

Fixed-Point Digital Predistortion System for Nonlinear High Power Amplifiers

Namrata Dwivedi

Department of Electronics and Communication Engineering

5 June, 2014

Thesis Committee:

Dr. Vivek Ashok Bohara (Chair)

Dr. Mazen Abi Hussein

Dr. Mohammad S. Hashmi

A thesis submitted to Indraprastha Institute of Information
Technology in partial fulfillment of the requirement for the
degree of Master of Technology

Abstract

Radio Frequency (RF) High Power Amplifiers (HPAs) are one of the basic building blocks of modern wireless communication system. But most of these broadband wireless communication systems such as Universal Mobile Telecommunications System (UMTS) and Long Term Evolution -Advanced (LTE-Advanced) employ transmission formats such as wideband code division multiple access (WCDMA) or orthogonal frequency division multiplexing (OFDM) which have high peak-to-average power ratio (PAPR). The HPAs generally operate close to the saturation region to attain maximum efficiency, however when driven with signals having high PAPR and wide bandwidth the PA might cross over to the saturation region causing out-of-band distortions (resulting into adjacent channel interference) and in-band distortions (increase in bit error rate of the receiver). Digital Predistortion (DPD) with its high implementation flexibility has emerged as a low cost high performance alternative for the linearization of power amplifiers in the past few years. DPD includes a functional block element prior to the PA which has an inverse characteristic to that of the PA such that the overall PD-PA combination is a linear one.

With the growth of wireless systems, energy usage and costs continue to increase. As a result there is an increased focus on energy efficient green radio communications. For low transmission powers, in order to achieve a noteworthy gain in power efficiency of the overall transmitter, the computational complexity of the utilized predistortion algorithms has to be kept as low as possible. Consequently, the use of fixed point arithmetic based implementation is desirable if not indispensable. In this work, we analyze the effects of fixed point implementation on DPD system. Unlike the floating point implementation, in fixed point implementation the digital predistorter and the coefficient estimation algorithm are implemented in fixed point arithmetic. We quantify the impact of this fixed point implementation on the overall performance of the digital predistorter system so that we can achieve good linearity performance with minimum number of bits for data, coefficients and arithmetic operations. The performance of the proposed fixed point digital predistorter system is evaluated in terms of adjacent channel power ratio (ACPR) and error vector magnitude (EVM) at the output of PA when a Long Term Evolution-Advanced (LTE-Advanced) signal is applied at the input.

Acknowledgements

As I draw my M.Tech Thesis to a close, I feel extremely fortunate to have met so many skilful, accomplished and proficient people during the past two years. It feels really great having been guided by some of the top names in academia. I am highly thankful for their support and encouragement.

Firstly, I would like to thank my advisor, Assistant Professor Dr. Vivek Ashok Bohara for giving me the generous opportunity to work under him. His constant support, encouragement and belief in my abilities is the sole factor behind the successful completion of this work. I would also like to thank him for being so patient and understanding throughout this period. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. His dedication towards his own work and the perfection that he demands has always been very inspiring for me.

Next, I would like to express my heart-felt thanks to Assistant professors Mazen Abi Hussein and Olivier Venard, Systems Engg. Dept., ESIEE, Paris for their insightful suggestions and comments which helped me understand the topic even better and enrich my thought process. I am deeply grateful to them for taking out time from their respective schedules and providing their invaluable feedback whenever I approached them. I wish to express my extended thanks to Mazen sir for accepting to be the external examiner in my thesis committee.

I would also like to thank Assistant professor Dr. Mohammad S. Hashmi for accepting to serve as the internal examiner on my thesis evaluation committee and for his constructive comments and effort. I have learnt a lot from his courses in early semesters.

My sincere appreciation is due to all my professors at IIITD who were a great source of motivation all this while. Their teachings and constant guidance kept me going. Last but not the least, I would like to thank my family and friends for their unconditional love and support in all my endeavors.

Contents

Acknowledgments	3
List of Abbreviations	3
1 Introduction	5
1.1 Motivation	5
1.2 Objectives	7
1.3 Outline	7
2 Power Amplifier Behavioral Models	9
2.1 Overview	9
2.2 Polynomial Models	10
2.2.1 Memory polynomial	10
2.2.2 Generalized Memory Polynomial	11
2.2.3 Two-Dimensional Memory Selective Polynomial	11
2.3 Two-Box and Three-Box Models	12
2.3.1 Hammerstein	13
2.3.2 Wiener	13
2.3.3 Parallel Wiener	14
2.3.4 Wiener-Hammerstein	14
2.3.5 Hammerstein-Wiener	15
2.4 Conclusion	15
3 Indirect Learning Architecture (ILA) for DPD	17
3.1 Overview	17
3.1.1 Adaptive Digital Baseband Predistortion	18
3.2 Learning Architectures	18
3.2.1 Direct Learning Architecture (DLA)	19
3.2.2 Indirect Learning Architecture (ILA)	20
3.3 Simulation results	22
3.3.1 AM/AM and AM/PM Characteristics	22
3.3.2 Spectral Density	23
3.3.3 ACPR and EVM	24
4 Fixed Point DPD System based on ILA	25
4.1 Overview	25

4.2	FXP DPD System	26
4.2.1	Coefficient Extraction Procedure	27
4.3	Simulation results	28
4.3.1	Spectral Density	28
4.3.2	ACPR	29
4.3.3	EVM	31
4.3.4	Comparitive Analysis	31
5	Conclusions and Future Work	34
5.1	Future Work	35
	Bibliography	38

List of Figures

1.1	PA Nonlinearity	6
2.1	Memory Polynomial Model	10
2.2	Hammerstein Model	13
2.3	Wiener Model	14
2.4	Parallel Wiener Model	14
2.5	Wiener-Hammerstein Model	15
2.6	Hammerstein-Wiener Model	15
3.1	DPD-PA cascade combines two nonlinear systems into one linear result	17
3.2	Adaptive Digital Baseband Predistortion	18
3.3	Direct Learning Architecture - DLA	19
3.4	Indirect Learning Architecture - ILA	20
3.5	Flowchart for QR decomposition based on Modified Gram Schmidt algorithm	21
3.6	AM/AM Characteristic	22
3.7	AM/PM Characteristic	23
3.8	Power Spectral Density Plot	23
3.9	ACPR Performance	24
3.10	EVM Performance	24
4.1	DPD system for PA linearization	27
4.2	The spectral regrowth suppression performance with 16,18 and 20 bit data word lengths for Wiener Model	29
4.3	The spectral regrowth suppression performance with 16,18 and 20 bit data word lengths for WH PA	29
4.4	ACPR Performance for Wiener PA: System level Iterations	30
4.5	ACPR Performance for WH PA: System level Iterations	30
4.6	EVM Performance for Wiener PA: System level Iterations	31
4.7	EVM Performance for WH PA: System level Iterations	32

List of Tables

3.1	COMPUTATIONAL COMPLEXITY	22
4.1	PARAMETERS FOR FXP IMPLEMENTATION	28
4.2	WIENER MODEL	33
4.3	WIENER-HAMMERSTEIN MODEL	33

List of Abbreviations

2-D MSP	Two-Dimensional Memory selective Polynomial
ACI	Adjacent Channel Interference
ACPR	Adjacent Channel Power Ratio
ADC	Analog-to-digital converter
AM/AM	Amplitude to amplitude distortion
AM/PM	Amplitude to phase distortion
ARM	Advanced RISC Machines
ASSP	Application Specific Standard Product
CC	computational complexity
DAC	Digital-to-analog converter
DLA	Direct Learning Architectures
DPD	Digital Predistortion
DSP	Digital Signal Processor
EVM	Error Vector magnitude
FCC	Federal Communications Commission
FIR	Finite Impulse Response
FL	Fractional Length
FLP	Floating-point
FPGA	Field programmable gate array
FXP	Fixed-point
GaN	Gallium Nitride

GMP	Generalized Memory Polynomial
HPA	High Power Amplifiers
HW	Hammerstein-Wiener
ILA	Indirect Learning Architecture
LDMOS	Laterally diffused Metal oxide semiconductor
LMS	Least Mean Square
LS	Least Square
LTE	Long Term Evolution
LTI	Linear time invariant
MP	Memory Polynomial
OFDM	Orthogonal Frequency Division Multiplexing
PA	power amplifier
PAPR	Peak-to-average power ratio
PD	predistorter
PH	Parallel Hammerstein
RF	Radio Frequency
RISC	Reduced Instruction Set Computing
RLS	Recursive Least Square
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
WH	Wiener-Hammerstein
WL	Word Length

1 Introduction

1.1 Motivation

In the present wireless industry scenario, the increasing quest for higher data rates, spectral efficiency and the integration of voice and data services has made the RF spectrum invaluable. All the efforts are directed towards squeezing in as much data as possible in a given portion of the RF spectrum. This has led to the development of digital modulation schemes that are more bandwidth efficient and support increased transmission rates. Consequently, most of the broadband wireless communication systems today, such as Universal Mobile Telecommunications System (UMTS) and Long Term Evolution-Advanced (LTE-Advanced) employ transmission formats such as wideband code division multiple access (WCDMA) or orthogonal frequency division multiplexing (OFDM). The resulting baseband signals of these modulation schemes, have high peak-to-average power ratio (PAPR) and non-constant envelope. Since, these signals carry information in the signal amplitude, they are quite sensitive to nonlinear amplification as it directly affects the amplitude of the signal.

Conventional RF HPA is a basic building block of all the modern wireless communication systems. The HPAs generally operate close to the saturation region to attain maximum efficiency, however when driven with signals having high PAPR and wide bandwidth the power amplifier (PA) might cross over to the saturation region causing out-of-band and in-band distortions [1]. Out-of-band distortions result from adjacent channel interference (ACI) due to widening of the spectrum whereas in-band-distortions mainly refer to the degradation in Error Vector magnitude (EVM). Linearity in PAs can be achieved by operating the amplifier backed off from the saturation region so that the signal level is confined to the linear region, but this leads to reduced power efficiency. Thus, linearity in PAs must be achieved without sacrificing efficiency to a great extent.

This typical tradeoff between PA linearity and efficiency is illustrated in Fig. 1.1 The red curve shows the nonlinear behaviour of the PA. Initially the output power increases in proportion with the input power but after a certain point it tends to show nonlinearity. The desired linear response of the PA is illustrated by the linear gain curve. When the amplifier is operating in compression, the output versus input power curve falls below the Linear gain curve, hence, the actual output power of the PA is not sufficient for linear operation. This can be compensated by introducing the effect of expansion which is essentially the case of predistortion, such that the amplitude of the input signal is increased so that the desired output power (falling

on the linear gain curve) is achieved. The expansion effect can be observed where the input power resulting in "Pactual" is increased to "Pin needed to achieve Pdesired" so that the PA output power is raised to "Pdesired" which coincides with the Linear gain curve.

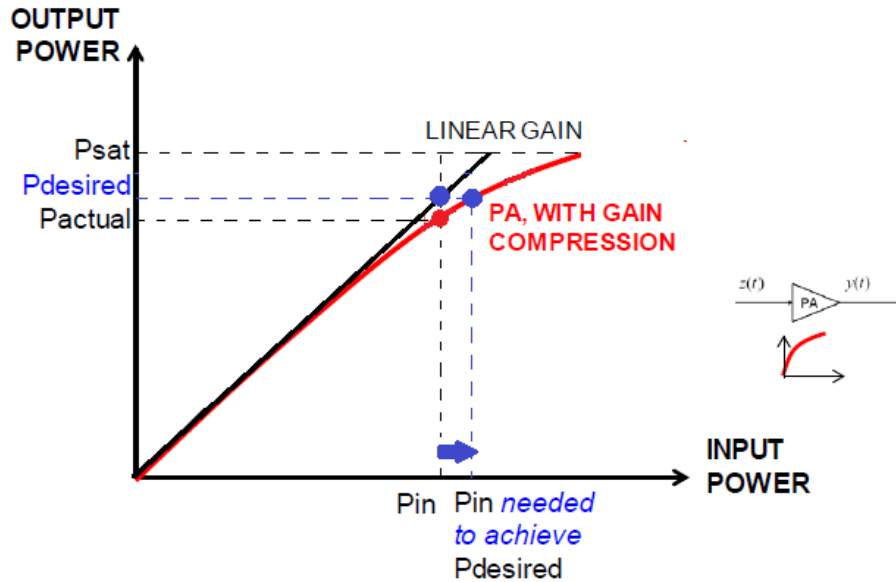


Figure 1.1: PA Nonlinearity

Various techniques have been proposed in literature for the linearization of PAs. Out of these, the three major ones include feedback [2], feedforward [3] and predistortion. In the feedback method, a correcting function is generated based on the complex input and output signal envelopes which is then applied to the input signal envelope to compensate for nonlinearities. In case of feedforward technique instead of feeding the difference signal to the input of PA, it is directly subtracted from the PA output. While Feedback technique suffered from the challenging stability issues, feedforward had its own constant analog delay problems [4]. Digital Predistortion (DPD) with its high implementation flexibility has emerged as a low cost high performance alternative for the linearization of PAs in the past few years. Digital Predistortion includes a functional block element prior to the PA which has an inverse characteristic to that of the PA such that the overall PD-PA combination is a linear one.

In the current industry scenario, digital hardware has gained significant attention for the efficient implementation of signal processing algorithms. Such digital hardware use numbers which can be represented in either fixed-point (FXP) or floating point (FLP) data types. Although, FLP processors can simplify the real time implementation of signal processing algorithms to a great extent, there is a pressing need for FXP hardware based microcontrollers or processors because they are better than their FLP counterparts in terms of power consumption and hardware complexity.

Consequently, FXP arithmetic based implementation is essential, as such a system can achieve better speed of computation and thus eases off hardware implementation.

The above points motivated us to dig deeper into the digital predistortion of RF PAs in order to alleviate the nonlinearities associated with them that lead to unwanted distortion of the signals, and propose a FXP DPD system with lesser power and area requirements that can be easily implemented on field programmable gate array (FPGA) or other hardware units.

1.2 Objectives

The main objectives of this thesis can be summarized as follows:

- Investigate the various behavioral models of PAs and outline their respective mathematical input-output relationship.
- Propose a FXP based DPD system and compare it with the conventional FLP DPD system through simulation results.
- Quantify the impact of this FXP implementation on the overall performance of the DPD system so that we can achieve good linearity performance with minimum number of bits for data, coefficients and arithmetic operations.
- Validate the performance of the system using nonlinearity performance metrics such as adjacent channel power ratio (ACPR) and Error Vector Magnitude (EVM).
- Highlight the obtained simulation results for different wordlengths, and discuss the potential impact of such results.
- State the key lessons and challenges learned while designing the proposed FXP DPD system.

1.3 Outline

The remainder of the thesis is organized as follows. Chapter 2 provides the necessary background knowledge about the behavioral modeling of PAs. The various existing behavioral models mainly Volterra series based models and two-box and three-box models are discussed. This involves a discussion on the input-output mathematical relationship and the number of coefficients required for estimation of each of the models. It also gives insights on the correlation between these models.

Chapter 3 showcases a comprehensive overview of the DPD technique. It touches on the theory related to adaptive digital baseband PD. Furthermore, it discusses the

background related to the two learning architectures i.e. DLA and ILA. Analytical and simulation results for FLP ILA for DPD are also presented.

In Chapter 4, a FXP DPD system based on ILA for nonlinear HPAs is proposed. The drawbacks of FLP based DPD approaches are highlighted and compared with the given system. Analytical and simulation results are shown to validate the proposed scheme.

Finally, chapter 5 provides a summary of the results obtained, draws conclusions, and outlines possible directions for future work.

2 Power Amplifier Behavioral Models

2.1 Overview

Behavioral Modeling [5-7] allows us to mathematically relate the input and output of the device under test, which in our case is the PA. In such kind of system-level modeling, the modeled device is considered as a “black-box,” i.e., we have no knowledge of the internal structure and the modeling information exists completely in the external responses of the device. We can thus, estimate the parameters of the model from measured transient responses or simulated results from detailed reference transistor-level models. Moreover, these models capture the nonlinearity and memory effects of PAs very effectively [6]. Predistortion can be considered as an important behavioral modeling problem, as in predistortion it is crucial to predict the nonlinearity of the PA. PD has inverse function to that of the PA hence, the synthesis of the predistortion function is equivalent to the behavioral modeling of the PA’s reverse function. This section complements previous work overviewing and comparing various behavioral models available in theory.

Over the years various single-box, two-box and three-box models for RF HPAs have been proposed in literature. Polynomial models are mainly based on series of Volterra functional [8], such as the compact memory polynomial (MP) model [9], the two-dimensional memory selective polynomial (2-D MSP) [10], the orthogonal memory polynomial [11] and the generalized memory polynomial (GMP) model [12]. Two box models consist of a Linear time invariant (LTI) system connected in tandem to a static nonlinearity and vice versa. These include the Wiener model [13], the Hammerstein model [14,15], the augmented Wiener [16], the augmented Hammerstein [15] and the twin nonlinear two-box models [17]. Three box models, such as Wiener-Hammerstein (WH) [18] and Hammerstein-Wiener (HW) [19] on the other hand, comprise of the static nonlinearity cascaded between two LTI systems or an LTI system in between two memoryless nonlinearity blocks respectively. In case of Volterra models, the complexity of the model increases immensely with the length of the system memory and the nonlinearity order whereas, the two-box and three-box models effectively capture the memory effects in PA modelling and thus overcome the requirement of large number of coefficients.

2.2 Polynomial Models

Volterra model has been used extensively by the researchers to model the nonlinearity of PAs including memory effects. A general Volterra model consists of large number of parameters and the complexity of determining these parameters is quite high. A simplified Volterra model is a MP model consisting of memory polynomials with less number of coefficients. Modeling the behavior of a PA by means of a MP for digital predistortion, has been covered in [6]. We discuss few of these models in this section.

2.2.1 Memory polynomial

According to Volterra series the most general form of nonlinearity with memory for a baseband input signal $x(n)$ can be represented in discrete time as [8]:

$$y(n) = \sum_{k=0}^K y_k(n) \quad (2.1)$$

where

$$y_k(n) = \sum_{l_1=0}^L \cdots \sum_{l_k=0}^L h_k(l_1, \cdots, l_k) \times \prod_{m=1}^k x(n - l_m)$$

$h_k(l_1, \cdots, l_k)$ denotes the k th order Volterra kernel, L represents the memory depth and K is the nonlinearity order. The measurement of such Volterra kernels involves high computational complexity (CC) which complicates the parameter identification process. But, if we leave only the diagonal terms in (2.1) and force all other coefficients to zero, we get a simplified structure in the form of MP model (refer to Appendix A) also known as Parallel Hammerstein (PH) as shown in Fig. 2.1.

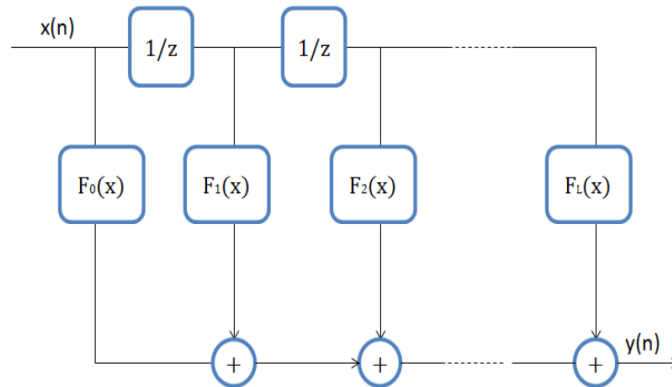


Figure 2.1: Memory Polynomial Model

The output of MP model can be given as:

$$y_{MP}(n) = \sum_{k=0}^{K-1} \sum_{l=0}^L a_{kl} x(n-l) |x(n-l)|^k \quad (2.2)$$

The main advantage of this model is that it is able to capture the memory effects while keeping the number of coefficients on the order of $K \times (L + 1)$ against the full Volterra system which restricts this number to $(L + 1)^K$, where K is the nonlinearity order.

2.2.2 Generalized Memory Polynomial

In [12] a direct way of adding cross terms i.e. the product terms with different time-shifts to an MP model is introduced without complicating the extraction procedure, known as the GMP model. In the GMP model the output is linearly dependent on its coefficients and it has been shown to outperform the MP model when used as a PD. In addition to the diagonal terms, the GMP model includes cross terms of the form $x(n-l)|x(n-m)|^k$ where $l = 0, \dots, L$, $m = -M, \dots, 0, \dots, M$ and $m \neq l$. The output for the GMP model can be expressed as

$$y_{GMP}(n) = \sum_{k \in K_a} \sum_{l \in L_a} a_{kl} x(n-l) |x(n-l)|^k + \sum_{k \in K_b} \sum_{l \in L_b} \sum_{m \in M_b} b_{klm} x(n-l) |x(n-l-m)|^k + \sum_{k \in K_c} \sum_{l \in L_c} \sum_{m \in M_c} c_{klm} x(n-l) |x(n-l+m)|^k \quad (2.3)$$

where K_a , K_b and K_c are the index arrays for nonlinearity, and L_a , L_b , L_c , M_b and M_c are the index arrays for memory. a_{kl} , b_{klm} and c_{klm} are the complex coefficients. The total number of coefficients is equal to $\overline{\overline{K_a L_a}} + \overline{\overline{K_b L_b M_b}} + \overline{\overline{K_c L_c M_c}}$ where $\overline{\overline{X}}$ denotes the cardinality (number of elements) of X .

2.2.3 Two-Dimensional Memory Selective Polynomial

From the MP and the GMP models discussed above we can observe that the terms that have a significant impact on modeling performance are of the form: $x(n-l) |x(n-m)|^k$, where $l = 0, \dots, L$, $m = -M, \dots, 0, \dots, M$. In [10] a memory selectivity has been introduced over a two-dimensional memory space, where nonlinearity index arrays are selected judiciously for each pair of delays (l, m) , instead of keeping same arrays of nonlinearity for all memory delays. Moreover, instead of restricting the terms $x(n-l) |x(n-m)|^k$ for $l = 0, \dots, L$, $m = -M, \dots, 0, \dots, M$ as in the case of GMP, other cross terms like $x(n-l) |x(n-m)|^k$ where $l = -L, \dots, -1$, $m = -M, \dots, 0, \dots, M$ were incorporated. Such memory selectivity could achieve

a performance comparable to the traditional models but with number of terms reduced significantly. The 2-D MSP model can be expressed as given below:

$$y(n) = \sum_{l=-L}^L \sum_{\substack{m=-M \\ A[i,j]=1 \\ i=l+L+1 \\ j=m+M+1}}^M \sum_{\substack{k=0 \\ B[i,j]=1}}^{K-1} a_{klm} x(n-l) |x(n-m)|^k \quad (2.4)$$

where

$$A = \begin{bmatrix} A_3 & m_n & A_4 \\ l_n & c & l_p \\ A_2 & m_p & A_1 \end{bmatrix} \quad (2.5)$$

Elements in A , $A[i, j]$, take values equal to 1 to select the corresponding terms in the two-dimensional memory space and 0 otherwise, with A_1, A_2, A_3 and A_4 as the $L \times M$ matrices controlling the terms in the four quadrants ($l \neq 0, m \neq 0$). m_n and m_p are $1 \times M$ column vectors controlling terms with delays $l = 0, m = -1 \dots -M$ and $l = 0, m = 1 \dots M$, respectively. Similarly, l_n and l_p are $L \times 1$ row vectors controlling terms with delays $l = -1 \dots -L, m = 0$ and $l = 1 \dots L, m = 0$, respectively. The value in c controls the terms of the form $x(n) |x(n)|^k$, i.e., $l = m = 0$. B_k has the same structure as A . However, for particular value of k it introduces the selectivity on the elements of A . The structure of B_k can be written as

$$B_k = \begin{bmatrix} B_3 & s_t & B_4 \\ u_t & d & u_v \\ B_2 & s_v & B_1 \end{bmatrix} \quad (2.6)$$

2.3 Two-Box and Three-Box Models

A two-box model for a nonlinear PA or PD usually consists of an input filter cascaded with a memoryless nonlinearity or vice-versa. The filter box is the small signal frequency response of the nonlinear device and the nonlinearity box characterizes the amplitude to amplitude distortion (AM/AM) and amplitude to phase distortion (AM/PM) conversion functions which operate on the instantaneous envelope of the input signal. Important two box-models that have been mentioned the most in theory include Wiener, Hammerstein, their augmented versions and the twin nonlinear two-box models [15-17].

On the other hand, a three-box model consists of a memoryless nonlinearity in between two filters characterising the frequency response of the nonlinear block or vice-versa i.e. a filter in between two memoryless nonlinearity blocks. WH [18] and HW [19] models are two of the most used three-box PA/PD models. We discuss few of these models in this section.

2.3.1 Hammerstein

Hammerstein model is a two box model that consists of a nonlinearity followed by a linear filter as can be seen from Fig. 2.2. By imposing special conditions on the extraction procedure of a_{kl} in (2.2) we can obtain the Hammerstein model if all the terms (diagonal terms) are maintained as it is. Hammerstein model is related to the MP model in the sense that, it is restricted to be the product of two other coefficients, splitting the 2-D array $\{a_{kl}\}$ in case of MP into two 1-D arrays $\{a_k\}$ and $\{b_l\}$, as given in (2.7) below:

$$y_H(n) = \sum_{k=0}^{K-1} \sum_{l=0}^L a_k b_l x(n-l) |x(n-l)|^k \quad (2.7)$$

Basically, the separation of static nonlinearity from linear filtering in the PH model results into this restriction on Hammerstein model parameters. As a result, there is a considerable reduction in the number of coefficients to $K + L + 1$ which might affect the behavioral modeling capability.



Figure 2.2: Hammerstein Model

2.3.2 Wiener

Wiener model is another important two-box model that is composed of a linear time invariant (LTI) system followed by a memoryless nonlinearity as shown in Fig. 2.3. If we use a finite impulse response (FIR) filter and the nonlinearity is modeled by a simple polynomial function, then the Wiener model output can be represented as:

$$y_W(n) = \sum_{k=0}^{K-1} a_k \sum_{l=0}^L b_l x(n-l) \left| \sum_{l=0}^L b_l x(n-l) \right|^k \quad (2.8)$$

One of the main advantages of Wiener model is that it can efficiently model the nonlinear memory effects of the PA with lesser coefficients but, unlike the Hammerstein and PH models it is quite difficult to identify its parameters. This is because the output depends nonlinearly on the coefficients, thus limiting its use.

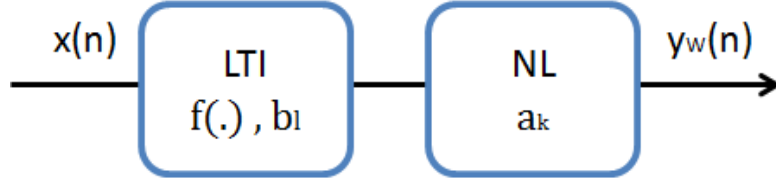


Figure 2.3: Wiener Model

2.3.3 Parallel Wiener

The outputs from several wiener models are combined to form the Parallel Wiener model as illustrated in Fig. 2.4. It represents the behavior of a PA at different envelope frequencies [20]. If we simply add the kernels of each subblock of the parallel wiener model, we will get the general Volterra series representation. Moreover, if the nonlinear functions following the linear filters are all coupled as a multiple-input, single-output memoryless nonlinearity it will result into a general Wiener model. The parallel wiener model can be seen as a generalization of the MP if each of the linear filters is specified as a simple delay element and each of the nonlinearities is specified as a polynomial with coefficients a_{kl} .

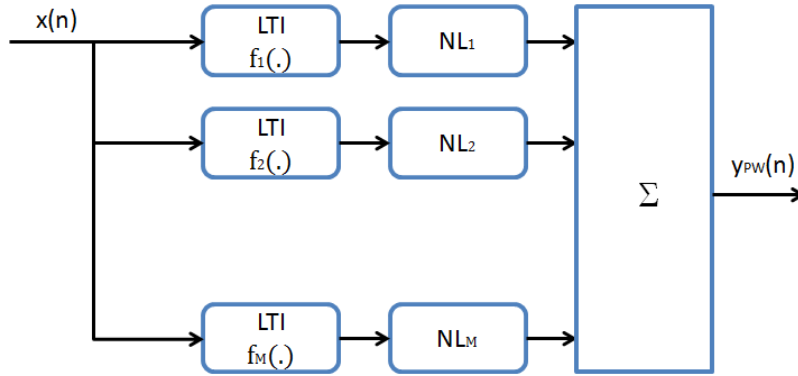


Figure 2.4: Parallel Wiener Model

2.3.4 Wiener-Hammerstein

WH model is a three-box model that consists of an LTI system connected in tandem with a memoryless nonlinearity which in turn is followed by another LTI system as illustrated in Fig. 2.5. It can be considered as a simple Wiener model with an additional filter at the output of static nonlinearity block. The output for the WH model can be written as:

$$y_{WH}(n) = \sum_{m=0}^{M-1} c_m \sum_{k=1}^K a_k \times \left[\sum_{l=0}^{L-1} b_l x(n-l-m) \right]^k \quad (2.9)$$

where a_k are the polynomial coefficients of nonlinearity and b_l and c_m are the filter coefficients. As it can be seen from (2.9), WH model is nonlinear in the parameters b_l even though it is more general than either Wiener or Hammerstein model.

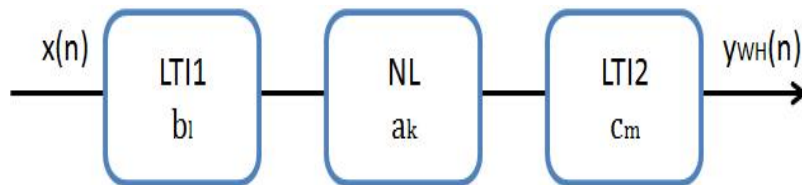


Figure 2.5: Wiener-Hammerstein Model

2.3.5 Hammerstein-Wiener

HW is another box oriented model with three boxes wherein a single LTI system is surrounded by two memoryless nonlinearities at its input and output. It can be considered as a Hammerstein model with an additional nonlinearity block at the output of LTI system. The equation representing the HW model as given in Fig. 2.6 is as follows:

$$y_{HW}(n) = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} c_m a_k b_l x(n-k) |x(n-k)|^l \left| \sum_{k=0}^{K-1} a_k \sum_{l=0}^{L-1} b_l x(n-k) |x(n-k)|^l \right|^m \quad (2.10)$$

where b_l are the polynomial coefficients of the input nonlinearity and c_m are the output nonlinearity polynomial coefficients, whereas a_k are the coefficients of the LTI system cascaded in between the two nonlinear systems.

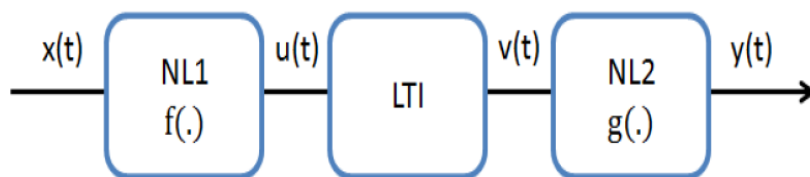


Figure 2.6: Hammerstein-Wiener Model

2.4 Conclusion

The presented chapter discusses the need for behavioral modeling of PAs for digital predistortion. PAs need to be modeled so that we can effectively calculate and

simulate the reverse function of the PA to be used in the PD. Another noteworthy point is that, behavioral modeling has the ability to capture nonlinearity as well as the memory effects which the transistor level modeling can not. Various such models that take into account the memory of PA were discussed. Volterra series forms the basis for most of the polynomial model PAs. But the major drawback associated with the Volterra models is that the number of parameters increases drastically with the nonlinearity order and memory depth. This led the researchers towards two-box and three-box models that could model the memory effects of the PAs with lesser number of parameters and thus, reduced CC. The mathematical system analysis of several such models is presented. One of the major findings during the study came out to be that, behavioral model quality is nowadays more based on the adopted parameter extraction algorithm that determines the CC, than by the model topology itself.

3 Indirect Learning Architecture (ILA) for DPD

3.1 Overview

The basic principle of predistortion is illustrated in Fig. 3.1. A PD block having transfer characteristics inverse to that of the PA is inserted prior to the PA in the transmit path. The PD has expanding characteristics which expands the input signal and when this "pre-expanded" signal is fed to the PA having compressive characteristics, it is rendered back to its original envelope without much distortion. As a result, the cascade of PD-PA ideally provides the desired linear gain and thus forms a linear system. The linearization function of the PD-PA system for an input signal $x(t)$ and an output signal $y(t)$ can be represented in the form of an equation as given below:

$$y(t) = ax(t) \tag{3.1}$$

where a is a real valued constant representing the desired linear gain of the PD-PA system. Mathematically, it can be stated that the PD described by the mapping

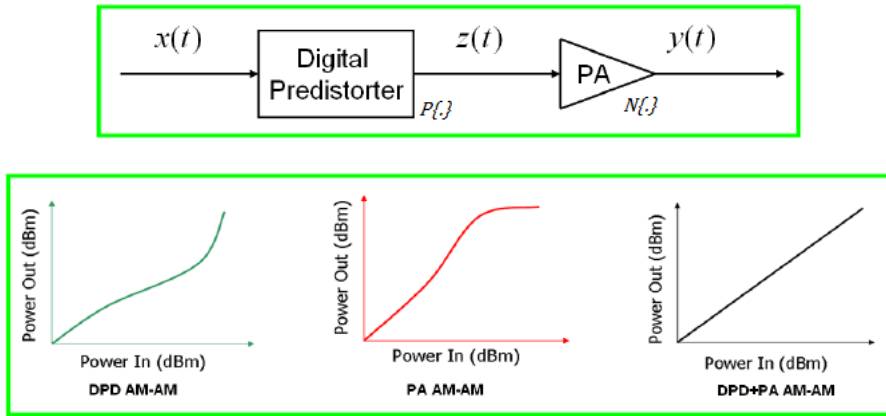


Figure 3.1: DPD-PA cascade combines two nonlinear systems into one linear result

$P\{.\}$ has to precisely invert the behaviour $N\{.\}$ of the PA upto a constant linear gain a as shown:

$$y(t) = N\{P\{x(t)\}\} = ax(t) \tag{3.2}$$

3.1.1 Adaptive Digital Baseband Predistortion

An adaptive digital baseband PD is shown in Fig. 3.2. When deployed signals are band-pass, which is the case of most of the wireless communication signals, we can determine an equivalent baseband system where the PD with an equivalent baseband model of the PA forms a linear system overall. Thus predistortion may be performed in the base band and block diagram of the linearization can be represented as in the following Fig. 3.2. There are two paths in the DPD system: feedback or the identification path and implementation or the observation path. The RF transmission signal at the output of the PA is downconverted to baseband i.e. the inphase and quadrature components in the feedback path which are then digitized by an analog-to-digital converter (ADC). The baseband samples are then processed in a digital signal processor (DSP) with an identification algorithm which compares them with the corresponding samples of the reference input signal. The PD parameter identification process is performed digitally, seeking to minimize the error between the input and the output, or another appropriate cost function. After a short time of convergence, the algorithm identifier characterizing the PD can operate as the pre-inverse of the PA. There are several approaches for the implementation of PD if PA has low memory effects where PD can be implemented by a lookup table or a non-parametric memory model. If memory effects are important, more complex model structures are to be used as discussed in chapter 2.

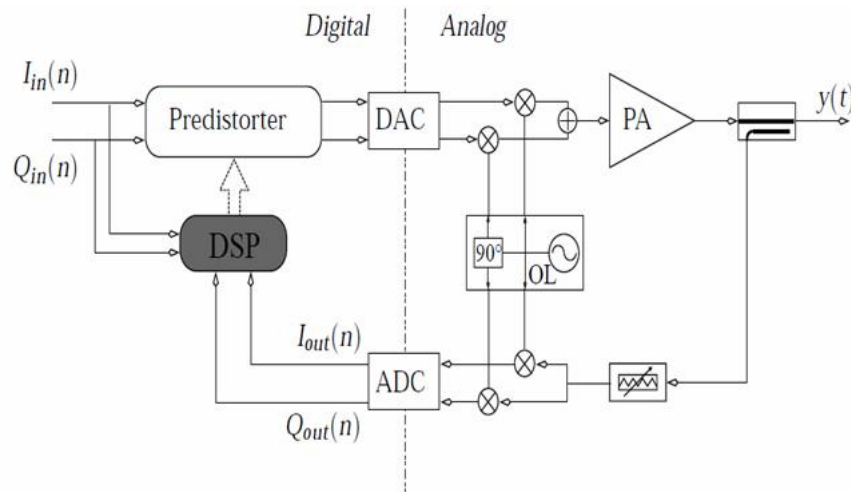


Figure 3.2: Adaptive Digital Baseband Predistortion

3.2 Learning Architectures

There are two main aspects in adaptive predistortion that need to be considered for finding the coefficients of the PD : Learning architecture and Adaptation Algorithm.

On the basis of learning architectures, DPD can be classified into Indirect learning architectures (ILA) [12], [21] and Direct learning architectures (DLA) [9], [22]. In the DLA approach, a model for the PA is first identified. The PD is then obtained based on the extracted PA model and a reference error between the input to the PD and output of the PA. Depending upon the extracted PA model and the reference error, various algorithms have been proposed to identify a PD in DLA. Prominent among them are the Analytical Method proposed by Kim et al. [9] and nonlinear filtered algorithms proposed in [22], [23]. In the ILA approach, a post-inverse of the PA is first identified and then just used as a PD [21]. The post-inverse can be identified by using either least mean square (LMS), recursive least square (RLS) or least squares (LS) approach.

3.2.1 Direct Learning Architecture (DLA)

The basic block diagram of DLA approach is illustrated in Fig. 3.3. As mentioned before, the identification of the PD based on DLA is done in two steps. First, the parameters of a predefined nonlinear model for the PA are extracted, and then in the second step, the identified model of the PA is used for the estimation of the PD. PA inverse may not exist analytically and which must then be approximated. In [9], an analytical method is used to compute the output of the PD using the extracted MP model of the PA.

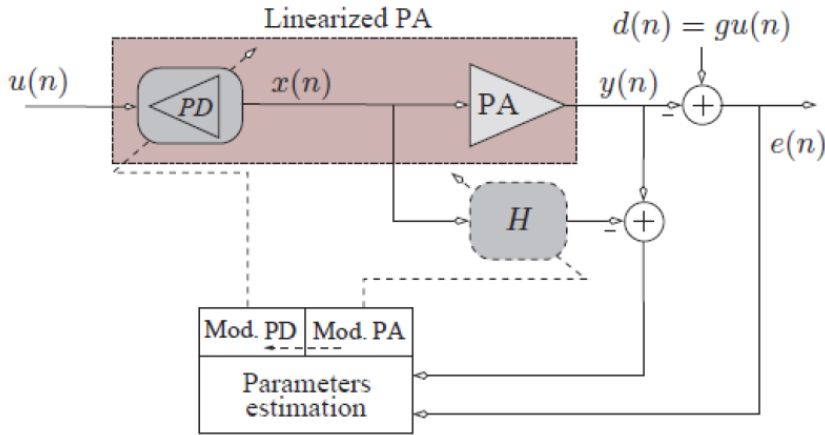


Figure 3.3: Direct Learning Architecture - DLA

Often the parameters of PA are usually extracted from one particular set of input and output data. However, after the 1st PD identification the characteristics of the input signal will change substantially. In fact the PD itself being a nonlinear system, after the 1st system level identification the spectrum of the PD output signal i.e. the input of PA will have wider bandwidth. Therefore, the behavior of the PA will likely

change and a new model should be “re-extracted” and used for the identification of the PD. This process should be repeated until the complete PD-PA system converges to the best possible solution.

3.2.2 Indirect Learning Architecture (ILA)

The PD identification in ILA is done in a single step as shown in Fig. 3.4. A post-inverse of the PA is identified and used as a PD. If the post-inverse is modeled as an MP, then its output can be written as [12]

$$z_p(n) = \sum_{k \in K} \sum_{l \in L} c_{pm} \phi_{kl}[z(n)] \quad (3.3)$$

$z(n) = y(n)/g$ is the input to the post-inverse block as shown in Fig. 3.4, g is the gain of the linearized PA, K is the index array for nonlinearity and L is the index array for memory. c_{kl} , $k \in K$ and $l \in L$ are the complex coefficients and $\phi_{kl}[z(n)] = z(n-l)|z(n-l)|^k$. The total number of coefficients is $J = \overline{K} \overline{L}$ with \overline{X} denoting the cardinality (number of elements) of X .

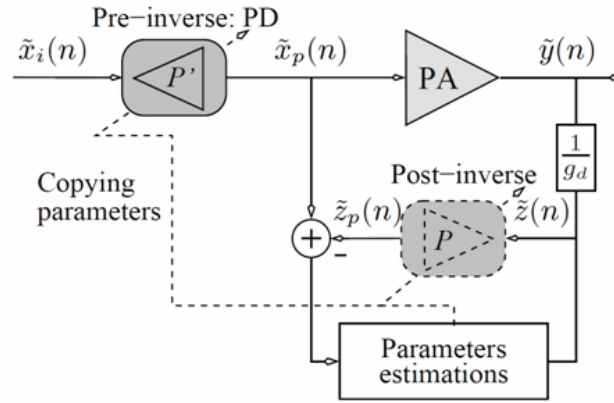


Figure 3.4: Indirect Learning Architecture - ILA

After convergence, we should have $z_p(n) = x(n)$ and hence $z(n) = u(n)$. For a total number of samples equal to N , we can write

$$z_p = \mathbf{Z} \mathbf{c} \quad (3.4)$$

where $z_p = [z_p(1), \dots, z_p(N)]^T$, \mathbf{c} is $J \times 1$ vector containing the set of coefficients c_{kl} , \mathbf{Z} is $N \times J$ matrix containing $\phi_{kl}[z]$ where $z = [z(1), \dots, z(N)]^T$. The LS solution for (3.4) will be

$$\hat{\mathbf{c}} = (\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H z_p \quad (3.5)$$

The following briefly illustrates the steps in computation of $\hat{\mathbf{c}}$

- Step 1: Define a new compound matrix

$$[\mathbf{Z}^H \mathbf{Z} | \mathbf{Z}^H \mathbf{z}_p] = \mathbf{QR} = \mathbf{Q}[\mathbf{U} | \mathbf{w}] \quad (3.6)$$

- Step 2: Compute the QR decomposition of the compound matrix by Gram Schmidt process [24] as illustrated in the flowchart in Fig. 3.5. r_{ii} are the nonzero diagonal entries of \mathbf{R} obtained by normalization of $[\mathbf{Z}^H \mathbf{Z} | \mathbf{Z}^H \mathbf{z}_p]$ and $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_i$ are the orthonormal vector columns of \mathbf{Q} . a_i is the i th column vector of the matrix that should be inverted, i.e., \mathbf{Z} . and n is the number of system level iterations [25].
- Step 3: Substitute the result of Step 2 into Step 1 to obtain

$$\mathbf{w} = \mathbf{U} \hat{\mathbf{c}} \quad (3.7)$$

which could be solved using back substitution [26].

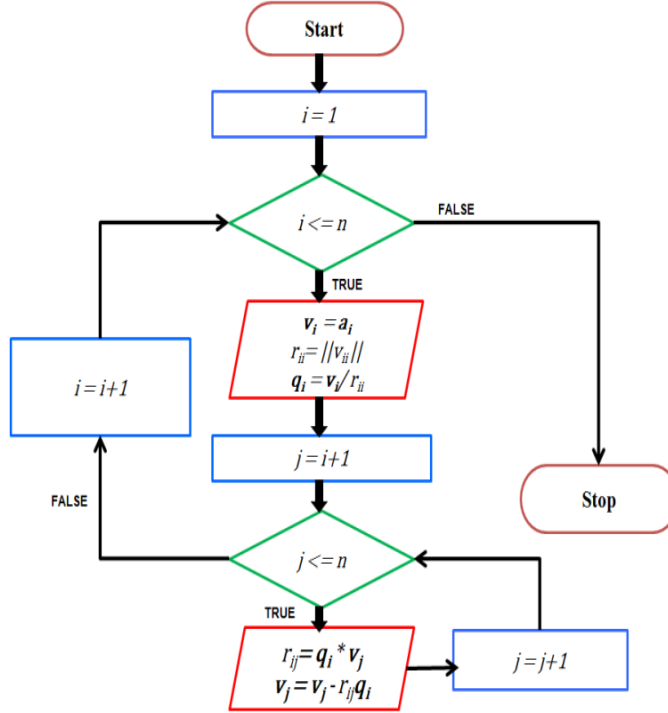


Figure 3.5: Flowchart for QR decomposition based on Modified Gram Schmidt algorithm

Table 3.1 summarizes roughly the CC needed in the computation of $\hat{\mathbf{c}}$. The CC is measured by computing the number of multiplication needed for each step. Hence, total CC needed for computation of $\hat{\mathbf{c}}$ would be $N(J + J^2) + J^3 + J^2$ where $J = P \times (M + 1)$. However, since $\hat{\mathbf{c}}$ and \mathbf{Z} are complex, the total number of real multiplication operations needed will be [24]:

$$CC = 4(N(J + J^2) + J^3 + J^2) \quad (3.8)$$

Table 3.1: COMPUTATIONAL COMPLEXITY

Step number	CC
Step 1	$N(J + J^2)$
Step 2	J^3
Step 3	J^2

In comparison to ILA, the CC needed for computation of \hat{c} for DLA is

$$CC = N(J + J^2) + J^3 + J^2 + N(5J^2 + 3J + D + JL) \quad (3.9)$$

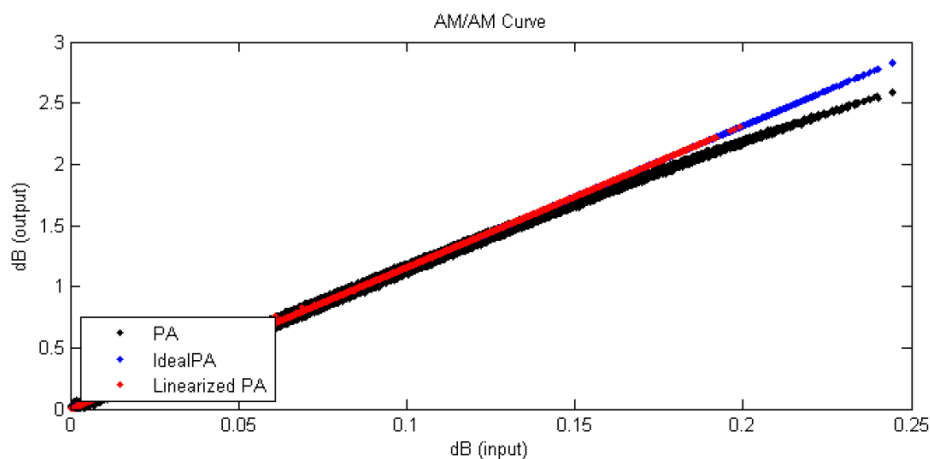
where $D = K \times (L + 1)$ and L is the memory depth [26].

3.3 Simulation results

In our study, we are going to focus only on ILA as it is computationally less complex than DLA. Thus, in this section, we present and discuss the simulation results for DPD based on ILA. For this purpose, we use Wiener model as the reference PA model as given in [13]. The PA is driven by an LTE-Advanced signal with bandwidth 10 MHz, sampling frequency 122.88 MHz and PAPR of approximately 11dB. We have simulated with nonlinearity array $K = [0 \ 2 \ 4]$ and memory array $L = [0 \ 2 \ 4]$. An MP model is used for the PD. Please note that, the PD-PA system requires more than one system level iteration to converge to the best possible solution. Depending upon the model of PA, the number of system level iterations needed for the convergence of ILA might vary as will be seen in the simulation results for ACPR and EVM [25].

3.3.1 AM/AM and AM/PM Characteristics

Fig. 3.6 shows the simulated AM/AM curve.

**Figure 3.6:** AM/AM Characteristic

It can be seen that the PA tends to show nonlinear behaviour after around 0.155 dB input amplitude.

Fig. 3.7 shows the AM/PM curve wherein the changes in output phase as a result of the changes in input amplitude are illustrated. The cloud like response for the non-linear PA for lower input levels is mostly due to the presence of memory effects.

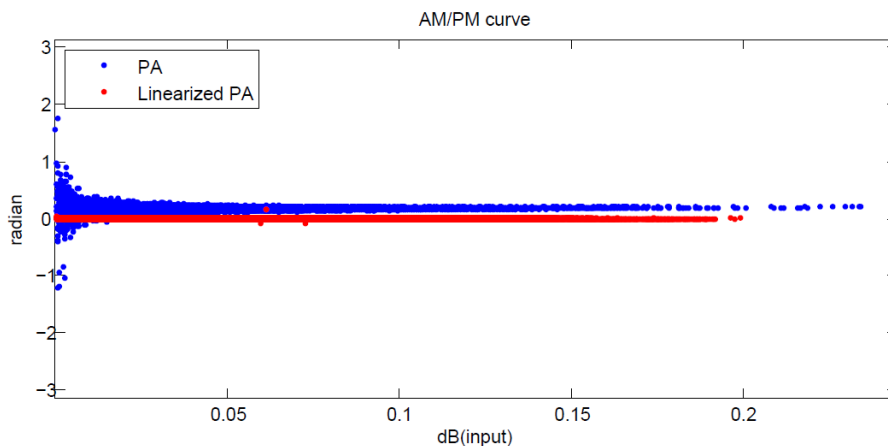


Figure 3.7: AM/PM Characteristic

3.3.2 Spectral Density

The power spectral density performance of the PA input and output signals before and after predistortion is shown in Fig. 3.8. The PD seems to fairly compensate for the spectral regrowth caused by the PA non-linearity. This corresponds to approximately 30 dB improvement in ACI. This improvement can be effectively used to enhance the PA efficiency for a given Federal Communications Commission (FCC) spectral mask.

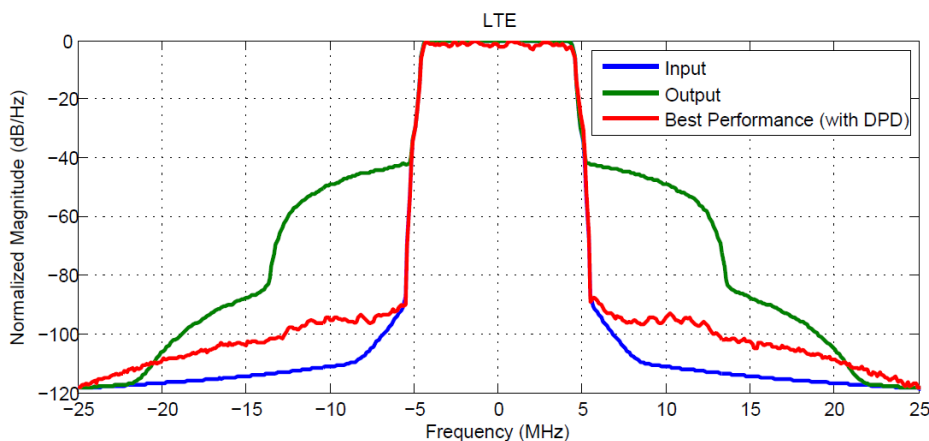


Figure 3.8: Power Spectral Density Plot

3.3.3 ACPR and EVM

Fig. 3.9 shows the ACPR performance of ILA for DPD for different system level iterations [25] for a Wiener PA model. ACPR measures the ratio of power in adjacent channel with respect to the amount of power in the main channel and thus, indicates the out-of-band distortions. For ACPR measurements we have considered 10 MHz bandwidth on both sides of the main channel. As can be observed, the identification algorithm converges after 2-3 system level iterations to achieve an ACPR of approximately -93 dBc.

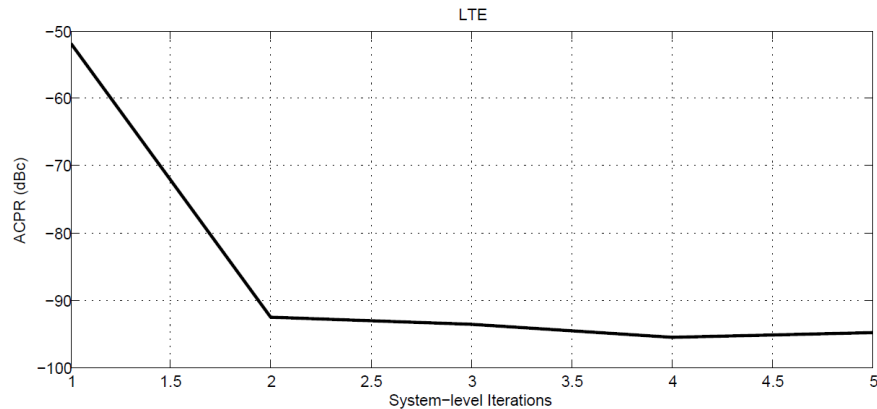


Figure 3.9: ACPR Performance

Fig. 3.10 shows the EVM performance of our DPD algorithm for different system level iterations for Wiener PA. EVM is a metric used for measuring the in-band distortion. We are able to achieve an EVM of approximately 0.05%.

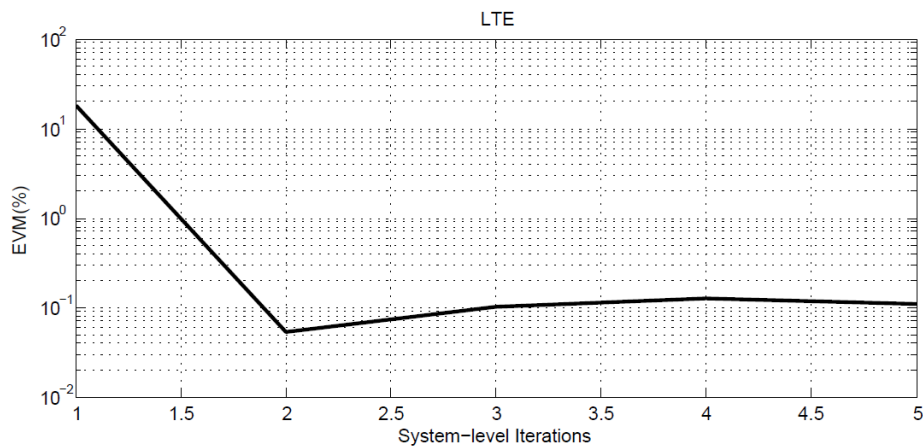


Figure 3.10: EVM Performance

4 Fixed Point DPD System based on ILA

4.1 Overview

Signal processing algorithms can be implemented in any of the two arithmetic formats available i.e. either FLP or FXP. These formats are used to store and manipulate numeric representations of data. FXP DSPs are designed to represent and manipulate integers – positive and negative whole numbers. On the other hand, FLP DSPs are designed to represent and manipulate rational numbers where a number is represented with a mantissa and an exponent. It is called 'Fixed-Point' because the numbers are represented with a fixed number of digits after, and sometimes before, the decimal point. In case of FLP representation, the decimal point can 'float' relative to the significant digits of the number.

FLP arithmetic offers a large dynamic range as it is determined by the size of the exponent, whereas in case of FXP the dynamic range is the range of numbers that can be represented in the available word length. If we consider the precision with which numbers can be represented, it is determined by the word length in the FXP format, and by the number of bits in the mantissa in the FLP format. For instance, in a 32 bit FLP DSP the mantissa is usually 24 bits, so the precision of such a DSP is the same as that of a 24 bit FXP processor. However, FLP has one further advantage over FXP that is, each number is scaled by the hardware automatically to use the full word length of the mantissa. As a result, full precision is maintained even for small numbers. Despite of all these advantages that the FLP offers, FXP data type is used widely in DSP applications, where performance is more important than precision because of the reasons discussed henceforth.

The FXP chip size is smaller with less power consumption in comparison to FLP as the logic circuits of FXP hardware are much less complicated than those of FLP hardware [28]. Calculations in FXP require less memory and less processor time to perform. Hence, when considering performance metrics such as cost, ease of use and area requirements FXP processors are a favourable choice for high-volume general purpose applications. In [28] the authors have shown an extensive comparison of the FXP and FLP units for a wordlength of 32 and 64. We can see that as we move from 64 bit FLP to 32 bit FXP addition units, we are able to achieve savings of approximately upto 90 percent in both area and power. It is also shown that 64 bit

FLP multiplier takes up almost four times more embedded multipliers than 32 bit FXP and consumes approximately three times more power.

Although much research has been done to investigate the performance of FLP digital predistortion systems[10,25,27,30], to the best of our knowledge not enough literature is publicly available to demonstrate the performance of FXP DPD systems. In [31], different numerical methods for the parameters extraction from the Volterra series PA are used to analyze the performance of a DPD, and the corresponding FXP numerical format implementation for only those numerical methods has been studied. Although, significant results have been achieved but the authors have restricted themselves only to the Volterra pruned series PA model for the analysis of their study [31], [32]. Also, the comparison results have been demonstrated only in terms of AM/AM and spectral density. In none of the work on FXP DPD that is available, ACPR and EVM performance metrics which are critical for the study of spectral regrowth and bit error rate have been considered.

In this section, we analyze the effects of FXP implementation of DPD system based on ILA. Both the digital PD and the parameter identification block have been implemented in FXP. The performance of the given algorithm is evaluated in terms of ACPR and EVM improvements using an LTE-Advanced signal. Simulation results show that the FXP implementation can achieve comparable performance to that of FLP reference model in terms of both ACPR and EVM.

4.2 FXP DPD System

This section deals with the detailed description of the proposed FXP implementation with respect to the overall DPD system design. A block diagram of DPD system for a PA along with technology mapping for the same is shown in Fig. 4.1. There are two computation paths for the implementation of digital PD, namely the predistorter path and the identification path. The PD lies in the digital domain before the digital to analog converter (DAC). It targets Field Programmable Gate Array (FPGA) or Application Specific Standard product (ASSP) for digital processing of the signal as the dynamic range or the resolution sizing of this block may affect the power consumption of the related hardware function. A sample by sample processing is required for the PD path to feed the DAC. Roughly speaking the sample rate of this path is constrained by the highest non linearity order taken into account. The only hardware that is able to run at such system rate is FXP processing hardware.

The output from the PD is converted to analog domain, passed through small signal RF upconversion blocks and then finally fed into the PA. PA can be a Gallium Nitride (GaN) or Laterally diffused Metal oxide semiconductor (LDMOS) PA. Similarly, the passband signal is converted back to baseband via down conversion blocks. The parameter identification algorithm is run only from time to time when new parameter identification is triggered by some system measurement [26,33]. The identification

path, in contrast to the PD path, handles block computation so the only constraint is that it is able to handle the computation at a rate equal to $(Sample\ Rate)/N$, where, N is the block size of the data taken into account for the identification algorithm. The hardware target for such processing is typically a microcontroller like Advanced RISC (Reduced Instruction Set Computing) Machines (ARM) core that is able to provide fast FXP computation and could be embedded for instance in an FPGA [34]. In such hardware the resolution of the data that is processed is based on a quantum which is the native data resolution of the processor, classically 32 bits. Based on this quantum, data format can range from simple resolution FXP to double precision FLP. The latter choice has a strong impact on the power consumption and the time required to process the algorithm and thus the PD and the identification block have been implemented in FXP.

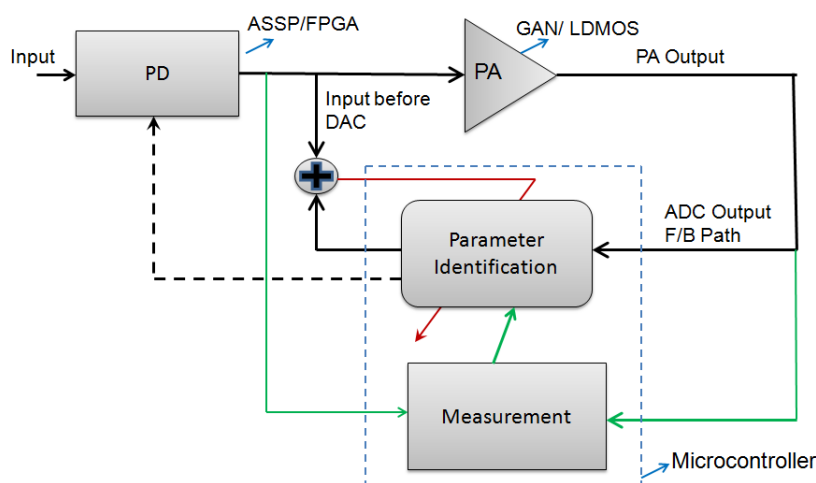


Figure 4.1: DPD system for PA linearization

4.2.1 Coefficient Extraction Procedure

The following steps describe the procedure to extract coefficients of a FXP DPD system using ILA algorithm:

1. Define the FXP Logic that determines the FXP data types to be used, their global and local settings for performing FXP arithmetic, data logging, data-type override etc.
2. Define the input and output data word length. Data word length has been defined assuming that it will be an input to 16, 18 or 20 bit DAC. Our simulation results showed that the overflow and underflow were minimized when the fractional length is kept same as data word length, hence in this implementation fractional length (FL) is kept same as word length (WL).

3. Specify the size of the FXP version of the MP model of the PD. Coefficient word length is taken as 32 assuming 32 bit processor and the coefficient fractional length is taken as 27. Table 4.1 lists the parameters used in the FXP implementation.
4. Determine the coefficients for the PD as already discussed in section II by computing the QR decomposition of the compound matrix given in (3.6) by modified Gram Schmidt process. The word length used for both Q and R is 32 and the fraction length used is 30 and 16 respectively. Such word lengths and fractional lengths have been chosen so as to optimize the performance with minimum overflows and underflows in FXP DPD algorithm. The estimated coefficients are recursively updated according to the damped Newton method [12].

Table 4.1: PARAMETERS FOR FXP IMPLEMENTATION

Parameters	Bit-width
Data WL	16, 18, 20
Data FL	16, 18, 20
Coefficient WL	32
Coefficient FL	27
Q WL	32
Q FL	30
R WL	32
R FL	16

4.3 Simulation results

In this section, we present and discuss the simulation results for the proposed FXP implementation. For this purpose, we use two different reference PA models, Wiener model as given in [13] and WH model as given in [18]. The PA is driven by an LTE-Advanced signal with bandwidth 10 MHz, sampling frequency 122.88 MHz and PAPR of approximately 11 dB. The number of input samples for each system level iteration is 20,000 [25]. An MP model is used for the extraction of PD parameters for both FLP as well as FXP implementation.

4.3.1 Spectral Density

Fig. 4.2 and Fig. 4.3 show the spectral regrowth suppression performance for Wiener and WH PA models respectively with the proposed FXP identification algorithm for data word lengths of 16, 18 and 20 bit. As observed all the data bitwidths are able to achieve sufficient amount of spectral regrowth reduction. As expected the

20 bit word length PD is able to achieve the best spectral regrowth suppression performance among all the different PD, hence proving to be more robust among all the word lengths considered because the input dynamic range is least limited in this case. Another point worth noting is that, as the data word length decreases from 20 to 16 bits the irreducible error floor increases which degrades the spectral regrowth suppression hence resulting in more ACPR.

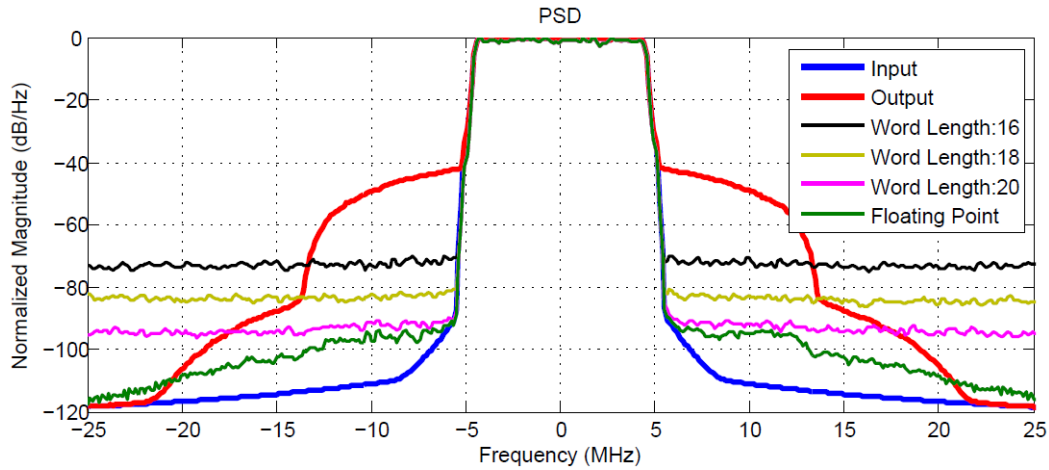


Figure 4.2: The spectral regrowth suppression performance with 16,18 and 20 bit data word lengths for Wiener Model

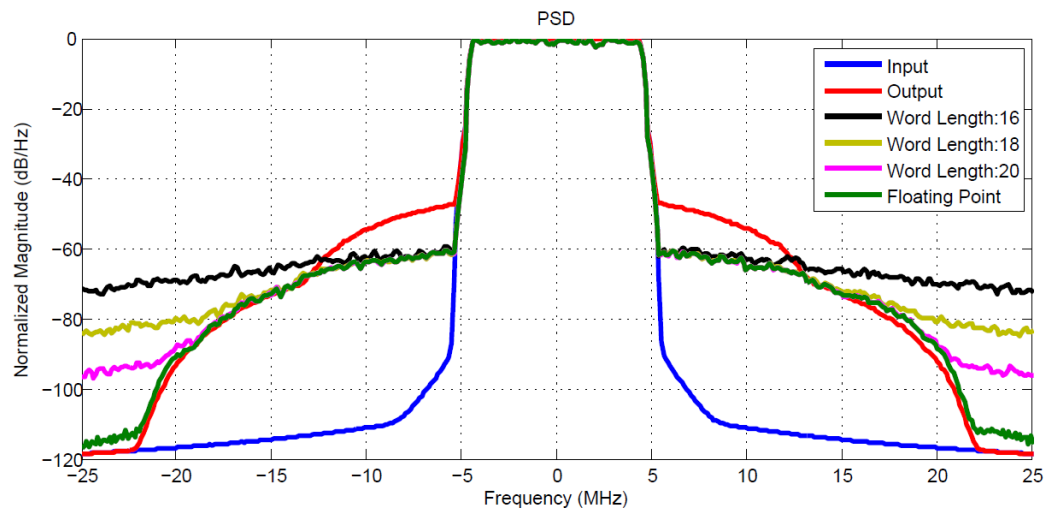


Figure 4.3: The spectral regrowth suppression performance with 16,18 and 20 bit data word lengths for WH PA

4.3.2 ACPR

Fig. 4.4 and Fig. 4.5 show the ACPR performance of FXP PD for different system level iterations for Wiener and WH PA respectively. These figures denote the

performance with the given algorithm for different data word lengths. As observed the identification algorithm for all the data word lengths converges after 2-3 system level iterations. However, it is quite evident that 20 bit word length system performs almost similar to its FLP counterpart. The 20 bit word length FXP MP PD converges after 2nd system level iteration to achieve an ACPR of approximately -90dBc for Wiener PA and -67dBc for WH PA.

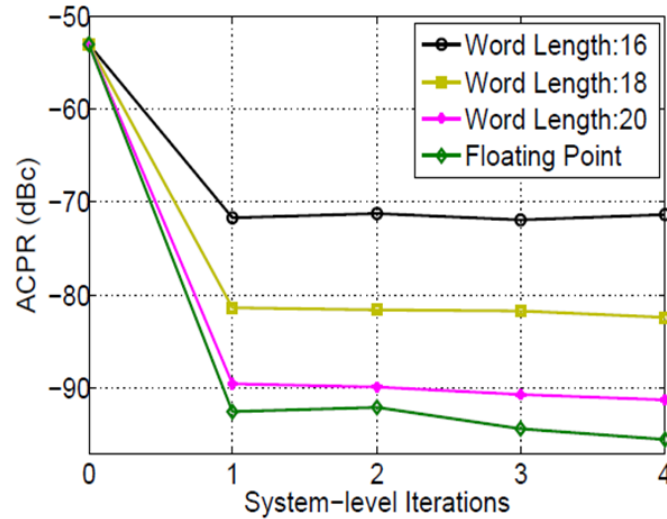


Figure 4.4: ACPR Performance for Wiener PA: System level Iterations

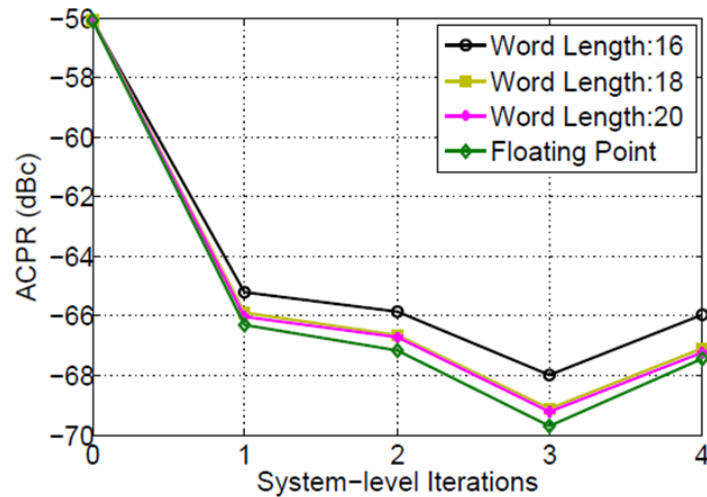


Figure 4.5: ACPR Performance for WH PA: System level Iterations

The difference in performance of both these models might be due to the difference in the characteristics and inherent nonlinearity behaviours of these PA models, since we are using only the behavioral models. The choice of input and output data word

lengths can be justified depending upon our requirement and the amount of tradeoff we are willing to tolerate. For example, the proposed system can be highly beneficial for a user who has a stringent ACPR requirement of approximately -80 dBc which can be achieved efficiently with our 18 bit data word length FXP DPD system. On the other hand if the primary user requirement is lesser area and power, we can achieve it with the 16 bit data word length FXP DPD, though compromising a bit on ACPR performance.

4.3.3 EVM

Fig. 4.6 and Fig. 4.7 show the EVM performance of our FXP DPD algorithm for different system level iterations for Wiener and WH PA respectively. The 20 bit word length FXP MP PD achieves an EVM of approximately 0.05% and 0.19% for Wiener and WH PA respectively. It is quite obvious from the figure that the EVM performance for almost all the cases is precisely the same for a given number of system level iterations. This might be due to the lack of significant in-band distortion in the PA model. Thus, if the user requirement is for comparable EVM performance then either a 16 bit, 18 bit or 20 bit data word length can be used but if the requirement is for robust ACPR performance then 20 bit data word length needs to be used.

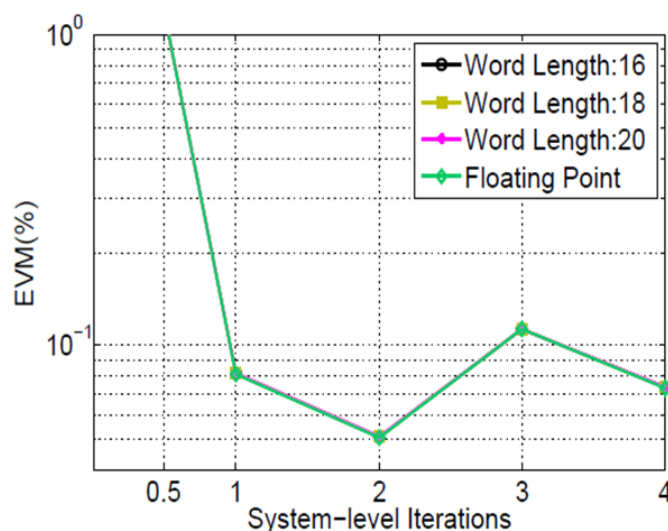


Figure 4.6: EVM Performance for Wiener PA: System level Iterations

4.3.4 Comparative Analysis

Table 4.2 and 4.3 show the results for Wiener and WH PA models respectively. We have compared our design with FLP simulations. The performance of FXP MP

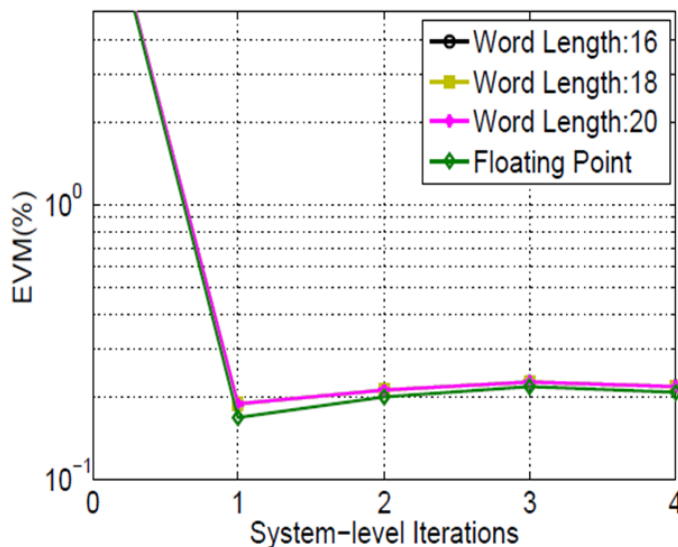


Figure 4.7: EVM Performance for WH PA: System level Iterations

PD has been demonstrated by considering data word lengths of 16, 18 and 20 bit. ACPR measures the ratio of power in adjacent channel with respect to the amount of power in the main channel. For ACPR measurements we have considered 10 MHz bandwidth on both sides of the main channel. EVM has been calculated by dividing the difference between the received and transmitted symbols by the average value of the input signal as given in [34]. K and L are vectors containing k and l values as given in (3.3). As seen from Table III and IV, all the three word lengths in FXP and the FLP DPD are able to achieve sufficient improvement in ACPR and EVM at the output of the PA. However, the DPD system with 20 bit data word length outperforms all the other PDs as it is able to achieve the best ACPR and EVM performance, hence proving to be more robust among all others. This indicates that as the dynamic range (no. of bits) of the data word length increases the performance of the system in terms of reducing the spectral regrowth and bit error rate improves. Thus, we can infer that as the data word length decreases from 20 to 16 bits the irreducible error floor increases. Also, it is worth noting that the overall performance of FXP DPD is comparable to that of FLP DPD with respect to both ACPR and EVM.

Table 4.2: WIENER MODEL

Parameters	Without DPD	FLP DPD	FXP DPD
ACPR(dBc)	-49.68	-95.51	wl 16: -71.95 wl 18: -82.44 wl 20: -91.33
EVM(%)	18.05	0.05	wl 16: 0.106 wl 18: 0.058 wl 20: 0.0511
Index array for nonlinearity and memory	NA	$K = [0 \ 2 \ 4]$ $L = [0 \ 2 \ 4]$	$K = [0 \ 2 \ 4]$ $L = [0 \ 2 \ 4]$

Table 4.3: WIENER-HAMMERSTEIN MODEL

Parameters	Without DPD	FLP DPD	FXP DPD
ACPR(dBc)	-54.78	-69.72	wl 16: -68.007 wl 18: -69.122 wl 20: -69.196
EVM(%)	19.58	0.169	wl 16: 0.19967 wl 18: 0.19043 wl 20: 0.1901
Index array for nonlinearity and memory	NA	$K = [0 \ 1 \ 2]$ $L = [0 \ 1 \ 2 \ 4]$	$K = [0 \ 1 \ 2]$ $L = [0 \ 1 \ 2 \ 4]$

5 Conclusions and Future Work

DPD is one of the most popular methods to linearize the PA. Although it is well established in theory and simulation, its performance with respect to the quantization effects and the relative impact on wordlength is still less known. Based on the simulation results of our proposed FXP DPD system for two of the behavioral PA models, it can be inferred that the given analysis can be used to optimize the performance of the linearizer with respect to the wordlength based on the maximum allowable ACI or other parameters like amplifier gain characteristic.

The research work done in the thesis can be categorized into three parts. In the first part fundamentals of behavioral modeling of PAs were discussed. The understanding of the behavioral modeling of PAs is critical for our FXP system implementation of DPD as the knowledge of the nonlinearity of the PA is essential in the calculation of the inverse function of the PA which is actually the predistortion function. The various behavioral models were investigated. This included a brief overview of the existing PA models based on Volterra series and other variants, two-box and three-box models and a discussion on their correlation.

In the second part, DPD theory and background was explored. Adaptive digital baseband PD functionality was discussed. The two learning architectures DLA and ILA for learning the coefficients of PD were investigated. It was shown that how the ILA for a FLP DPD performs in terms of AM/AM and AM/PM characteristics. The power spectral density curve highlighted the spectral regrowth suppression performance. ACPR and EVM at the output of the PA were also measured to analyse the performance of FLP DPD system

In the third part, FXP system for DPD was presented. The algorithm was shown to identify the PD by ILA with both the PD and the parameter identification block implemented in FXP rather than FLP. The performance of the proposed algorithm was evaluated by measuring the ACPR and EVM at the output of the PA for different data word lengths for an LTE-Advanced input signal. Simulation results demonstrated that FXP DPD performed closely to the FLP DPD when a 20 bit data word length was assumed. From the power spectral density simulation we could also conclude that as the word length increases the dynamic range increases, hence the irreducible error floor decreases.

5.1 Future Work

Finally we discuss some issues which need further investigations. These can be interesting directions for future work.

- For the simulation results, only two PA behavioral models are considered. In practice, various other models can be used to validate the performance of the given system.
- It will be interesting to observe the performance enhancement (if any) when signals other than LTE-Advanced are used.
- The performance of the given FXP based DPD system has been validated by the simulation results. For future research, the next step would be the implementation of this DPD system on hardware such as FPGA to obtain the on chip performance assessment.
- Another important area where the current research can be extended is the complexity study of the various behavioral models.

Appendix A

Volterra series represents the most general form of K^{th} order nonlinearity with L -tap memory for a baseband input signal $x(n)$ as [8]:

$$y(n) = \sum_{k=0}^K y_k(n) \quad (5.1)$$

where

$$y_k(n) = \sum_{l_1=0}^L \cdots \sum_{l_k=0}^L h_k(l_1, \cdots, l_k) \times \prod_{m=1}^k x(n - l_m) \quad (5.2)$$

We can expand equation (5.1) for $K = 2$ as follows:

$$y(n) = \sum_{k=0}^2 y_k(n) = y_0(n) + y_1(n) + y_2(n) \quad (5.3)$$

From (5.2) we have,

$$\begin{aligned} y_0(n) &= h_0(n) \\ y_1(n) &= \sum_{l_1=0}^L h_1(l_1)x(n - l_1) \\ y_2(n) &= \sum_{l_1=0}^L \sum_{l_2=0}^L h_2(l_1, l_2)x(n - l_1)x(n - l_2) \end{aligned}$$

where h_0 is a constant and h_1, h_2 are the set of first and second order Volterra kernel coefficients. For $K = 2$ let us take $L = 2$ sample memory. Equation (5.3) can then be expressed as given below:

$$y(n) = h_0 + \sum_{l_1=0}^2 h_1(l_1)x(n - l_1) + \sum_{l_1=0}^2 \sum_{l_2=0}^2 h_2(l_1, l_2)x(n - l_1)x(n - l_2) \quad (5.4)$$

Now, we can expand equation (5.4) to get the number of coefficients required for Volterra series for $K = 2$ and $L = 2$ as:

$$\begin{aligned} y(n) &= h_0 + h_1(0)x(n) + h_1(1)x(n - 1) + h_1(2)x(n - 2) + h_2(0, 0)[x(n)]^2 \\ &+ h_2(0, 1)x(n)x(n - 1) + h_2(1, 0)x(n - 1)x(n) + h_2(1, 1)[x(n - 1)]^2 \\ &+ h_2(0, 2)x(n)x(n - 2) + h_2(2, 0)x(n - 2)x(n) + h_2(2, 2)[x(n - 2)]^2 \\ &+ h_2(1, 2)x(n - 1)x(n - 2) + h_2(2, 1)x(n - 2)x(n - 1) \end{aligned} \quad (5.5)$$

For baseband applications, we are only interested in terms generating signals around the center frequency, hence only odd order terms are considered. Looking at expression (5.5) we can clearly count the number of such unknowns to be 9. This validates the expression for number of coefficients for a full Volterra system (considering only odd order terms) that is $(L + 1)^K$.

If $h_2(l_1, l_2) = 0$ except along the diagonal $l_1 = l_2$, then (5.5) becomes:

$$y(n) = h_0 + h_1(0)x(n) + h_1(1)x(n-1) + h_1(2)x(n-2) + h_2(0,0)[x(n)]^2 + h_2(1,1)[x(n-1)]^2 + h_2(2,2)[x(n-2)]^2 \quad (5.6)$$

We can clearly observe that now the number of coefficients required reduces to 3 i.e. $(L + 1)$ for odd order terms. Equation (5.6) can be generalised to give the expression for a MP as:

$$y(n) = \sum_{l=0}^L [h_1(l)x(n-l) + h_2(l,l)\{x(n-l)\}^2] \quad (5.7)$$

Bibliography

- [1] S. C. Cripps, *RF Power Amplifiers for Wireless Communications, Second Edition* (Artech House Microwave Library (Hardcover)). Norwood, MA, USA: Artech House, Inc., 2006.
- [2] Raab, Frederick H., et al. "Power amplifiers and transmitters for RF and microwave." *Microwave Theory and Techniques, IEEE Transactions on* 50.3 (2002): 814-826
- [3] P. B. Kenington, *High-Linearity RF Amplifier Design*. Boston, MA: Artech House, 2000.
- [4] Smith, A. M., & Cavers, J. K. (1998, May). A wideband architecture for adaptive feedforward linearization. In *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE* (Vol. 3, pp. 2488-2492). IEEE.
- [5] Isaksson, Magnus, David Wisell, and Daniel Ronnow. "A comparative analysis of behavioral models for RF power amplifiers." *Microwave Theory and Techniques, IEEE Transactions on* 54.1 (2006): 348-359.
- [6] Pedro, José C., and Stephen A. Maas. "A comparative overview of microwave and wireless power-amplifier behavioral modeling approaches." *Microwave Theory and Techniques, IEEE Transactions on* 53.4 (2005): 1150-1163.
- [7] Ghannouchi, Fadhel M., and Oualid Hammi. "Behavioral modeling and predistortion." *Microwave Magazine, IEEE* 10.7 (2009): 52-64.
- [8] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.
- [9] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electronics Letters*, vol. 37, no. 23, pp. 1417–1418, 2001.
- [10] Abi Hussein, Mazen, Vivek Ashok Bohara, and Olivier Venard. "Two-dimensional memory selective polynomial model for digital predistortion." *New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International*. IEEE, 2012.
- [11] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1468–1479, 2004.

- [12] D. R. Morgan, Z. X. Ma, J. Kim, M. G. Zierdt, and J. Z. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3852–3860, 2006.
- [13] S. Chen, "An efficient predistorter design for compensating nonlinear memory high power amplifiers," *Broadcasting, IEEE Transactions on*, vol. 57, no. 4, pp. 856–865, dec. 2011.
- [14] P. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *Microwave Conf. Proc., APMC 2005, Asia-Pacific*, Dec. 2005.
- [15] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Augmented Hammerstein predistorter for linearization of broad-band wireless transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 4, pp. 1340–1349, Apr. 2006.
- [16] Liu, Taijun, Slim Boumaiza, and Fadhel M. Ghannouchi. "Pre-compensation for the dynamic nonlinearity of wideband wireless transmitters using augmented Wiener predistorters." *Microwave Conference Proceedings, 2005. APMC 2005. Asia-Pacific Conference Proceedings*. Vol. 5. IEEE, 2005.
- [17] O. Hammi and F. M. Ghannouchi, "Twin nonlinear twobox models for power amplifiers and transmitters exhibiting memory effects with application to digital predistortion," *IEEE Microwave and Wireless Components Letters*, vol. 19, no. 8, pp. 530–532, 2009.
- [18] Kibangou, Alain Y., and Gérard Favier. "Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients." *IEEE signal processing letters* 13.6 (2006): 381.
- [19] Vörös, Jozef. "An iterative method for Hammerstein-Wiener systems parameter identification." *Journal of electrical engineering* 55.11-12 (2004): 328-331.
- [20] H. Ku, M. D. Mckinley, and J. S. Kenney, "Extraction of accurate behavior models for power amplifiers with memory effects using twotone measurements," in *Proc. IEEE MTT-S Int. Microwave Symp. Dig.*, 2002, vol. 1, pp. 139–142.
- [21] L. Ding, G. Zhou, D. Morgan, Z. Ma, J. Kenney, J. Kim, and C. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *Communications, IEEE Transactions on*, vol. 52, no. 1, pp. 159–165, Jan. 2004.
- [22] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *Signal Processing, IEEE Transactions on*, vol. 46, no. 6, pp. 1726–1730, Jun 1998.
- [23] D. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *Signal Processing, IEEE Transactions on*, vol. 55, no. 1, pp. 120–133, Jan. 2007.
- [24] L. Trefethen and D. Bau III, *Numerical linear algebra*. Society for Industrial Mathematics, 1997, no. 50.

-
- [25] M. Abi Hussein, V. A. Bohara, and O. Venard, "On the system level convergence of ila and dla for digital predistortion," in *Wireless Communication Systems (ISWCS)*, 2012 International Symposium on, Aug. 2012, pp. 870–874.
- [26] V. A. Bohara, A. H. Mazen, and O. Venard, "Digital Predistortion Algorithms Complexity and Sensitivity Study," in *Proceedings of Par4CR workshop on cognitive radio*, Kista, Sweden, June 2013.
- [27] V. A. Bohara, M. Abi Hussein, and O. Venard, "A parameter identification algorithm for multi-stage digital predistorter". In *Microwave Conference (EuMC)*, 2013 European (pp. 416-419). IEEE.
- [28] Frantz, Gene, and Ray Simar. "Comparing fixed-and floating-point DSPs." Texas Instruments, Dallas, TX, USA (2004).
- [29] M. Moonen and I. Proudler, "Introduction to adaptive signal processing," Department of Electrical Engineering ESAT/SISTA KU Leuven, Leuven, Belgium, pp. 105–107, 1998.
- [30] M. Abi Hussein, V. A. Bohara, and O. Venard, "Multi-stage digital predistortion based on indirect learning architecture," in *38th International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, IEEE, Vancouver, Canada, 2013, May 2013.
- [31] M. G. Hernandez, A. P. Guerrero, G. A. Laguna-Sanchez and P. M. Valencia, "Fixed Point implementation for parameters extraction in a digital predistorter using adaptive algorithms" in *11th International conference on Information Sciences, Signal Processing and their applications: Main Tracks*.
- [32] N. Lashkarian and C. Dick, "FPGA implementation of digital predistortion linearizers for wideband power amplifiers", Signal Processing Division, Xilinx Inc, San Jose, USA.
- [33] M. A. Hussein, O. Venard, B. Feuvrie, and Y. Wang, "Digital predistortion for RF power amplifiers: State of the art and advanced approaches," in *New Circuits and Systems Conference (NEWCAS)*, 2013 IEEE 11th International, 2013, pp. 1–4.
- [34] <http://arm.com/products/processors/cortex-m/cortex-m4-processor.php>
- [35] Q. Gu, "RF System Design of Transceivers for Wireless Communications", Springer, 2005.