as illustrated in Figure 2.9. The IMD products are usually expressed as InterModulation distortion Ratio (IMR) which is equal to the power ratio of these mixing products to that of the fundamental tones.

$$y = a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 \tag{2.18}$$

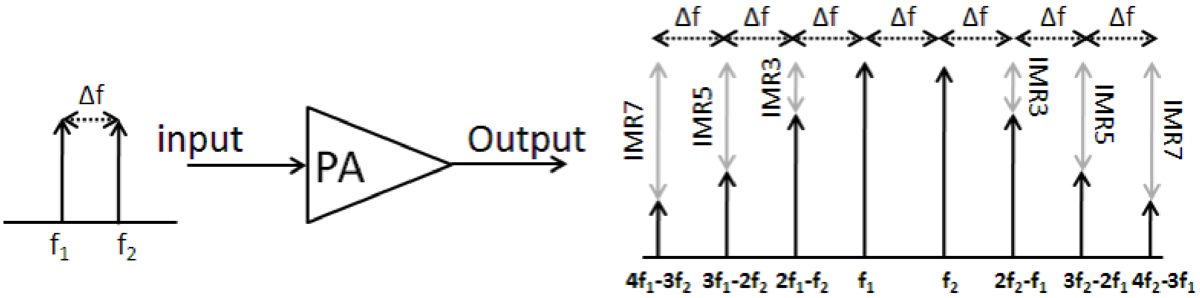

Figure 2.9: InterModulation Distortion for a Nonlinear Power Amplifier

2.3 RF Power Amplifier Nonlinearity Characterizations using Modulated Signals

For simple nonlinearity, the one- and two-tone tests are good and simple tools to characterize PA nonlinearity. However, under modulated signals, the PA does not present simple nonlinearity, and other criteria are needed to determine the degree of the unwanted signal engendered by the nonlinear PA. Two established measures exist for this purpose, namely the Adjacent Channel Power Ratio (ACPR) and the Error Vector Magnitude (EVM) measures.

2.3.1 Adjacent Channel Power Ratio

The ACPR, which is also called the Adjacent Channel Leakage Ratio (ACLR), is the ratio between the total power of the neighbouring channels to the signal power in the main channel. The ACPR describes the degree of the signal regrowth into adjacent channels. The ACPR is a major criterion for the standards of regulatory bodies, because it characterizes the radio system interference with other adjacent radios. The ACPR can be measured as the power contained in a defined bandwidth, B_1, at a defined offset, f_0, from the centre frequency, f_c, of the main channel by the power in a defined bandwidth, B_2, around f_c. This concept is illustrated in Figure 2.10 and Equation 2.19.

$$ACPR = \frac{P_{B_1}}{P_{B_2}} \tag{2.19}$$

where P_{B_1} is the power contained in the adjacent bandwidth, B_1, and P_{B_2} is the power contained in the main bandwidth, B_2.

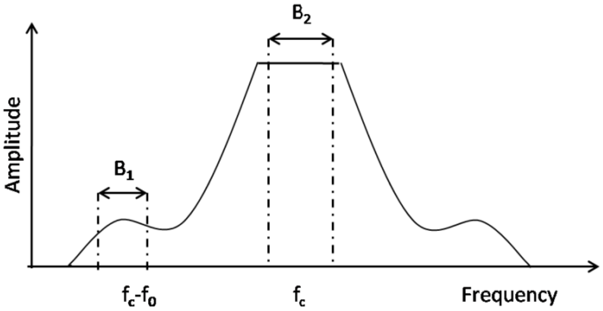

Figure 2.10: Adjacent Channel Power Ratio

The spectral regrowth happens as a result of the presence of intermodulation nd leads to the generation of an out-of-band signal, which causes harsh adjacent channel interference. The spectrum regrowth is shown in Figure 2.11.

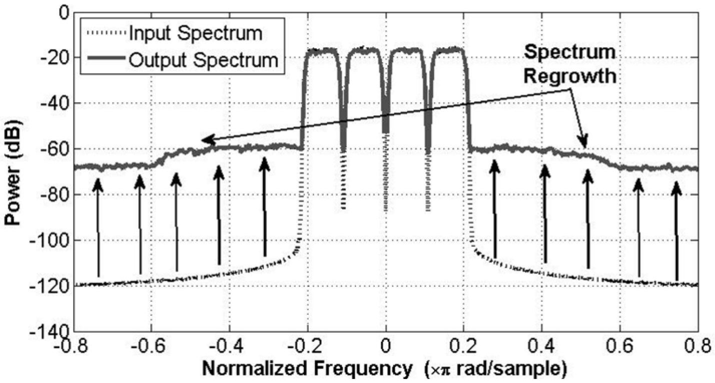

Figure 2.11: Spectrum Regrowth

2.3.2 Error Vector Magnitude

The EVM is a measure that quantifies the performance of both the transmitter and the receiver. The EVM characterizes the imperfection of the constellation, which is distorted due to the nonlinearity in the PA. The EVM measures the quality of the output signal and its effects in the in-band signal. The EVM is illustrated in Figure 2.12, where v is the ideal symbol vector, w is the measured symbol vector, e is the magnitude error, θ is the phase error, and $w - v$ is the EVM.

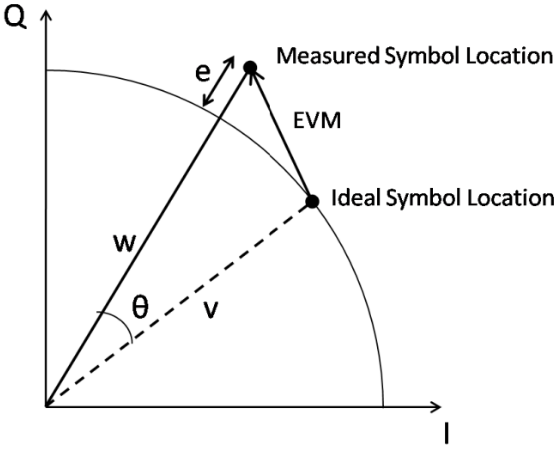

Figure 2.12: Error Vector Magnitude

For a sequence of length N, the EVM is defined in Equation 2.20. The EVM is normalized here by $|v_{max}|$ to eliminate the dependency on the symbol being transmitted.

$$EVM = \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left[(I_j - \tilde{I}_j)^2 + (Q_j - \tilde{Q}_j)^2\right]}}{|v_{max}|} \tag{2.20}$$

where I_j and Q_j are the in-phase and quadrature components of the desired signal and $\tilde{I}_j$ and $\tilde{Q}_j$ are their transmitted counterparts.

The EVM can also be defined as a percentage, as in Equation 2.21, where P_{error} is the Root Mean Square (RMS) power of the error vector and $P_{reference}$ is the RMS power of ideal transmitted signal.

$$EVM(\%) = 100\%\sqrt{\frac{P_{error}}{P_{reference}}} \tag{2.21}$$

The EVM can also be defined in decibels (dB) as in Equation 2.22.

$$EVM(dB) = 10\log_{10}\left(\frac{P_{error}}{P_{reference}}\right) \tag{2.22}$$

The EVM characterizes the in-band distortion and causes a deformation of the constellation, which leads to distortion of the signal being transmitted. This constellation deformation consists of an InterSymbol Interference (ISI) and a warping effect. The ISI is responsible for spreading the constellation point in small clusters, a phenomenon known as a clustering effect. On the other hand, the warping effect results from the memoryless nonlinear behaviour of the PA and manifests in the form of a drift of the constellation from the original lattice arrangement. The clustering and warping effects are illustrated in Figure 2.13.

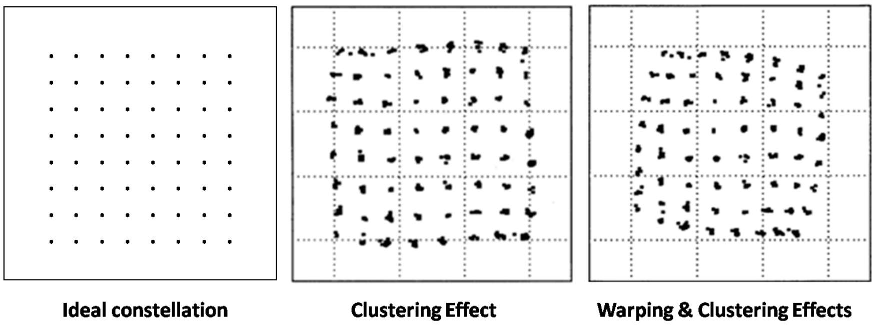

Figure 2.13: Warping and Clustering Effects

An alternative to the EVM measure for the characterization of the in-band distortion is the Noise Power Ratio (NPR). The NPR can be deduced by eliminating a portion of the input signal, using a notch filter, and examining the level of distortion, i.e. filling in the space, within the resulting gap due to the nonlinear behaviour of the PA. Thus, the NPR is defined as the ratio between the noise Power Spectral Density (PSD) when the filter is applied, to the noise PSD without the notch filter, when the PA is driven at the same power level.

2.4 Memory Effects in the Power Amplifier

In the previous section, only distortions caused by changes of the current input amplitude were discussed, i.e. static distortions. However, as the bandwidth of the input signal gets wider, new sources of distortion happen, which are known as Memory Effects (MEs). These MEs happen when the amplitude and phase of the output distorted components vary as functions of modulation frequency, e.g. the tone spacing in a two-tone signal. Unlike static nonlinearity, MEs do not generate new spectral components, but rather change the shapes of the existing signal components. Indeed, this can be explained by the fact that the output of the PA with MEs is no longer a function of only the current input values, but also a function of the previous input and output values. MEs are classified in two categories: Short-Term Memory (STM) and Long-Term Memory (LTM) effects. Since the PA is considered with a quasi-static response, the LTM effects are negligible compared to STM; and, only STM effects are discussed in the following paragraphs.

For a typical nonlinear PA with MEs, its transistor can be approximated as a quasi-memoryless nonlinear device, assuming that the signal bandwidth (few MHz) is negligible compared to the carrier frequency (few GHz). On the other hand, the PA output matching and biasing networks are subject to the transistor distorted output, as shown in Figure 2.14. These previous networks are designed to operate at a specific range of frequencies and are characterized by constant reflection coefficients over that range. However, since the transistor generates IMD due to static nonlinearity the reflection coefficients of the biasing and the matching networks can no longer be considered constant, especially when the bandwidth is getting wider. Therefore, these reflection coefficients depend on the range of frequencies seen at the output of the transistor, which include the signal bandwidth and its corresponding IMD.

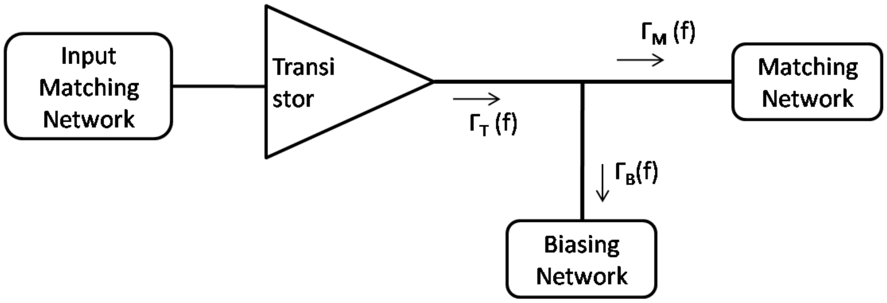

Figure 2.14: Physical Dominant Sources of Short-Term Memory Effects in a Power Amplifier

When the PA is excited with a modulated signal, the static transistor generates certain IMD, as shown earlier for the case of the two-tone test in Figure 2.9. These IMDs are seen at the input of the biasing and matching networks and the non-ideal reflection, initiating

STM effects. The matching network reflects several frequency components present at its input back into the transistor, due to the non-constant reflection over a large range of frequencies. These reflected components are recombined with the next input signal and also excite the static nonlinear transistor. The biasing network also contributes to this input mixture with previous transistor output signals by reflecting the lower range of frequencies. These reflected components remix with the carrier band frequencies and fall within this carrier band.

When adding the reflection effects from both the matching and biasing networks, the resulting reflected components generate new IMDs around the carrier. However, the new IMDs occur at the same IMDs of those caused by static nonlinearity, since no new frequency components are added, but rather remixed the already existing frequencies. Therefore, MEs do not introduce new components, but alter the already existing ones, as illustrated in Figure 2.15.

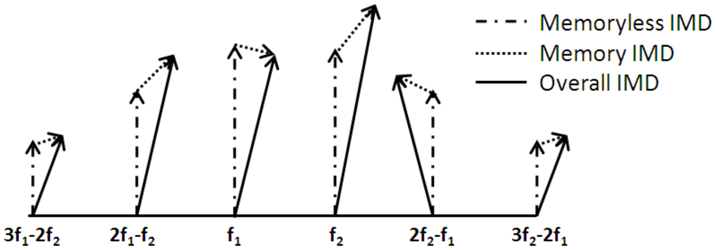

Figure 2.15: InterModulation Distortion for a Nonlinear Power Amplifier with Memory

2.5 Linear and Nonlinear Power Amplifiers

2.5.1 Power Amplifiers Mode of Operations

A linear PA is a continuously driven linear device. Classes A, AB, and B are the three main classes of linear PAs. This category of PAs achieves low efficiency, due to the large dissipation of the DC source power resulting from the continuous operation of the PA, even under unexcited input power. Figure 2.16 illustrates efficiency for classes A, AB and B. Class A achieves a peak efficiency of 50%, while class B attains a maximum of 78%. For the two latter classes, the efficiency falls rapidly, as the back-off increases. Class B remains linear, while improving its efficiency, compared to class A, but inherently has lower gain compared to class A. Class AB is a class between class A and class B and is achieved by saturating class A to attain the maximum efficiency or by reducing the bias of class B to improve its gain.

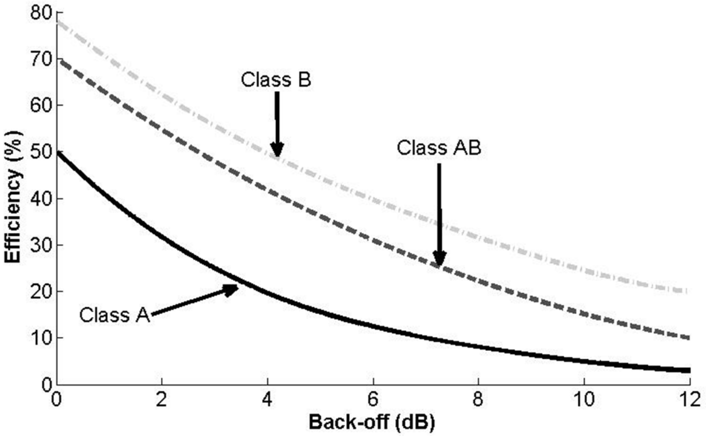

Figure 2.16: Classes A, AB and B Power Amplifier Efficiency Performance

2.5.2 Nonlinear Power Amplifiers

Nonlinear PAs are devices with high efficiency and high gain. The main nonlinear classes are C, D, E, F, G and S. Classes D and E are very nonlinear PAs and are used only when the distortion requirement in the radio system is flexible, or when the input signal has a constant envelope. Classes E, D and S are highly efficient switching PA, where the switching time, i.e. operation time of the PA, becomes comparable to the duty time, i.e. when necessary for transmission. Classes C, F and G are voltage controlled source PAs. Nonlinear PAs achieve higher efficiency compared to linear PAs; however, they are inherently nonlinear. Hence, linearization schemes are required to extend the linear region of the PA, allowing for a more efficient and linear PA.

2.5.3 Doherty Power Amplifier

The Doherty PA consists of a two PAs, as shown in Figure 2.17, with a class AB main stage and a class C auxiliary stage, also called a peaking stage. The input signal in the Doherty PA is split between the two PAs and combined at the output with a correction for the phase differences between the two PAs. For low input power, only the main PA operates and amplifies the signal efficiently, while the auxiliary PA remains 'OFF' and consumes no power.

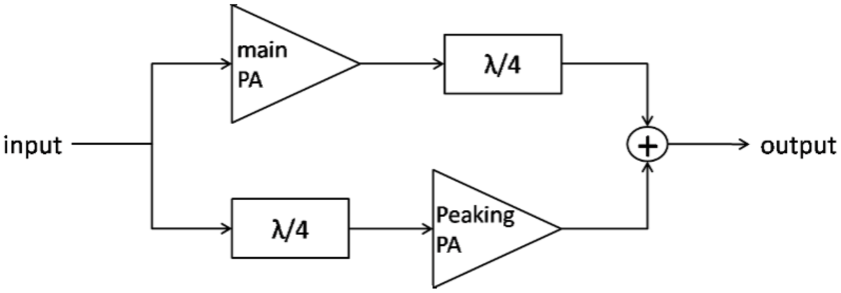

Figure 2.17: Doherty Power Amplifier Diagram

When the main PA saturates at a predetermined input power threshold, the auxiliary PA kicks in and compensates the saturation of the main PA, thus improving the overall linearity, as shown in Figure 2.18. Figure 2.19 shows the typical improvement in efficiency that the Doherty PA brings compared to a regular class AB PA.

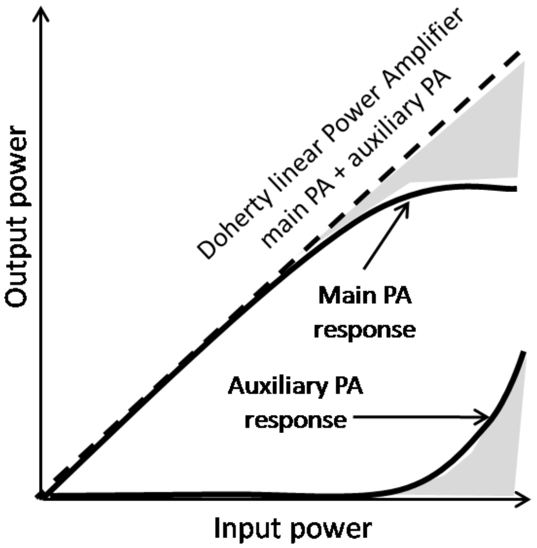

Figure 2.18: Doherty Power Amplifier Basic Operation Characteristic

2.6 Conclusion

Linearity and efficiency are the two major criteria for PAs. Many techniques have been tried to enhance the linearity versus efficiency trade-off. Improving the linearity of the PA by only changing its mode of operation comes at the expense of its efficiency. On the other

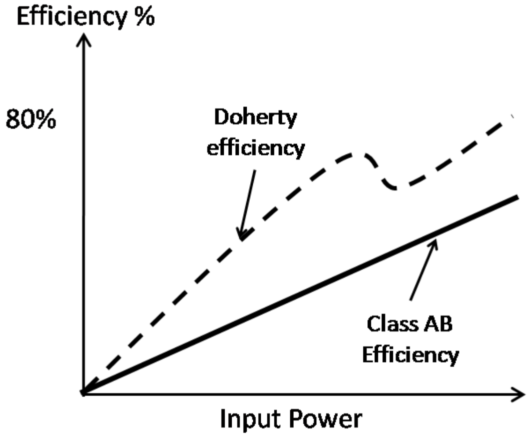

Figure 2.19: Improved Efficiency of the Doherty Power Amplifier over Class AB

hand, improving the efficiency at the back-off power, as in the case of the Doherty PA, is still not sufficient for accurate transmission for wideband and high varying envelope signals (large PAPR), because of the large back-off needed. Therefore, a number of linearization techniques have been suggested in the literature to extend the linear range of the PA, thereby reducing the necessary power back-off to meet the linearity requirement. The next chapter introduces and discusses the main linearization techniques with a focus on digital predistortion schemes.

Chapter 3

Behavioural Modeling and Linearization of RF Power Amplifier

3.1 Linearization Technique

As discussed in the last chapter, efficiency and linearity are two critical design factors that cannot be satisfied simultaneously. Indeed, poor efficiency, i.e. when a large back-off is required in order to be in the linear region, significantly escalates the power dissipation. Consequently, PA designers aim first to achieve peak efficiency with less constraints on linearity. Subsequently, a linearization technique is applied to enhance the linearity in the efficient region. Several linearization techniques are reported in the literature that deal with the nonlinearity in PAs, namely feedback [2, 13, 25, 55, 69], feed-forward [27, 29, 33, 36, 61], and DPD [24, 53] linearizers. In the next subsections, a detailed discussion is dedicated to these techniques with a focus on DPD, since it is the subject of this thesis.

3.1.1 Feedback Linearization

The feedback approach is a method based on algebraic transformation of nonlinear systems into linear ones [52], so that linear control techniques can be applied. Feedback is achieved by selecting different state representations of the system, in order to end up with an equivalent linear system. The feedback linearization principle consists of compensation for the output signal of the nonlinear system by the implementation of a feedback loop, so that the current output signal is used to adjust future inputs. This is achieved by comparing the output signal to a desired output, then deducing an error signal that will be substituted from the next input signal. The block diagram of a typical feedback linearizer is shown in Figure 3.1, and its transfer function is given in Equation 3.1:

$$y\left(t\right)=\frac{G}{1+GA}x\left(t\right) \tag{3.1}$$

where $x(t)$ and $y(t)$ represent the input and the output of the feedback system, respectively, and G and A are the transfer function of the PA and the feedback loop, respectively.

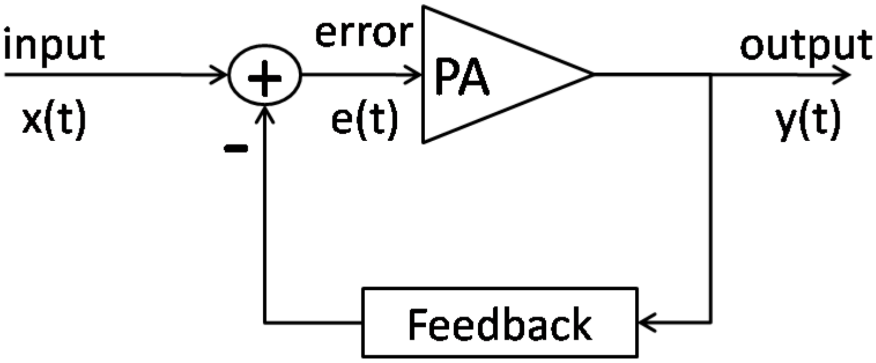

Figure 3.1: Block Diagram of Feedback Linearization

Feedback linearization, although reasonably simple to implement, suffers from many drawbacks when applied to RF PA systems. The major problem with the feedback linearizer gains importance when it is applied to high-frequency and wideband systems. Indeed, the feedback linearizer reduces the gain of the linear PA and suffers from serious potential instability. In fact, the penalty for better linearity when using the feedback linearizer is that the gain of the feedback system is reduced by the factor $\frac{1}{1+AG}$, as shown in Equation 3.1. On the other hand, when applying wideband signals, it is generally difficult to preserve the stability of the feedback system. In fact, the stability of the system is determined by the manner in which the feedback loop gain, AG, varies with frequency, and this is difficult to control over a wide bandwidth signal.

3.1.2 Feed-Forward Linearization

Figure 3.2 illustrates the feed-forward linearizer. The principle of feed-forward linearization is based on the generation of an error signal, $e(t)$, that is intended to represent the additive distortion products that are present at an attenuated version of PA output signal, $y_{PA_1}(t)$. This error, after being scaled up to its original magnitude using an Error Power Amplifier (EPA), is then subtracted from the PA output, $y_{PA_2}(t)$. The theoretical result is a linear PA, and there is no bandwidth limitation that is directly attributed to the linearization effort. Note that, because of the phase shifts introduced in the carrier by the PA and EPA, two delay blocks, PA Equal Delay and EPA Equal Delay, are introduced to compensate for the propagation delays in the two previous mentioned PAs.

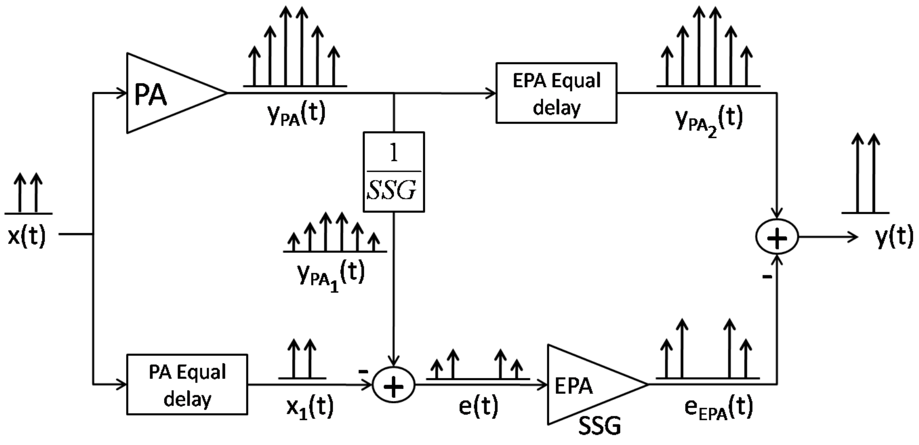

Figure 3.2: Feed-Forward Diagram

As always, things are not so perfect in practice. In fact, the two delay blocks must be very tightly tuned, in order to realize good performance. In addition, the gain of the EPA must be very well matched to the gain of the PA, that good cancellation of the distortion can be realized. Moreover, the EPA should be also linear to amplify the error signal, $e(t)$, with no additional distortions. Finally, the analogue delays and the EPA in the output path of the PA must be extremely low-loss, in order for the whole linearization effort to be worthwhile in terms of efficiency.

3.1.3 Digital Predistortion

The predistortion principle is based on intentionally introducing a nonlinear function in the base-band part of the transmitter, commonly called digital predistorter, to generate additional distortions that are complementary to those introduced by the PA, thereby leading to a linear cascade. Note that it is also possible to introduce a postdistortion in such a way that the complementary distortion element is placed after the PA. However, postdistortion is undesirable, because it leads to processing the signal at a high power level, which results on a high power consumption and decreases the overall efficiency.

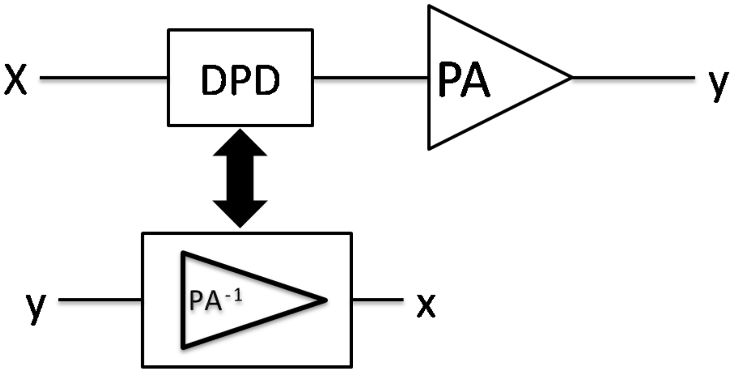

Figure 3.3: Digital Predistortion Followed by Power Amplifier

The linearity of the cascaded system (DPD plus PA) relies primarily on the ability of the DPD function to produce nonlinearities that are complementary in magnitude and out of phase to those generated by the PA. Hence, successful deployment of the digital predistorter requires proper choice of the predistortion function and accurate modeling of the PA behaviour to accurately predict and reproduce the PA distortions.

The DPD corrects the power level of a primary input signal, so that the output of the corrected amplified input is a simple amplification of that primary input. The DPD concept is shown in Figure 3.4, where the PA transfer function, the DPD transfer function, and an ideal PA transfer function are illustrated. Note here that the gain of the PA is set to 1 for the simplicity of illustration; however, results remain true for any given gain. The DPD principle works as follows: to achieve a power level at the output of the PA,V_{out_A}, an adjustment of the input signal amplitude from V_{in_A} to V_{in_B} is required. This correction can be deduced by projecting the input V_{in_A} in the DPD transfer function, which provides the V_{in_B} level. The DPD is, therefore, responsible for the transformation from level V_{in_A} to level V_{in_B}, in order to achieve the desired output level, V_{out_A}. The DPD transformation can be constructed by simply reversing the PA transfer function. Consequently, the resulting DPD plus PA cascade results in a linear amplification of the input V_{in_A} to the desired output V_{out_A}. The DPD plus PA cascade is illustrated in Figure 3.5.

Predistortion main attribute is the simplicity of its principle. In addition, due to the open loop-scheme, the DPD is unconditionally stable and does not suffer from a bandwidth limitation as the feedback linearizer does. (Note, however, that baseband predistortion is bandwidth limited, because of the limitation of the analogue-to-digital converter in the bandwidth). Moreover, due to the low power consumption of the DPD relative to the PA, there is no significant degradation in the transmitters? power efficiency, as in the

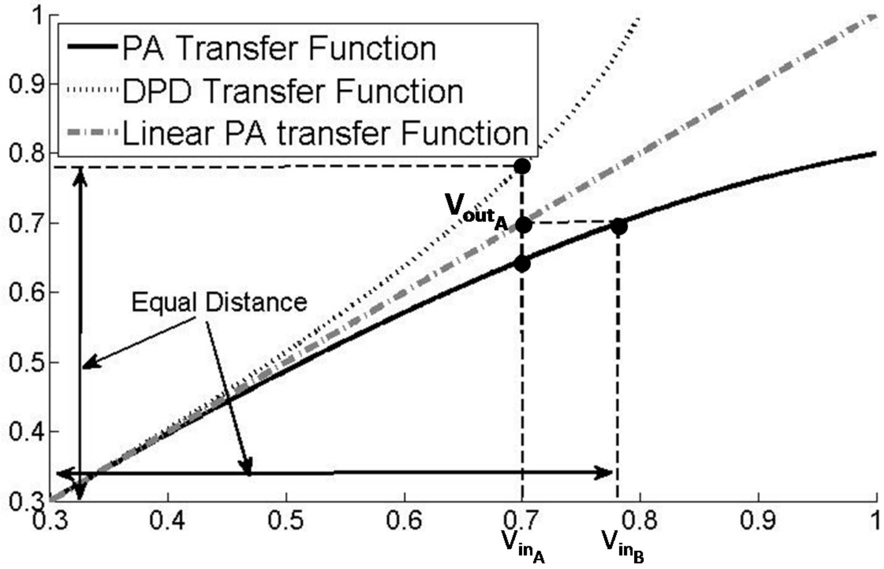

Figure 3.4: Digital Predistortion Principle 1

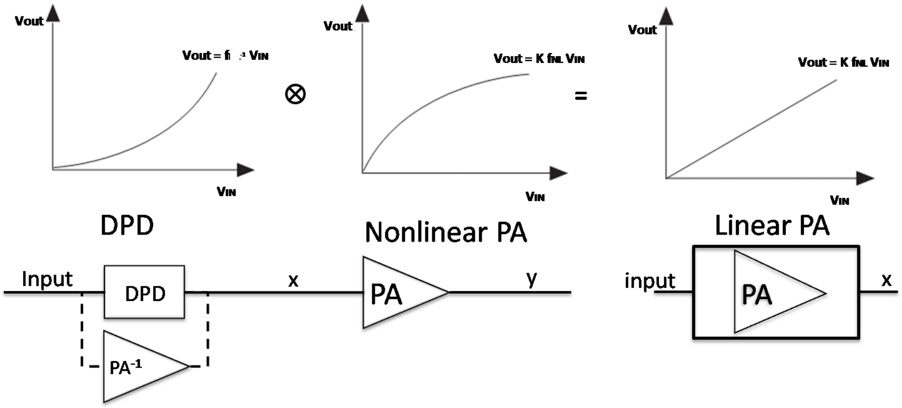

Figure 3.5: Digital Predistortion Principle 2

feed-forward linearizer. On the other hand, disadvantages of predistortion include the fact that predistorters are normally optimized for specific PA conditions (output power level, signal type, bandwidth, etc.); and, as these conditions change, a fixed predistorter tends to become misaligned with the nonlinearity of the PA. Another drawback to the DPD is its dependence on a good model of the PA. In fact, PAs are difficult to model, and DPDs are based into this tricky issue.

Performance Comparison of the Three Main Linearization Techniques

Table 3.1 presents a comparison of the typical performances of the three linearization schemes previously discussed [42]. The feedback linearizer is narrowband and, therefore, is not suitable for most multi-carrier applications. On the other hand, feed-forward linearizer is usually inefficient compared to other linearizers because of the use of EPA that reduces the overall efficiency. Alternatively, DPD shows good achievable efficiency thanks to the limited extra power consumption along with its applicability to large bandwidth signal.

Table 3.1: Comparison of the Three Linearization Techniques

Linearization Method	**Cancellation Performance**	**Bandwidth**	**Achievable Efficiency**	**Complexity**
Feed-Back	Good	Narrow	Medium	Medium
Feed-Forward	Good	Widest	Low	Large
Predistortion	Moderate	Wide	High	Small

Thus, DPD was selected from among the linearization techniques to be the focus of this thesis, since it offers the highest efficiency along with low complexity and great flexibility, due to the use of advanced digital signal processing techniques.

Power Amplifier Digital Predistortion and Behavioural Modeling

As explained previously, the DPD principle relies on producing the inverse function of the PA. To synthesize this function, a good understanding and accurate modeling of PA distortion is essential, in order to generate the predistorter function. In fact, the DPD construction is achieved through two steps:

1. Model the PA behaviour to identify the nonlinearity characteristics of the PA.

2. Construct the DPD function by modeling the inverse of the PA.

The problem of modeling the PA behaviour is not a trivial one, and different PA model schemes have been suggested in the past. In the literature, nonlinear PA behavioural models can be classified in two categories: memoryless models and memory capable models.

With a memoryless model, the current output depends only on the current inputs through a nonlinear mechanism. This instantaneous nonlinearity is refereed to as static nonlinearity. The Saleh [59], memoryless polynomial [64] and the LUT [7] models were the first models that aimed to fit this static nonlinearity of the PA.

As the signal bandwidth gets wider, however, such as with WCDMA signals, the PA begins to exhibit MEs. Accordingly, the output of the PA no longer depends only on the current input, but also on past input and output values. In other words, the PA becomes a nonlinear system with memory. For such a PA, memoryless predistortion can achieve only a very limited linearization performance. Therefore, DPD also needs to capture these MEs, by introducing structures capable of modeling them. These models are refereed as memory capable models.

The Volterra series [58, 68, 73], due to its richness, is considered as a reference for modeling dynamic nonlinear systems. Nevertheless, the Volterra series has always been criticized for its prohibitive complexity and restricted applicability to high nonlinear PAs. The MP model [31] is currently the most popular derivation of the Volterra series that exclude cross terms to alleviate complexity. However, the identification of the MP model parameters using Least Square Error (LSE) algorithm is computationally demanding. Additionally, two-box models [17, 22, 41, 47, 72], known as Wiener and Hammerstein models, employ a cascade of a static nonlinear function and a linear filter to model dynamic nonlinear systems. Alternatively, the excellent capability of ANNs to accurately approximate continuous functions [11, 14, 20] has been successfully exploited to model RF and microwave devices/circuits [1, 5, 9, 12, 23, 40, 45, 46, 51, 62, 63, 70, 71].

In the next sections, detailed descriptions of the main behavioural models are presented.

3.2 Memoryless Power Amplifier Behavioural Modeling and Digital Predistortion

3.2.1 Saleh Model

The Saleh model [59] consists of two parametric frequency-independent functions to fit the AM/AM and AM/PM distortions as shown in Equations 3.3 and 3.3:

$$A(r) = \frac{\alpha_a r}{1 + \beta_r r^2} \tag{3.2}$$

$$\Phi(r) = \frac{\alpha_\phi r}{1 + \beta_\phi r^2} \tag{3.3}$$

where $A(.)$ and $\Phi(.)$ represent the AM/AM and AM/PM distortions, respectively, and r is the input amplitude of the PA.

To further illustrate the Saleh model, let's consider the following input signal as in Equation 3.4:

$$x\left(t\right) = r\left(t\right)\cos\Big[w_0 t + \theta\left(t\right)\Big] \tag{3.4}$$

where w_0 is the carrier frequency, $r(t)$ and $\theta(t)$ are respectively the envelope and the phase components of the input signal. Therefore, the Saleh model applied to the above equation is given in Equation 3.5:

$$y\left(t\right) = A\left[r\left(t\right)\right]\cos\Big[w_0 t + \theta\left(t\right) + \Phi\big(r(t)\big)\Big] \tag{3.5}$$

Figure 3.6 is an example of the output of the Saleh model, when using the following set of parameters [39]:

$$\begin{cases} \alpha_a = 2.1587 & \alpha_\phi = 4.033 \\ \beta_a = 1.1517 & \beta_\phi = 9.1040 \end{cases} \tag{3.6}$$

Different models have been derived from the Saleh model by proposing different shapes for the AM/AM and the AM/PM characteristics [39]. Nevertheless, the simplicity of the model and few parameters make these models poor in terms of modeling dynamic nonlinear PAs.

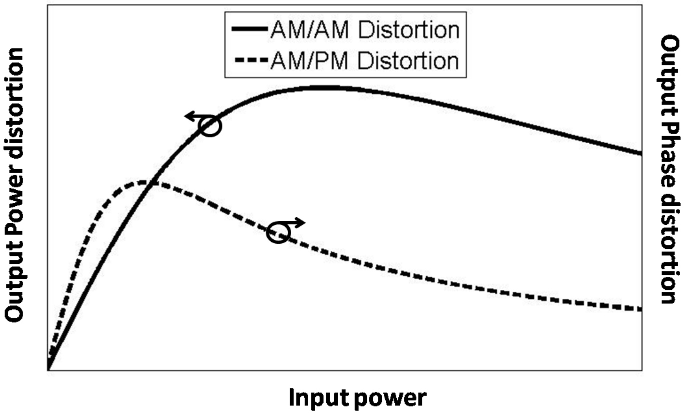

Figure 3.6: Saleh Function with the Parameters in Equation 3.6

3.2.2 Memoryless Polynomial Model

Polynomial models are broadly used to reproduce the nonlinear effect in PAs. Memoryless polynomial models consist of a polynomial formulation between the baseband input and the output signals of the PA. This relationship is denoted in Equation 3.7:

$$y\left(n\right) = \sum_{i=0}^{N} a_i \left|x\left(n\right)\right|^{i-1} x\left(n\right) \tag{3.7}$$

where x(n) and y(n) are the input and output of the PA, respectively, a_i are the complex-valued coefficients of the i^{th} power; and, N is the highest order of the memoryless polynomial model.

A well-known reduction of the polynomial model consists of the omission of the even order in the polynomial function [44]. In fact, as discussed in the two-tone test (Section 2.2.2), the even-order terms in the polynomial model produce distortions that are far away from the carrier frequency. Thus, they can be easily filtered out by a simple bandpass filter around the carrier frequency. In the memoryless polynomial, therefore, even-order coefficients can be eliminated. Equation 3.7 can be simplified to the Equation 3.8, where $2N + 1$ is now the highest order of the memoryless polynomial model.

$$y\left(n\right) = \sum_{i=0}^{N-1} a_{2i+1} \left|x\left(n\right)\right|^{2i} x\left(n\right) \tag{3.8}$$

The memoryless polynomial model is a very powerful model when the PA exhibits strong nonlinearity, but fails when the PA exhibits some sources of MEs. The identification of the memoryless model coefficients, a_i, is performed using the LSE algorithm, since this model coefficients can be written in a linear manner for a certain sequence of input/output samples of the PA. In fact, if we consider the following sequence of the input and output samples, $X = [X(n)\ X(n-1)\ \cdots\ X(N-P)]^T$ and $Y = [y(n)\ y(n-1)\ \cdots\ y(N-P)]^T$, where M is the length of the chosen sequence, $X(n) = [x(n)\ |x(n)|x(n)\ \cdots\ |x(n)|^{N-1}x(n)]$ and $[\,]^T$ is the matrix transpose, the coefficients of the memoryless polynomial model can be written as follow:

$$y(n) = \sum_{i=0}^{N} a_{2i+1}|x(n)|^{2i}x(n) \tag{3.9}$$

$$\Rightarrow y(n) = \begin{bmatrix} x(n)\ |x(n)|x(n)\ \cdots\ |x(n)|^{2N}x(n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2N+1} \end{bmatrix} \tag{3.10}$$

$$\Rightarrow \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(n-P) \end{bmatrix} = \begin{bmatrix} x(n) & \cdots & |x(n)|^{2N}x(n) \\ x(n-1) & \cdots & |x(n-1)|^{2N}x(n-1) \\ \vdots & \ddots & \vdots \\ x(n-P) & \cdots & |x(n-P)|^{2N}x(n-P) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2N+1} \end{bmatrix} \tag{3.11}$$

$$\Rightarrow Y = X \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2N+1} \end{bmatrix} \tag{3.12}$$

Therefore, using matrix pseudo-inversion, the coefficients of the memoryless polynomial can be deduced, as in Equation 3.13:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{2N+1} \end{bmatrix} = pinv(X)Y \tag{3.13}$$

where *pinv* of a matrix is the Moore Penrose pseudo-inverse [50, 54] of that matrix.

3.2.3 Look Up Table

When models represent a relative simple architecture along with few parameters, it is often less expensive to process them off-line and save the characteristic model into a LUT.

Therefore, there is no need for real-time computing and a simple memory access for each processed value is enough. The LUT consists of the accumulation of the output values of a specific model for the different possible input values. LUTs are not preferred when adaptation is required, as they consist of static memory that needs to be replaced in the case of any update. The main drawback of the LUT method is high memory occupation, which grows exponentially with the number of precision bits in the input variable.

Two LUT approaches are described, i.e. the mapping predistorter and the gain-based predistorter:

- In the mapping LUT, the input signal is represented by its Cartesian components, i.e. in-phase x_I and the phase-quadrature x_Q. Two separate LUTs are used to map the new constellation of Cartesian components, $y_I = x_I + f_I\left(x_I, x_Q\right)$ and $y_Q = x_Q + f_Q\left(x_I, x_Q\right)$. The advantage of this method is its simplicity; however, is suffers from a sensitivity to the signal to quantization noise ratio, which is proportional to the number of table entries, leading to high memory occupation. The principle of the mapping LUT is illustrated in Figure 3.7.

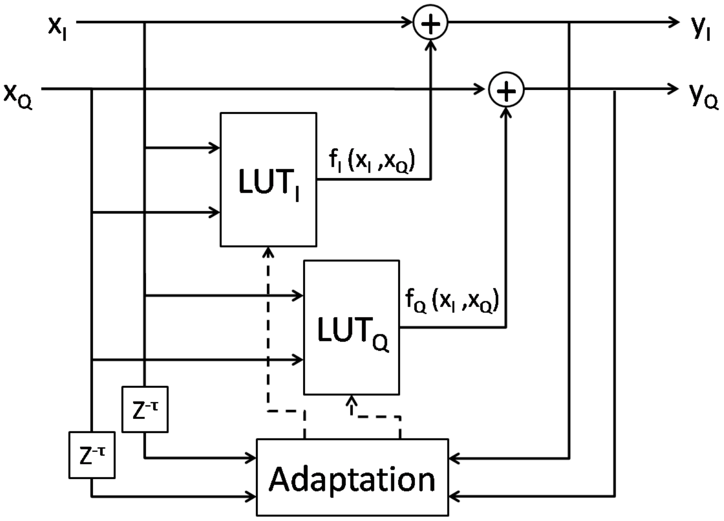

Figure 3.7: Block Diagram of Mapping Look-Up Table

- Unlike the mapping LUT, the gain-based LUT uses the power of the input signal, $R = |x|^2$, to create a unique table, where $x = x_I + jx_Q$ represents the input signal. This single LUT contains the complex gains of a predistortion function, $G_{LUT}(R)$, which is used to predistort the input signal, x, by computing the complex product $y =$

$xG_{LUT}(R)$. Compared to the mapping LUT, the complexity of the gain-based LUT is reduced, due to the use of a single table. Note that the use of only the amplitude of the input signal to linearize the PA is permitted, since the PA nonlinearity is only a function of the input amplitude; therefore, we can safely eliminate the input phase information. The principle of the mapping LUT is illustrated in Figure 3.8.

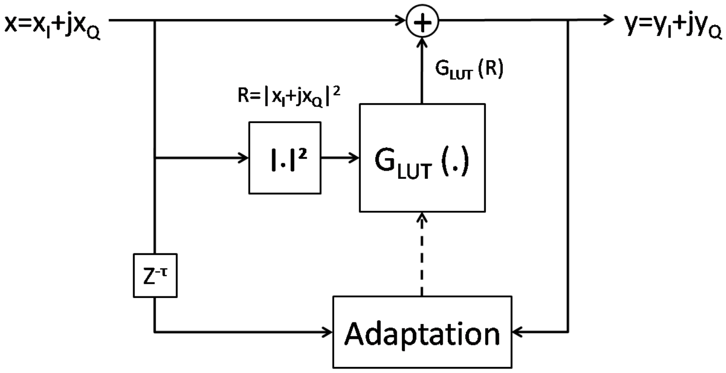

Figure 3.8: Block Diagram of Gain-Based Look-Up Table

3.3 Power Amplifiers Behavioural Modeling with Memory and Digital Predistortion

3.3.1 Volterra Series

The Volterra series [68] are comprehensive nonlinear modeling schemes that are suitable for predicting the response of a dynamic nonlinear system, such as a PA with memory. They are generic models that describe the output of nonlinear systems with memory as the sum of the response of a number of Volterra kernels [60]. The Volterra series have been used in several applications, e.g. biological and physiological systems, nonlinear satellite links, microwave circuits, etc. Many efforts have been made to prove that the Volterra series is capable of accurately modeling any dynamic nonlinear system with any arbitrary accuracy [60].

The Volterra series can be seen as a Taylor series with memory [60]. It expresses, for a causal system, the output signal, $y(t)$, as an infinite series of the input signal, $x(t)$, as in

Equation 3.15:

$$
\begin{aligned}
y(t) &= \sum_{n=1}^{\infty}\left[\int_0^{\infty}\cdots\int_0^{\infty} h_n(\tau_1 \cdots \tau_n)\prod_{j=1}^{n} x(t-\tau_j)d\tau_j\right] && (3.14)\\
y(t) &= \int_0^{\infty} h_1(\tau)x(t-\tau)d\tau + \\
&\int_0^{\infty}\int_0^{\infty} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 + \cdots \\
&\int_0^{\infty}\cdots\int_0^{\infty} h_P(\tau_1,\cdots,\tau_P)x(t-\tau_1)\cdots x(t-\tau_P)d\tau_1 d\tau_2 && (3.15)
\end{aligned}
$$

where h_1 , h_2 , $\cdots$, h_p are the Volterra kernels of the Volterra series, and P is the Volterra series order. A discretized form of the Volterra series is given in Equation 3.16, where $x(n)$ and $y(n)$ are the discretized input and output signals.

$$
\begin{aligned}
y(n) &= \sum_{q=0}^{\infty} h_1(q)x(n-q) + \\
&\sum_{q_1=0}^{\infty}\sum_{q_2=0}^{\infty} h_2(q_1,q_2)x(n-q_1)x(n-q_2) + \cdots \\
&\sum_{q_1=0}^{\infty}\cdots\sum_{q_P=0}^{\infty} h_P(q_1,\cdots,q_P)x(n-q_1)\cdots x(n-q_P) && (3.16)
\end{aligned}
$$

The Fourier transformation of the kernel functions gives the transfer functions of the modeled system. Indeed, h_1 gives the linear response; whereas, in general, h_i , $wherei > 1$, represent higher order nonlinear distortions of the nonlinear system. Figure 3.9 illustrates a representation of the Volterra series model, where the order of the Volterra kernels are restricted to 3, and H_1, H_2 and H_3 represent the Fourier transfer functions of the kernels, h_1, h_2 and h_3.

The Volterra series coefficients can be written in a linear manner, thereby allowing the use of the LSE algorithm. In fact, the Volterra series system in Equation 3.16 can be rewritten as Equation 3.17, where the Volterra kernels are restricted to 2 and M represents the memory depth of the input signals:

$$Y = UH \tag{3.17}$$

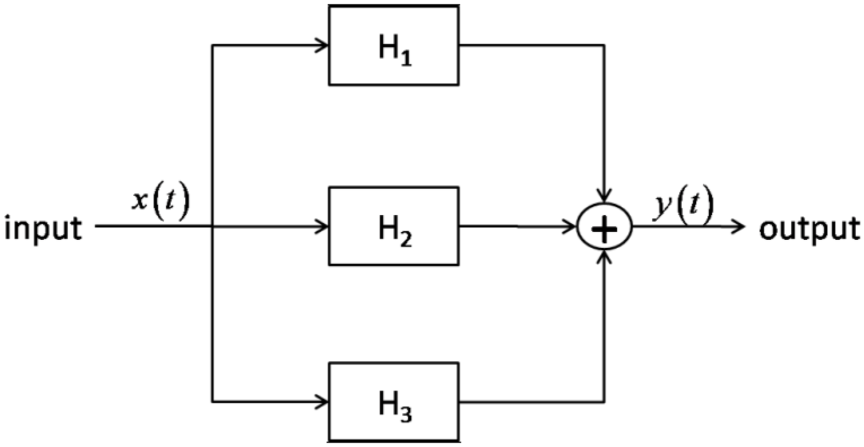

Figure 3.9: Schematic Representation of a Volterra Series Model

where

$$Y = [y(n) \ \cdots \ y(n-P)]^T \tag{3.18}$$

$$U = \begin{bmatrix} u(n) & \cdots & x(n-M) & x^2(n) & \cdots & x(n-q_1)x(n-q_2) & \cdots & x^2(n-M) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x(n-P) & \cdots & x(n-P-M) & x^2(n-P) & \cdots & x(n-P-q_1)x(n-P-q_2) & \cdots & x^2(n-P-M) \end{bmatrix} \tag{3.19}$$

$$H = [h_1(1) \ \cdots \ h_1(M) \ \ h_2(0,0) \ \cdots \ h_2(q_1,q_2) \ \cdots \ h_2(M,M)]^T \tag{3.20}$$

where, $P+1$ is the length of the sequence used for the identification. Thus, the solution using the LSE algorithm is given in Equation 3.21:

$$H = pinv(U)Y \tag{3.21}$$

3.3.2 Memory Polynomial Model

The MP model [31] is currently the most popular derivation of the Volterra model that excludes cross terms to alleviate the complexity of the Volterra series. In other words, MP exploits the diagonal kernels of the Volterra series corresponding to the pure powers of the input signals samples. MP is expressed in Equation 3.22:

$$y(n) = \sum_{j=0}^{M-1} \sum_{i=1}^{N} a_{i,j} |x(n-j)|^{i-1} x(n-j) \tag{3.22}$$

where $x(n)$ and $y(n)$ are the discretized input and output signals of the model, respectively; M is the memory depth; N is the highest polynomial order of the model; and, $a_{i,j}$ are the complex coefficients of the MP model.

A typical MP structure is shown in Figure 3.10, where M is equal to the number of branches and N is the highest order of the different polynomials in each branch. Figure 3.10 suggests that the order of nonlinearity in the different branches does not have to be the same. However, for ease of representation, orders of the different polynomials of the MP model are considered to be equal.

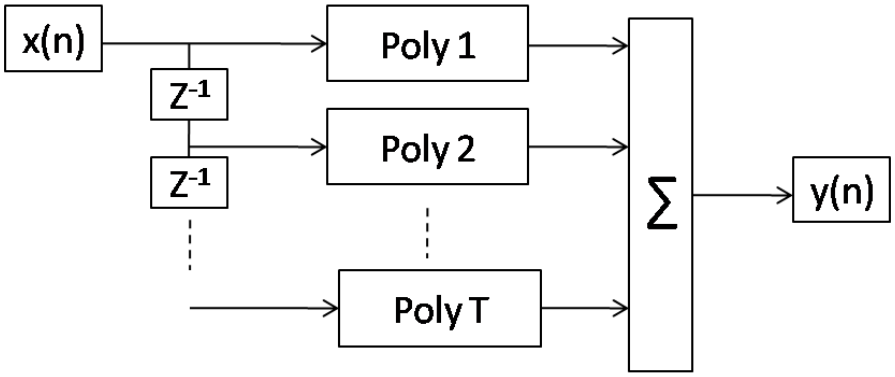

Figure 3.10: Memory Polynomial Structure

As was the case of the memoryless polynomial, the MP coefficients can be written in a linear manner, thereby allowing the use of the LSE algorithm. In fact, by defining the new states $u_{i,j}(n)$, as in Equation 3.23:

$$u_{i,j}(n) = |x(n-j)|^{i-1}x(n-j) \tag{3.23}$$

The MP system in Equation 3.22 can be rewritten as Equation 3.24:

$$Y = Ua \tag{3.24}$$

where

$$\begin{aligned}
Y &= [y(n)\ y(n-1)\ \cdots\ y(n-P+1)]^T & (3.25)\\
U &= [u_{1,0}\ \cdots\ u_{N,0}\ \cdots\ u_{1,M-1}\ \cdots\ u_{N,M-1}] & (3.26)\\
u_{i,j} &= [u_{i,j}(n)\ \cdots\ u_{i,j}(n-P+1)]^T & (3.27)\\
a &= [a_{1,0}\ \cdots\ a_{N,0}\ \cdots\ a_{1,M-1}\ \cdots\ a_{N,M-1}]^T & (3.28)
\end{aligned}$$

where P is the length of the sequence used for the identification. Thus, the solution using the LSE algorithm is given in Equation 3.29:

$$a = pinv(U)Y \tag{3.29}$$

MP was defined as a reduction of the Volterra series through the elimination of all cross terms of the samples $x(n)$. On the other hand, MP can also be defined as an expansion of the memoryless polynomial modeling by adding similar branches that process the past input samples, $x(n-j)$, where $1 < j \leq M-1$.

The MP model is considered to be the gold standard for PA behavioural modeling and linearization. The small number of coefficients and the relative simple structure of the MP make it a very attractive model. At the same time, the computational complexity in the LSE algorithm makes the MP not very simple to implement in a real-time system. Indeed, many reductions and simplifications of the MP model have been suggested in the literature to alleviate the identification task. One of the well-known simplifications is the elimination of the even-order coefficients from the MP formulation, since they only affect harmonics, i.e. very far for the carrier frequency of the PA [44]. Another attempt to reduce the number of coefficients of the MP model was proposed in [4], which consists of reducing the individual nonlinearity orders in the MP different branches. The authors in [4] succeed in decreasing the number of coefficients to almost one third and the conditioning number by three orders of magnitude, while maintaining the same modeling and linearization capabilities.

3.3.3 Hammerstein-Wiener Models

A few other derivations of the Volterra series have been suggested for PA behavioural modeling and linearization. Two of these derivations are Hammerstein and Wiener models, also called two-box models [17, 47]. These models employ a cascade of a nonlinear function and a linear filter to model dynamic nonlinear systems. In the case of the PA, the first box of the Hammerstein scheme captures its static nonlinear behaviour, while the second one is intended to take into account for its MEs, as illustrated in Figure 3.11. For the Wiener model, the two boxes are arranged differently than in the Hammerstein case, as shown in Figure 3.12.

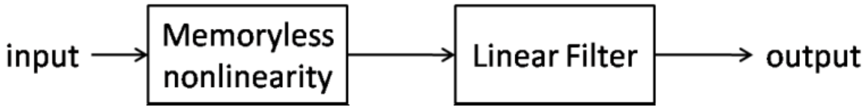

Figure 3.11: Block Diagram of the Hammerstein Model

In the literature, polynomial functions, LUTs, feed-forward ANNs, and neuro-fuzzy inference systems have been used to construct the static nonlinear function of the Hammerstein/Wiener model. However, the authors in [41] showed the limitation of conventional

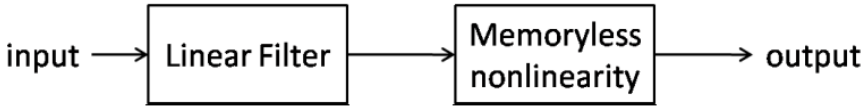

Figure 3.12: Block Diagram of the Wiener Model

Hammerstein/Wiener schemes in mimicking the behaviour of a wideband PA. They suggested an augmented version of the Hammerstein model, where the linear block is replaced with weakly nonlinear one with multiple filters to solve the modeling inaccuracy.

Parallel Hammerstein/Wiener [34] models have been also suggested to address the traditional schemes' limited capability in accounting for the MEs by stacking extra branches in parallel. The authors in [47] proposed an extended version of the Hammerstein model, where two different improvements were suggested: first, a polar feed-forward ANN was chosen for accurate construction of the static nonlinear part; and, second, the error signal between the input and output signals of the memoryless module was then computed and used to feed a second linear filter. The post-injection of the filtered error signal at the output of the conventional Hammerstein model allowed for substantial improvement of the modeling accuracy. Indeed, the extra error signal with the corresponding filter was found to complement the conventional Hammerstein model by adding an adequate mechanism to account for the MEs in the PA behaviour attributed to the dispersive bias network.

3.3.4 Artificial Neural Networks Based Models

A few alternatives to polynomial based models have been successful at capturing the PA's characteristics. These include ANNs, fuzzy logic systems, genetic algorithms, etc. In this thesis, an ANN was considered, due to its excellent capability in accurately approximating any dynamic nonlinear function [11, 14, 19, 20, 28]. In recent years, ANN has been successfully exploited to model RF and microwave devices/circuits and for PA behavioural modeling and for the construction of the corresponding DPD [1, 5, 9, 12, 23, 40, 45, 46, 49, 51, 62, 63, 70, 71].

Three different ANN structures and schemes are used for PA behavioural modeling:

- The simple structure and the Back Propagation Learning Algorithm (BPLA) make the Multi Layers Perceptron (MLP) very attractive for PA behavioural modeling [1, 5, 40, 70]. Furthermore, both complex and real-valued MLPs can be used, depending on the format of the input and output signals. As examples, the authors in

[1] used complex MLP to mimic the AM/AM and AM/PM characteristics; whereas, authors in [5, 40, 70] employed real-valued MLP to predict the in-phase and quadrature outputs of the Device Under Test (DUT). MEs are generally incorporated into the MLP structure through additional buffers in the input layer to account for the previous samples of the input and output signals. MLP has been also used for the identification of the parameters of the MP and Volterra series models [63].

- Alternatively, RBFNNs [23, 71] have been widely exploited to predict the behaviour of PAs, due to their capability in modeling dynamic nonlinear systems. The constant weights (equal to 1) in the first layer of RBFNNs make them easier to train than MLP. As an example, the authors in [23] used a signal RBFNN with complex signals in the input layers. However, because of the complex signal used, they also used the complex version of the BPLA, which made the training phase longer than for the classic BPLA case.
- RNNs are generally preferred when modeling dynamic nonlinear systems that include feedback paths and/or frequency-dependent mechanisms [9, 12, 51]. However, the complex and lengthy training algorithm (propagation through time [19, 28]) makes RNNs not very popular for real-time deployment. To mitigate this problem, attempts have targeted the modification of the RNN structure into feed-forward like scheme, so that a simpler BPLA can be applied for their training [46].

Example of ANN-Based Power Amplifier Model: Real-Valued Time Delay Neural Network

The conventional ANN model uses a complex envelope signal, like in the MP case, to model the complex output signal. However, this technique involves an ANN with complex valued parameters, which calls for complex BPLA. Alternatively, two separate real-valued ANNs have been used to model the AM/AM and AM/PM distortions and trained using the classic BPLA. However, this approach needs many more parameters than the first approach. For the case of Real-Valued Time Delay Neural Network (RVTDNN) [5], shown in Figure 3.13, a single ANN is applied to the in-phase (I) and quadrature (Q) components of the input signal to compute the output signal.

Considering the MEs of the PA, the baseband output components of the PA at instant n are a function of the current input signal and the M_1 past value of the in-phase signal and the M_2 past value of the quadrature signal. The network uses one hidden layer with nonlinear activation functions (NL), as shown in Figure 3.13. Therefore, the output of the

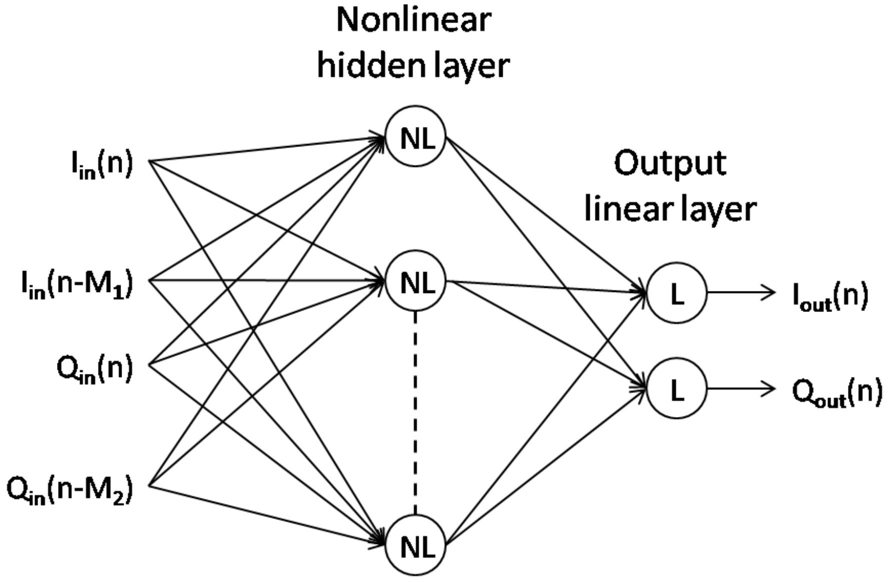

Figure 3.13: Block Diagram of the Real-Valued Time Delay Neural Networks Power Amplifier Behavioural Model

signal can be written as in Equations 3.30 and 3.31:

$$I_{out}(n) = f_1\left[I_{in}(n) \ \cdots \ I_{in}(n-M_1) \ Q_{in}(n) \ \cdots \ Q_{in}(n-M_2)\right] \quad (3.30)$$

$$Q_{out}(n) = f_2\left[I_{in}(n) \ \cdots \ I_{in}(n-M_1) \ Q_{in}(n) \ \cdots \ Q_{in}(n-M_2)\right] \quad (3.31)$$

Where f_1 and f_2 designate the dynamic nonlinearity relationship between the output and the input signals.

Knowing that the ANN uses a linear activation function (L) at the output layer, Equations 3.30 and 3.31 can be rewritten as Equations 3.32 and 3.33:

$$I_{out}(n) = \sum_{k=1}^{N} w_{1,k} f\left[\sum_{j=0}^{M_1} \alpha_{k,j} I_{in}(n-j) + \sum_{j=0}^{M_2} \beta_{k,j} Q_{in}(n-j) + b_k^1\right] + b_1^2 \quad (3.32)$$

$$Q_{out}(n) = \sum_{k=1}^{N} w_{2,k} f\left[\sum_{j=0}^{M_1} \alpha_{k,j} I_{in}(n-j) + \sum_{j=0}^{M_2} \beta_{k,j} Q_{in}(n-j) + b_k^1\right] + b_2^2 \quad (3.33)$$

where $k = 1\,,\,2\,,\,\cdots\,,\,N$ designate the nonlinearity order; $w_{1,k}$ and $w_{2,k}$ denote the weights of the second linear layer; $\alpha_{k,j}$ and $\beta_{k,j}$ represent the weight of the nonlinear hidden layer; b_1^2, b_2^2 and b_k^1 are the biasing points of the neurons in the second and the first layer, respectively; and, $f(x) = tansig(x) = \frac{1}{1+e^{-x}} - 1$ is the nonlinear activation function.

Note that the case where $M_1 = M_2$ can be used since the MEs on the in-phase component are, in general, the same as those of the quadrature component.

After the establishment of the RVTDNN structure is determined, the next step in achieving accurate modeling is to perform an appropriate training phase for the network. The training phase uses the BPLA. This algorithm adjusts the network parameters to minimize the cost function, E, over an epoch, defined as follows in Equation 3.34:

$$E = \frac{1}{2N_b} \sum_{n=1}^{N_b} \left[\left(I_{out}(n) - \hat{I}_{out}(n) \right)^2 + \left(Q_{out}(n) - \hat{Q}_{out}(n) \right)^2 \right] \tag{3.34}$$

where $I_{out}(n)$ and $Q_{out}(n)$ are the desired output, i.e. measured output; $\hat{I}_{out}(n)$ and $\hat{Q}_{out}(n)$ are the simulated RVTDNN output; and, N_b is the length of the training sequence.

The ANN models mentioned above are proven modeling structures that rely on the input and output signals relationship of the DUT. Their structures, whether a MLP, RBFNN, RVTDNN or RNN, are chosen separately from the sources of distortions and their origins in the PA. This factor negatively affects the performance of the ANN-based models, since the ANN performance directly depends on their structure [19, 28]. Therefore, in this thesis, the physical knowledge about the PA presented in [10, 43] was used to deduce an appropriate ANN structure for the behavioural modeling of the PA. This approach increases the performance of the ANN model, as well as improving the fitting of the PA behaviour, as shown in Chapter 5.

3.4 Experimental Model Identification

For the different models presented in this thesis, the following identification procedure was used: first, an experimental setup was established, as described in the next two subsections, to capture the envelope information about the input and the output signals of a given PA. Subsequently, the captured signals were used for the identification of the forward model, i.e. PA behavioural model, and the reverse model, i.e. DPD function construction, to asses the model's capability of reproducing and linearizing the PA response, as described in Section 3.4.3.

3.4.1 Experimental Setup

Figure 3.14 presents the setup used for the measurement tests to collect the signals used for the model identification. Test signals were synthesized with Agilent Advanced

Design System (ADS) [66]. The signal was then transferred to the signal generator via the General Purpose Interface Bus (GPIB). The signal excited the PA, also described as the DUT, and the PA output was captured by the spectrum analyzer. Finally, the Vector Software Analyzer (VSA) saved the signal captured by the spectrum analyzer.

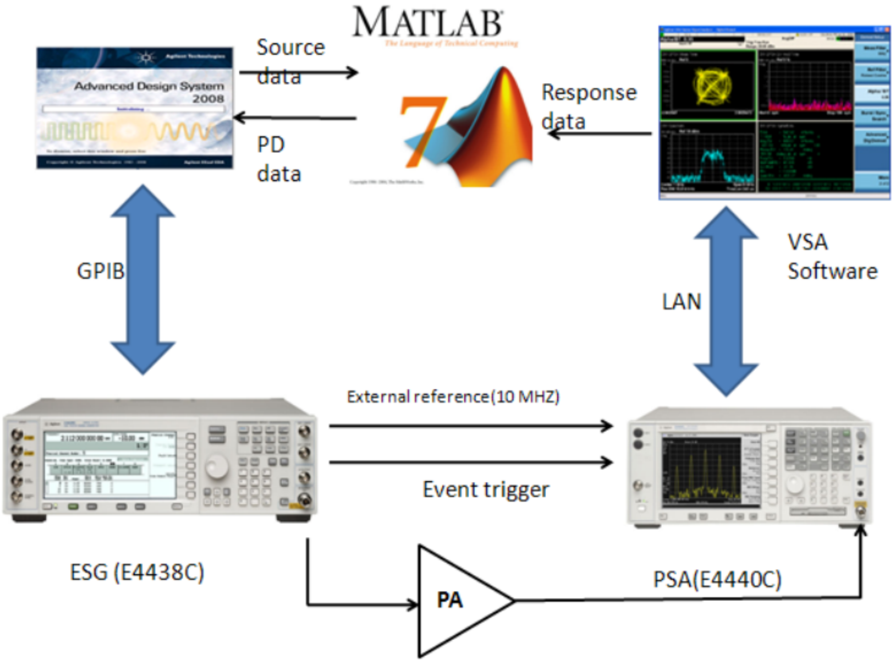

Figure 3.14: Experimental Setup

3.4.2 Signal and Device Under Test for Model Identification

Testing and validation of the proposed model was conducted using the following PAs:

- 250 Watts (W) Peak Envelop Power (PEP) Symmetrical Doherty Power Amplifier (SDPA) designed using the 80W high-power LDMOS (Laterally Diffused Metal Oxide Semiconductor) FET (Field-Effect Transistor) devices from Freescale Semiconductor, Inc., MRF7S21080H.
- 250W PEP Asymmetrical Doherty Power Amplifier (ADPA) designed using the 150W high-power LDMOS FET devices from Freescale Semiconductor, Inc., MRF7S21150H.

The input and output signals of the PA setup are measured in both forward and reverse modeling scenarios. Then, the construction of the dynamic, AM/AM, AM/PM, PSD and the PA/DPD models are performed in MATLAB.

A 4C WCDMA signal with a PAPR of 7.412dB and a 1001 WCDMA signal with a PAPR of 7.24dB, synthesized in ADS, were used as the test signals, as shown in Figures 3.15 and 3.16.

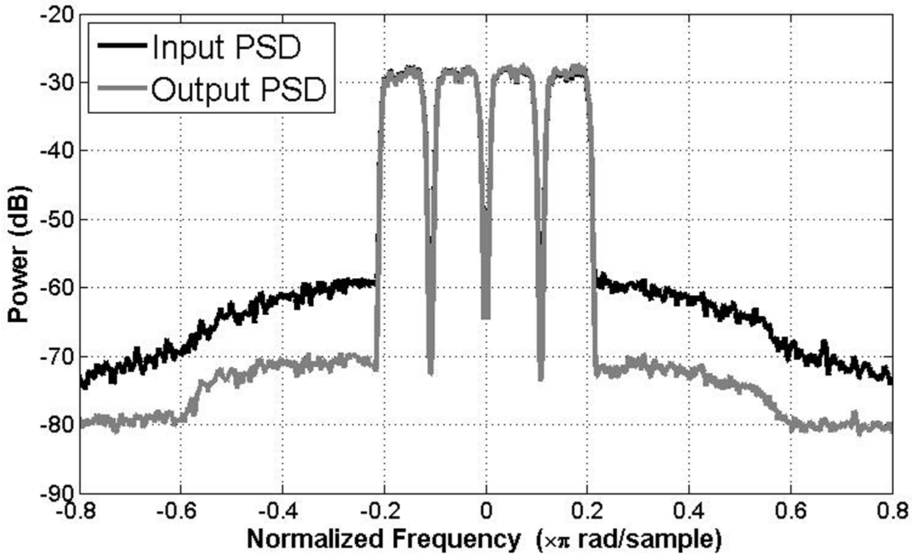

Figure 3.15: Typical DUT Input and Output Spectra of a 4-Carrier WCDMA Signal

3.4.3 Forward and Reverse Model Validation Tests

The forward model and DPD validation were performed to test the model accuracy for fitting the input and output responses of the DPD and its capacity to create an accurate linearizer.

Forward Validation Test

Forward validation aims to validate a model's capabilities in capturing and characterizing the linear and nonlinear distortions observed in the DUT. This validation was carried out by comparing the measured output response of the DUT against the output of the model. This validation was done in the frequency and time domains. For the time domain, the

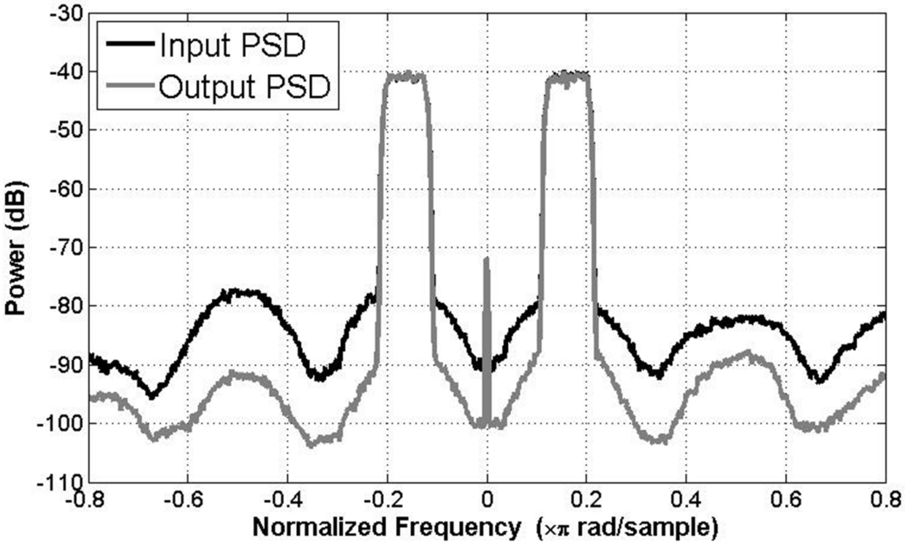

Figure 3.16: Typical DUT Input and Output Spectra of a 1001 WCDMA Signal

Normalized Mean Square Error (NMSE), shown in Equation 3.35, between the measured and the modeled signal gives the ability of the model to reproduce the output signal components of the DUT. The time domain validation can also be carried out by comparing the AM/AM and AM/PM distortions generated by the DUT and the ones produced by the model. For the frequency domain, the validation was carried out by comparing the matching between the respective PSD of the measured and modeled outputs.

$$NMSE = 10\, log_{10} \left\{ \frac{1}{2N} \sum_{n=1}^{N} \left[\left(\frac{I(n) - \hat{I}(n)}{I(n)} \right)^2 + \left(\frac{Q(n) - \hat{Q}(n)}{Q(n)} \right)^2 \right] \right\} \tag{3.35}$$

where I and Q are the measured in-phase and quadrature components of the output of the DUT, $\hat{I}$ and $\hat{Q}$ are the modeled in-phase and quadrature components, and N is the length of the sequence used for comparison.

For a typical DUT, the generated MEs are usually a small perturbation around the static nonlinear distortion. Therefore, assessing the fidelity of the model to reproduce these ME perturbation when comparing the complete model and the DUT output signal is difficult, since the MEs perturbation is hidden by the memoryless nonlinear distortion. Consequently, a model validation approach suggested in [65] was used as it emphasizes the capability of the modeling scheme in predicting the MEs. This method consists of applying a memoryless digital predistorted function, and its respective output is used to drive the

DUT. Hence, the focus is on assessing the capability of the model in predicting the residual out-of-band emission dominated by the MEs.

Reverse, or DPD, Validation Test

DPD validation requires the synthesis of the predistorted function. For an analytical model, the predistorted function can be analytically deduced. However, a easier way to find the inverse function consists of identifing the predistortion by simply reversing the input and output data in the identification process. In fact, this method results in the inverse model, because if a model consists of transforming X to Y, the inverse model is just a function that transforms back Y to X. Therefore, inverting the input and the output signal yields the construction of the inverse function, i.e. the DPD function. Subsequently, validation of the reverse model is carried out by identification the DPD's ability to cancel the unwanted distortion, especially out-of-band. Thus, the calculation of the ACPR reduction is a good measure to assess the DPD performance for correction of PA nonlinearity.

Chapter 4

Overview of Artificial Neural Networks

The term 'neural network' has been employed for two different usages:

- Biological neural networks are a set of large numbers of biological neurons that are highly connected in the human brain. They perform physiological functions and are responsible of the activity of the human brain, including processing and producing information.
- Artificial Neural Networks (ANNs), which were inspired from biological neural networks, are mathematical connectionist models that consist of connections between artificial neurons organized in a specific manner. ANNs have been successfully applied to diverse applications, including financial, medical, industrial, data mining, sales and marketing, operational analysis, human resource management, science, energy, education, etc. [19, 28].

The focus, in this thesis, is on ANNs and their use for the modeling and linearization of dynamic nonlinear PA. The structures and parameters of ANNs are discussed in this chapter.

4.1 Biological Inspiration

ANNs were inspired by the structure and functionality of biological neural networks. In fact, ANNs are an attempt to emulate the computational ability of the human brain. Therefore, a brief description of the human brain is required to understand the principle of the construction and functionality of the ANN.

The development of biological neural networks is performed in two phases. The first phase happens at birth, where the human is provided with a specific network structure of neurons with their respective connections. This structure, i.e. the human brain, is composed of a massive network of parallel and distributed computational elements called neurons (approximately 10^{11}). These neurons are highly connected (approximately 10^4 connections per element) making it capable of very powerful learning, storage and processing of information. These neurons are composed of three main components, namely dendrites, cell body and axon, as shown in Figure 4.1. Note that the artificial neuron adopts these three components in the construction of its structure.

- The dendrites are collector connections that bring electrical signals into the cell body.
- The cell body obtains the electrical signals from the dendrites and then sums and thresholds these signals.
- The axon is an output of the cell body that carries the signal to other neurons to become dendrites for these neurons through points of contact called synapses.

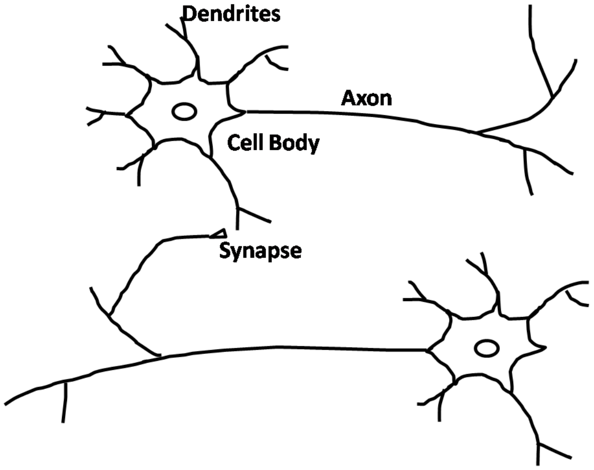

Figure 4.1: Schematic Drawing of Biological Neurons

The second phase of the biological neural network is performed throughout human life, where the network continues changing by creating or cancelling some neurons and strengthening or weakening their connections. The second phase is known as the learning phase, where the human updates his biological network, while learning and storing new information and knowledge.

ANNs use the concept of biological neural networks. Nevertheless, ANN does not approach the high complexity of the human brain, but tends to reproduce the same structure and learning process of the biological neural networks. In fact, similar to biological neural networks, the development of the ANNs is done through two steps. First, an a priori structure with some predefined connections (called weights in the case of ANN), is chosen where the structure of the artificial neuron resembles the biological neuron. Second, through a learning process, the ANN updates its structure and weights using some information for the purpose of saving, learning and/or processing this information.

Note that the principle of ANN is original and innovative compared to the traditional way of computing. Indeed, ANNs' usage is motivated by the fact that, although human brains are much slower than electrical circuits (10^{-3} versus 19^{-9} seconds), they are capable to solve much more complex problems faster and more efficiently than any conventional computer. This is due to the highly connected structure of the neurons in the human brain, allowing these neurons to work simultaneously, which is much more powerful than the sequential and/or recursive procedures of traditional computers.

The next section describes the artificial neuron architecture and its similarity to the biological neurons. Section 4.3 discusses the learning process and its similarity to human learning. Finally, the last two sections (4.4. and 4.5) present the ANN structure and learning parameters.

4.2 Neuron Model and Network Architecture

4.2.1 Neuron Model

Single Neuron

A typical artificial neuron is shown in Figure 4.2. Like the biological neuron, the artificial neuron is composed of three elements:

- A set of input weights, similar to the dendrites, which represents the connection between neurons in the network.
- An Activation Function (AF), also called a transfer function, similar to the cell body, is responsible for summing and processing the input weights, resulting in a single output.

– Output weights, similar to the axon, are responsible for transferring the output of the neuron to the other neurons in the ANN, which will represent either input weights for different neurons or the output of the network.

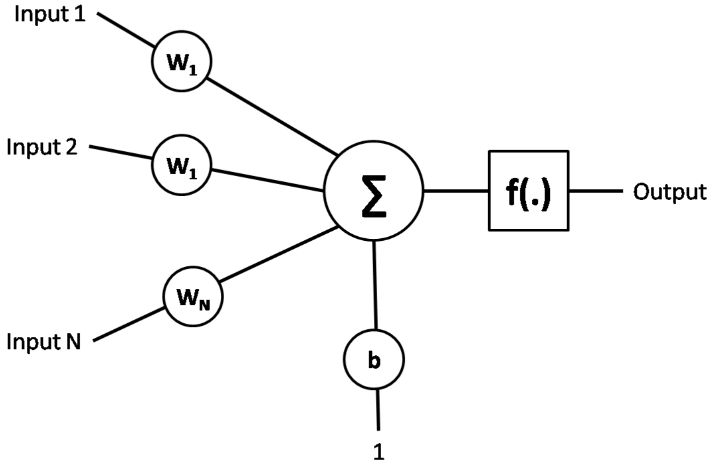

Figure 4.2: Single Neuron

From a mathematical point of view, the artificial neuron works as follows. Outputs, o_i, from others neurons that are connected to specific neurons are multiplied by certain weights, w_i. In general, a special neuron is connected to all neurons in the network and has a constant output equal to 1, called bias; and, this neuron has a special weight, b. All the resulting values at the input of the neuron are summed and transferred to AF which produces an output, o. This mechanism is summarized in Equation 4.1.

$$o = f\left(\sum_{i=1}^{N} w_i o_i + b\right) \tag{4.1}$$

where N represent the number of neurons connected to the inputs of the neuron, f designates the AF and w_i and b are adjustable parameters of the neuron that a learning algorithm sets, so that the ANN converges to a specific task. On the other hand, AFs are invariable functions that are set when choosing the network structure. There exist different types of AFs, which are described in the next subsection.

Activation Function

There exist two types of AFs used in ANN, linear and nonlinear:

- A linear AF is used to mimic linear systems. For example, an equivalent of the Finite Impulse Response (FIR) filter can be modeled using an ANN with a set of liner neurons.
- A nonlinear AF is used in the modeling of nonlinear systems. For example, a static nonlinear system can be modeled by an ANN with a set of nonlinear neurons.

Figures 4.3 and 4.4 and Equations 4.2-4.6 represent the most commonly used linear and nonlinear AFs used in ANNs.

$$Purelin(x) = x \tag{4.2}$$

$$Satlin(x) = \begin{cases} 0, & if \quad x \leq 0 \\ x, & if \quad 0 \leq x \leq 1 \\ 1, & if \quad 1 \leq x \end{cases} \tag{4.3}$$

$$Satlins(x) = \begin{cases} -1, & if \quad x \leq -1 \\ x, & if \quad -1 \leq x \leq 1 \\ 1, & if \quad 1 \leq x \end{cases} \tag{4.4}$$

$$Tansig(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{4.5}$$

$$Logsig(x) = \frac{1}{1 + e^{-x}} \tag{4.6}$$

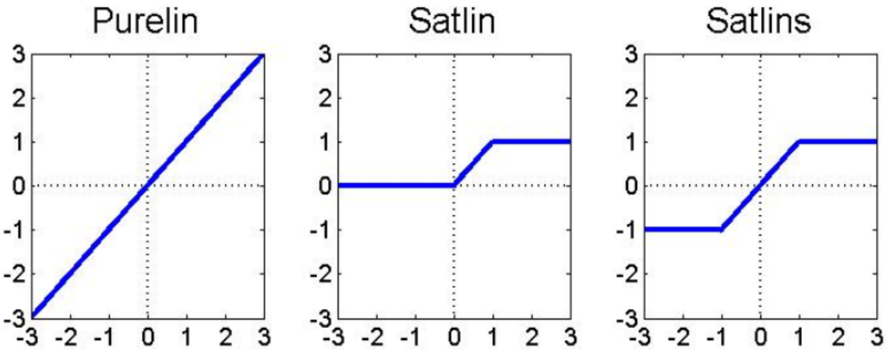

Figure 4.3: Most Commonly Used Linear Activation Functions

4.2.2 Network Architecture

As in the case of the human brain, one neuron is not sufficient to approximate an input-output relationship. In general, an arrangement of parallel neurons, called layers, is set to accurately model an input-output relationship.

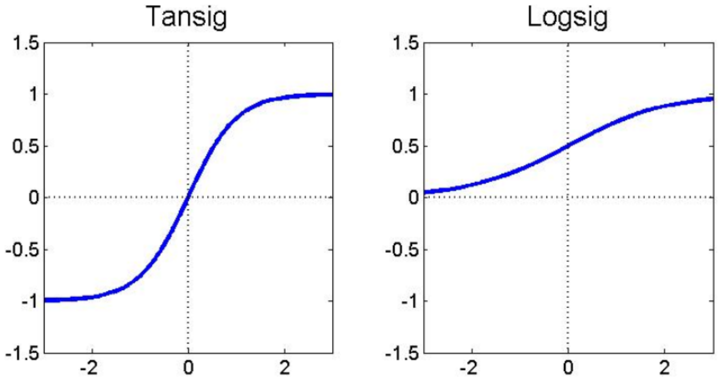

Figure 4.4: Most Commonly Used Nonlinear Activation Functions

One Layer of Neurons

A set of N neurons arranged in parallel, as shown in Figure 4.5, represents one layer of neurons. Input neurons are connected to each neuron of the one-layer network and outputs of these neurons represent outputs of the ANN, i.e. there is no hidden layer in this type of network. The one layer of neurons works as follows: input elements ,x_j, where $j = 1 \cdots M$, are connected to the single layer through a weight matrix, W, where M is the number of inputs. Each neuron has a bias, b_i, where $i = 1 \cdots N$; an adder; an AF, f_i, where $i = 1 \cdots N$; and, an output, o_i, where $i = 1 \cdots N$ and N represents the number of the neurons in the one-layer network. These ANN outputs are expressed in Equation 4.7:

$$o_i = f_i \left(\sum_{j=1}^{M} w_{i,j} o_j + b_i \right) \; ; \, i = 1 \cdots N \tag{4.7}$$

Equation 4.7 can be rewritten in matrix form, as in Equation 4.8, where O is the output vector of the ANN, W is the weight matrix, X is the input vector and B is the bias vector.

$$O = F\left(WX + B\right) \tag{4.8}$$

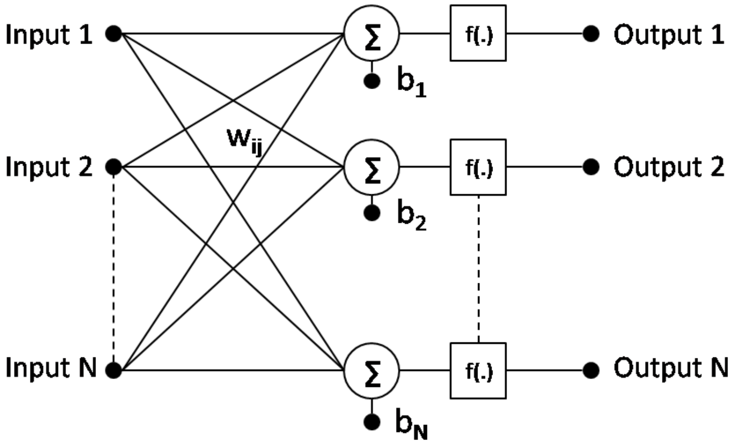

Figure 4.5: One Layer of Neurons

Where

$$
\begin{aligned}
O &= [o_1\ o_2\ \cdots\ o_N]^T && (4.9)\\
F\left([a_1\ a_2\ \cdots\ a_N]^T\right) &= [f_1(a_1)\ f_2(a_2)\ \cdots\ f_N(a_N)]^T && (4.10)\\
W &= \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,M} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N,1} & w_{N,2} & \cdots & w_{N,M} \end{bmatrix} && (4.11)\\
X &= [x_1\ x_2\ \cdots\ x_M] && (4.12)\\
B &= [b_1\ b_2\ \cdots\ b_N] && (4.13)\\
& && (4.14)
\end{aligned}
$$

It is common for the neuron?s AFs in the same layer to be set the same; however, in general, there is no restriction to having different AFs in the same layer.

Multiple Layers of Neurons

For a multiple layer ANN, which is shown in Figure 4.6, each layer consists of a different weight matrix, W^i, where $i = 1 \cdots NL$ and NL is the number of layers; different bias vectors, b^i where $i = 1 \cdots NL$; and, an output vector, o. The layer whose output is the output of the ANN is called an output layer, where the other ones are called hidden layers.

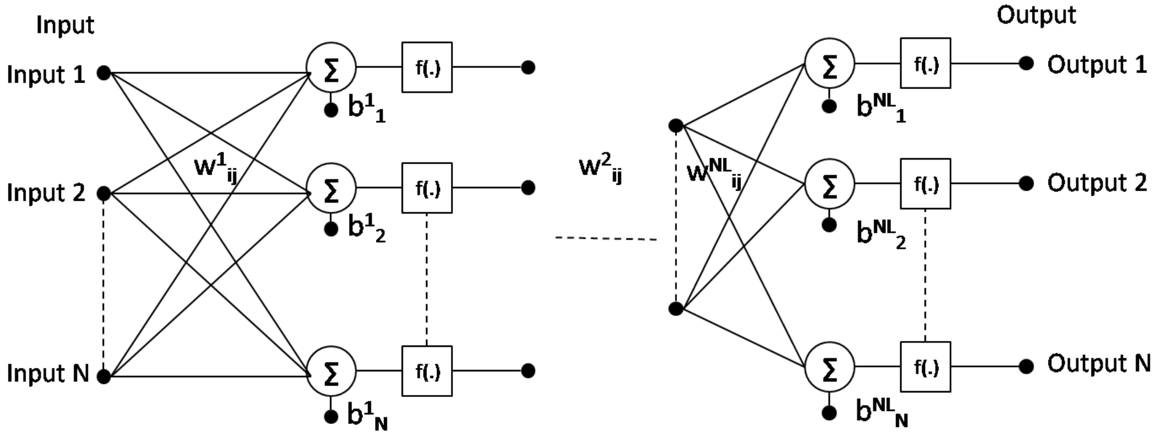

Figure 4.6: Multiple Layers of Neurons

Multilayer ANNs are more general than the one-layer network; and, they are able to perform better representations than the one-layer network, in terms of modeling input-output relationships. For example, a one hidden layer network with nonlinear AFs in the hidden layer can approximate most nonlinear functions arbitrarily well [11, 14, 19, 20, 28]. Moreover, Chester in [8] established that the use of two hidden layers instead of one, makes the ANN stronger, in terms of accuracy and stability. In the Section 4.4, a more detailed discussion about the number of hidden layers is provided.

With the use of a multilayer ANN the choice of the number of neurons and the AFs becomes more difficult. For the one-layer network, the number of inputs to the network and the number of outputs from the network are equal to the system inputs and outputs being modeled. The choice of AFs in the one-layer network depends on the relationship between the inputs and the outputs. Therefore, the system to be modeled can be used to determine almost completely the network architecture and parameters of the one-layer network.

In the case of networks with hidden layers, however, the problem specification, i.e. the input-output relationship, does not provide all the ANN parameters. One of the tedious parameters to find is the required number of neurons in the hidden layers. In fact, there is no direct information that can provide the exact or even approximate number of neurons in the different hidden layers, since these parameters are hidden to the system being modeled.

In multilayer networks, there exist two types of ANN structures: feed-forward and recurrent ANN:

- Feed-Forward Neural Networks (FFNNs) are ANNs that allow signals to travel only in one direction, from the input layer to the output layer. In FFNNs, hidden layers, including the output layer, get their input from only the previous layers. This type of network is extensively used in pattern recognition and as an input-output modeling approach.

- Recurrent Neural Networks (RNNs), also called feedback ANNs, allow signals to travel both ways. This type of network allows feedback loops, which make them very powerful in the modeling of dynamic nonlinear systems with memory. However, RNNs are very complicated for training and present serious problems in terms of stability. Many applications use RNNs, but many works preferred an FFNN or a modified a priori RNN structure to a feed-forward structure [46].

4.3 Artificial Neural Networks Training Algorithm

As in the case of biological neural networks, an a priori structure is set at the time of birth, but a learning process is needed to acquire knowledge. The same process is works for ANNs. In fact, after setting the structure of the ANN, a training phase is required to make the ANN learn the relationships between its inputs and outputs. The training of the ANN is performed by adjusting the values of the connections (weights and biases) between the neurons to approach a certain behaviour. The training process is classified into two categories:

- Supervised learning, in which learning is applied by practice through learning, i.e. learn by example.

- Unsupervised learning, in which the system is presented with a number of patterns and the network has to group these patterns into categories, i.e. learn from the environment.

These two algorithms are discussed in the following subsections.

4.3.1 Supervised Learning Algorithm

The supervised learning algorithm relies on an external agent that provides the network with a set of training signals, i.e. a set of inputs and their respective outputs, also called the desired outputs. The principle of the supervised algorithm is based on error correction. In fact, the network adjusts its parameters (weights and biases) with respect to the error between the network outputs and the desired outputs. One of the most powerful algorithms in the class of supervised algorithms is the BPLA [19, 28], which uses the error between the network outputs and the desired outputs to update the network parameters in the backward direction. The BPLA is comprised of two steps: first, forward propagation, where an error signal is computed; and, second, a backward propagation of the error adjusts the network parameters starting with the output layer down to the input one.

The BPLA works as follows [19, 28]:

1. Starts with an initialization phase, in which the selection of random values for the weights and the biases is performed.
2. Selects an input pattern from the training data and feeds it to the inputs of the ANN.
3. Activates neurons in the following way: multiplies the weight values of the connections leading to a neuron with the output values of the preceding neurons; adds up these values; passes the result to the AF; and, the AF processes these inputs, resulting in the output value of this neuron.
4. Repeats step 3 until the output layer is reached.
5. Computes the error between the outputs of the ANN and the desired outputs.
6. In a backward manner, adjusts the ANN parameters to reduce the local error.
7. If the error reaches a desired level, the algorithm ends; otherwise, the algorithm restarts from step 2.

There are different variations of the BPLA. The most widely used version is based on the descend gradient method. This algorithm updates the network parameters in the direction that the performance function decreases most rapidly, i.e. the negative of the error gradient. The gradient descendent BPLA is based on the iteration of the network parameters in Equation 4.15:

$$x_{k+1} = x_k - \alpha \frac{\partial e_k}{\partial x_k} \tag{4.15}$$

where k represents the iteration index, x is the network parameter (weight or bias), α is the learning rate, and e is the error between the network outputs and the targeted outputs.

There exist two ways to compute the error function and its gradient, which are needed to update the network parameters:

- Sequential mode, where the gradient is computed and the network parameters are updated after each single input pattern is applied to the network.
- Batch mode, where the gradient is computed and the network parameters are updated after the processing of all inputs patterns.

4.3.2 Unsupervised Learning Algorithm

Unlike the supervised algorithm, the unsupervised one does not use an external agent that provides the network with a set of training data. The unsupervised learning algorithm, also called a self-organizing algorithm, clusters the network input data with respect to their resemblance and emergent collective properties. The unsupervised algorithm does not have any knowledge about what the output should be. This type of algorithm is very useful in many applications, where there is no a priori information.

For this thesis, the focus is on the supervised learning algorithm, since the input the output of the PA being modeled or linearized are measured quantities.

4.4 Artificial Neural Network Structure Parameters

4.4.1 Number of Hidden Layers

The number of hidden layers in an ANN has received a great deal of interest in ANN research. In fact, some theories [11, 14, 20] have proven that a feed-forward ANN with a single hidden layer with nonlinear AFs can approximate any nonlinear function with any desired error. This result is known as the universal approximation theorem. This theorem was published by Cybenko [11], Funahashi [14], and Hornik, Stinchcourbe and White [20]. This theorem was inspired by the Weierstrass theorem [57].

The universal approximation theorem states that, given a:

- $\phi(.)$ as a non constant, bounded and monotone-increasing continuous function,
- I_{m_0} as an m_0-dimensional unit hypercube $[0, 1]^{m_0}$ function, and

- $C\left(I_{m_0}\right)$ as space of continuous function,

for any function, $f \in C\left(I_{m_0}\right)$, and any accuracy, $\epsilon > 0$, there exists an integer, m_1, and a set of real constants, α_i, b_i, and $w_{i,j}$ where $i = 1 \cdots m_1$ and $j = 1 \cdots m_0$, such that:

$$F\left(x_1, \cdots, x_{m_0}\right) = \sum_{i=1}^{m_1} \alpha_i \phi \left(\sum_{j=1}^{m_0} w_{i,j} x_j + b_i \right) \tag{4.16}$$

is an approximate realization of the function $f(.)$; that is:

$$\left| f\left(x_1, \cdots, x_{m_0}\right) - F\left(x_1, \cdots, x_{m_0}\right) \right| < \epsilon \tag{4.17}$$

for all $x_1, \cdots, x_{m_0}$ that lie in the input space, I_{m_0}.

Equation 4.16 represents the structure of a feed-forward ANN with one hidden layer. The universal approximation theorem implies that a single hidden layer is sufficient for a multilayer perceptron to compute a uniform approximation to a given training set represented by the set of inputs, $x_1, \cdots, x_{m_0}$ and a desired output, $f\left(x_1, \cdots, x_{m_0}\right)$. The universal approximation theorem was later generalized to the two hidden layer case by Kurkova [35].

The universal approximation theorem has received a lot of interest when dealing with ANN as modeling tool; and, consequently, many applications restricted the choice of number of hidden layers to 1. However, the theorem does not say if the single hidden layer is optimal in the sense of learning time (number of epochs), ease of implementation and, in particular, generalization (behaviour with non trained data). In addition, the theorem assumes that the training data are noise free, an assumption that is violated in most practical applications, such as the case of the PA modeling problem.

Some alternatives studies extended the use of ANN to two hidden layers to provide a more general and stable ANN, compared to the case of a single hidden layer. Nevertheless, most practical ANNs have just one or two hidden layers. Three or more hidden layers are rarely used; and, until now, there has been no theoretical study that proves the necessity of more than two hidden layers. The real issue at present is whether one or two hidden layers should be used. In [8], Chester established that with a finite sampling of points and with a two hidden layer ANN, results are better than those from the universal approximation theorem; and, even if similar, the number of hidden neurons grows without bound in the case of the single hidden layers.

The authors in [21] proved that a small network with two hidden layers can be used in most of approximation and modeling problems, where a network with one hidden layer would require an infinite number of nodes. In fact, Chester criticized the use of the single hidden layer, in the sense that it is true that the universal approximation theorem can produce any level of accuracy for the training data, but with new data nothing is guaranteed. In other words, with a finite number of neurons in a single hidden layer, no generality is guaranteed for non trained data. However, the single hidden layers can sometimes be useful. Indeed, the author in [18] suggested that the single hidden layer is the best choice when the system being modeled has one input and/or if data are noise free.

Yet, there is no deterministic and clear approach to set the number of hidden layers in a particular problem; and, an empirical study remains the only way to deal with this matter. Hence, the choice of the number of hidden layers can be based on the following criteria: the size of the ANN, the learning time, the implementation capability of the hardware, the targeted accuracy and, especially, the system structure. The system structure can inspire that of the ANN, as is the case in the model proposed in this thesis [46].

4.4.2 Number of Neurons in the Hidden Layers

The number of neurons in the input and output layers is straight-forward. In fact, the number of neurons in the input layer depends on the number of possible inputs of the system being modeled or approximated; and, similarly, the number of neurons in the output layer depends on the number of desired outputs. However, there exists no concrete rule for the selection of the number of neurons in the hidden layers, since they are hidden to the system being modeled. Moreover, these numbers are the most important and crucial parameters in an ANN [19, 28]. When not enough neurons are used, the ANN filters out the underlying function; whereas, when too many neurons are used, the ANN starts to model the noise and/or overfits the data. Thus, the real power of the ANN in its generality is primarily determined by the number of neurons in the hidden layers. Generally, users of ANN turn to empirical studies for the determination of the number of neurons in the hidden layer.

The empirical study for the number of neurons consists of changing the number of neurons in each hidden layer and analyzing the results using the bias-variance dilemma [16], in order to determine the most suitable number of neurons. The bias error corresponds to the error relative to the training data; whereas, the variance error is relative to the overall data, including data not provided during the training. The bias-variance dilemma represents a good indicator for the overfitting problem in ANNs, since it presents a good measure for the generality of the ANN assessed by measuring the ANN performance with non trained data. One of the techniques used for the bias-variance error is the cross-validation [19], which balances the ANN between a good interpolator and a good generalizer.

A first guess of the number of hidden layers can be set to the average of the number of input and output attributes [37]. The adjustment of the number of neurons is then performed to obtain the better trained network, until the optimal number of neurons is attained. To accomplish this, two different strategies in learning algorithms can be used:

– The pruning procedure [38], which consists of starting with a large number of neurons and gradually reducing the connections between neurons to reach the optimal network.

– The growing algorithm [6], which starts with a few neurons and adds more connections and neurons to reach the optimal network.

4.4.3 Type of Activation Function

As in the case of almost all ANN parameters, there exists no straightforward method to determine the suitable AFs. For the output layer and/or other hidden layers, the choice of the AFs depends on the nature of the problem. Usually, linear AFs are used for regression problems, while nonlinear AFs are used for classification or pattern recognition [28]. For linear and nonlinear AFs, the use of an antisymmetrical function often results in faster training [37]. A popular choice of AFs is the antisymmetric hyperbolic tangent, which is defined in Equation 4.18 [37]:

$$\phi(x) = a\,tansig(bx) = a\,\frac{e^{2bx}-1}{e^{2bx}+1} \tag{4.18}$$

where a and b are constants. LeCun [37] proposed the following values for a and b:

$$a = 1.7159 \tag{4.19}$$

$$b = \frac{2}{3} \tag{4.20}$$

4.4.4 Input and Target Values

Preprocessing the training data can sometime be very crucial to good ANN performance. In fact, in some extreme cases, the ANN fails to converge to a good solution. For example, for very small targeted values, the ANN can easily diverge, especially when their respective inputs are large and/or the ANN has a large size.

LeCun in [37] proposed a number of preprocessing steps for the input and output training data, in order to facilitate the task of the learning algorithm:

- Uncorrelate the input using the principal component analysis.
- Scale the input variables, so that their covariances became equal; then, all input variables are trained at the same speed.
- Normalize the output in the range of the AF in such a way that the output values are offset by some amount away from the saturation of the nonlinear AF.

Also, it is always preferable to maximize the information content in the training data. This can be done by using training data that results in the largest training error and/or training data that are radically different from all those previously used. LeCun [37] proposed to randomize the order in which the examples are presented to the ANN from one epoch to the other, in order to maximize the information content.

4.5 Artificial Neural Networks Training Parameters

The following subsections discuss some heuristics that can improve the learning process for ANN.

4.5.1 Learning Rate and Momentum Term

In the BPLA, the learning rate controls the speed of convergence, i.e. too small a learning rate value makes the network learn very slowly, while too large a learning rate value makes the network diverge. In [3], the authors demonstrated that the best learning rate can be deduced from the Hessian matrix. However, since the Hessian matrix changes dramatically, because the objective function has many local and global optima, a constant learning rate is the preferred solution.

The rule to select the right learning rate is to have all the neurons learn at the same rate; and, since the gradient of the output layer is larger than the one in the hidden layer, a smaller learning rate is usually set for the output layer. In general, a good value for the learning rate should be smaller than $\frac{2}{\lambda_{max}}$, where λ_{max} is the maximal eigenvalue of the error function of the Hessian matrix of second derivatives [19]. A simpler rule for the choice of the learning rate is:

- The smaller the learning rate value is, the smoother the convergence is; however, a large number of iterations is required.
- The larger the learning rate value is, the rougher the convergence is; however, a small number of iterations is required.

.

The BPLA, even with a good choice of learning rate, suffers from the convergence into local optima. Many algorithms have been proposed to optimize the learning rate, in order to approach the global optima. Alternatively, a simple and powerful method has been introduced by authors in [56] to improve the learning process by adding a momentum term, as in Equation 4.22:

$$x_{k+1} = x_k - \alpha \frac{\partial e_k}{\partial x_k} + \eta \left[x_k - x_{k-1} \right] \qquad (4.21)$$

$$x_{k+1} = x_k - \alpha \frac{\partial e_k}{\partial x_k} + \eta \left[-\alpha \frac{\partial e_{k-1}}{\partial x_{k-1}} \right] \qquad (4.22)$$

where k represents the iteration index, x is the network parameter (weight or bias), α is the learning rate, e is the error between the network outputs and the desired outputs, and η represents the momentum term.

The momentum term adds prehistory to the current update of the parameters, involving previous parameter changes to the current changes. In other words, the momentum term determines the relative contribution of the current and past errors to the current change of the parameters. Thus, the momentum rate is a solution of oscillatory descent, when applying a large learning rate, and a solution of slow convergence, when using a small learning rate, due to the prehistory added to the learning by the momentum term [19].

The learning rate and momentum term values can be determined by the Hessian matrix of the error function. Nevertheless, because the computation of the Hessian matrix is expensive and non constant throughout the learning, the optimal choice of these two values remains based on trial and evaluation techniques. In general, a good start for these two values is $]0\,,\,0.3]$ for the learning rate and $[0.5\,,\,0.7]$ for the momentum term [19].

4.5.2 Error Function for Stopping Criterion at Learning

The BPLA is controlled by the maximum allowed error function, i.e. the desired error. Therefore, the modeling capability of an ANN is defined by the stopping criterion of the learning. Usually, the desired error is a function of the amount of noise present in the data, which can, in general, be approximated. Thus, it is a good idea to incorporate information about the noise for the assessment of the network's performance, especially when the cross-validation technique is considered. In fact, involving the amount of the noise in the error function provides an idea about when the ANN turns from a good generalizer to curve fitting (data and noise modeling). The traditional NMSE function assumes the data to be noise free; therefore, scaling of this error function with the amount of noise is required to take into account the noise information in the data.

Several functions that incorporate information about the noise have been developed in the literature. These functions consist of variations of the widely used Root Mean Square Error (RMSE) defined in Equation 4.23:

$$E_{RMSE} = \sqrt{\frac{\sum_{p=1}^{P}\sum_{k=1}^{K}\left(d_{p,k} - o_{p,k}\right)^2}{PK}} \tag{4.23}$$

Where $d_{p,k}$ are the desired outputs, $o_{p,k}$ are the ANN outputs, P is the length of training data, and K is the number of output neurons. In the following equations, four error functions that use information about the amount of noise present in the training data in their definition are defined as:

$$\begin{aligned}
E_{\sigma} &= \frac{E_{RMSE}}{\sigma_d} && (4.24)\\
E_{d} &= \frac{E_{RMSE}}{\left|\bar{d}\right|} && (4.25)\\
E_{(\sigma+d)} &= \frac{E_{RMSE}}{0.5\left(\sigma_d + \left|\bar{d}\right|\right)} && (4.26)\\
E_{(exp\, d\sigma)} &= \frac{E_{RMSE}}{e^{-\left|\bar{d}\right|}\sigma_d + \left|\bar{d}\right|} && (4.27)
\end{aligned}$$

where σ_d and $\bar{d}$ denote the standard deviation and the mean of the desired values, respectively.

4.5.3 Weight Initialization

A good choice of the initial weight and bias values can be a crucial step for a successful network design. In fact, with inappropriate initial network parameters, the training of the network can easily diverge or last forever, without any significant learning or adapting [19]. For example, with large weights, the neurons in the network are driven into saturation, which leads to a very small gradient, making the learning very slow and sometimes ineffective. On the other hand, if the weights are assigned to be very small, the AFs operate at a flat region, near zero, which makes the network unable to model a nonlinear function. Thus, when the initialization is unsuitable, the input of the nonlinear neurons are far from the nonlinear shape of the AF; therefore, the gradient of the error is very small which, resulting in no change in the adaptation. Consequently, the proper choice of initialization should lie between these two extreme cases.

Usually, a random initialization leads to a good starting point for network learning. In addition, some studies have advanced some standard strategies for the initialization of the weights for any ANN applications. For example, Gallant in [15] proposed a uniform distribution of the weights in the interval $\left[\frac{-2}{I}, \frac{2}{I}\right]$, where I represents the dimension of the input vector. Alternatively, LeCun [37] proposed a mean zero distribution of the weights and a variance equal to $m^{-0.5}$, where m represents the average number of synaptic connections of a neuron.

In general, the key to a good choice of the network parameters is avoidance of the two extreme cases previously presented, as well as avoidance of a symmetrical distribution of weights. In fact, symmetry in the network parameters (or worse, the same distribution) for the different neurons usually results in all neurons performing similarly.

4.5.4 Maximizing the Information Content

In general, good training of a particular ANN depends on the information content present in the training data. Thus, using the largest possible information available in the system being modeled results in a good bias error, but also in a good variance error. Two strategies can be adopted for maximizing the information content in the training data [37]:

- Select the training data that results on the largest possible training error.
- Avoid training data that are highly redundant.

These two techniques result in training of the ANN in the largest space, covering the largest functionality of the system being modeled.

4.5.5 Sequential versus Batch Update

A sequential update is a pattern-by-pattern updating, while a batch update consists of one update after processing all the training data. Usually, for large and highly redundant data, the sequential update is faster than a batch update. However, the batch mode is more resistant to imprecise and distorted examples (noisy data) than the sequential mode.

Chapter 5

Two Hidden Layers Artificial Neural Networks for Behavioural Modeling and Linearization of RF Power Amplifier

In this chapter, a novel Two Hidden Layers Artificial Neural Networks (2HLANN) model is suggested to predict the dynamic nonlinear behaviour and linearization of wideband RF PAs. Starting with a generic low-pass equivalent model of the PA [10, 43], an appropriate ANN structure is deduced. This structure is then optimized to form a real-valued and feed-forward 2HLANN based model that is capable of predicting the nonlinear behaviour and MEs of wideband PAs.

The validation of the proposed model in mimicking the behaviour of a DUT is carried out, in terms of its accuracy in predicting the output spectrum, the dynamic AM/AM and AM/PM characteristics and the NMSE. The forward validation is carried out using two PAs, namely a 130W class AB PA and a 250W PEP Doherty PA, both driven with a 20MHz bandwidth signal. In addition, the 2HLANN model is used to linearize two 250 Watt PEP Doherty PAs driven with 20 MHz bandwidth signals. The 2HLANN model performance is compared to a number of previously published behavioural models. This extensive validation demonstrates the excellent modeling accuracy and linearization capability of the proposed 2HLANN model.

5.1 Two Hidden Layers Artificial Neural Network

To identify a suitable ANN structure, an in-depth study of PA behaviour is needed. Figure 5.1 shows the block diagram of a typical RF PA [10], describing its band-pass behaviour. The transistor, represented by a nonlinear function, $g(.)$, designates the static nonlinear distortion of the PA. However, three extra linear filters represent the frequency-dependent behaviour of the networks around the transistor, which are generally found to be the main sources behind the mechanism of MEs. Generally, contributions from the input and output FIR filters are less pronounced than the feedback FIR filter; therefore, the diagram can be simplified into the PA model shown in Figure 5.2, where only the feedback FIR filter remains [10, 43].

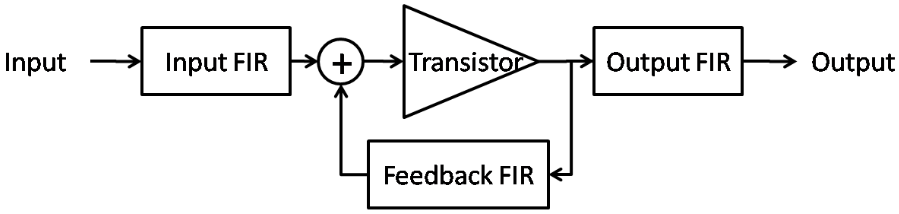

Figure 5.1: Band-Pass Model of a Power Amplifier

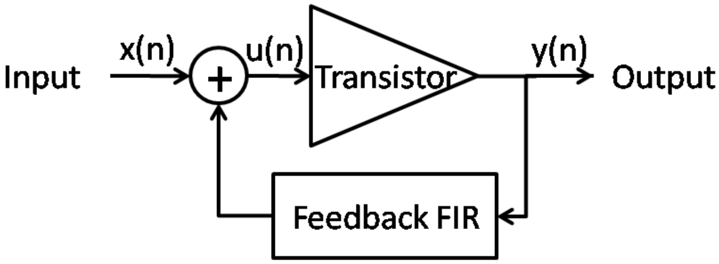

Figure 5.2: Low-Pass Equivalent Model of a Power Amplifier

Despite their completeness, band-pass equivalent models have always been approximated by their low-pass counterparts, when dealing with behavioural modeling for predicting the DUT around the carrier or in the process of constructing adequate digital predistortion schemes. This is attributed to unavailability of the necessary measurement data needed to deduce the band-pass models and to the complexity of their parameter identification. Hence, attempts have focused on developing low-pass equivalent models that would capture the DUT behaviour around the carrier.

Defining signals $y(n)$ and $x(n)$ as the output and input signals of the circuit, respectively, the nonlinear PA behaviour is formulated in Equation 5.1:

$$\begin{cases} y\left(n\right) & = & g\left(u\left(n\right)\right) \\ u\left(n\right) & = & x\left(n\right)+f\left(y\left(n\right)\right) \end{cases} \tag{5.1}$$

where $g(.)$ is the static nonlinearity function of the transistor and $f(.)$ is the function defining the feedback path.

Using these two expressions in Equation 5.1, the general form is written in Equation 5.2:

$$y\left(n\right)=h\big[x(n),y(n-1),\cdots,y(n-Q)\big] \tag{5.2}$$

where $h(.)$ represents a dynamic nonlinearity, and Q designates the memory depth of the output signal.

Equations 5.1 and 5.2 form the basis for determining the ANN structure. These equations suggest the use of an RNN structure; however, to avoid the training complexity incurred by this type of network, a corresponding feed-forward structure is derived.

Figure 5.3 corresponds to the PA model shown in Figure 5.2, where nonlinear AFs (NL) are used to mimic the transistor's static nonlinear behaviour and linear AFs (L) are utilized to incorporate the feedback mechanism.

A few transformations are applied to convert the RNN structure of Figure 5.3 to a feed-forward one, for which the BPLA can be used in the training phase. The first transformation applied to the ANN is outlined in Figure 5.4. Previous samples of the output signal, $y(n)$, are fed back to the input layer through buffers. The resulting ANN is a pure feed-forward structure with signals traveling only from the input layer to the output layer. However, this structure is not a typical feed-forward structure, in that non-adjoining layers are connected as shown in Figure 5.4. Hence, an additional transformation is still needed.

The second transformation consists of adding a neuron to the first layer through which the input signal samples, $x(n)$, are reported to the second hidden layer. The connection weight between the $x(n)$ signal and the first neuron is equal to 1, so that the input signal is not altered, as shown in Figure 5.5. Now that all the signals flow in one forward direction and te neurons of one layer are connected only to neurons of adjacent ones, the ANN can be considered as a typical feed-forward network, such as shown in Figure 5.5.

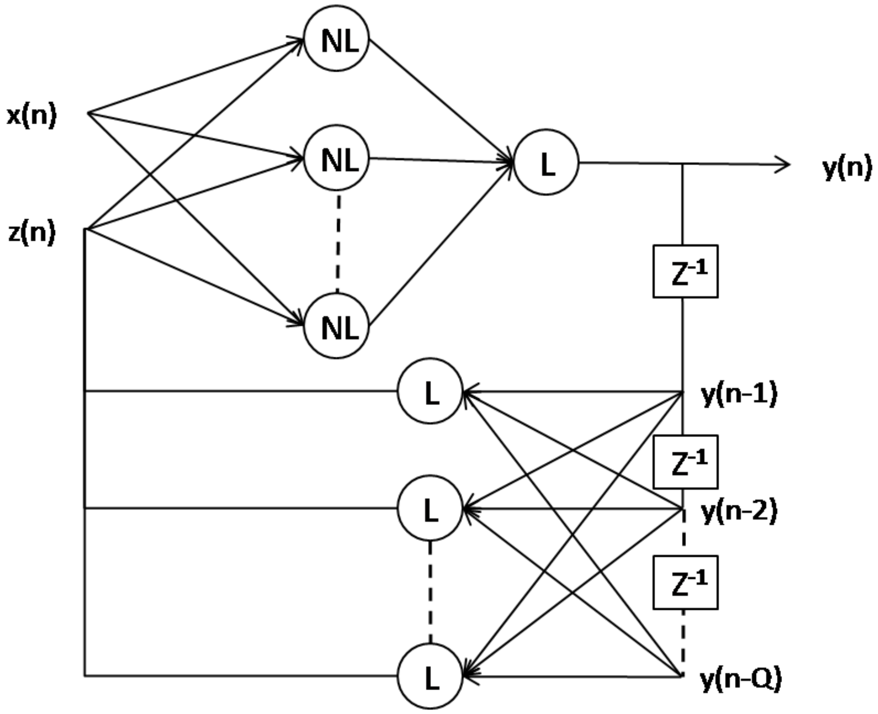

Figure 5.3: Recurrent Neural Network Based Model for a Power Amplifier with Memoryless Input

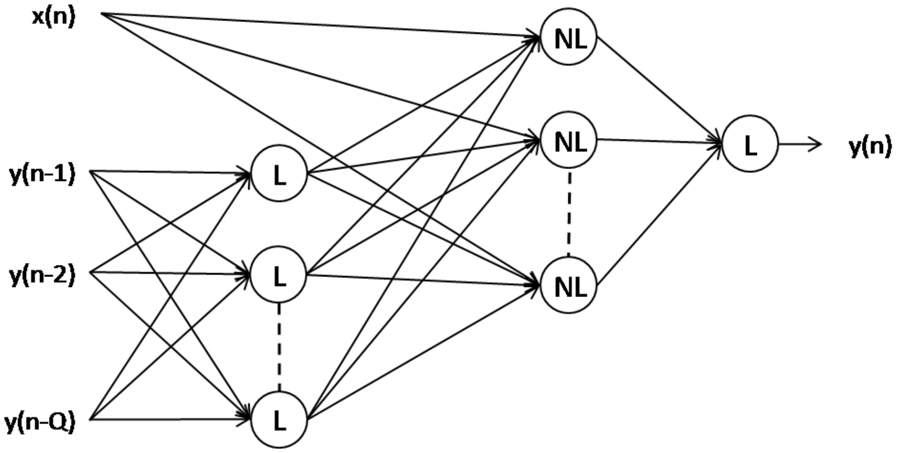

Figure 5.4: Feed-forward ANN Based Model for a Power Amplifier with Memoryless Input

As shown in Figure 5.5, the input layer now consists of both buffered samples of the output signal and the current sample of the input signal. The AFs of the first hidden layer

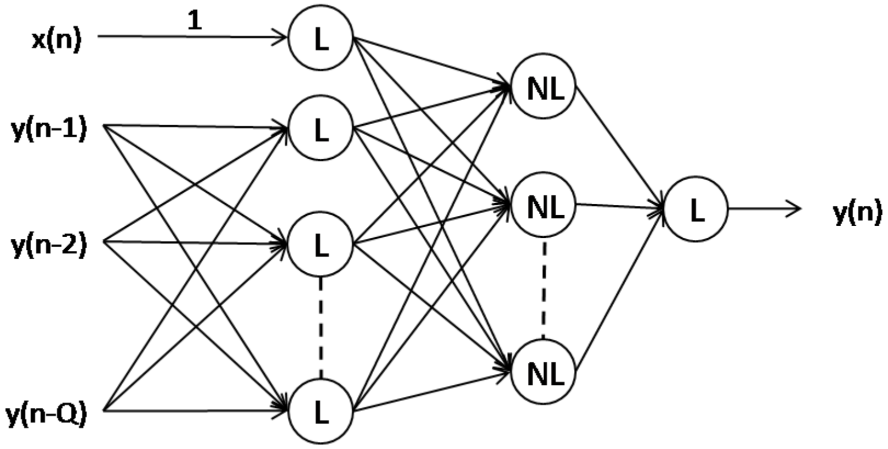

Figure 5.5: Multilayer ANN Based Model for a Power Amplifier with Memoryless Input

are the linear implementation of the feedback FIR filter. The number of neurons in this layer is $Q + 1$, where Q designates the memory depth of the output signal. Neurons of the second hidden layer consist of a nonlinear AF. The number of neurons on the second hidden layer depends on the degree of the static nonlinearity of the PA. Therefore, the stronger the PA nonlinearity is, the larger the number of neurons that are needed in the nonlinear layer. Therefore, to achieve the same modeling performance, highly nonlinear PAs require a structure with a higher number of neurons, compared to that required for a lower nonlinearity order.

To account for the MEs of the input signals, a set of neurons can be added to the input layer, as shown in Figure 5.6, and the resulting network will be a MLP that takes into account the contribution of MEs from the input and output signals, where P designates the memory depth of the input signal.

Figure 5.7 depicts the ANN structure, where the in-phase (I) and quadrature (Q) components of the input and output signals at the PA, (I_{in}, Q_{in}) and (I_{out}, Q_{out}), respectively, are used, resulting in real-valued networks.

Now that the structure of the MLP is determined, the next crucial step in accurate modeling is the training, which is described in the following section.

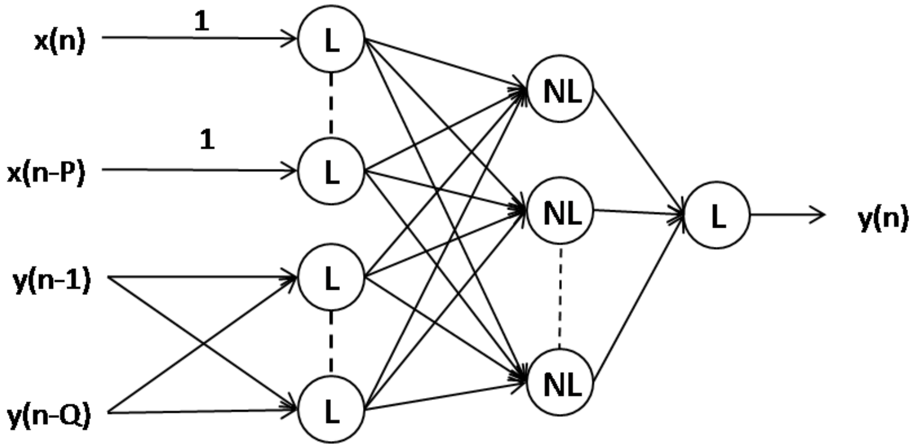

Figure 5.6: Multilayer Perceptron Artificial Neural Network Based Model for a Power Amplifier with Memory Input

5.2 Two Hidden Layers Artificial Neural Network Training

A 4C WCDMA signal, with a PAPR of 7.41dB, and synthesized in ADS, was applied to a LDMOS 250 Watt PEP Doherty PA, designated as the first PA in Section 5.3. The measured input and output envelope signals were used to train the 2HLANN structure with the BPLA available in MATLAB.

Following the cross-validation technique [19], the training and validation of the ANN were performed using different segments of the input and output signals to ensure the generality of the model. Hence, 10,000 samples were used for the training phase while a different set of 20,000 samples served in the validation phase. In addition, another set of data, different from the ones used during the training and validation phases, was used for the testing of the capacity of the model in predicting the output of the PA. The same training and validation sets were used later for comparison with MP, RVTDNN and memoryless polynomial models. It is worth mentioning that the training for the 2HLANN model was done using the measured past samples of $y(n)$ values; however, during the validation, the newly computed past values of $y(n)$ were used to demonstrate the correctness of the model.

The model validation approach suggested in [65] was used, since it puts the emphasis on the capability of the modeling scheme in predicting the MEs. A memoryless digital predistortion function was initially applied to the 4C WCDMA signal, and its output was

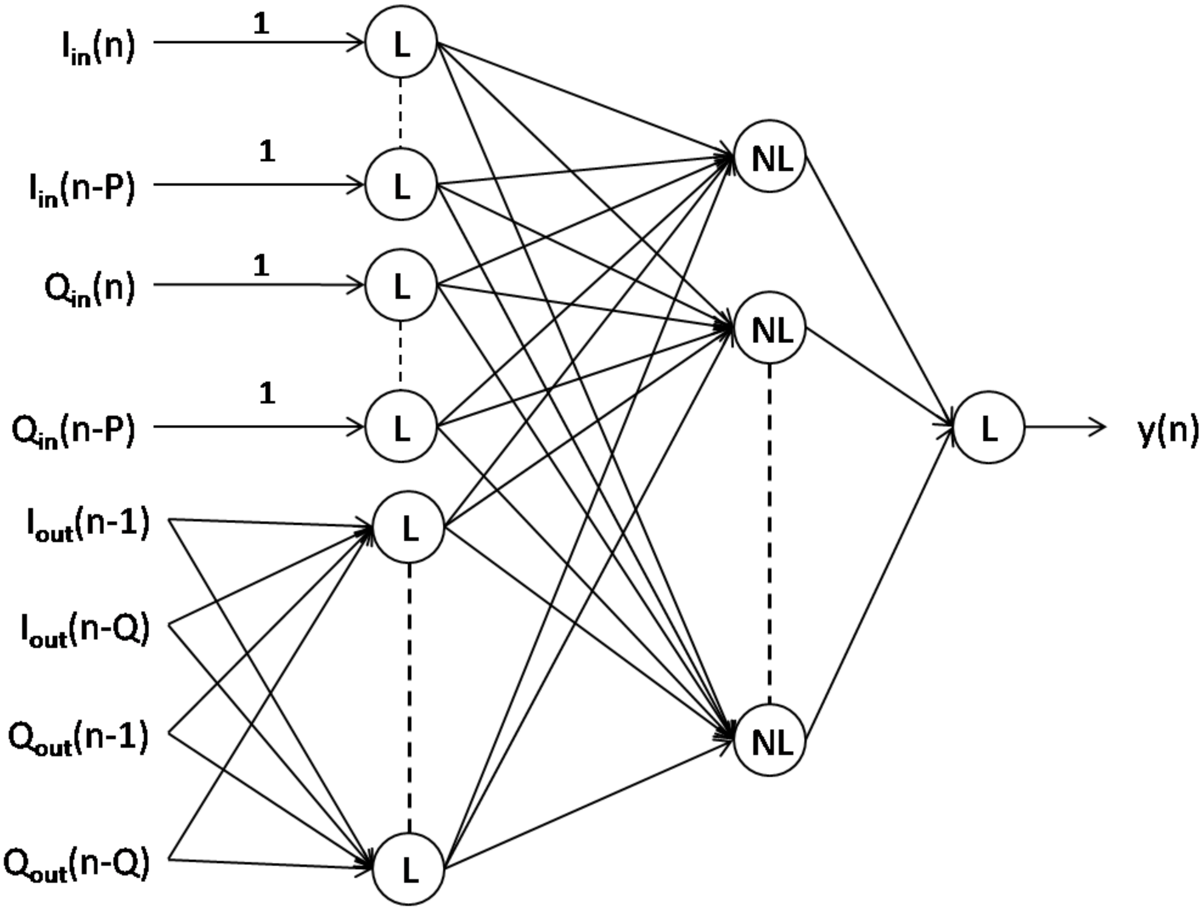

Figure 5.7: Multilayer Perceptron Artificial Neural Network Based Model for a Power Amplifier

used to drive the RF PA under test. Hence, the focus is in assessing the capability of the models in predicting the residual out-of-band emission dominated by the MEs. The resulting PSD of the input and output signals of the DUT are shown in Figure 5.8.

As previously mentioned, the number of neurons, in the first layer, is determined by the number of the inputs $(1 + P + Q)$, which is directly related to the memory depth in the input and output signals. Furthermore, the number of neurons in the second layer is related to the order of the static nonlinearities' exhibited by the transistor. The determination of its value can be achieved separately, by finding the required number of neurons required by a memoryless ANN model to predict the static nonlinearity of the DUT. Hence, the only remaining unknown parameters of the proposed model are P and Q, which denote

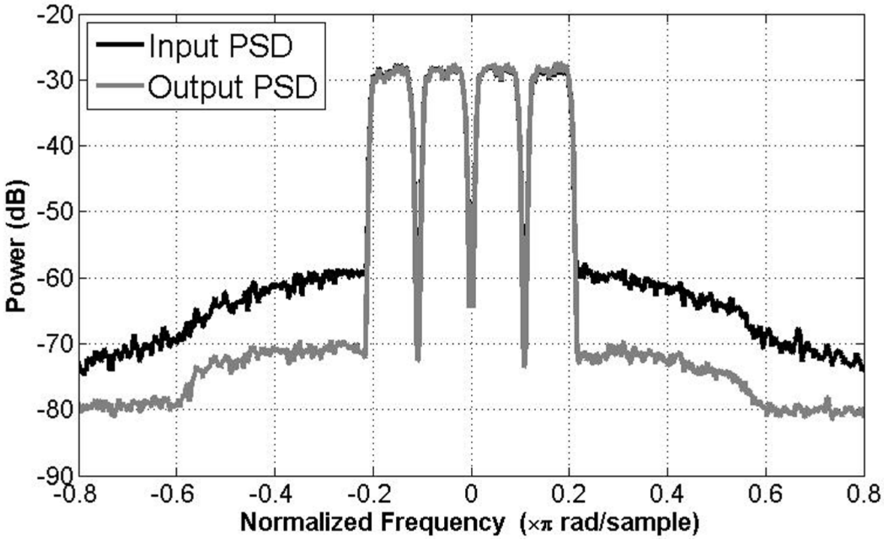

Figure 5.8: Power Spectral Density of the Power Amplifier's Input and Output

the input and output memory depths, respectively. This can be achieved by assessing the modeling accuracy, i.e. NMSE, for number of P and Q, as will be shown later on.

Various parameters pertinent to the structure of the ANN and its training algorithm were adjusted to identify an optimal combination.

Table 5.1 recapitulates the ANN model performance as a function of input and output memory depths, corresponding to number values of P and Q, respectively. Based on Table 5.1, one can conclude that higher memory depth does not imply higher performance. Hence, the case where $P = 0$ and $Q = 4$ was chosen as the optimal setting, as it yields the lowest NMSE versus minimum complexity. The result of $P = 0$ was an expected result from physical model of the PA given in Figure 5.2. In fact, the physical model states that the PA behaviour is a function only of the current input signal and the previous output signals, and this assumption was validates experimentally, as mentioned in Table 5.1.

Table 5.2 corresponds to the model performance when different numbers of neurons was used in the linear layer. One can observe the performance degradation if more than 5 neurons are used. Again, this result was expected from the physical model of the PA. In fact, as previously mentioned, the number of neurons, in the first layer, is determined by the number of the inputs and outputs used for modeling $(1 + P + Q)$. On the other hand, P and Q were found to be equal respectively to 0 and 4. Thus the expected number of

Table 5.1: 2HLANN Model Performance vs. Input Output Memory Depth

Memory Depth	Normalized Mean Square Error
P = 0 Q = 1	-37.77
P = 0 Q = 2	-37.13
P = 0 Q = 3	-37.37
P = 0 Q = 4	**-39.23**
P = 0 Q = 5	-37.30
P = 1 Q = 1	-34.17
P = 1 Q = 2	-34.40
P = 1 Q = 3	-34.79
P = 1 Q = 4	-36.98
P = 1 Q = 5	-37.07
P = 2 Q = 1	-33.08
P = 2 Q = 2	-33.38
P = 2 Q = 3	-33.98
P = 2 Q = 4	-35.5
P = 2 Q = 5	-35.32
P = 3 Q = 1	-33.06
P = 3 Q = 2	-32.53
P = 3 Q = 3	-33.53
P = 3 Q = 4	-35.63
P = 3 Q = 5	-35.1
P = 4 Q = 1	-32.6
P = 4 Q = 2	-34.04
P = 4 Q = 3	-34.59
P = 4 Q = 4	-34.66
P = 4 Q = 5	-35.22
P = 5 Q = 1	-32.99
P = 5 Q = 2	-33.20
P = 5 Q = 3	-34.63
P = 5 Q = 4	-34.99
P = 5 Q = 5	-35.59

neurons in the first layer is 5 ($5 = 1+P+Q = 1+0+4$) which was validated experimentally through Table 5.2.

Table 5.3 corresponds to the model performance when different numbers of neurons were used in the nonlinear layer. According to Table 5.3, one can observe the performance

Table 5.2: 2HLANN Model Performance vs. Number of Linear Neurons

Number of Linear Neurons	Normalized Mean Square Error
5	**-37.79**
10	-37.33
14	-36.73
18	-37.69
20	-35.26

degradation if more than 25 neurons are used. This phenomenon is attributed to the over modeling problem [28]. In fact, ANN modeling where the needed neurons are exceeded or are not met degrades the ANN performance. Henceforward, 25 neurons will be used.

Table 5.3: 2HLANN Model Performance vs. Number of Nonlinear Neurons

Number of Nonlinear Neurons	Normalized Mean Square Error
5	-33.10
10	-34.99
15	-35.87
20	-35.67
25	**-39.42**
30	-38.96

Table 5.4 summarizes the performance of the 2HLANN model, in terms of NMSE, depending on the type of the AF in the linear and nonlinear hidden layers. According to Table 5.4, the best modeling accuracy is obtained when the AF in the first hidden layer is “satlins” and the nonlinear AF in the second hidden layer is “tansig”. These AFs will be used hereafter.

Table 5.4: 2HLANN Model Performance vs. Activation Functions

Activation Functions		Normalized Mean Square Error
L hidden layer	NL hidden layer	
Satlins	**Sigmoid**	**-43.05**
Satlins	Logsig	-41.91
Purelin	Sigmoid	-39.42
Purelin	Logsig	-39.47

Table 5.5 depicts the proposed model performance depending on the learning rate of the BPLA, where both constant and dynamic rates are used. Based on Table 5.5, one can conclude that a moderated learning rate equal to 0.1 leads to a good trade-off between accuracy and convergence rate. In addition, when the learning rate was set to be dynamic, a slight improvement was seen, as can be observed in Table 5.5.

Table 5.5: 2HLANN Model Performance vs. Learning Rate

Learning rate	Normalized Mean Square Error
Constant lr = 0.2	-42.67
Constant lr = 0.1	**-43.1**
Constant lr = 0.05	-42.87
Constant lr = 0.01	-43.09
Constant lr = 0.005	-42.70
Inc. rate = 1.05 Dec. rate = 0.9 Initial value = 0.1	-43.1
Inc. rate = 1.05 Dec. rate = 0.8 Initial value = 0.1	**-43.55**
Inc. rate = 1.05 Dec. rate = 0.7 Initial value = 0.1	-42.97
Inc. rate = 1.05 Dec. rate = 0.6 Initial value = 0.1	-42.70

Table 5.6 depicts the proposed model performance depending on the momentum term of the BPLA, where both constant and dynamic rates are used. Based on Table 5.6, one can conclude that a momentum term equal to 0.1 leads to a good trade-off between accuracy and convergence rate. In addition, when the momentum term is dynamic, a slight improvement is seen, as can be observed in Table 5.6.

Table 5.6: 2HLANN Model Performance vs. Momentum Term

Momentum Term	Normalized Mean Square Error
Constant mr = 0.7	-41.73
Constant mr = 0.6	**-42.25**
Constant mr = 0.5	-42.16
Inc. rate = 1.05 Dec. rate = 0.9 Initial value = 0.6	-43.98
Inc. rate = 1.05 Dec. rate = 0.8 Initial value = 0.6	-44.41
Inc. rate = 1.05 Dec. rate = 0.7 Initial value = 0.6	**-43.05**
Inc. rate = 1.05 Dec. rate = 0.6 Initial value = 0.6	-44.25

As discussed in the previous chapter, the learning algorithm and ANN performance can be improved by proper choices of the ANN structure and learning parameters. The structure of the ANN that yields to the 2HLANN model was inspired from the physical behaviour f PAs; therefore, the network structure was optimized. This was proven through Tables 5.1 and 5.2, where the memory depth and the number of neurons in the linear layer were found experimentally to be equal to the ones predicted by the PA behaviour.

For the learning algorithm, a set of experiences were performed to find the best learning rate and the momentum term. Also, based on the study presented in the previous chapter, a set of heuristics were chosen to make the BPLA perform better. In fact, a proper normalization of the input neuron was set in such a way that the nonlinear neurons in the second hidden layer do not go into saturation. In addition, a couple of BPLA versions were tried, and the most suitable one for PA modeling was chosen. Finally, the training data were chosen to cover the PA behaviour over all possible input powers. A large set was used, approximately $10,000$.

This extensive study of the ANN structure and learning parameters yielded a powerful 2HLANN model that accurately models the PA behaviour and results in a good linearizer, as it is shown in the next Section. To further demonstrate the necessity of this analysis of the ANN parameters, the behaviour of nonlinear neurons in the second hidden layer is drawn in Figures 5.9 and 5.10, in the cases of a PA with non-optimized and optimized parameters, respectively.

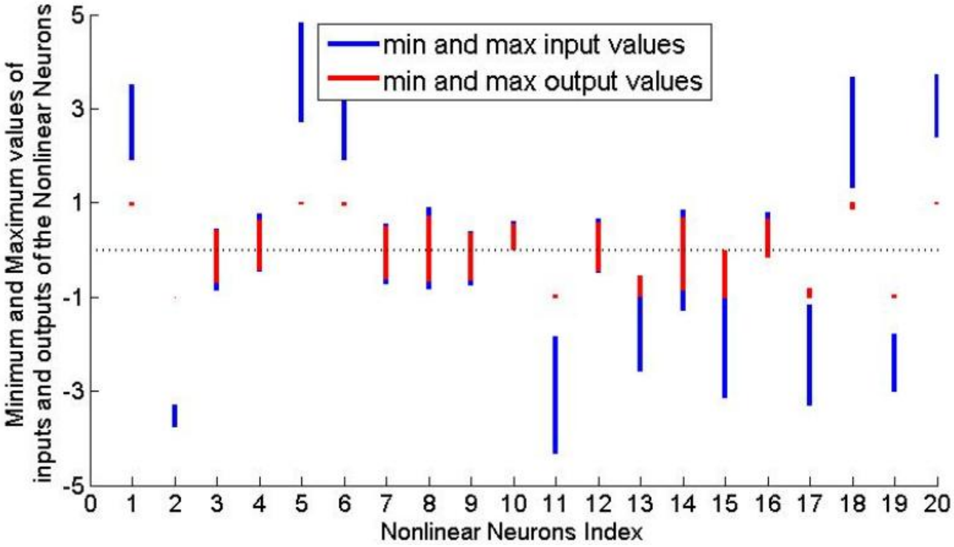

Figure 5.9: Nonlinear Neurons Response When the ANN Parameters Are Not Optimized

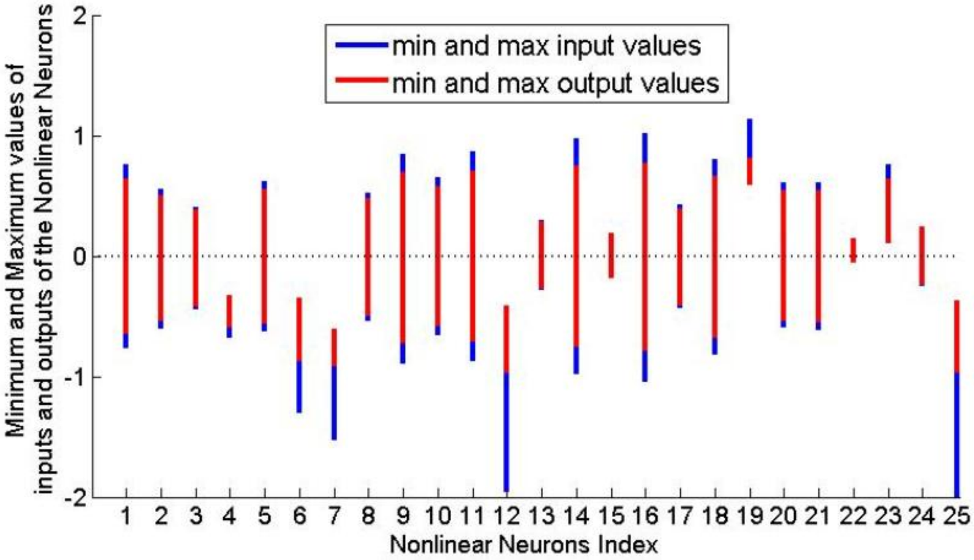

Figure 5.10: Nonlinear Neurons Response When the ANN Parameters Are Optimized

In the case where the network parameters were not set properly, it is clear from Figure 5.9 that most of the nonlinear neurons are saturated and, therefore, yield to a constant response at their outputs. Thus, their functionality are restricted to modeling a constant response; and, the overall ANN will fail in accurately modeling the dynamic nonlinear behaviour of the PA. On the other hand, it is clear from Figure 5.10, that when the parameters of the ANN were chosen through experimental studies or through the physical behaviour of the PA, almost all the nonlinear neurons use their maximum capabilities by performing along their nonlinear region. In fact, in the optimized selection of the network parameters, all nonlinear neurons participate in the modeling of the dynamic nonlinearity; and, unlike the non-optimized parameters, there exists no constant response.

5.3 Two Hidden Layers Artificial Neural Network Forward Validation

To train the proposed 2HLANN in Figure 5.7, the number of neurons in the first and second layers was set to 5 and 25, respectively. Figure 5.11 shows the performance of the 2HLANN during the training phase, where the goal or tolerated error was set to -70 dB for the training samples. The performance of the ANN improves with the number of training epochs; however, overtraining the network can lead to erroneous parameters and, consequently, compromise the accuracy and generality of the ANN. Hence, a validation step was added to detect the training point at which the training error started deteriorating and stop the ANN training. Figure 5.11 shows the training with the validation step; therefore, the overtraining problem is avoided.

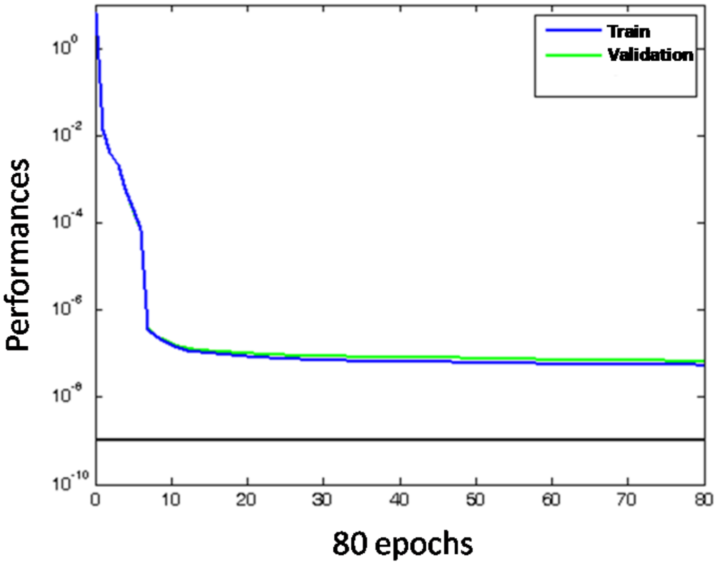

Figure 5.11: Training Performance of the Proposed 2HLANN Model

To assess the accuracy of the trained 2HLANN model in predicting the behaviour of the PA under test a number of tests were conducted. In a first step, the in-phase and quadrature components of the constructed model output signal were compared to the measured ones, as shown in Figure 5.12. The model demonstrated good fitting of the measurement data, indicating that the 2HLANN structure is capable of mimicking actual DUT behaviour. Furthermore, Figures 5.13 and 5.14 compare the measured and modeled AM/AM and AM/PM characteristics and the capacity of the 2HLANN model in producing a similar scattering as those exhibited by the measurement data. It is clear that the model has the ability to fit both the nonlinear and linear distortions, even for the low input signal power range.

Figure 5.15 shows the PSD of the measured and 2HLANN modeled output signals. It is clear that the 2HLANN model accurately predicted the measured PSD. It is worth mentioning that, since a memoryless predistortion was already applied a priori to the input signal, the proposed model allowed for excellent prediction of the MEs behind the residual out-of-band spectrum emission in the measured signal. To further highlight the capability

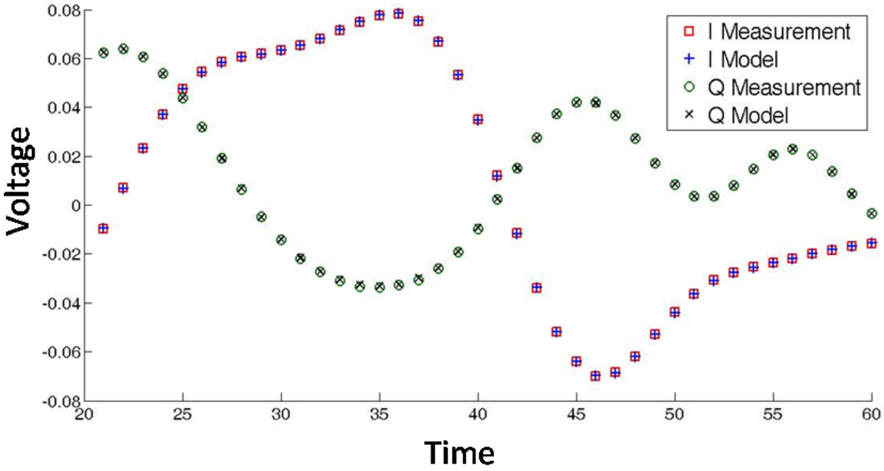

Figure 5.12: Measured and 2HLANN Modelled Output In-Phase and Quadrature Components

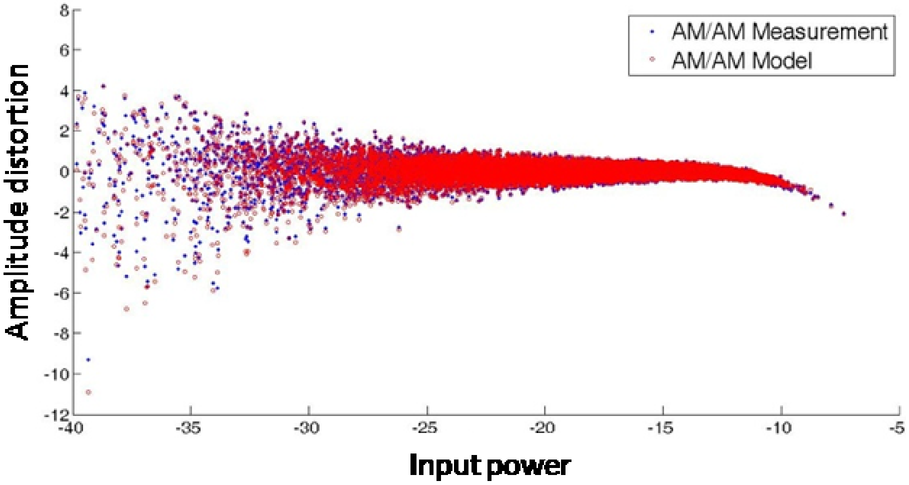

Figure 5.13: Measured and 2HLANN Modeled AM/AM Distortion

of the proposed model, the error PSD between the model and the measurement is shown in Figure 5.16

To further evaluate the performance of the 2HLANN, its modeling accuracy was compared to that of three proven modeling schemes, namely memoryless polynomial model with 9^{th} degree of nonlinearity, a MP of 9^{th}-order nonlinearity and memory depth of 8,

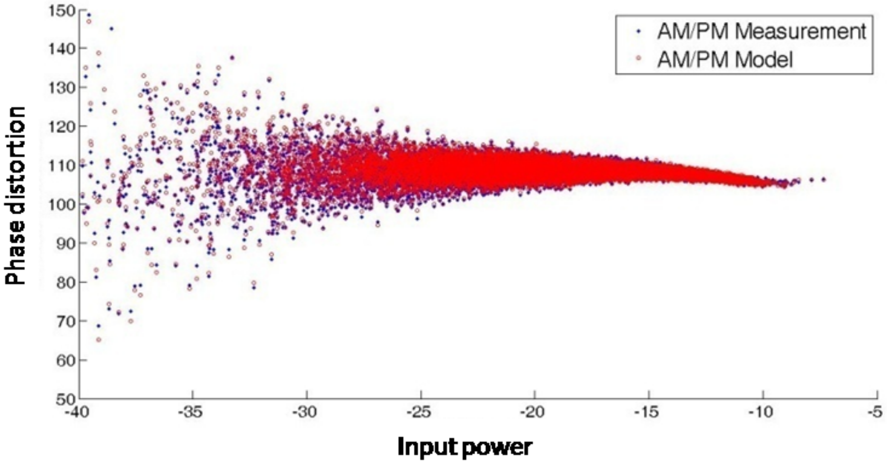

Figure 5.14: Measured and 2HLANN Modeled AM/PM Distortion

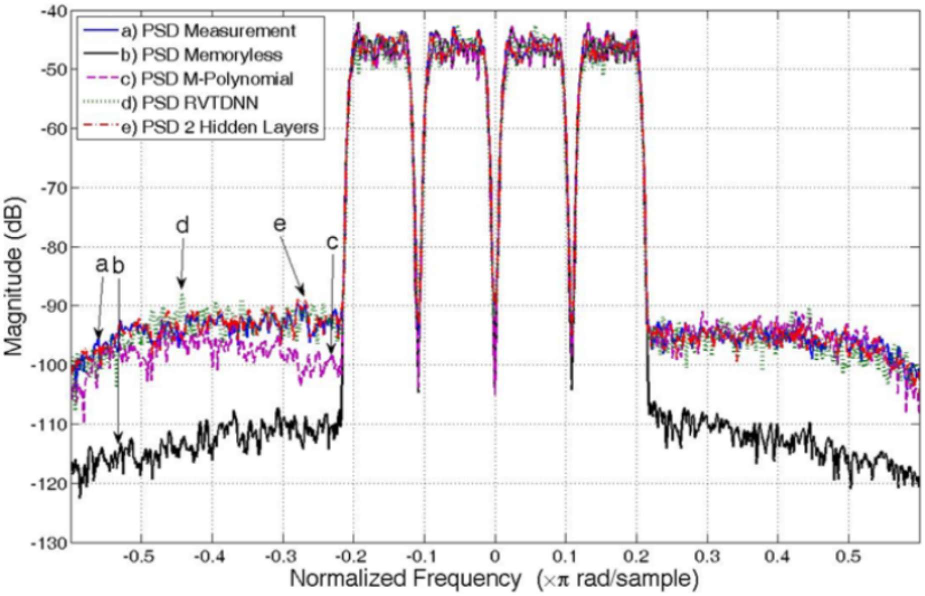

Figure 5.15: Comparison of the Power Spectrum Densities for the First PA for the (a) Measured, (b) Memoryless Polynomial, (c) Memory Polynomial, (d) RVTDNN, and (e) 2HLANN Models

and a RVTDNN with 20 neurons and 8 tapped inputs. It is worth mentioning that models involved in this comparison were closely analyzed, so that the configurations that lead to their best performance were retained. The predicted output signals obtained out of the four different models are all plotted in Figure 5.15. Since the three models with mem-

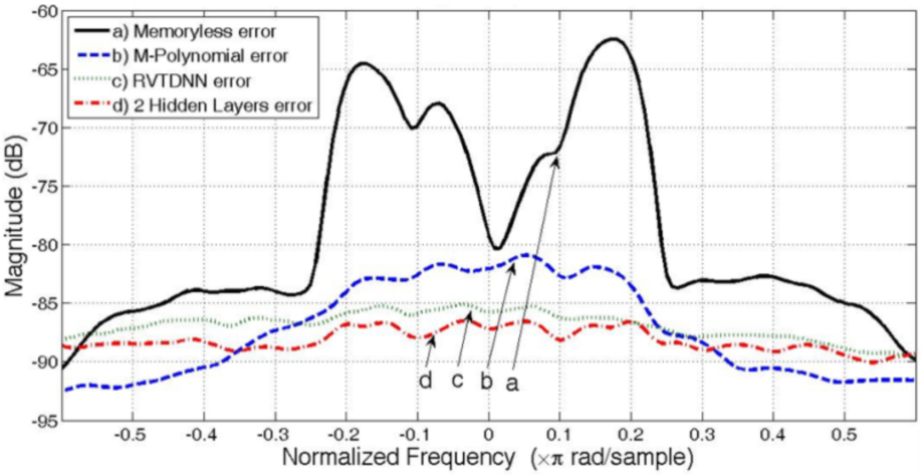

Figure 5.16: Comparison of the Normalized Error Power Spectrum Densities of the First PA for (a) Memoryless Polynomial, (b) Memory Polynomial, (c) RVTDNN, and (d) 2HLANN models

ory achieved very similar PSDs, the PSD of the error signals between the modeled and measured signals was drawn and is shown in Figure 5.16.

It is clear that the 2HLANN model outperformed the MP in the in-band region, where it allows for a reduced error of more than 5dB. However, the proposed model shows about an increased error of 2 to 3dB in the out-of-band region, compared to the MP. The 2HLANN model also outperformed RVTDNN model, as its error spectrum exhibited a decreased error of a few dB. This superiority can be attributed to its physical underpinnings and the more suitable ANN structure. Table 5.7, which reports the NMSE of the four models, confirms the superior modeling accuracy of the proposed model.

Table 5.7: 2HLANN Model vs. Memoryless Polynomial, MP, and RVTDNN Models

Model Name	Normalized Mean Square Error		
	1st PA	2nd PA	
		1st Biasing Point	2nd Biasing Point
Memoryless Polynomial	-28.3342	-31.5073	-34.7672
MP	-42.0965	-40.4349	-38.0825
RVTDNN	-42.7307	-41.0545	-38.4205
2HLANN	**-43.2855**	**-43.5653**	**-42.1142**

To further demonstrate the capability and the generality of the new 2HLANN model, a 130W Class AB PA, designated the second PA, which was biased in the two different conditions of a class AB close to A (Idq=0.9A) and a deeper class AB (Idq=0.55A), was also used as a DUT. Figures 5.17 and 5.18 show the comparisons of the error spectra of the 2HLANN model with the RVTDNN, memoryless polynomial and MP models for the second PA at the first and second biasing conditions, respectively. As demonstrated for the first PA, the 2HLANN model again outperformed the other models, by allowing for better modeling accuracy of the PSD in the in-band and out-of-band regions. According to Table 5.7, the new model lowered the NMSE by about 3dB, when compared to the RVTDNN or the MP models.

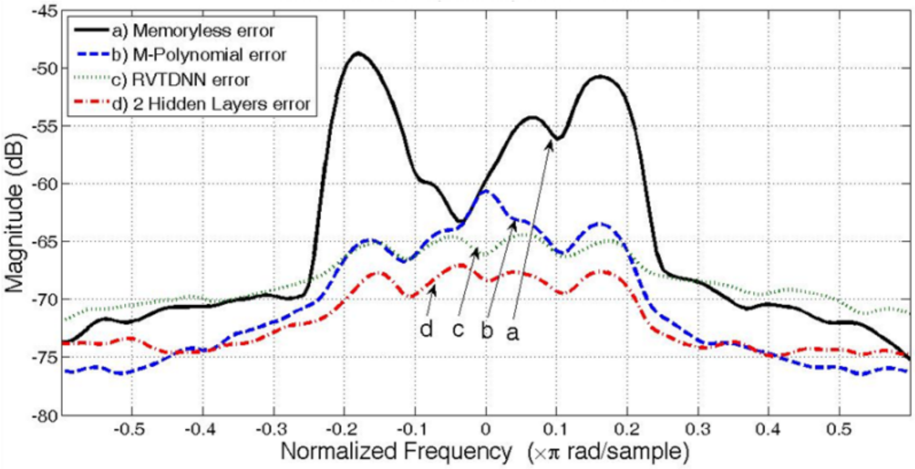

Figure 5.17: Comparison of the Normalized Error Power Spectrum Densities of the Second PA with the First Biasing Point for (a) Memoryless Polynomial, (b) Memory Polynomial, (c) RVTDNN and (d) 2HLANN Models

The validation of the proposed model was conducted in two steps. First, its excellent modeling accuracy was indicated through (i) the negligible error spectrum when predicting the measured output spectrum of the two PAs under test both driven with wideband signals, (ii) the very small NMSE (about -43dB), and (iii) the good agreement between the measured and modeled AM/AM and AM/PM characteristics. Second, it was demonstrated that, by combining the appropriate structure and excellent modeling accuracy of ANN, the 2HLANN model outperformed the well established modeling approaches, namely the MP and RVTDNN models.

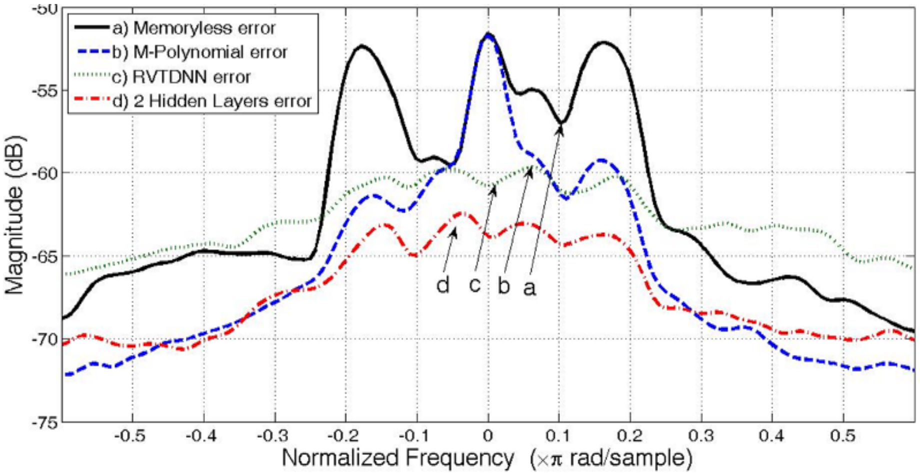

Figure 5.18: Comparison of the Normalized Error Power Spectrum Densities of the Second PA with the Second Biasing Point for (a) Memoryless Polynomial, (b) Memory Polynomial, (c) RVTDNN, and (d) 2HLANN models

5.4 Two Hidden Layers Artificial Neural Network Linearization Capability

In the previous section, the 2HLANN model demonstrated excellent accuracy in predicting the SDPA response, noted as the first PA in the previous section. This section assesses the 2HLANN performance when used as a DPD. Its linearization capability is also compared to that of three types of linearization schemes.

Figures 5.19 and 5.20 depict the linearized output spectrum of the previously used SDPA and an Asymmetrical Doherty PA (ADPA), both driven with a 20MHz 1001 WCDMA signal. Compared to the SDPA, the ADPA uses 150 W high power LDMOS FET devices from Freescale Semiconductor, Inc., MRF7S21150H, for the peaking amplifier, while the main amplifier was kept the same. This special arrangement was used to enhance the power efficiency, as explained in [30].

Figures 5.19 and 5.20 show the significant reduction in the out-of-band spectrum emission. The 2HLANN allowed for an ACPR of higher than 50dBc at the three frequency offsets (10MHz, 15MHz and 20MHz). According to Figure 5.20, the 2HLANN DPD outperformed the three other linearizers. In the particular case of the ADPA, which exhibited stronger nonlinearity, the 2HLANN DPD succeeded in mitigating the sources of distortions, where the other schemes suffered from limited linearization capabilities. This is

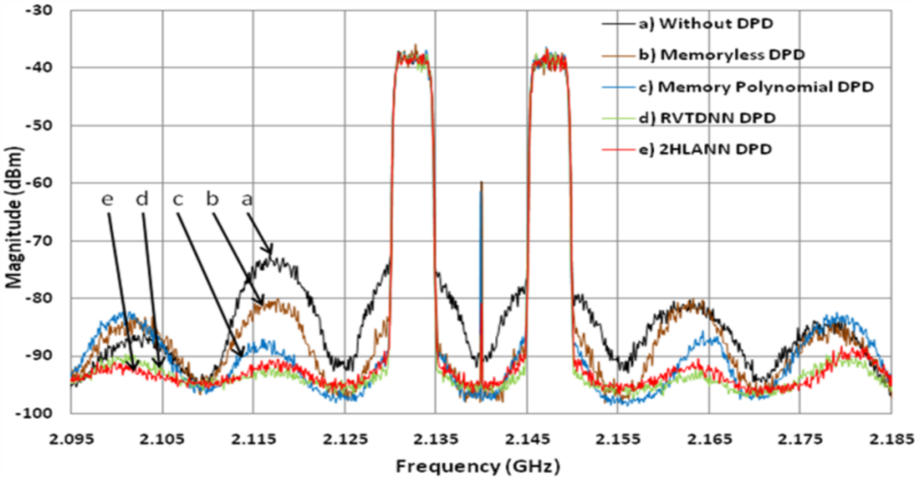

Figure 5.19: SDPA Output Driven with the 20MHz 1001 WCDMA Signal: (a) without DPD and with (b) Memoryless, (c) Memory Polynomial, (d) RVTDNN and (e) 2HLANN DPDs

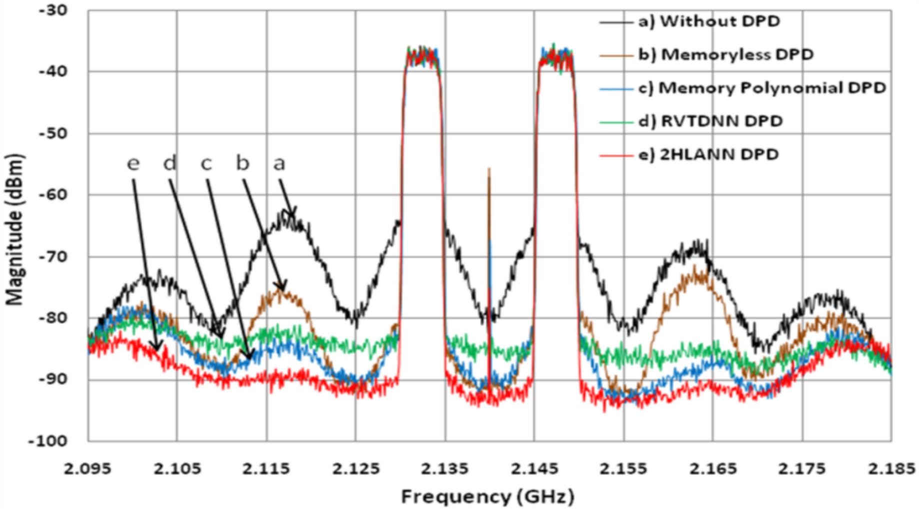

Figure 5.20: ADPA Output Driven with the 20MHz 1001 WCDMA Signal: (a) without DPD and with (b) Memoryless, (c) Memory Polynomial, (d) RVTDNN and (e) 2HLANN DPDs

mainly attributed to the completeness of the 2HLANN that was inspired from the PA model presented in Figure 5.2.

Figure 5.21 presents a comparison of the linearization capabilities between the 2HLANN and the three other DPD schemes, when the SDPA is driven with a typical 4C WCDMA model. The 2HLANN model maintained a comparable linearization capability. Therefore, the 2HLANN was able to maintain excellent linearization capabilities in both scenarios (1001 and 4C WCDMA signals); whereas, the conventional linearization models presented limited capacity when the PA exhibited stronger nonlinearity and MEs.

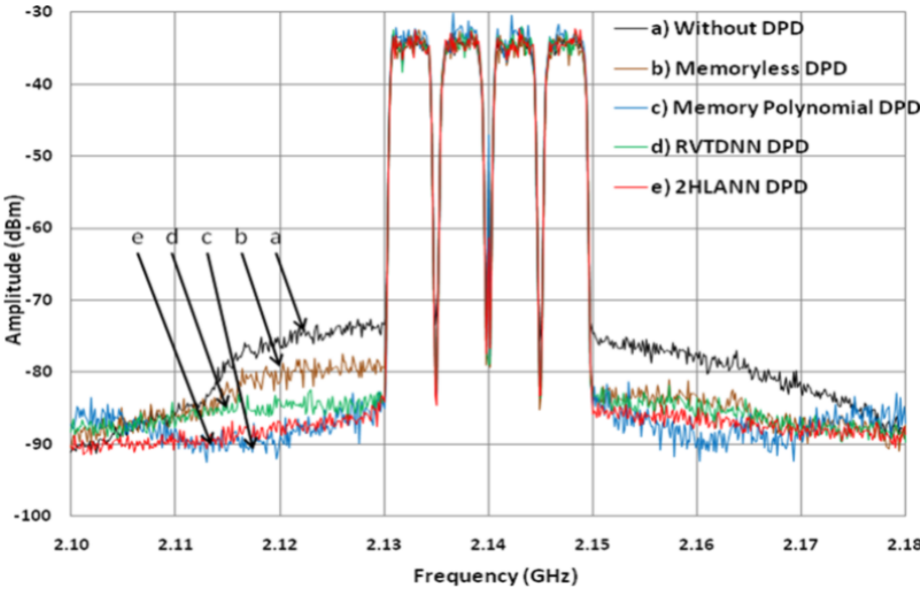

Figure 5.21: SDPA Output Driven with the 20MHz 4C WCDMA Signal: (a) without DPD and with (b) Memoryless, (c) Memory Polynomial, (d) RVTDNN and (e) 2HLANN DPDs

Table 5.8 illustrates the amount of power back-off required for the DUTs without linearization to achieve the same ACPR as the linearized DUTs. It is shown that the DUTs without linearization achieved a similar linearity at a back-off of about 10dB with a significant degradation in the drain efficiency of about 30%.

Table 5.8: Drain Efficiency Comparison when DPD and when no DPD is Applied

DUT and Signal Type	Drain Efficiency DPD ON	Drain Efficiency DPD OFF
SDPA & 4C WCDMA	35.5%	11.7% (9dB back-off)
SDPA & 1001 WCDMA	40.3%	11.0% (10.4dB back-off)
ADPA & 1001 WCDMA	47.5%	11.2% (12dB back-off)

The 2HLANN model was successfully applied in the linearization of the two Doherty PAs driven with a two different WCDMA signals. The model demonstrated an excellent linearization capability by allowing for an ACPR greater than 50dBc over the three different frequencies offsets (10MHz, 15MHz and 20MHz). This linearity improvement significantly increased the effective linear output power and, consequently, the power efficiency of both Doherty PAs.

Chapter 6

Conclusion and Future Works

Conclusion

The main goal of this thesis was the development of a general method to generate spectrally pure signals using a model based on predistortion. The technique developed allows the Power Amplifier (PA) to operate in its nonlinear region, thereby significantly increasing its efficiency. As seen in the last chapter, an improvement of about 30% was achieved when Digital Pre-Distortion (DPD) was applied. The efficiency gain translates into electricity and cooling cost savings for service providers and longer battery life for mobile terminal users. Memory Effects (MEs), which are exhibited by highly nonlinear PAs or PAs with wideband signals, were taken into account, thus increasing the DPD performance and allowing for better linearity with higher efficiency when compared to other models.

In this thesis, an overview of the most common system-level solutions to the problems of PA linearization was presented. Theories about model-based predistortion, nonlinear behaviour and intermodulation products were reviewed, in order to explain and design the model. Characteristics of the PA were obtained, using Artificial Neural Network (ANN) identification. The new model technique used in this thesis offers simplicity and the lowest cost, compared to analogue techniques or such complex models as the Volterra series.

In order to validate this study's model, MATLAB and Advanced Design System (ADS) simulations were conducted, along with real measurements of PAs, confirming that the model works well. Some of the results of these simulations have been presented in this thesis to show the accurate model obtained when using ANNs, instead, for example, the memory polynomial model.

Future Works

With the development of new PA topologies, linear and nonlinear MEs are becoming more dominant in the PA response. In fact, with broadband modulated signals, classical techniques, such as memoryless polynomial, Memory Polynomial (MP) and Wiener/Hammerstein models, are no longer sufficient for accurate PA behavioural modeling and for suitable DPD linearizer construction. On the other hand, ANN and Volterra series architectures present superior results compared to the MP model when the PA exhibits important MEs. This leads to the conclusion that these two models present mechanisms that leads to a better representation of the PA behaviour over the MP models. In fact, the Volterra series and ANNs, when approximated by polynomial functions, show the presence of cross terms, which are likely behind the superior results when the PA exhibits significant MEs. An investigation on the effect of cross terms is thus essential to determining the effect and importance of these terms compared to the models that do not include these terms.

The investigation could consist of a systematic methodology to identify the weights of the different cross terms and the most important terms. This step can be based on experimentation and/or theoretical analysis that determine the necessity and influence in the absence or presence of various cross terms. However, identification of these cross term coefficients will not be an easy task. In fact, nonlinear identifications can be difficult and costly, when the number of coefficient is significant.

Several algorithms and approaches are present in the literature for this problem, but the complexity of these methods grows exponentially when the number of coefficient increases. On the other hand, an interesting idea could be to compare the ANN structure and the Volterra series with the goal of identifying mirroring parameters. In fact, since ANN involves cross terms, as is the case in the Volterra series, we can link the ANN weights and structure to the corresponding terms in the Volterra series. Afterwards, an identification of the ANN model will result in the identification of the Volterra series? coefficients. Moreover, thanks to the powerful Back Propagation Learning Algorithm (BPLA) developed for ANN, the identification of the Volterra series will be much easier and efficient than the classic nonlinear identification.

Finally, the success of any research work depends on its implementability on physical devices. Thus, the next step in my work, in addition to the new directions presented above, is the implementation of the 2HLANN DPD and the other previously published DPDs in an Field-Programmable Gate Array (FPGA) evaluation platform. Therefore, a more detailed and realistic validation of my research can be conducted.

References

[1] A. Ahmed, E.R. Srinidhi, and G. Kompa. Efficient PA modeling using neural network and measurement setup for memory effect characterization in the power device. *IEEE MTT-S International Microwave Symposium Digest*, Jun. 12 – Jun. 17 2005. 3, 30, 40, 41

[2] E. Ballesteros, F. Perez, and J. Perez. Analysis and design of microwave linearised amplifiers using Active Feedback. *IEEE Transaction on Microwave Theory and Techniques*, 36(3):499–504, Mar. 1988. 24

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Belmont, MA, Athena Scientific, 1996. 62

[4] S. Boumaiza. Advanced Memory Polynomial Linearization Techniques. *IEEE MTT 2009 International Microwave Symposium Workshop, Anaheim, CA*, 2009. 39

[5] S. Boumaiza and F. Mkadem. Wideband RF Power Amplifier Predistortion using Real-Valued Time-Delay Neural Networks. *European Microwave Conference, EuMC2009*, pages 1449–1452, Sept. 29 – Oct. 1 2009. 3, 30, 40, 41

[6] Carlevarino, R. Martinotti, G. Metta, and G. Sandini. An incremental growing neural network and its application to robot control. *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 5, 2000. 61

[7] H. H. Chen, C. S. Maa, Y. C. Wang, and J. T. Chen. Dynamically Optimum Lookup-Table Spacing for Power Amplifier Predistortion Linearization. *IEEE Transaction on Microwave Theory and Techniques*, 54(5):2118–2127, May 2006. 2, 30

[8] D. L. Chester. Why two hidden layers are better than one. *in Proceeding International Joint Neural Networks Conference, Washington, DC*, pages 265–268, Jan. 1990. 55, 59

[9] A. S. Ciminski. Recurrent neural networks usefulness in digital predistortion of power amplifiers. *15th International Conference on. Microwaves, Radar and Wireless Communications, MIKON2004*, 1:249–252, May 17 – May 19 2004. 3, 30, 40, 41

[10] T. R. Cunha, J. C. Pedro, and E. G. Lima. Low-Pass Equivalent Feedback Topology for Power Amplifier Modeling. *IEEE MTT-S International Microwave Symposium Digest*, pages 1445–1448, Jun. 15 – Jun. 20 2008. 43, 67, 68

[11] G. CYBENKO. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems, MCSS*, 2(4):303–314, Feb. 1989. 3, 30, 40, 55, 58

[12] Y. H. Fang, M.C.E. Yagoub, F. Wang, and Q.J. Zhang. A New Macromodeling Approach for Nonlinear Microwave Circuits Based on Recurrent Neural Networks. *IEEE Transaction on Microwave Theory and Techniques*, 48(12):2335–2344, Dec. 2000. 3, 30, 40, 41

[13] M. Faulkner. Amplifier linearization using RF feedback and feedforward techniques. *IEEE Transaction on Vehicular Techniques*, 47(1):209–215, Feb. 1998. 24

[14] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, Sept. 1989. 3, 30, 40, 55, 58

[15] S. I. Gallant. *Neural Network Learning and Expert Systems.* Cambridge, Mass. : MIT Press, 1993. 65

[16] S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4:1–58, 1992. 60

[17] P. Gilabert, G. Montoro, and E. Bertran. On the Wiener and Hammerstein models for power amplifier predistortion. *Asia-Pacific Conference Proceedings on Microwave Conference Proceedings, APMC2005*, 2, Dec. 4 – Dec. 7 2005. 3, 30, 39

[18] Y. Hayashi, M. Sakata, and S. I. Gallant. Multi-layer versus single-layer neural networks and an application to reading hand-stamped characters. *Proceedings of the International Neural Network Conference INNC1990*, pages 781–784, Jul. 9 – Jul. 13 1990. 60

[19] S. Haykin. *Neural Networks: A Comprehensive Foundation, 2nd edition.* Prentice Hall, 1999. 3, 40, 41, 43, 48, 55, 57, 60, 61, 62, 63, 64, 72

[20] K. Hornik, M. Stinchcombe, and H. White. Multilayer Feedforward Neural Networks Are Universal Approximators. *Neural Networks*, 2(5):359–366, May 1989. 3, 30, 40, 55, 58

[21] D.R. Hush and W.G. Horn. Progress in Supervised Neural Networks. *IEEE Signal Processing Magazine*, 10:8–39, 1993. 60

[22] M. Isaksson and D. Wisell. Extension of the Hammerstein model for power amplifier applications. *63rd ARFTG Conference Digest Spring*, pages 131–137, June 11 2004. 30

[23] M. Isaksson, D. Wisell, and D. Ronnow. Wide-Band Dynamic Modeling of Power Amplifiers Using Radial-Basis Function Neural Networks. *IEEE Transaction on Microwave Theory and Techniques*, 53(11):3422–3428, Nov. 2005. 3, 30, 40, 41

[24] M. Isaksson, D. Wisell, and D. Ronnow. A Comparative Analysis of Behavioral Models for RF Power Amplifiers. *IEEE Transaction on Microwave Theory and Techniques*, 54(1):348–359, Jan. 2006. 24

[25] F. M. Gannouchi J. S. Cardinal. A new Adaptive Double Envelope Feedback (ADEF) lineariser for mobile radio power amplifiers. *IEEE MTT-S International Microwave Symposium Digest*, 2:573–576, May 23 – May 27 1994. 24

[26] J. Kang, D. Yu, K. Min, and B. Kim. A Ultra-High PAE Doherty Amplifier Based on 0.13μm CMOS Process. *IEEE Microwave and Wireless Components Letters*, 16(9):505–507, Sept. 2006. 1

[27] S. Kang and I. Lee. Analysis and design of feedforward power amplifiers. *IEEE MTT-S International Microwave Symposium Digest*, pages 1519–1522, Jun. 1997. 24

[28] F. O. Karray and C. De Silva. *Soft Computing and Intelligent Systems Design : Theory, Tools and Applications.* Addison Wesley, 2004. 3, 40, 41, 43, 48, 55, 57, 60, 61, 76

[29] P.B. Kenington, R.J. Wilkinson, and J.D. Marvill. Broadband linear amplifier design for a PCN basestation. *41st IEEE Vehicular Technology Conference, Gateway to the Future Technology in Motion*, pages 155–160, May 19 – May 22 1991. 24

[30] J. Kim, B. Fehri, S. Boumaiza, and J. Wood. Power Efficiency and Linearity Enhancement using Optimized Asymmetrical Doherty Power Amplifier. *submitted to IEEE Transaction on Microwave Theory and Techniques.* 85

[31] J. Kim and K. Konstantinou. Digital Predistortion of Wide-Band Signals based on Power Amplifier Model with Memory. *Electronic Letters*, 37(23):1417–1418, Nov. 2001. 3, 30, 37

[32] D. F. Kimball, J. Jeong, C. Hsia, P. Draxler, S. Lanfranco, W. Nagy, K. Linthicum, L.E. Larson, and P. M. Asbeck. High-Efficiency Envelope-Tracking W-CDMA Base-Station Amplifier Using GaN HFETs. *IEEE Transaction on Microwave Theory and Techniques*, 54(11):3848–3856, Nov. 2006. 1

[33] K. Konstantinou and D. K. Paul. Analysis and design of broadband high-efficiency feedworward amplifiers. *IEEE MTT-S International Microwave Symposium Digest*, 2:867–870, Jun. 17 – Jun. 21 1996. 24

[34] H. Ku and J. S. Kenney. Behavioral Modeling of Nonlinear RF Power Amplifiers Considering Memory Effects. *IEEE Transaction on Microwave Theory and Techniques*, 51(12):2495–2504, Dec. 2009. 40

[35] V. Kurkova. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5:501–506, 1992. 59

[36] C. L. Larose and F.M. Ghannouchi. Optimization of feedforward amplifier power efficiency on the basis of drive statistics. *IEEE Transaction on Microwave Theory and Techniques*, 51(1):41–54, January 2003. 24

[37] Y. LeCun. Efficient Learning and Second-order Methods. *A Tutorial at NIPS 93*, 1993. 61, 62, 65

[38] Y. LeCun, J. S. Denker, and S. Solla. Optimal Brain Damage. *Advances in Neural Information Processing Systems in Touretzky, David (Eds)*, 1990. 61

[39] J. Liebetreu. Proposed System Impairment Models. *Cover Sheet for Presentation to IEEE 802.16 Broadband Wireless Access Working Group*, 1:00–15, March 2000. 31

[40] T. Liu, S. Boumaiza, and F. M. Ghannouchi. Dynamic Behavioral Modeling of 3G Power Amplifiers Using Real-Valued Time-Delay Neural Networks. *IEEE Transaction on Microwave Theory and Techniques*, 52(3):1025–1033, Mar. 2004. 3, 30, 40, 41

[41] T. Liu, S. Boumaiza, and F. M. Ghannouchi. Augmented Hammerstein Predistorter for Linearization of Broad-Band Wireless Transmitters. *IEEE Transaction on Microwave Theory and Techniques*, 54(4):1340–1349, April 2005. 3, 30, 39

[42] K. Madani. Reducing the intermodulation distortion in multi-carrier microwave power amplifiers. *Symposium on High Performance Electron Devices for Microwave and Optoelectronic Applications, EDMO1999*, pages 153–157, Nov. 22 – Nov. 23 1999. 29

[43] C. Maziere, A. Soury, E. Ngoya, and J. M. Nebus. A system level model of solid state amplifiers with memory based in a nonlinear feedback loop principle. *European Microwave Conference*, 1:4–6, Oct. 2005. 43, 67, 68

[44] N. Messaoudi, M. C. Fares, S. Boumaiza, and J. Wood. Complexity Reduced Odd-Order Only Memory Polynomial Pre-distorter for Multi-Carrier Doherty Power Amplifier Linearization. *IEEE MTT-S International Microwave Symposium Digest*, pages 419–422, Jun. 15 - Jun. 20 2008. 13, 32, 39

[45] F. Mkadem, M. B. Ayed, S. Boumaiza, J. Wood, and P. Aaen. Behavioral Modeling and Digital Predistortion of Power Amplifiers with Memory using Two Hidden Layers Artificial Neural Networks. *IEEE MTT-S International Microwave Symposium Digest*, May 23 – May 28 2010. 3, 4, 30, 40

[46] F. Mkadem and S. Boumaiza. Behavioral Modeling of Power Amplifier with Memory using Two Hidden Layers Artificial Neural Network. *submitted to IEEE Transaction on Microwave Theory and Techniques.* 3, 4, 30, 40, 41, 56, 60

[47] F. Mkadem and S. Boumaiza. Extended Hammerstein Behavioral Model Using Artificial Neural Networks. *IEEE Transaction on Microwave Theory and Techniques*, 57(4):745–751, Apr. 2009. 3, 30, 39, 40

[48] F. Mkadem and S. Boumaiza. Extended Hammerstein Model for RF Power Amplifier Behavior Modeling. *European Microwave Conference, EuMC2009*, pages 1066–1069, Sept. 29 – Oct. 1 2009. 3

[49] F. Mkadem and S. Boumaiza. Power Amplifier Linearization using Two Hidden Layers Artificial Neural Network. *IEEE Topical Symposium on Power Amplifiers for Wireless Communication*, (Sept. 14 – Sept. 15), 2009. 3, 4, 40

[50] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26:394–395, 1920. 33

[51] B. O'Brien, J. Dooley, and T.J. Brazil. RF Power Amplifier Behavioral Modeling using a Globally Recurrent Neural Network. *IEEE MTT-S International Microwave Symposium Digest*, pages 1089–1092, Jun. 11 – Jun. 16 2006. 3, 30, 40, 41

[52] H. Ozbay. *Introduction to Feedback Control Theory.* CRC Press, 1999. 24

[53] J. C. Pedro and S. A. Maas. A Comparative Overview of Microwave and Wireless Power-Amplifier Behavioral Modeling Approaches. *IEEE Transaction on Microwave Theory and Techniques*, 53(4):1150–1163, Apr. 2005. 24

[54] R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955. 33

[55] V. Petrovic. Reduction of spurious emission from radio transmitters by means of modulation feedback. *IEE Conference On Radio Spectrum Conservation Techniques*, pages 44–49, Sept. 1983. 24

[56] D. C. Plaut, S. J. Nowlan, and G. E. Hinton. Experiments on Learning by Back Propagation. *Technical Report: CMU-CS-86-126, Carnegie-Mellon University*, 1986. 63

[57] J. B. Prolla. *Weierstrass-Stone, the theorem* . Frankfurt am Main ; New York : P. Lang, 1993. 58

[58] N. Safari, T. Roste, P. Fedorenko, and J. S. Kenney. An Approximation of Volterra Series Using Delay Envelopes, Applied to Digital Predistortion of RF Power Amplifiers With Memory Effects. *Microwave and Wireless Components Letters*, 18(2):115–117, Feb. 2008. 3, 30

[59] A. M. Saleh. Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers. *IEEE Transaction on Communication*, 29:1715–1720, Nov. 1981. 2, 30, 31

[60] M. Schetzen. *Volterra and Wiener Theories of Nonlinear Systems.* John Wiley & Sons, 1980. 35

[61] H. Seidel. A feedforward experiment applied to an L-4 carrier system amplifier. *IEEE Transaction on Communication Technology*, 19(3):320–325, Jun. 1971. 24

[62] H. Songbai, Y. Xiaohuan, and B. Jingfu. Applications of Feed-forward Neural Networks to WCDMA Power Amplifier Model. *Asia-Pacific Conference Proceedings Microwave Conference Proceedings, APMC2005*, 5, Dec 4 – Dec. 7 2005. 3, 30, 40

[63] A. G. Stegmayer and O. Chiotti. Volterra NN-based behavioral model for new wireless communications devices. *Neural Computing and Applications*, 18(3):283–291, Apr. 2009. 3, 30, 40, 41

[64] J. T. Stonick, V. L. Stonick, J. M. F. Moura, and R. S. Zborowski. Memoryless Polynomial Adaptive Predistortion. *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95*, 2:981–984, May 1995. 2, 30

[65] T. Taijun, S. Boumaiza, and F. M. Ghannouchi. Deembedding static nonlinearities and accurately identifying and modeling memory effects in wide-band RF transmitters. *IEEE Transaction on Microwave Theory and Techniques*, 53(11):3578–3587, Nov. 2005. 46, 72

[66] Agilent Technologies. *Advanced Design System.* `http://www.home.agilent.com/agilent/product.jspx?cc=US&lc=eng&ckey=1297113&nid=-34346.0.00&id=1297113`. 44

[67] J. Vuolevi. Analysis, measurement and cancellation of the bandwidth and amplitude dependence of intermodulation distortion in RF power amplifiers. *PhD Dissertation at the Department of Electrical Engineering, University of Oulu*, 2001. 3

[68] J. Vuolevi and T. Rahkonen. *Distortion in RF Power Amplifiers*. Artech House, 2003. 3, 14, 30, 35

[69] R. Wilkinson and A. Bateman. Linearisation of Class C amplifiers using Cartesian Feedback. *IEEE 1989 Workshop on Mobile and Cordless Telephone Communications*, pages 62–66, Sept. 1989. 24

[70] J. Wood, M. LeFevre, D. Runton, J.C. Nanan, B. H. Noori, and P. H. Aaen. Envelope-Domain Time Series (ET) Behavioral Model of a Doherty RF Power Amplifier for System Design. *IEEE Transaction on Microwave Theory and Techniques*, 54(8):3163–3172, Aug. 2006. 3, 30, 40, 41

[71] G. Xu, M. Li, and Y. Xi. Radial basis function neural network models for power-amplifier design. *International Conference on Communications, Circuits and Systems, ICCCAS2004*, 2:1066–1070, Jun. 27 – Jun. 29 2004. 3, 30, 40, 41

[72] Y. Ye, T. Liu, X. Zeng, and J. He. Generalized Hammerstein-based Dynamic Nonlinear Behavior Models for Wideband RF Transmitters. *International Conference on Wireless Communication, Networking and Mobile Computing, WiCom2007*, pages 684–687, Sept. 21 – Sept. 25 2007. 3, 30

[73] A. Zhu and T. J. Brazil. RF Power Amplifier Behavioral Modeling Using Volterra Expansion with Laguerre Function. *IEEE MTT-S International Microwave Symposium Digest*, Jun. 2005. 3, 30

[74] A. Zhu, J. C. Pedro, and T. J. Brazil. Dynamic Deviation Reduction-Based Volterra Behavioral Modeling of RF Power Amplifiers. *IEEE Transaction on Microwave Theory and Techniques*, 54(12):4323–4332, Dec. 2006. 3