Digital Predistortion of Power Amplifiers for Wireless Applications

A Thesis
Presented to
The Academic Faculty

by

Lei Ding

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
March 2004

Petitioners Exhibit 1005
Page 1 of 117

Digital Predistortion of Power Amplifiers for Wireless Applications

Approved by:

Dr. G. Tong Zhou, Advisor

Dr. Ye (Geoffrey) Li

Dr. J. Stevenson Kenney

Dr. Jianmin Qu

Dr. W. Marshall Leach

Date Approved: April 1, 2004

Petitioners Exhibit 1005
Page 2 of 117

For my family.

Petitioners Exhibit 1005
Page 3 of 117

ACKNOWLEDGEMENTS

It is time to draw a period to my PhD endeavor. I feel extremely lucky to meet so many talented people during these years. I am grateful for their friendship, help, encouragement, and support.

First, I would like to thank my advisor, Dr. G. Tong Zhou, for providing this opportunity for me to study at Georgia Tech. Her constant encouragement and valuable advices are essential for the completion of this thesis. I will be always inspired by the high standards she sets for herself and her sharp focus and deep devotion to whatever she works on.

I would like to thank my thesis committee members: Drs. J. Stevenson Kenney, W. Marshall Leach, Ye (Geoffrey) Li, and Jianmin Qu for taking time to serve on my committee and for their questions and suggestions. My special thanks also go to Drs. Monson Hayes and Douglas B. Williams for the excellent classes they offered, which benefit me a lot for my research work and job search.

I also benefit a lot from my two summer internships at Bell Labs, Lucent Technologies. I would like to thank Drs. Zhengxiang Ma and Dennis R. Morgan for making this possible. I enjoyed every aspect of my internship experience: talented colleagues, excellent research opportunities, cool weather, nice work environment, and the list goes on. My special thanks also go to Dr. Yiteng Huang for his friendship and help during my stay in New Jersey.

I would like to thank my group members: Yongsub, Krishna, Muhammad, Raviv, Gail, Ning, Hua, Chunpeng, Yuan, Thao, Vincent, and Bob for all the technical discussions and help along the way. I also enjoyed interactions with the nice Chinese student community here at CSIP.

A lot of credit goes to my parents. Their unconditional support has always helped me in all my endeavors. I would also like to thank my sister for her support and feeding me with great dishes from time to time. Last but not least, I would like to thank my wife, wenjin, for her love, encouragement, and support.

Petitioners Exhibit 1005
Page 4 of 117

TABLE OF CONTENTS

Dedication . . . . . **iii**

Acknowledgements . . . . . **iv**

List of Tables . . . . . **viii**

List of Figures . . . . . **ix**

Summary . . . . . **xiii**

Chapter 1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 3

1.3 Outline 3

Chapter 2 Background **5**

2.1 Modeling Memoryless Power Amplifiers 5

2.2 Predistortion of Memoryless Power Amplifier 7

2.2.1 Data Predistortion for Memoryless Power Amplifiers 7

2.2.2 Signal Predistortion for Memoryless Power Amplifiers 8

2.3 Modeling Power Amplifiers with Memory Effects 10

2.4 Predistortion of Power Amplifiers with Memory Effects 14

Chapter 3 Digital Predistorter Design **17**

3.1 Hammerstein Predistorter Design 17

3.1.1 Hammerstein Predistorter Training 17

3.1.2 Hammerstein Predistorter Simulation 22

3.2 Memory Polynomial Predistorter Design 25

3.2.1 Memory Polynomial Predistorter Training 25

3.2.2 Memory Polynomial Predistorter Simulation 26

3.2.3 Memory Polynomial Predistorter Discussion 33

3.3 A New Combined Predistorter Design 35

3.3.1 Combined Predistorter Model 36

3.3.2 Combined Predistorter Training 37

3.3.3 Effects of Noise and Initial Estimates 40

Petitioners Exhibit 1005
Page 5 of 117

3.3.4 Combined Predistorter Performance 45

Chapter 4 Effects of Even-order Nonlinear Terms 47

4.1 Passband and Baseband Nonlinearities 47

4.1.1 Memoryless Case 47

4.1.2 Quasi-memoryless Case 48

4.2 Even-Order Terms in the Baseband Power Amplifier Model 50

4.3 Even-Order Terms in the Baseband Predistorter Model 53

4.4 Extensions to Power Amplifiers and Predistorters with Memory . . . 58

Chapter 5 Analog Imperfection Compensation 61

5.1 Two-Stage Upconversion Transmitter 61

5.1.1 System Setup 61

5.1.2 Channel Estimation 62

5.1.3 Equalizer Design 65

5.1.4 Experimental Results 67

5.2 Direct Upconversion Transmitter 71

5.2.1 Channel Models 72

5.2.2 Channel Estimation 76

5.2.3 Compensator Construction 79

5.2.4 Simulations 84

Chapter 6 Real-time Implementations 87

6.1 Memory Polynomial Model 87

6.2 Indirect Learning Architecture 88

6.3 Predistorter Construction 88

6.4 DSP Implementation 91

6.4.1 Implementation Details 91

6.4.2 Performance Evaluation 92

6.5 Testbed Measurements 92

Chapter 7 Conclusions 98

7.1 Contributions 98

7.2 Suggestions for Future Research 99

Petitioners Exhibit 1005
Page 6 of 117

References **100**

Petitioners Exhibit 1005
Page 7 of 117

LIST OF TABLES

Table 1 NMSE for $F(|z(t)|)$ 52
Table 2 NMSE for $G(|x(t)|)$ 55
Table 3 NMSE when the power amplifier is modeled by memory polynomial. . . . 58
Table 4 Ideal Model Coefficients 84
Table 5 Real-time performance of the predistorter training algorithm. 94

Petitioners Exhibit 1005
Page 8 of 117

LIST OF FIGURES

Figure 1 Digital Predistortion System Diagram 2

Figure 2 The AM/AM and AM/PM responses of a Class AB power amplifier. . . . 6

Figure 3 Block diagram of a data predistortion system with the pulse shaping filter implemented after the power amplifier. 7

Figure 4 Block diagram of a data predistortion system with the pulse shaping filter placed in the baseband after the predistorter. 8

Figure 5 Block diagram of a signal predistortion system. 9

Figure 6 The Wiener Model. 12

Figure 7 The Hammerstein Model. 12

Figure 8 The Wiener-Hammerstein model. 12

Figure 9 The indirect learning architecture for the predistorter. 15

Figure 10 The indirect learning architecture for the Hammerstein predistorter. . . . 18

Figure 11 Comparison of the PSDs for the pole/zero Wiener power amplifier and the pole/zero Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion, NG and LS/SVD algorithms (similar performance). 23

Figure 12 Comparison of the PSDs for FIR Wiener power amplifier and 15-tap FIR Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion (NG); (d) Output with Hammerstein predistortion (LS/SVD). . . . 24

Figure 13 Comparison of the PSDs for full Volterra power amplifier and 15-tap FIR Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion (NG); (d) Output with Hammerstein predistortion (LS/SVD); (e) Input signal. 24

Figure 14 The indirect learning architecture for the memory polynomial predistorter. 26

Figure 15 Wiener-Hammerstein model diagram. 27

Figure 16 Effectiveness of predistortion in suppressing spectral regrowth when the power amplifier is modeled by a Wiener-Hammerstein system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$) (d) Original input. (c) and (d) almost coincide. 28

Petitioners Exhibit 1005
Page 9 of 117

Figure 17 Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier itself is modeled by a memory polynomial. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$, odd order); (d) Output with memory polynomial predistortion ($Q = 2$, $K = 5$, even and odd orders); (e) Original input. 29

Figure 18 Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier is modeled by a perturbed Wiener (hence Volterra) system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$); (d) Output with memory polynomial predistortion ($Q = 10, K = 5$); (e) Original input. 29

Figure 19 Parallel Wiener model diagram. $H_i(\cdot)$ is an LTI block, and $F_i(\cdot)$ is a memoryless nonlinear block. 32

Figure 20 Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier is modeled by a parallel Wiener system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$); (d) Output with memory polynomial predistortion ($Q = 5, K = 5$); (e) Original input. . . 32

Figure 21 Parallel Hammerstein model diagram. $F_i(\cdot)$ is a memoryless nonlinear block, and $H_i(\cdot)$ is an LTI block. 34

Figure 22 The indirect learning architecture of the new combined predistorter model. 37

Figure 23 Block diagram of the predistorter model simulation. 41

Figure 24 Trajectories of (a) mean squared error, (b) c_l coefficients, (c) amplitudes of a_{kp} coefficients, and (d) amplitudes of b_q coefficients vs. number of iterations in the noiseless setting. 43

Figure 25 Trajectories of (a) mean squared error, (b) c_l coefficients, (c) amplitudes of a_{kp} coefficients, and (d) amplitudes of b_q coefficients vs. number of iterations in the noisy setting. 43

Figure 26 Comparison of the power spectrum of the noiseless predistorter model output $z_0(n)$ with the power spectra of noiseless residue $e_0(n)$ in noiseless and noisy settings. 44

Figure 27 Comparison of the power spectrum of the noiseless predistorter model output $z_0(n)$ with the power spectra of noiseless residue $e_0(n)$ for different initialization $\mathbf{c}$'s in the noisy setting. 44

Figure 28 Overall system performance of the least-squares/Newton algorithm on the predistortion testbed, showing the power spectra of the output signal (a) without predistortion, (b) with predistorter ($\mathbf{m}$ = [7 4 0 0]), (c) with predistorter ($\mathbf{m}$ = [7 4 7 3]), and (d) with predistorter ($\mathbf{m}$ = [7 4 7 35]). 46

Petitioners Exhibit 1005
Page 10 of 117

Figure 29 Fitting $F(|z(t)|)$ using polynomials of different orders. (a) and (b) show the real and imaginary parts of the measured $F(|z(t)|)$ and its polynomial approximations $\hat{F}(|z(t)|)$. (c) and (d) show the real and imaginary parts of the fitting error $\hat{F}(|z(t)|) - F(|z(t)|)$. In all figures, LUT refers to the measured power amplifier data, and the polynomial order sets $\mathcal{K}_1 = \{1,\ 3,\ 5\}$, $\mathcal{K}_2 = \{1,\ 3,\ 5,\ 7,\ 9\}$, $\mathcal{K}_3 = \{1,\ 2,\ 3,\ 4,\ 5\}$. 52

Figure 30 The predistorter precedes the power amplifier, and the objective is to have $y(t) \approx Cx(t)$, where C is a constant. 53

Figure 31 The errors $\hat{G}(|x(t)|) - G(|x(t)|)$ for three sets of polynomial orders $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$, $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$, and $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$. The real part of the error is shown in Figure (a), and the imaginary part of the error is shown in Figure (b). $\mathcal{L}_2$ and $\mathcal{L}_3$ resulted in comparable amount of error and both outperform $\mathcal{L}_1$. 56

Figure 32 Predistortion linearization performance in terms of spectral regrowth suppression. power amplifier output power spectral density (PSD) is shown for the following cases: (a) there is no predistorter; (b) predistorter (119) with $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$ is used; (c) predistorter (119) with $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$ or predistorter (119) with $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ is used (the two lines coincide). (d) PSD of the original input. 57

Figure 33 Predistortion linearization performance in terms of spectral regrowth suppression. Power amplifier output power spectral density (PSD) is shown for the following cases: (a) there is no predistorter; (b) predistorter (123) with $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$ and $Q = 4$ is used; (c) predistorter (123) with $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$ and $Q = 4$ is used; (d) predistorter (123) with $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ and $Q = 4$ is used. (e) PSD of the original input. 60

Figure 34 Two-stage upconversion transmitter. 62

Figure 35 Block diagram of the baseband predistortion system with equalization. The dashed lines refer to the feedback path for equalizer training. 62

Figure 36 Comparison between noncausal system response $H(e^{j\omega})$ (solid line) and causal system response $H_{\rm c}(e^{j\omega})$ (dotted-line). Note that the mean slopes of the phase responses have been removed. The phase responses of $H(e^{j\omega})$ and $H_{\rm c}(e^{j\omega})$ coincide. 65

Figure 37 Block diagram of the test bed. 68

Figure 38 Example of a training signal from predistorted waveform with the equalizer turned off. 69

Figure 39 Magnitude and phase responses of the estimated channel (solid line), the equalizer (dotted line), and the overall cascade of the channel and the equalizer (dashed line). The estimated channel has 51 taps. The equalizer has 14 taps and is designed with $\omega_{\rm p}= 0.7\pi$, $w = 10^{-3}$. 69

Petitioners Exhibit 1005
Page 11 of 117

Figure 40 Comparison of power spectral densities (PSDs). (a) Training signal; (b) Power Amplifier output with predistortion but no equalizer; (c) PA output with predistortion and equalizer; (d) PA output with the same predistorter as in (c) but no equalizer. 70

Figure 41 Block diagram of the baseband predistortion system. The dashed lines refer to the feedback loop for I/Q compensator training. 73

Figure 42 Detailed block diagram of the path from the DAC input $x(n)$ to the baseband feedback data samples $y(n)$. 73

Figure 43 Block diagrams of three channel models: (a) real I/Q model; (b) complex I/Q model; (c) direct/image model. 74

Figure 44 Cascade of the I/Q compensator and the channel. 79

Figure 45 Comparison of the direct upconverter outputs without I/Q compensation and with different I/Q compensators. (a) Without I/Q compensation; (b) With a 1-tap I/Q compensator; (c) With a 9/9 I/Q compensator constructed using the two-step approach; (d) With a 9-tap I/Q compensator constructed using the one-step approach; (e) Original input. Here, (c), (d), and (e) coincide. 85

Figure 46 Comparison of the power amplifier output with different I/Q compensators and predistorters. (a) Without predistortion but with a 9-tap I/Q compensation from the one-step approach; (b) With predistortion and a 1-tap I/Q compensator; (c) With predistortion and a 9/9 I/Q compensator from the two-step approach; (d) With predistortion and a 9-tap I/Q compensator from the one-step approach; (e) Original input. Here, (c), (d), and (e) almost coincide. 86

Figure 47 The indirect learning architecture for the predistorter. 88

Figure 48 Condition number of the correlation matrix with different Q values and different input signals: (a) three-carrier WCDMA with $K = 5$ conventional polynomials; (b) three-carrier WCDMA with $K = 5$ orthogonal polynomials; (c) a complex random signal (amplitude uniformly distributed in [0,1]) with $K = 5$ conventional polynomials; (d) a complex random signal (amplitude uniformly distributed in [0,1]) with $K = 5$ orthogonal polynomials. 91

Figure 49 Flow chart of the algorithm. 93

Figure 50 Block diagram of the testbed. 94

Figure 51 Measured power amplifier output PSD: (a) without predistortion; (b) with $K = 5$ memoryless predistorter trained by 5,000 data samples; (c) with $K = 5$ memoryless predistorter trained by 20,000 data samples. (b) and (c) coincide. 96

Figure 52 Measured power amplifier output PSD: (a) without predistortion; (b) with $K = 5$, $Q = 4$ memory polynomial predistorter trained by 5,000 data samples; (c) with $K = 5$, $Q = 4$ memory polynomial predistorter trained by 20,000 data samples. 97

Petitioners Exhibit 1005
Page 12 of 117

SUMMARY

Power amplifiers are essential components in communication systems and are inherently nonlinear. The nonlinearity creates spectral growth (broadening) beyond the signal bandwidth, which interferes with adjacent channels. It also causes distortions within the signal bandwidth, which decreases the bit error rate at the receiver. Newer transmission formats, such as wideband code division multiple access (WCDMA) or orthogonal frequency division multiplexing (OFDM), are especially vulnerable to the nonlinear distortions due to their high peak-to-average power ratios (PAPRs). If we simply back-off the input signal to achieve the linearity required for the power amplifier, the power amplifier efficiency will be very low for high PAPR signals.

Another choice is to linearize a nonlinear power amplifier so that overall we have a linear and reasonably efficient device. Digital predistortion is one of the most cost effective ways among all linearization techniques. However, most of the existing designs treat the power amplifier as a memoryless device. For wideband or high power applications, the power amplifier exhibits memory effects, for which memoryless predistorters can achieve only limited linearization performance.

In this dissertation, we propose novel predistorters and their parameter extraction algorithms. We investigate a Hammerstein predistorter, a memory polynomial predistorter, and a new combined model based predistorter. The Hammerstein predistorter is designed specifically for power amplifiers that can be modeled as a Wiener system. The memory polynomial predistorter can correct both the nonlinear distortions and the linear frequency response that may exist in the power amplifier. It is a robust predistorter, which has demonstrated good performance on several nonlinear system models. Real-time implementation aspects of the memory polynomial predistorter are also investigated in the dissertation. The new combined model includes the memory polynomial model and the Murray Hill model, thus extending the predistorter's ability to compensate for strong memory effects in

Petitioners Exhibit 1005
Page 13 of 117

the power amplifier. Performance of the new model is demonstrated through experimental measurements.

The predistorter models considered in this dissertation include both even- and odd-order nonlinear terms. In the literature, most of the power amplifier and predistorter models consider only the odd-order terms. Here, we show that it is beneficial to include even-order nonlinear terms in both the baseband power amplifier and predistorter models. By including these even-order nonlinear terms, we have a richer basis set, which offers appreciable improvement.

The ideal performance of digital predistortion certainly relies on robust predistorters that can completely compensate for the nonlinearities of the power amplifier. In reality, however, the performance can also be affected by the analog imperfections in the transmitter, which are introduced by the analog components; mostly analog filters and quadrature modulators. There are two common configurations for the upconversion chain in the transmitter: two-stage upconversion and direct upconversion. For a two-stage upconversion transmitter, we design a band-limited equalizer to compensate for the frequency response of the surface acoustic wave (SAW) filter which is usually employed in the IF stage. For a direct upconversion transmitter, we develop a model to describe the frequency-dependent gain/phase imbalance and dc offset. We then develop two methods to construct compensators for the imbalance and dc offset. These compensation techniques help to correct for the analog imperfections, which in turn improve the overall predistortion performance.

Petitioners Exhibit 1005
Page 14 of 117

CHAPTER 1

INTRODUCTION

1.1 Motivation

Power amplifiers are indispensable components in a communication system and are inherently nonlinear. The nonlinearity generates spectral regrowth, which leads to adjacent channel interference and violations of the out-of-band emission requirements mandated by regulatory bodies. It also causes in-band distortion, which degrades the bit error rate (BER) performance.

To reduce the nonlinearity, the power amplifier can be backed off to operate within the linear portion of its operating curve. However, newer transmission formats, such as wideband code division multiple access (WCDMA) and orthogonal frequency division multiplexing (OFDM), have high peak to average power ratios, i.e., large fluctuations in their signal envelopes. This means that the power amplifier needs to be backed off far from its saturation point, which results in very low efficiencies, typically less than 10% [50]; i.e., more than 90% of the dc power is lost and turns into heat. Considering the large number of wireless base stations deployed worldwide, improved power amplifier efficiency can substantially reduce the electricity and cooling costs incurred to the service providers. To improve the power amplifier efficiency without compromising its linearity, power amplifier linearization is essential.

Among all linearization techniques, digital predistortion is one of the most cost effective (see Fig. 1). It adds a digital predistorter in the baseband to create an expanding nonlinearity that is complementary to the compressing characteristic of the power amplifier. Ideally, the cascade of the predistorter and the power amplifier becomes linear and the original input is amplified by a constant gain. With the predistorter, the power amplifier can be utilized up to its saturation point while still maintaining a good linearity, thereby significantly increasing its efficiency. In reality, the power amplifier characteristics may change

Petitioners Exhibit 1005
Page 15 of 117

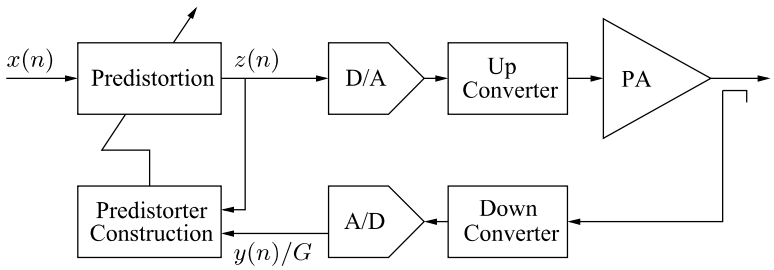


Figure 1: Digital Predistortion System Diagram

over time because of temperature drift, component aging, etc. Therefore, the predistorter should also have the ability to adapt to these changes.

Digital predistortion implementations in the current literature mostly focus on the power amplifier that has a memoryless nonlinearity; i.e., the current output depends only on the current input through a nonlinear mechanism. This instantaneous nonlinearity is usually characterized by the AM/AM and AM/PM responses of the power amplifier, where the output signal amplitude and phase deviation of the power amplifier output are given as functions of the amplitude of its current input. There has been intensive research on predistortion techniques for memoryless power amplifiers during the past decade [28].

As the signal bandwidth gets wider, such as in WCDMA, power amplifiers begin to exhibit memory effects. This is especially true for those high power amplifiers used in wireless base stations. The causes of the memory effects can be attributed to thermal constants of the active devices or components in the biasing network that have frequency-dependent behaviors [47]. As a result, the current output of the power amplifier depends not only on the current input, but also on past input values. In other words, the power amplifier becomes a nonlinear system with memory. For such a power amplifier, memoryless predistortion can achieve only very limited linearization performance [29], [17]. Therefore, digital predistorters also need to have memory structures. This dissertation investigates robust predistorter models that are capable of linearizing power amplifiers with memory effects. It also investigates system implementation issues related to these wideband digital predistortion systems.

Petitioners Exhibit 1005
Page 16 of 117

1.2 Objectives

The objective of this dissertation is to develop digital predistortion systems for linearization of power amplifiers with memory effects. Our research efforts focus on three areas:

- Predistorter models with memory structures;
- Digital compensation techniques of analog imperfections in the transmitters;
- Wideband digital predistortion testbed.

Volterra series is a general nonlinear model with memory. However, the large number of coefficients in the Volterra series makes it unattractive for practical applications. Instead, several special cases of the Volterra series are considered in this dissertation, which include the Hammerstein model [25], the memory polynomial model [30], the Murray Hill model [32], and possible combinations of these models.

The ideal performance of digital predistortion certainly relies on robust predistorters that can completely compensate for the nonlinearities in the power amplifier. In reality, however, the performance can also be affected by the analog imperfections in the transmitter, which are introduced by the analog components, such as mixers, analog filters, and quadrature modulators. The second focus of this dissertation is to investigate modeling and compensation techniques for these imperfections.

In this dissertation, a wideband digital predistortion testbed is also developed to evaluate the performance of digital predistortion systems on real power amplifiers.

1.3 Outline

The dissertation is organized as follows:

Chapter 2 reviews the literature in the field of modeling and predistortion of power amplifiers. A memoryless power amplifier can be characterized by its AM/AM and AM/PM responses. To linearize such a power amplifier, a memoryless predistorter is sufficient. For a power amplifier with memory effects, various models are available to capture the behavior of the power amplifier, which include the Volterra series, the Wiener model, the Hammerstein

Petitioners Exhibit 1005
Page 17 of 117

model, and the Wiener-Hammerstein model. The indirect learning architecture is then presented to construct the predistorter for a power amplifier with memory effects.

In Chapter 3, we present novel predistorters and their parameter extraction algorithms. A Hammerstein predistorter, a memory polynomial predistorter, and a new combined model based predistorter are considered. The parameters of these predistorters are extracted using the indirect learning architecture, eliminating the need for model assumption and parameter extraction of the power amplifier. Performance of these predistorters are demonstrated through computer simulations and/or experimental measurements.

Most existing literature considers only odd-order nonlinear terms when modeling power amplifiers and designing predistorters in the baseband. We show in Chapter 4 that it is beneficial to include even-order nonlinear terms in the baseband PA as well as predistorter models. By including these even-order nonlinear terms, we have a richer basis set, which offers appreciable improvement.

In Chapter 5, we study the analog imperfections in the transmitter and design compensation techniques. For a two-stage upconversion transmitter, we design a band-limited equalizer to compensate for the frequency response of the SAW filter, which is usually employed in the IF stage. For a direct upconversion transmitter, we develop a model to describe the frequency-dependent gain/phase imbalance and dc offset. We then develop two methods to construct compensators for the imbalance and dc offset. These compensation techniques help to correct for the analog imperfections, which in turn improve the overall predistortion performance.

In a practical implementation, predistorter training is performed on a digital signal processor, such as the Texas Instruments TMS320C6711. In Chapter 6, we investigate real-time implementation aspects of the memory polynomial predistorter. We implement the predistorter training algorithm on a Texas Instruments TMS320C6711 processor and evaluate the performance of the trained predistorter on our wideband digital predistortion testbed.

Finally, Chapter 7 concludes the dissertation and provides future research directions.

Petitioners Exhibit 1005
Page 18 of 117

CHAPTER 2

BACKGROUND

In this chapter, we review modeling techniques and predistorter design for memoryless power amplifiers, as well as power amplifiers with memory effects.

2.1 Modeling Memoryless Power Amplifiers

In the passband, a strictly memoryless power amplifier can be described as a nonlinear function that maps a real valued input to a real valued output. Over a closed interval for $\tilde{z}(t)$, this memoryless nonlinearity can be approximated by a power series; i.e.,

$$\tilde{y}(t) \;=\; \sum_{k=1}^{K} \tilde{b}_k \, \tilde{z}^k(t), \tag{1}$$

where $\tilde{b}_k$ are real-valued coefficients, $\tilde{z}(t)$ is the passband power amplifier input, and $\tilde{y}(t)$ is the passband power amplifier output. In the baseband, (1) becomes

$$y(t) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \, z(t) \, |z(t)|^{k-1} \tag{2}$$

[3, p. 69], [39], where

$$b_k \;=\; 2^{1-k} \binom{k}{\frac{k-1}{2}} \tilde{b}_k, \tag{3}$$

$z(t)$ is the baseband power amplifier input, and $y(t)$ is the baseband power amplifier output. The first observation from (2) is that it only contains odd order terms. This is because the signals generated from the even order terms in (1) are far from the carrier frequency. Thus, they do not contribute to the baseband output $y(t)$. The second observation is that b_k are real valued since $\tilde{b}_k$ are real valued. Therefore, if the power amplifier is strictly memoryless, it only introduces amplitude distortion to the input signal, giving rise to the so called AM/AM conversion of the power amplifier.

Petitioners Exhibit 1005
Page 19 of 117

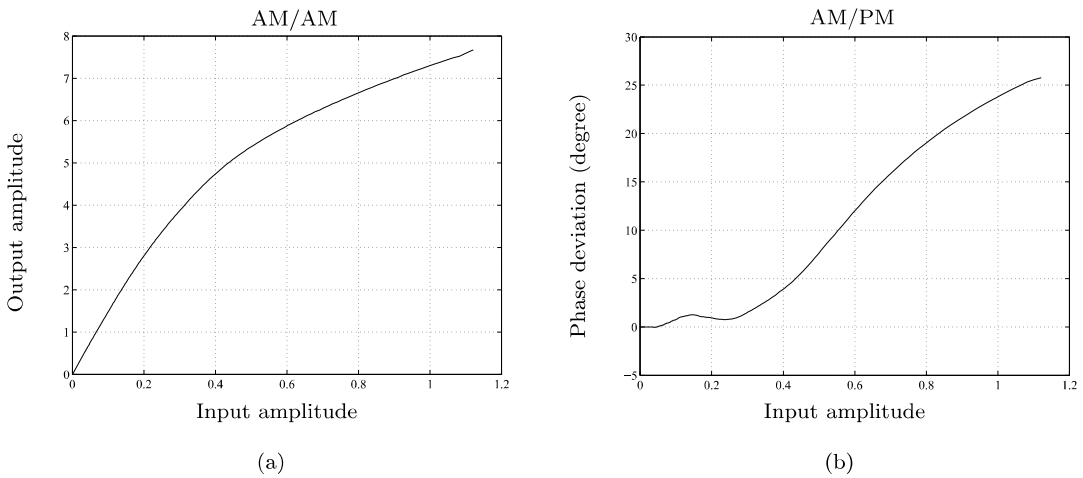


Figure 2: The AM/AM and AM/PM responses of a Class AB power amplifier.

Interestingly, if b_k in (2) are allowed to be complex, (2) can represent a much larger class of power amplifiers, often referred to as quasi-memoryless power amplifiers. In the passband, a nonlinear power amplifier with memory can be approximated by the Volterra series; i.e.,

$$\tilde{y}(t) = \sum_k \int \cdots \int \tilde{h}_k(\boldsymbol{\tau}_k) \prod_{i=1}^{k} \tilde{z}(t-\tau_i)\, \mathrm{d}\boldsymbol{\tau}_k, \tag{4}$$

where $\boldsymbol{\tau}_k = [\tau_1, \ldots, \tau_k]^T$, $\tilde{h}_k(\cdot)$ is the real-valued kth-order Volterra kernel, and $\mathrm{d}\boldsymbol{\tau}_k = \mathrm{d}\tau_1 \mathrm{d}\tau_2 \cdots \mathrm{d}\tau_k$. An important special case here is when the power amplifier in the passband has short-term memory; i.e., the time span of the memory is short compared to the time variations of the input signal envelope. With this assumption, it is shown in [39] that the baseband version of (4) has the same form as (2) except that b_k are complex valued. In this case, besides amplitude distortion, the power amplifier also introduces phase distortion to the input signal, which leads to the nonconstant AM/PM conversion of the power amplifier. However, the baseband representation in this case is still memoryless.

In summary, power amplifiers that are strictly memoryless or quasi-memoryless can be characterized by their AM/AM and AM/PM conversions. As an example, the AM/AM and AM/PM responses of a memoryless Class AB power amplifier are shown in Fig. 2.

Petitioners Exhibit 1005
Page 20 of 117

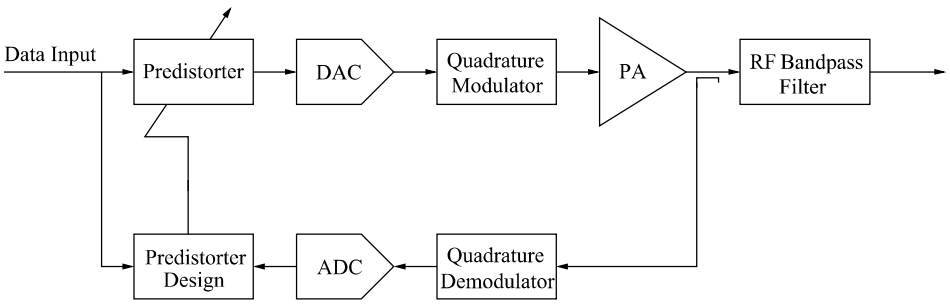


Figure 3: Block diagram of a data predistortion system with the pulse shaping filter implemented after the power amplifier.

2.2 Predistortion of Memoryless Power Amplifier

In the current literature, digital predistortion implementations mostly focus on memoryless power amplifiers.

2.2.1 Data Predistortion for Memoryless Power Amplifiers

Early digital predistorters mainly fall into the data predistorter category in the sense that predistortion is applied directly to each of the input signal constellation points. Depending on the location of the pulse shaping filter in the transmitter, there are two types of data predistorters.

The first type, exemplified by [24], [41], implements the pulse shaping filter using a radio frequency (RF) bandpass filter after the power amplifier. The schematic diagram of this type of predistortion system is shown in Fig. 3. Since the power amplifier is memoryless and there is no filtering between the predistorter and the power amplifier, it is sufficient to use a memoryless data predistorter here. The predistorter can be easily implemented as look-up tables (LUTs) that map the original input constellation points to the desired locations. Because of the small size of the input levels, this type of predistorters converges fast and requires very little memory. However, RF bandpass filters with sharp cut-offs are difficult to obtain and have relative large losses, thereby making this structure unattractive.

The second type, exemplified by [26], [27], [20], considers the case where the pulse shaping filter is placed in the baseband, immediately after the data predistorter (see Fig. 4). In such an arrangement, a memoryless data predistorter is not sufficient to fully linearize

Petitioners Exhibit 1005
Page 21 of 117

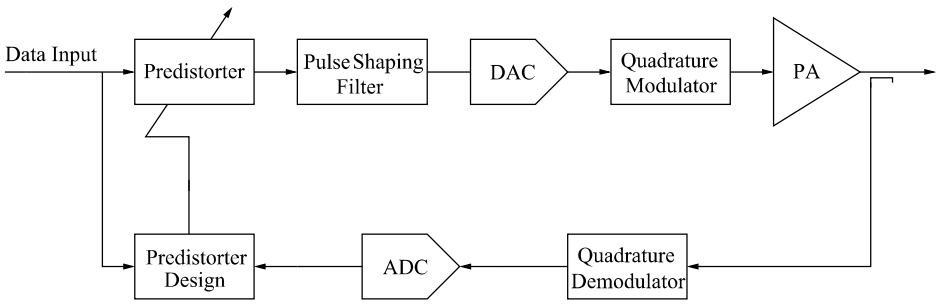


Figure 4: Block diagram of a data predistortion system with the pulse shaping filter placed in the baseband after the predistorter.

the memoryless power amplifier since it compensates only for those signal levels appearing in the signal constellation. The approach taken by [26] uses a specific pulse shaping filter to generate discrete values at two or three equally spaced data points per symbol interval. Each data point is then predistorted by a memoryless predistorter and combined with adjacent data points to reduce the nonlinear distortion at the power amplifier output. In [27], the current data input is considered together with its previous and following inputs as a signal point of a larger signal constellation, which is then predistorted by a memoryless data predistorter. Eun and Powers [20] proposed a Volterra series based data predistorter to compensate for the cascade of the pulse shaping filter and the power amplifier. The predistorter is trained using the indirect learning architecture, where the desired power amplifier output is set to be the original data after pulse shaping.

The main drawback of data predistorters is their dependence on the signal constellation and the pulse shaping filter. Moreover, data predistorters do not work well if the processing produces almost continuous input signal levels, e.g., in OFDM or WCDMA.

2.2.2 Signal Predistortion for Memoryless Power Amplifiers

To overcome these limitations, recent digital predistorters have usually been applied at the last stage of the baseband processing (see Fig. 5). In contrast to data predistorters, these predistorters can deal with arbitrary input waveforms, thus they are referred to as signal predistorters here. They can be divided into three categories [44]: mapping structure, polar structure, and complex gain structure.

The first signal predistorter for memoryless power amplifiers was proposed by Nagata

Petitioners Exhibit 1005
Page 22 of 117

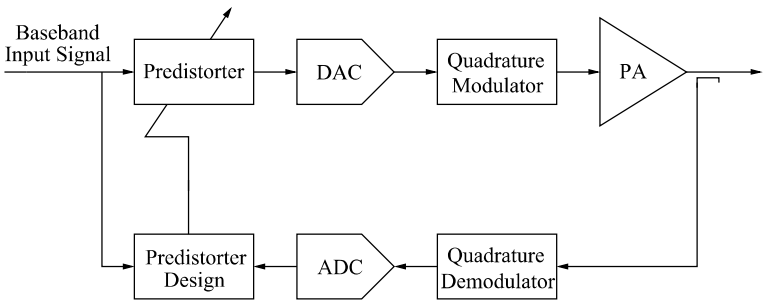


Figure 5: Block diagram of a signal predistortion system.

[36] using a mapping structure. Given a complex baseband input signal $x(t)$, the predistorter generates a complex correction signal $\Delta[x(t)]$ from a two-dimensional LUT indexed by the real and imaginary parts of $x(t)$. The predistorted signal $z(t)$ is given by

$$z(t) \quad = \quad x(t) + \Delta[x(t)]. \tag{5}$$

Thus, the predistorter maps each complex input point to its desired location, which is a generalization of the first type of data predistorters in Section 2.2.1. This nonlinear mapping can also correct for other types of memoryless distortions in the transmitter, such as the gain/phase imbalance in the quadrature modulator. The drawback of this approach is the large LUT size since the LUT needs to be two-dimensional and cover a large number of input levels.

Considering that the AM/AM and AM/PM responses of the memoryless power amplifier depend only on the input amplitude, the two-dimensional LUT in the mapping predistorter can actually be replaced by two one-dimensional LUTs. Indexed by the input amplitude $r_x(t)$, the one-dimensional LUTs specify the desired output amplitude, $A[r_x(t)]$, and the phase shift, $\phi_A[r_x(t)]$; i.e., the predistorted output is given by

$$z(t) \quad = \quad A[r_x(t)] \; e^{j\,\{\phi_x(t)+\phi_A[r_x(t)]\}}, \tag{6}$$

where $\phi_x(t)$ is the pase of the input signal $x(t)$. Since the predistortion LUTs are in polar form, this predistorter, proposed by by Faulkner *et al.* [22], is often referred to as the polar structure predistorter. In practical implementations, phase calculation in the polar conversion for each input complex point is quite computationally intensive. Faulkner *et*

Petitioners Exhibit 1005
Page 23 of 117

al. [22] also provided an efficient implementation to avoid the polar conversion during predistortion, which is a rearrangement of (6); i.e.,

$$z(t) \quad = \quad x(t)\,\frac{A[r_x(t)]}{r_x(t)}\,e^{j\,\phi_A[r_x(t)]}, \tag{7}$$

where $\frac{A[r_x(t)]}{r_x(t)}$ and $e^{j\,\phi_A[r_x(t)]}$ are given by the gain table and phase table, respectively. The polar conversions are still needed when updating the LUTs, but the update can be done much more slowly.

An approach similar to the polar predistorter was proposed by Cavers [6]. It employs a complex gain table instead of the gain and phase tables in the polar predistorter; i.e.,

$$F[r_x(t)] \quad = \quad \frac{A[r_x(t)]}{r_x(t)}\,e^{j\phi_A[r_x(t)]}. \tag{8}$$

The predistorted output is then given by

$$z(t) \quad = \quad x(t)\,F[r_x(t)]. \tag{9}$$

This structure avoids the conversions between polar and Cartesian forms. Therefore, it is more computationally efficient. Furthermore, only one complex multiplication is needed in (9) as opposed to two real and one complex multiplications in (7).

2.3 Modeling Power Amplifiers with Memory Effects

In Section 2.1, it is shown that power amplifiers with short-term memory effects in the passband have memoryless baseband representations. However, as the input signal bandwidth becomes wider, such as in WCDMA, the time span of the power amplifier memory becomes comparable to the time variations of the input signal envelope. Thus, the memory effects of the power amplifier in the passband can no longer be considered as short-term. Without the short-term memory assumption, (4) gives the full baseband Volterra series [39], [4], which is

$$y(t) = \sum_k \int \cdots \int h_{2k+1}(\boldsymbol{\tau}_{2k+1}) \prod_{i=1}^{k+1} z(t-\tau_i) \prod_{i=k+2}^{2k+1} z^*(t-\tau_i)\,\mathrm{d}\boldsymbol{\tau}_{2k+1}, \tag{10}$$

where

$$h_{2k+1}(\boldsymbol{\tau}_{2k+1}) = \frac{1}{2^{2k}}\binom{2k+1}{k}\,\tilde{h}_{2k+1}(\boldsymbol{\tau}_{2k+1})\,e^{-j2\pi f_o\left(\sum_{i=1}^{k+1}\tau_i - \sum_{i=k+2}^{2k+1}\tau_i\right)}, \tag{11}$$

Petitioners Exhibit 1005
Page 24 of 117

$(\cdot)^*$ denotes complex conjugation, and f_o is the carrier frequency. In discrete-time domain, (10) becomes

$$y(n) = \sum_{k} \sum_{l_1} \cdots \sum_{l_{2k+1}} h_{2k+1}(l_1, l_2, \cdots, l_{2k+1}) \prod_{i=1}^{k+1} z(n-l_i) \prod_{i=k+2}^{2k+1} z^*(n-l_i). \quad (12)$$

From (12), it can be seen that the number of coefficients of the Volterra series increases exponentially as the memory length and the nonlinear order increase. This drawback makes the Volterra series unattractive for real-time applications. This prompts us to consider several special cases of the Volterra series. The special cases considered here include the Wiener model, the Hammerstein model, the Wiener-Hammerstein model, the memory polynomial model, and the Murray Hill model.

The Wiener model is a linear time-invariant (LTI) system followed by a memoryless nonlinearity (NL) (see Fig. 6). The two subsystems are given by

$$u(n) = \sum_{l=0}^{L-1} a_l \, z(n-l), \quad (13)$$

$$y(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \, u(n)|u(n)|^{k-1}, \quad (14)$$

where a_l are the impulse response values of the LTI block and b_k are the coefficients of the odd-order polynomial describing the memoryless nonlinearity. Substituting (13) into (14) gives

$$y(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \left[\sum_{l=0}^{L-1} a_l \, z(n-l) \right] \left| \sum_{l=0}^{L-1} a_l \, z(n-l) \right|^{k-1}. \quad (15)$$

The Wiener model was used by Clark *et al.* [11] to model the power amplifier with memory effects, where improvements in modeling accuracy were observed when the Wiener model replaces the memoryless polynomial model.

Petitioners Exhibit 1005
Page 25 of 117

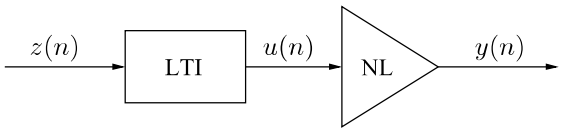


Figure 6: The Wiener Model.

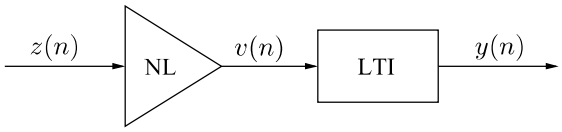


Figure 7: The Hammerstein Model.

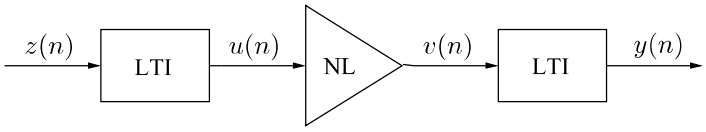


Figure 8: The Wiener-Hammerstein model.

Petitioners Exhibit 1005
Page 26 of 117

The Hammerstein model is a memoryless nonlinearity followed by an LTI system (see Fig. 7). The two subsystems in this model are described by

$$v(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \, z(n)|z(n)|^{k-1}, \tag{16}$$

$$y(n) = \sum_{l=0}^{L-1} c_l \, v(n-l), \tag{17}$$

where b_k are the coefficients for the memoryless nonlinearity and c_l are the impulse response values of the LTI system. Substitution of (16) into (17) leads to

$$y(n) = \sum_{l=0}^{L-1} c_l \sum_{\substack{k=0 \\ k \text{ odd}}}^{K} b_k \, z(n-l) \, |z(n-l)|^{k-1}. \tag{18}$$

The Wiener-Hammerstein (W-H) model (see Fig. 8) is an LTI system followed by a memoryless nonlinearity, which in turn is followed by another LTI system. Such a configuration is commonly used for satellite communication channels, where the power amplifier at the satellite transponder is driven near saturation to exploit the maximum power efficiency for the downlink [2]. The subsystems in this model are described by

$$u(n) = \sum_{l=0}^{L_a-1} a_l \, z(n-l), \tag{19}$$

$$v(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \, u(n)|u(n)|^{k-1}, \tag{20}$$

$$y(n) = \sum_{l=0}^{L_c-1} c_l \, v(n-l), \tag{21}$$

where a_l and c_l are, respectively, impulse response values of the LTI systems before and after the memoryless nonlinear block, and b_k are the coefficients of the nonlinear block. Combining (19), (20), and (21), we infer

$$y(n) = \sum_{l_1=0}^{L_c-1} c_{l_1} \sum_{\substack{k=0 \\ k \text{ odd}}}^{K} b_k \left[\sum_{l_2=0}^{L_a-1} a_{l_2} z(n-l_2-l_1) \right] \left| \sum_{l_2=0}^{L_a-1} a_{l_2} z(n-l_2-l_1) \right|^{k-1}. \tag{22}$$

The memory polynomial model uses the diagonal kernels of the Volterra series and can be viewed as a generalization of the Hammerstein model. In the discrete-time Volterra

Petitioners Exhibit 1005
Page 27 of 117

series (12), if $l_1 = \cdots = l_{2k+1} = l$, (12) becomes

$$y(n) = \sum_{\substack{p=1 \\ p \text{ odd}}}^{P} \sum_{l=0}^{L-1} b_{pl}\ z(n-l)|z(n-l)|^{p-1}, \tag{23}$$

where b_{pl} are equal to $h_{2k+1}(l, l, \cdots, l)$ in (12). This model was considered for modeling power amplifiers with memory effects in [30] and for data predistortion of the cascade of a pulse shaping filter and a memoryless power amplifier in [10].

The Murray Hill model, proposed by Ma *et al.* [32], introduces a set of nonlinear terms into the conventional memoryless polynomial model based on the underlying physical phenomena. For each input, these terms generate a complex gain that depends on the combination of the current and past input signal envelopes in a nonlinear fashion. The output of the model is given by

$$y(n) \quad = \quad \sum_{p=1}^{P} a_p\, z(n)|z(n)|^{p-1} + \sum_{q=2}^{Q} b_q\, z(n) \left[\sum_{l=0}^{L-1} c_l\, |z(n-l)| \right]^{q-1}, \tag{24}$$

where a_p and b_q are complex polynomial coefficients and c_l are real valued memory filter coefficients. Note that the b_q coefficients start with b_2 instead of b_1. This is because the term associated with b_1, i.e., $x(n)$, is already taken into account in (24) by a_1. Moreover, the c_l coefficients are assumed to be real valued instead of complex valued for easier implementation. This model was used in [32] as a predistorter model and achieved very good performance on the power amplifiers under test.

There is another large class of power amplifier models [40] that is based on frequency-dependent AM/AM and AM/PM conversions of the power amplifier. However, they are usually obtained from single tone measurements and are difficult to extract from practical baseband inputs and outputs. Therefore, they are not considered here.

2.4 Predistortion of Power Amplifiers with Memory Effects

For power amplifiers with memory effects, memoryless predistortion can achieve only very limited linearization performance [29], [17]. Thus, digital predistorters also need to have memory structures.

Petitioners Exhibit 1005
Page 28 of 117

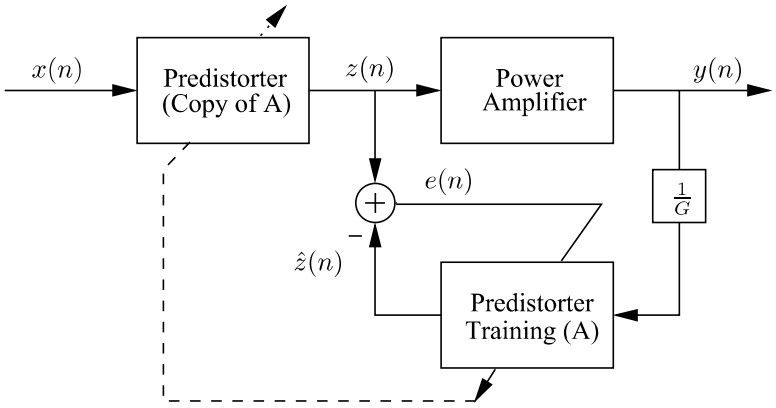


Figure 9: The indirect learning architecture for the predistorter.

In the current literature, digital predistorters with memory structures are mostly considered for data predistortion of the cascade of a pulse shaping filter and a memoryless power amplifier, as reviewed in Section 2.2.1. The models that have been considered for these predistorters include the Volterra series [20], [21], the Hammerstein model [25], and the memory polynomial model [10]. One exception is [32], in which Ma *et al.* applied a Murray Hill model based predistorter to a power amplifier with memory effects and achieved good linearization performance.

To construct digital predistorters with memory structures, there are two types of approaches. One type of approach is to first identify the power amplifier and then find the inverse of the power amplifier, which was used in [25]. However, obtaining the inverse of a nonlinear system with memory is generally a difficult task. Another type of approach is to use the indirect learning architecture to design the predistorter directly, as adopted in [20], [10]. The advantage of this type of approaches is that it eliminates the need for model assumption and parameter estimation of the power amplifier.

A block diagram of the indirect learning structure is shown in Fig. 9. The feedback path labeled "Predistorter Training" (block A) has $y(n)/G$ is its input, where G is the intended power amplifier gain, and $\hat{z}(n)$ as its output. The actual predistorter is an exact copy of the feedback path (copy of A); it has $x(n)$ as its input and $z(n)$ as its output. Ideally, we would like $y(n) = G\,x(n)$, which renders $z(n) = \hat{z}(n)$ and the error term $e(n) = 0$. Given $y(n)$ and $z(n)$, this structure enables us to find the parameters of block A directly, which yields the

Petitioners Exhibit 1005
Page 29 of 117

predistorter. The algorithm converges when the error energy $||e(n)||^2$ is minimized.

In general, power amplifier characteristics do not change rapidly with time; changes in the power amplifier characteristics are often due to temperature drift and aging, which have long time constants. After gathering a block of $y(n)$ and $z(n)$ data samples, the training branch (block A) can process the data off-line, which lowers the processing requirements of the predistortion system. Once the predistorter identification algorithm has converged, the new set of parameters is plugged into the high speed predistorter, which can be readily implemented using application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). When the predistorter coefficients have been found and it is believed that the power amplifier characteristics are hardly changing, the setup in Fig. 9 can be run in open loop; i.e., the training branch is temporarily shut down until changes in the power amplifier characteristics require a predistorter coefficient update.

Petitioners Exhibit 1005
Page 30 of 117

CHAPTER 3

DIGITAL PREDISTORTER DESIGN

Digital predistortion implementations in the current literature mostly focus on memoryless power amplifiers. For power amplifier with memory effects, memoryless predistortion can only achieve very limited linearization performance [29], [17]. Thus, digital predistorters also need to have memory structures. The most general way to introduce memory is to use the Volterra series, which has been considered in designing data predistorters [20], [21]. However, the large number of coefficients of the Volterra series makes it unattractive for practical applications. Therefore, we will investigate several special cases of the Volterra series, which include the Hammerstein model [25], the memory polynomial model [30], and the combination of the Memory polynomial and the Murray Hill models [32].

3.1 Hammerstein Predistorter Design

A Hammerstein predistorter is ideal for linearization of a Wiener system. The Wiener system can be either a power amplifier with memory effects [11] or the cascade of a pulse shaping filter and a memoryless power amplifier [25]. To construct a Hammerstein predistorter, the approach taken by [25] uses a gradient method to first identify the Wiener system and then find the Hammerstein predistorter as its inverse. An alternative approach is pursued here and [14], which generates the Hammerstein predistorter without first identifying the Wiener power amplifier by using the indirect learning architecture.

3.1.1 Hammerstein Predistorter Training

In the training branch (see Fig. 10), the Hammerstein predistorter is given by:

$$v(n) = \sum_{\substack{k=0 \\ k \text{ odd}}}^{K} c_k \, y(n) \, |y(n)|^{k-1} \tag{25}$$

$$z(n) = \sum_{p=1}^{P} a_p \, z(n-p) + \sum_{q=0}^{Q} b_q v(n-q), \tag{26}$$

Petitioners Exhibit 1005
Page 31 of 117

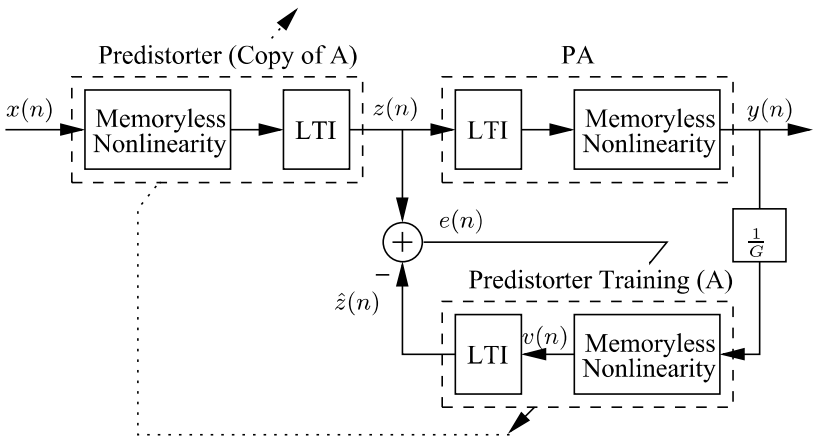


Figure 10: The indirect learning architecture for the Hammerstein predistorter.

where $y(n)$ and $z(n)$ are, respectively, the input and output of the predistorter in the training branch, c_k in (25) are the coefficients of the nonlinear block of the predistorter, and a_p and b_q in (26) are the coefficients of the LTI portion of the predistorter. Substituting (25) into (26), we obtain

$$z(n) \;=\; \sum_{p=1}^{P} a_p\, z(n-p) + \sum_{q=0}^{Q} b_q \left[\sum_{\substack{k=0 \\ k\,\text{odd}}}^{K} c_k\, y(n-q)\, |y(n-q)|^{k-1} \right]. \qquad (27)$$

In (25), the memoryless nonlinearity is modeled as an odd-order polynomial, and in (26), the LTI block is modeled as a general pole/zero system. Note that (27) generalizes the Hammerstein model described in Section 2.3 by using a pole/zero LTI system instead of an FIR LTI system.

Parameter estimation of the model in (27) is a classical Hammerstein system identification problem. If no additional assumptions are made on the system's input signal $y(n)$, iterative Newton and Narendra-Gallman algorithms are the two most popular iterative estimation methods [19]. The two algorithms exhibit similar performance as shown in [19]. The main drawback of these algorithms is that they are sensitive to the initial guess and may converge to a local minimum. A recent method proposed by Bai [1] uses a two stage least-squares/singular value decomposition (LS/SVD) algorithm, which can lead to a global optimum. Although the model structure considered in [1] is a Hammerstein system followed by a memoryless nonlinearity, the results there can be easily modified to suit the model in

Petitioners Exhibit 1005
Page 32 of 117

(27). Note that for a given set of $\{y(n),\ z(n)\}$ values, the b_q and c_k coefficients are not unique (i.e., multiplying b_q with a constant and dividing c_k by the same constant yields the same model). To avoid this problem, we constrain that

$$\sum_{q=0}^{Q} |b_q|^2 = 1$$

and the real part of b_0 is positive as suggested in [1].

Next, we will review the Narendra-Gallman (NG) and the optimal two stage identification (LS/SVD) algorithms.

Narendra-Gallman algorithm. The NG algorithm starts with initial guesses for the a_p and b_q coefficients, denoted by $a_p^{(0)}$ and $b_q^{(0)}$, respectively. At the ith iteration (27) can be rewritten as

$$\begin{aligned} z(n) - \sum_{p=1}^{P} a_p^{(i)} z(n-p) &= \sum_{k=0}^{(K-1)/2} c_{2k+1} u_{2k+1}(n) \qquad (28) \\ u_{2k+1}(n) &= \sum_{q=0}^{Q} b_q^{(i)} y(n-q)|y(n-q)|^{2k}. \end{aligned}$$

At this stage, our objective is to solve for c_{2k+1}. Using matrix notation we can reformulate (28) as

$$\mathbf{z}_0 - \mathbf{Z}\mathbf{a}^{(i)} = \mathbf{U}\mathbf{c}, \qquad (29)$$

where

$$\begin{aligned} \mathbf{Z} &= [\mathbf{z}_1, \ldots, \mathbf{z}_P], \\ \mathbf{z}_l &= [\mathbf{0}_l^T, z(1), \ldots, z(N-l)]^T \end{aligned}$$

with $\mathbf{0}_l$ is a $l \times 1$ all-zero vector, and

$$\begin{aligned} \mathbf{a}^{(i)} &= [a_1^{(i)}, \cdots, a_P^{(i)}]^T, \\ \mathbf{U} &= [\mathbf{u}_1, \cdots, \mathbf{u}_K], \\ \mathbf{u}_{2k+1} &= [u_{2k+1}(1), \cdots, u_{2k+1}(N)]^T, \\ \mathbf{c} &= [c_1, \cdots, c_K]^T. \end{aligned}$$

The least-squares solution for (29) is

$$\hat{\mathbf{c}}^{(i+1)} = (\mathbf{U}^H\mathbf{U})^{-1}\mathbf{U}^H\left(\mathbf{z}_0 - \mathbf{Z}\mathbf{a}^{(i)}\right), \qquad (30)$$

Petitioners Exhibit 1005
Page 33 of 117

where $(\cdot)^H$ denotes Hermitian transpose. In the second step, based on the $c_{2k+1}^{(i+1)}$'s obtained, we rewrite (27) as,

$$\mathbf{z}_0 = \mathbf{Z}\mathbf{a} + \mathbf{V}\mathbf{b} = [\mathbf{Z}\ \mathbf{V}] \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \tag{31}$$

where

$$\begin{aligned} \mathbf{V} &= [\mathbf{v}_0 \mathbf{v}_1, \cdots, \mathbf{v}_Q], \\ \mathbf{v}_l &= [\mathbf{0}_l^T, v(1), \cdots, v(N-l)]^T, \\ \mathbf{b} &= [b_0, \cdots, b_Q]^T, \end{aligned} \tag{32}$$

and $v(n)$ is given in (51). The least-squares solution for (31) is,

$$\begin{bmatrix} \hat{\mathbf{a}}^{(i+1)} \\ \hat{\mathbf{b}}^{(i+1)} \end{bmatrix} = \left([\mathbf{Z}\ \mathbf{V}]^H [\mathbf{Z}\ \mathbf{V}]\right)^{-1} [\mathbf{Z}\ \mathbf{V}]^H \mathbf{z}_0, \tag{33}$$

With the new $\hat{\mathbf{a}}^{(i+1)}$ and $\hat{\mathbf{b}}^{(i+1)}$ estimates, we can go back to the first step and continue until the algorithm converges.

Optimal two stage identification algorithm.

Since the difficulty in estimating the b_q's and c_{2k+1}'s is that they appear together as the coefficient on the r.h.s. of (27), if we define

$$d_{q,2k+1} = b_q\, c_{2k+1}, \tag{34}$$

we can first estimate $d_{q,2k+1}$ using least-squares and then find b_q and c_{2k+1} from $d_{q,2k+1}$. Substituting (34) into (27), we obtain

$$z(n) = \sum_{p=1}^{P} a_p z(n-p) + \sum_{q=0}^{Q} \sum_{k=0}^{(K-1)/2} d_{q,2k+1}\ g_{q,2k+1}(n), \tag{35}$$

where

$$g_{q,2k+1}(n) = y(n-q)|y(n-q)|^{2k}. \tag{36}$$

Rewriting in matrix form, we obtain

$$\begin{aligned} \mathbf{z}_0 &= \mathbf{Z}\mathbf{a} + \mathbf{G}\mathbf{d} \\ &= [\mathbf{Z}\ \mathbf{G}] \begin{bmatrix} \mathbf{a} \\ \mathbf{d} \end{bmatrix}, \end{aligned} \tag{37}$$

Petitioners Exhibit 1005
Page 34 of 117

where

$$\begin{aligned} \mathbf{G} &= [\mathbf{g}_{01}, \cdots, \mathbf{g}_{0K}, \cdots, \mathbf{g}_{Q1}, \cdots, \mathbf{g}_{QK}], \\ \mathbf{g}_{q,2k+1} &= [g_{q,2k+1}(1), \cdots, g_{q,2k+1}(N)]^T, \\ \mathbf{d} &= [d_{01}, \cdots, d_{0K}, \cdots, d_{Q1}, \cdots, d_{QK}]^T. \end{aligned} \tag{38}$$

The least-squares solution for (37) is

$$\begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{d}} \end{bmatrix} = \left([\mathbf{Z}\ \mathbf{G}]^H [\mathbf{Z}\ \mathbf{G}]\right)^{-1} [\mathbf{Z}\ \mathbf{G}]^H \mathbf{z}_0, \tag{39}$$

Equation (34) can be alternatively expressed as

$$\mathbf{D} = \begin{bmatrix} d_{01} & d_{03} & \cdots & d_{0K} \\ d_{11} & d_{13} & \cdots & d_{1K} \\ \vdots & \vdots & & \vdots \\ d_{Q1} & d_{Q3} & \cdots & d_{QK} \end{bmatrix} = \mathbf{b}\,\mathbf{c}^T,$$

where $\mathbf{b} = [b_0, \ldots, b_Q]^T$, $\mathbf{c} = [c_1, \ldots, c_K]^T$. Since the matrix $\mathbf{D}$ has rank one, a natural way to estimate $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ from $\hat{\mathbf{D}}$ is to perform a singular value decomposition (SVD) on $\hat{\mathbf{D}}$ and then find the eigenvectors corresponding to the largest singular value. Let the SVD of $\hat{\mathbf{D}}$ be given by,

$$\hat{\mathbf{D}} = \sum_{i=1}^{\min[(Q+1),(K+1)/2]} \sigma_i\, \boldsymbol{\mu}_i\, \boldsymbol{\nu}_i^H, \tag{40}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\nu}_i$ are $Q+1$ and $(K+1)/2$ dimensional orthonormal vectors, respectively. Then $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ can be estimated as

$$\hat{\mathbf{b}} = s_\mu\, \boldsymbol{\mu}_1 \quad , \quad \hat{\mathbf{c}} = s_\mu\, \sigma_1\, \boldsymbol{\nu}_1^*, \tag{41}$$

where * denotes complex conjugate and s_μ is the first non-zero element of $\boldsymbol{\mu}_1$. These estimates can be shown to be the closest $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$ to $\hat{\mathbf{D}}$ in the least-squares sense [1].

In summary, the NG algorithm is a simple and robust algorithm. Although it may have convergence problems, it can perform well in many cases as will be shown in the next section. The LS/SVD algorithm avoids the potential local minimum problem of the NG algorithm. However, using SVD to find the b_q's and c_{2k+1}'s may not result in the best b_q's

Petitioners Exhibit 1005
Page 35 of 117

and c_{2k+1}'s that minimize the squared error criterion. Our examples in the next section will show that both work well for identifying the Hammerstein predistorter although one may outperform the other in a particular scenario.

3.1.2 Hammerstein Predistorter Simulation

The performance of the Hammerstein predistorter identified using the indirect learning architecture is illustrated in following examples.

Example 3.1 The LTI portion of the Wiener power amplifier model has a pole/zero form, whose system function is given by

$$H(z) = \frac{1 + 0.3z^{-2}}{1 - 0.2z^{-1}}. \tag{42}$$

For the memoryless nonlinear portion of the Wiener power amplifier model, (25) is assumed with $K = 5$ and

$$\begin{aligned} c_1 &= 14.9740 + 0.0519j \\ c_3 &= -23.0954 + 4.9680j \\ c_5 &= 21.3936 + 0.4305j, \end{aligned} \tag{43}$$

which were extracted from an actual Class AB power amplifier.

The baseband input signal was a 3-carrier WCDMA signal. Hammerstein predistorter identification was carried out based on 8000 data samples. Usually within a few iterations, the predistorter parameter estimation algorithm converges. The power spectral densities (PSDs) of the input and output signals were then compared to assess the effectiveness of the predistorter in reducing spectral regrowth. In this example, the LTI portion of the Hammerstein predistorter is assumed to be a pole/zero system with two poles and one zero (correct model orders for the inverse of $H(z)$ in (42)). The nonlinear block of the predistorter uses a 5th odd-order polynomial.

The performance of the predistorter identified with the LS/SVD algorithm [1] and the Narendra-Gallman (NG) algorithm [19] is demonstrated in Fig. 11. Both algorithms fully suppress the spectral regrowth exhibited in power amplifier output. In contrast, it can

Petitioners Exhibit 1005
Page 36 of 117

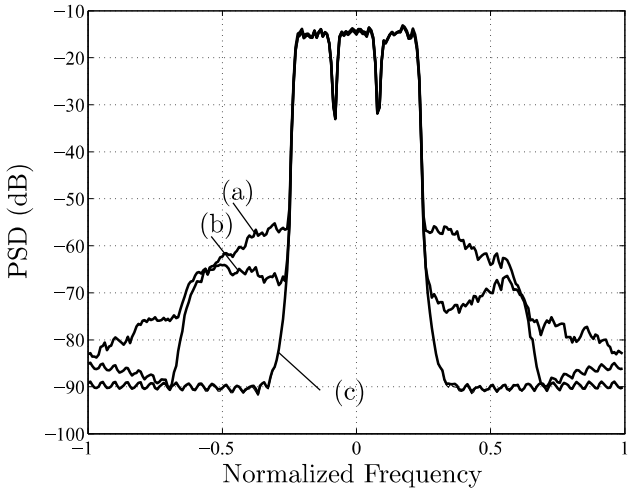


Figure 11: Comparison of the PSDs for the pole/zero Wiener power amplifier and the pole/zero Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion, NG and LS/SVD algorithms (similar performance).

be observed in Fig. 11 that 5th order memoryless predistortion cannot fully suppress the spectral regrowth.

Example 3.2 The LTI portion of the Wiener power amplifier is

$$H(z) \quad = \quad 1 + 0.3z^{-2} \tag{44}$$

(FIR), and the LTI portion of the Hammerstein predistorter is assumed to be FIR as well. The objective here is to see whether the algorithms can correctly identify an FIR filter that approximates the inverse of the FIR system in the power amplifier. When the FIR system in the predistorter has 15 taps, the performance of the predistorter is shown in Fig. 12. In this case, the NG algorithm performs worse than the LS/SVD algorithm. When examining the concatenated response of the two LTI blocks (one from the Wiener power amplifier and the other from the Hammerstein predistorter), it is observed that the predistorter's LTI system identified by the NG algorithm can only compensate for the power amplifier's LTI system within the signal bandwidth. However, the LS/SVD algorithm is able to find a good FIR system for the predistorter, both within and outside of the signal bandwidth.

Example 3.3 The Hammerstein predistorter was used to predistort a perturbed Wiener

Petitioners Exhibit 1005
Page 37 of 117

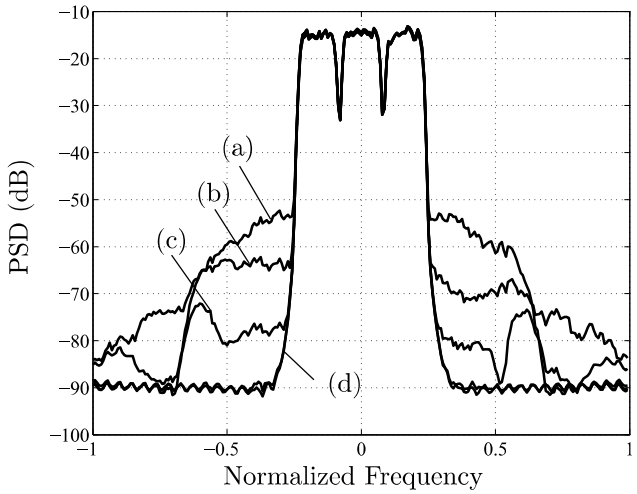


Figure 12: Comparison of the PSDs for FIR Wiener power amplifier and 15-tap FIR Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion (NG); (d) Output with Hammerstein predistortion (LS/SVD).

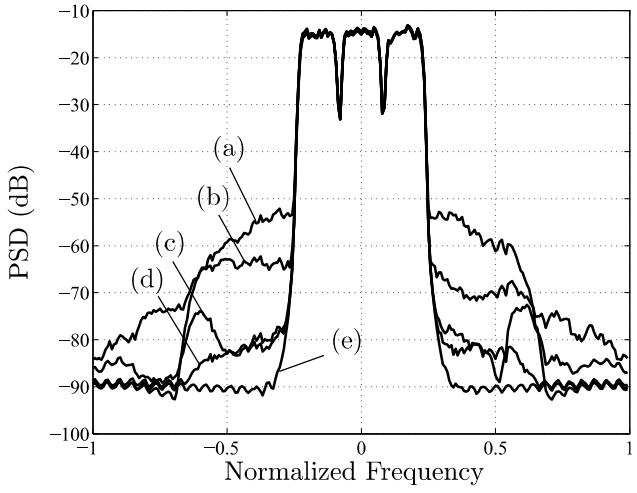


Figure 13: Comparison of the PSDs for full Volterra power amplifier and 15-tap FIR Hammerstein predistorter. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with Hammerstein predistortion (NG); (d) Output with Hammerstein predistortion (LS/SVD); (e) Input signal.

Petitioners Exhibit 1005
Page 38 of 117

power amplifier to test the robustness of the predistorter. The perturbed Wiener power amplifier was constructed from the Wiener power amplifier in Example 3.2. First the Volterra kernels of the Wiener power amplifier was found. Then zero mean complex Gaussian noise with variance 2×10^{-4} was added to the Volterra kernels, which turns the original Wiener model into a full Volterra system. From Fig. 13, it can be seen that although the power amplifier is not exactly a Wiener system, significant reduction of spectral regrowth can still be obtained by using the Hammerstein predistorter.

In all three examples, memoryless predistortion is not very effective in suppressing spectral regrowth, which underscores the notion that power amplifier memory effects must be taken into account when designing the predistorter.

3.2 Memory Polynomial Predistorter Design

A memory polynomial predistorter uses the diagonal kernels of the Volterra series and can be viewed as a generalization of the Hammerstein predistorter. In this section [17, 18], the memory polynomial predistorter is used to linearize power amplifiers with memory effects. The predistorter is constructed using the indirect learning architecture, thereby eliminating the need for model assumption and parameter estimation of the power amplifier. Comparing with the Hammerstein predistorter, the memory polynomial predistorter has slightly more terms. However, it is much more robust and its parameters can be easily estimated by way of least-squares.

3.2.1 Memory Polynomial Predistorter Training

In the training branch (see Fig. 14), the memory polynomial predistorter can be described by

$$z(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} \sum_{q=0}^{Q} a_{kq}\, y(n-q)|y(n-q)|^{k-1}, \tag{45}$$

where $y(n)$ and $z(n)$ are, respectively, the input and output of the predistorter in the training branch, and a_{kq} are the coefficients of the predistorter. Since the model in (45) is linear with respect to its coefficients, the predistorter coefficients a_{kq} can be directly obtained by

Petitioners Exhibit 1005
Page 39 of 117

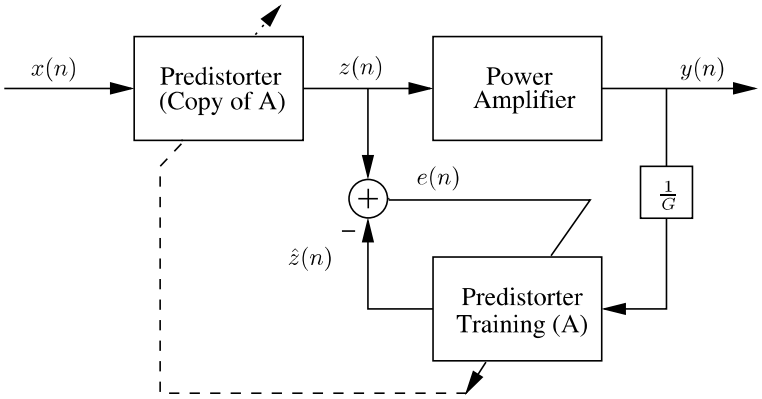


Figure 14: The indirect learning architecture for the memory polynomial predistorter.

least-squares. By defining a new sequence

$$u_{kq}(n) \quad = \quad \frac{y(n-q)}{G}\left|\frac{y(n-q)}{G}\right|^{k-1}, \tag{46}$$

at convergence, we should have

$$\mathbf{z} \quad = \quad \mathbf{U}\,\mathbf{a}, \tag{47}$$

where

$$\begin{aligned}
\mathbf{z} &= [z(0), \cdots, z(N-1)]^T, \\
\mathbf{U} &= [\mathbf{u}_{10}, \cdots, \mathbf{u}_{K0}, \cdots, \mathbf{u}_{1Q}, \cdots, \mathbf{u}_{KQ}], \\
\mathbf{u}_{kq} &= [u_{kq}(0), \cdots, u_{kq}(N-1)]^T, \\
\mathbf{a} &= [a_{10}, \cdots, a_{K0}, \cdots, a_{1Q}, \cdots, a_{KQ}]^T.
\end{aligned}$$

The least-squares solution for (47) is

$$\hat{\mathbf{a}} \quad = \quad (\mathbf{U}^H\mathbf{U})^{-1}\mathbf{U}^H\mathbf{z}, \tag{48}$$

where $(\cdot)^H$ denotes complex conjugate transpose.

3.2.2 Memory Polynomial Predistorter Simulation

The performance of the memory polynomial predistorter constructed using the indirect learning architecture is demonstrated through the following examples. These examples

Petitioners Exhibit 1005
Page 40 of 117

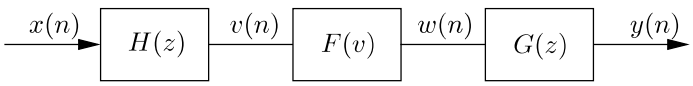


Figure 15: Wiener-Hammerstein model diagram.

show that the same memory polynomial structure can effectively linearize several different nonlinear models with memory, thereby demonstrating the robustness of memory polynomial predistortion.

Example 3.4 Here, the nonlinearity to be compensated for is assumed to obey a Wiener-Hammerstein (W-H) model (see Fig. 15); i.e., an LTI system followed by a memoryless nonlinearity, which in turn is followed by another LTI system. Such a configuration is commonly used for satellite communication channels where the power amplifier at the satellite transponder is driven near saturation to exploit the maximum power efficiency [2]. The LTI blocks before and after the memoryless nonlinearity, which are denoted by $H(z)$ and $G(z)$, respectively, are assumed to be

$$H(z) = \frac{1 + 0.5z^{-2}}{1 - 0.2z^{-1}}, \tag{49}$$

$$G(z) = \frac{1 - 0.1z^{-2}}{1 - 0.4z^{-1}}. \tag{50}$$

For the memoryless nonlinear portion of the W-H model,

$$w(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \; v(n)|v(n)|^{k-1}, \tag{51}$$

where $v(n)$ and $w(n)$ are, respectively, input and output of the memoryless nonlinear block. For the coefficients, we had

$$b_1 = 1.0108 + 0.0858j, \quad b_3 = 0.0879 - 0.1583j, \quad b_5 = -1.0992 - 0.8891j, \tag{52}$$

which were extracted from an actual Class AB power amplifier.

The baseband input was a 3-carrier WCDMA signal. Memory polynomial predistorter identification was carried out based on 8000 data samples. Next, the power spectral densities (PSDs) of the input and output signals were compared to evaluate the effectiveness of the predistorter in reducing spectral regrowth. Here, the predistorter (45) has two delay taps

Petitioners Exhibit 1005
Page 41 of 117

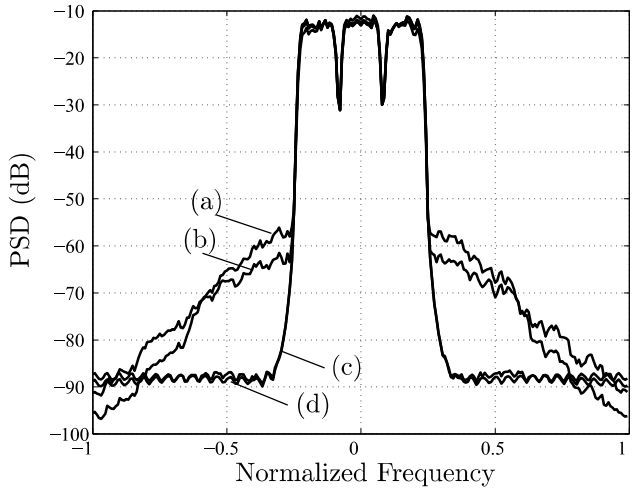


Figure 16: Effectiveness of predistortion in suppressing spectral regrowth when the power amplifier is modeled by a Wiener-Hammerstein system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$) (d) Original input. (c) and (d) almost coincide.

($Q = 2$) and 5th odd-order nonlinearity ($K = 5$). The performance of the predistorter is shown in Fig. 16. Spectral regrowth is almost fully suppressed with only two delay taps, even though the LTI portions of the W-H system have much longer impulse responses.

Example 3.5 Here, the power amplifier is also assumed to obey a memory polynomial model similar to (45); i.e.,

$$y(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} \sum_{q=0}^{Q} b_{kq}\ z(n-q)|z(n-q)|^{k-1} \tag{53}$$

The coefficients,

$$\begin{aligned} b_{10} &= 1.0513 + 0.0904j, \quad b_{30} = -0.0542 - 0.2900j, \quad b_{50} = -0.9657 - 0.7028j, \\ b_{11} &= -0.0680 - 0.0023j, \quad b_{31} = 0.2234 + 0.2317j, \quad b_{51} = -0.2451 - 0.3735j, \\ b_{12} &= 0.0289 - 0.0054j, \quad b_{32} = -0.0621 - 0.0932j, \quad b_{52} = 0.1229 + 0.1508j, \end{aligned} \tag{54}$$

were extracted from the same power amplifier as in Example 3.1. Fig. 17 shows the performance of various predistorters. The memory polynomial predistorter with $Q = 2$ and $K = 5$ is able to suppress most of the spectral regrowth. However, when both even- and odd-order nonlinearities are included in the predistorter, an additional 3-5 dB suppression

Petitioners Exhibit 1005
Page 42 of 117

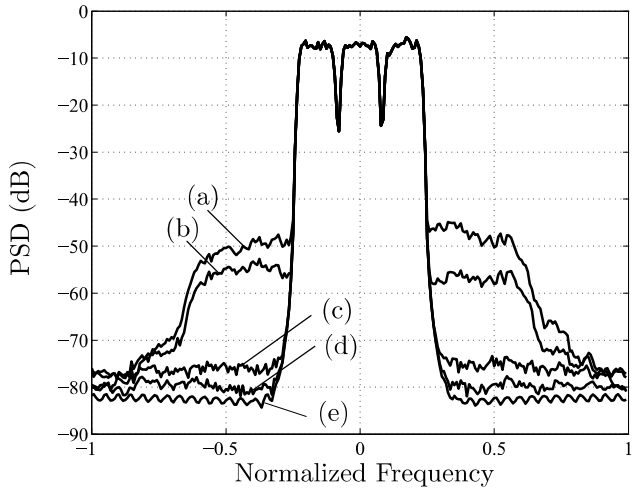


Figure 17: Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier itself is modeled by a memory polynomial. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$, odd order); (d) Output with memory polynomial predistortion ($Q = 2$, $K = 5$, even and odd orders); (e) Original input.

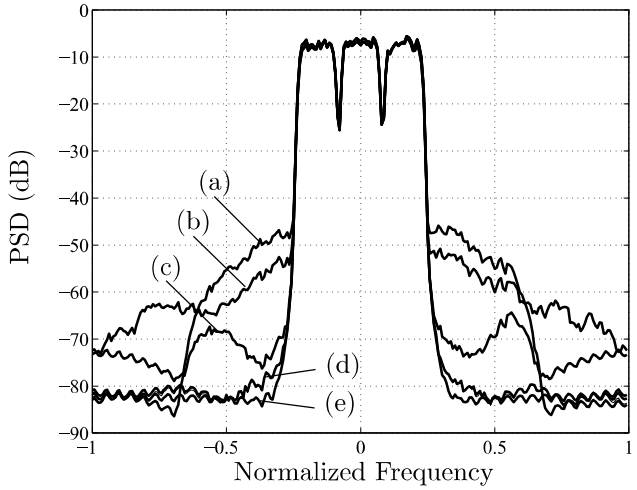


Figure 18: Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier is modeled by a perturbed Wiener (hence Volterra) system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$); (d) Output with memory polynomial predistortion ($Q = 10, K = 5$); (e) Original input.

Petitioners Exhibit 1005
Page 43 of 117

can be achieved. Spectral regrowth can be further suppressed by increasing the memory of the predistorter to $Q = 10$.

Example 3.6 In this example, a perturbed Wiener system was used as the power amplifier model. First, we constructed a Wiener model, whose

$$H(z) \quad = \quad 1 + 0.5z^{-2}, \tag{55}$$

$$b_1 \quad = \quad 1.0108 + 0.0858j, \quad b_3 \quad = \quad 0.0879 - 0.1583j. \tag{56}$$

Next, zero mean complex Gaussian noise with variance 2×10^{-4} was added to the Volterra kernels of the Wiener system, which turns the original Wiener system into a full Volterra system. The results are shown in Fig. 18. We still observe significant reduction in spectral regrowth with the memory polynomial predistorter ($Q = 2, K = 5$). With the maximum delay increased to $Q = 10$, the predistorter almost fully suppressed the spectral regrowth.

Example 3.7 The power amplifier here is assumed to follow a 3-branch parallel Wiener model (sum of Wiener sub-systems; see Fig. 19). The LTI blocks in the model are defined by

$$\begin{aligned} H_1(z) &= 1, \\ H_2(z) &= \frac{1 + 0.3z^{-1}}{1 - 0.1z^{-1}}, \\ H_3(z) &= \frac{1 - 0.2z^{-1}}{1 - 0.4z^{-1}}. \end{aligned} \tag{57}$$

The memoryless nonlinearity in the ith branch has input/output relationship

$$y_i(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} d_{ki} \; v_i(n)|v_i(n)|^{k-1}, \tag{58}$$

where $v_i(n)$ and $y_i(n)$ are the input and output of the nonlinearity $F_i(v)$, respectively. The

Petitioners Exhibit 1005
Page 44 of 117

d_{ki} coefficients used were

$$\begin{aligned} d_{11} &= 1.0108 + 0.0858j, \quad d_{31} = 0.0879 - 0.1583j, \\ d_{51} &= -1.0992 - 0.8891j, \quad d_{12} = 0.1179 + 0.0004j, \\ d_{32} &= -0.1818 + 0.0391j, \quad d_{52} = 0.1684 + 0.0034j, \\ d_{13} &= 0.0473 - 0.0058j, \quad d_{33} = 0.0395 + 0.0283j, \\ d_{53} &= -0.1015 - 0.0196j. \end{aligned} \tag{59}$$

Since $H_1(z)$=1, the first branch is actually a memoryless nonlinearity here. This reflects some belief that the dominating type of nonlinearity in a power amplifier is memoryless. The second and third branches both exhibit memory nonlinearity, with 10 dB and 13 dB less power than the first branch, respectively.

Fig. 20 shows the performance of our memory polynomial predistorter in linearizing such a power amplifier. With the memory polynomial predistorter ($Q = 2$, $K = 5$), there is a significant decrease in spectral regrowth, and the result is further improved when Q is increased to 5.

In all of the above cases, memoryless predistortion is not very effective in suppressing spectral regrowth, which underscores the notion that power amplifier memory effects must be taken into account when designing the predistorter.

The objective of power amplifier linearization is two-fold: suppression of spectral regrowth to reduce adjacent channel interference and minimization of in-band distortion to improve BER. Although only PSD plots are shown here, this does not mean that in-band distortion is left un-checked. Recall that in the indirect learning architecture, our convergence criterion requires the mean squared error between $y(n)$ and $Gx(n)$ to be minimized. Therefore, at convergence, the power amplifier is linearized, which automatically ensures the suppression of both in-band and out-of-band distortions. The PSD plots are shown for verification purposes. Because the PSD is phase blind, if one were to define a linearization criterion solely in terms of the PSD, the resulting predistorter may not be a true linearizer.

Petitioners Exhibit 1005
Page 45 of 117

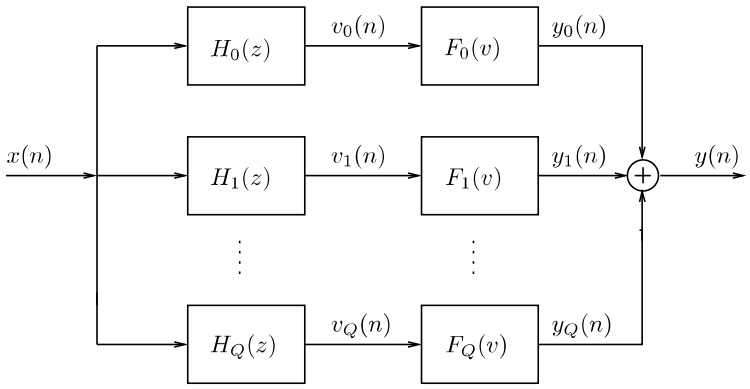


Figure 19: Parallel Wiener model diagram. $H_i(\cdot)$ is an LTI block, and $F_i(\cdot)$ is a memoryless nonlinear block.

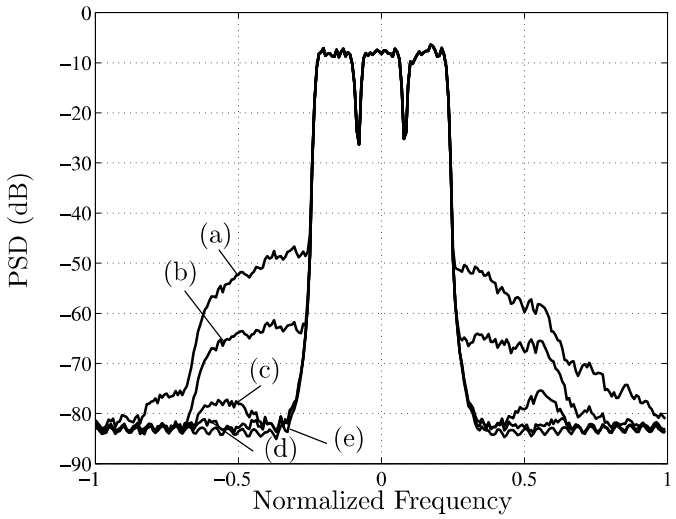


Figure 20: Effectiveness of predistortion in suppressing spectral regrowth, when the power amplifier is modeled by a parallel Wiener system. (a) Output without predistortion; (b) Output with memoryless predistortion; (c) Output with memory polynomial predistortion ($Q = 2, K = 5$); (d) Output with memory polynomial predistortion ($Q = 5, K = 5$); (e) Original input.

Petitioners Exhibit 1005
Page 46 of 117

3.2.3 Memory Polynomial Predistorter Discussion

The Volterra series is the most general polynomial type of nonlinearity with memory. In this section, we have encountered the memory polynomial, the Wiener, the Hammerstein, and the parallel Wiener models as special cases of the Volterra model. Next, we would like to point out some interesting links among these models.

For the memory polynomial model (45), let us collect the coefficients in a matrix

$$\mathbf{A} = \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1Q} \\ a_{30} & a_{31} & \cdots & a_{3Q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K0} & a_{K1} & \cdots & a_{KQ} \end{bmatrix}. \tag{60}$$

A Hammerstein model on the other hand, can be described by

$$v(n) = \sum_{k=1}^{K} c_k \; x(n)|x(n)|^{k-1} \tag{61}$$

$$y(n) = \sum_{q=0}^{Q} h(q) \; v(n-q). \tag{62}$$

Let us collect the coefficients in (61) and (62) in vectors

$$\mathbf{c} = [c_1, \ldots, c_K]^T,$$

$$\mathbf{h} = [h(0), h(1), \ldots, h(Q)]^T.$$

Substitution of (61) into (62) yields

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} c_k \; h(q) \; x(n-q)|x(n-q)|^{k-1}. \tag{63}$$

Comparing (63) with (45), we can see that the Hammerstein system (61)-(62) is a special case of the memory polynomial model (45) with

$$a_{kq} = c_k \, h(q). \tag{64}$$

In other words,

$$\mathbf{A} = \mathbf{c} \, \mathbf{h}^T \tag{65}$$

Petitioners Exhibit 1005
Page 47 of 117

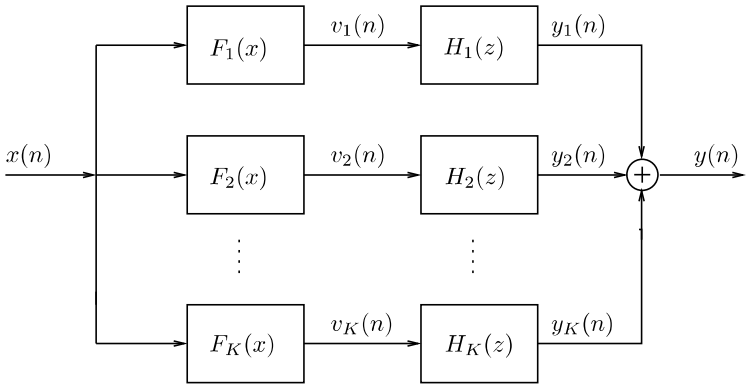


Figure 21: Parallel Hammerstein model diagram. $F_i(\cdot)$ is a memoryless nonlinear block, and $H_i(\cdot)$ is an LTI block.

and hence r$ank(\mathbf{A}) = 1$. This implies that as a predistorter, the memory polynomial is expected to work well with a Wiener power amplifier. On the other hand, the memory polynomial predistorter is expected to be more robust than the Hammerstein predistorter. Interestingly, although the memory polynomial model is more general than the Hammerstein model, its parameter estimation is actually more straightforward (c.f., via the least-squares solution (48)).

Let

$$v_k(n) \;=\; F_k(x(n)) \;=\; x(n)|x(n)|^{k-1}, \tag{66}$$

$$y_k(n) \;=\; v_k(n) * h_k(n), \tag{67}$$

where $h_k(n) = a_{kn}$, and $*$ denotes convolution. We can rewrite the memory polynomial model as

$$y(n) = \sum_{k=1}^{K} y_k(n) = \sum_{k=1}^{K} v_k(n) * a_{kn} \tag{68}$$

$$= \sum_{k=1}^{K} \sum_{q=0}^{Q} a_{kq}\; x(n-q)|x(n-q)|^{k-1}. \tag{69}$$

Therefore, a memory polynomial model is also a parallel Hammerstein model (see Fig. 21) where the memoryless nonlinear block is a polynomial.

Petitioners Exhibit 1005
Page 48 of 117

In [10], the authors tried to linearize a Wiener system with a Hammerstein predistorter using the indirect learning architecture. They adopted the least squares approach[1] to solve for the predistorter coefficients, although the parameters of the memoryless nonlinear and the LTI blocks of the Hammerstein model are not explicitly recovered.

Alternatively, we can also rewrite the memory polynomial model as

$$v_q(n) = x(n) * h_q(n), \quad h_q(n) = \delta(n-q), \tag{70}$$

$$F_q(v) = \sum_{k=1}^{K} a_{kq}\, v|v|^{k-1}, \tag{71}$$

$$y(n) = \sum_{q=0}^{Q} F_q[v_q(n)], \tag{72}$$

where $*$ in (70) denotes convolution. Comparing with the parallel Wiener model (see Fig. 19), we observe that a memory polynomial is also a special parallel Wiener model with

$$H_q(z) = z^{-q}, \quad \text{or} \quad h_q(n) = \delta(n-q). \tag{73}$$

In summary, when considering polynomial type of nonlinearities, both the parallel Wiener and parallel Hammerstein models are special cases of the Volterra series. In fact, it can be shown that the memory polynomial model is equivalent to the parallel Hammerstein model. We have also shown that a memory polynomial model is a special case of the parallel Wiener model. Obviously, the parallel Hammerstein model includes the Hammerstein model as a special case, and the parallel Wiener model includes the Wiener model as a special case. Hammerstein and Wiener models are the most "specialized" with the least number of coefficients, but are by no means the easiest to identify. The memory polynomial model, however, offers a good compromise between generality and ease of parameter estimation and implementation.

3.3 A New Combined Predistorter Design

Although the memory polynomial model has been shown to be robust for predistorting several types of nonlinear models with memory [17], the Volterra kernel support in the model

[1]There is a typo in Equation (13) of [10]: $|x[n]|^2x^*[n-1]$ should be $|x[n-1]|^2x^*[n-1]$; $|x[n]|^2x^*[n-2]$ should be $|x[n-2]|^2x^*[n-2]$. Moreover, we believe that the baseband expression (9) should be in terms of $x[n-i]|x[n-i]|^{j-1}$ instead of $x^j[n-i]$.

Petitioners Exhibit 1005
Page 49 of 117

is very limited. Therefore, when predistorting a nonlinear system with complex memory structures, the model may not be adequate. The Murray Hill model suggested by [32] offers a memory structure that is not present in the memory polynomial model. This model introduces a nonlinearity that depends on a linear combination of past input amplitudes and adds it to the conventional memoryless polynomial model. Good performance was achieved by using the model to predistort a Class AB power amplifier. However, unlike the memory polynomial model, this model does not contain terms of an LTI system. Therefore, it does not have the intrinsic capability of compensating for a linear frequency response that may exist in the RF upconverter or the power amplifier. Moreover, the least-squares/simplex approach used in [32] for estimating the model coefficients converges very slowly when the number of memory taps becomes relatively large.

Here, we combine the memory polynomial model [30] and the model of [32] to obtain a more robust new model. We also develop a fast-converging least-squares/Newton algorithm for estimating the coefficients of the new model.

3.3.1 Combined Predistorter Model

Given an input $x(n)$, the output of the model of [32], $z(n)$, is defined as

$$z(n) = \sum_{p=1}^{P} a_p\, x(n)|x(n)|^{p-1} + \sum_{q=2}^{Q} b_q\, x(n) \left[\sum_{l=0}^{L-1} c_l\, |x(n-l)|\right]^{q-1}, \tag{74}$$

where a_p and b_q are complex polynomial coefficients and c_l are real envelope filter coefficients. Note that the b_q coefficients start with b_2 instead of b_1 to avoid redundancy with a_1. The first term in the model is the conventional memoryless polynomial, where the nonlinearity depends on the current input amplitude $|z(n)|$. The nonlinearity generated by the second term, however, depends on combinations of the current and past input signal amplitudes. Moreover, by restricting c_l to be real, this nonlinearity can be represented by a one-dimensional look-up table (LUT) indexed by $\left[\sum_{l=0}^{L-1} c_l\, |x(n-l)|\right]$, which makes it very easy to implement the model in hardware.

Petitioners Exhibit 1005
Page 50 of 117

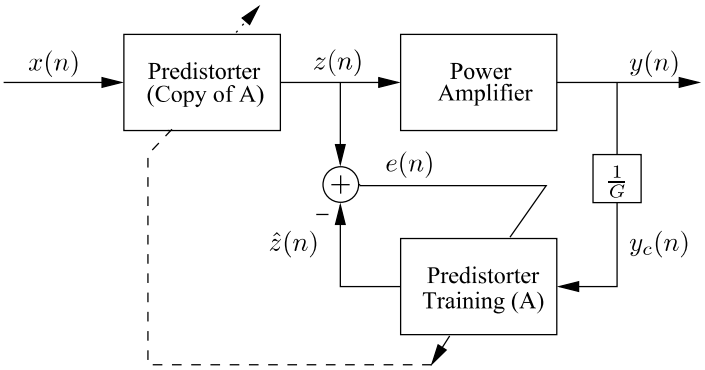


Figure 22: The indirect learning architecture of the new combined predistorter model.

If we replace the conventional memoryless polynomial in (74) with the memory polynomial proposed in [30], we arrive at the new model

$$z(n) \;=\; \sum_{k=0}^{K-1}\sum_{p=1}^{P} a_{kp}\, x(n-k)|x(n-k)|^{p-1} + \sum_{q=2}^{Q} b_q\, x(n)\left[\sum_{l=0}^{L-1} c_l\,|x(n-l)|\right]^{q-1}, \quad (75)$$

where a_{kp} are the coefficients of the memory polynomial. The new model combines the terms in both the memory polynomial model and the model of [32], which captures a wider class of nonlinear behavior of a wideband predistorter and can be implemented easily in hardware.

3.3.2 Combined Predistorter Training

In the training branch of Fig. 22, we have

$$\hat{z}(n) \;=\; \sum_{k=0}^{K-1}\sum_{p=1}^{P} a_{kp}\, y_c(n-k)|y_c(n-k)|^{p-1} + \sum_{q=2}^{Q} b_q\, y_c(n)\left[\sum_{l=0}^{L-1} c_l\,|y_c(n-l)|\right]^{q-1}. \quad (76)$$

To estimate the model parameters, we first define the instantaneous error as

$$e(n) \;=\; z(n) - \hat{z}(n). \quad (77)$$

For a block of N data samples, the least-squares cost function is then given by

$$J \;=\; \sum_{n=1}^{N} |e(n)|^2 = \sum_{n=1}^{N} e(n)\, e^*(n) \quad (78)$$

where $(\cdot)^*$ denotes complex conjugate.

Petitioners Exhibit 1005
Page 51 of 117

The optimum a_{kp}, b_q, and c_l that minimize J can be found by setting the partial derivatives of J with respect to a_{kp}^*, b_q^* (pretending that a_{pk} and b_q are constants [5]), and c_l equal to zero, i.e.,

$$\frac{\partial J}{\partial a_{kp}^*} = -\sum_{n=1}^{N} e(n)\, u_{kp}^*(n) = 0 \tag{79}$$

$$\frac{\partial J}{\partial b_q^*} = -\sum_{n=1}^{N} e(n)\, v_q^*(n) = 0 \tag{80}$$

$$\frac{\partial J}{\partial c_l} = -\sum_{n=1}^{N} 2\ \mathrm{Re}\left[e(n) s_l^*(n)\right] = 0. \tag{81}$$

where

$$u_{kp}(n) = y_c(n-k)\,|y_c(n-k)|^{p-1} \tag{82}$$

$$v_q(n) = y_c(n)\left(\sum_{l=0}^{L-1} c_l\,|y_c(n-l)|\right)^{q-1} \tag{83}$$

$$s_l(n) = \sum_{q=2}^{Q} b_q\, y_c(n)\,(q-1) \times \left(\sum_{l_1=0}^{L-1} c_{l_1}\,|y_c(n-l_1)|\right)^{q-2} |y_c(n-l)|. \tag{84}$$

However, (79), (80), and (81) can not be solved simultaneously. The coefficients a_{kp} and b_q are coupled with c_l. We propose an iterative method to solve this problem. First, from a set of initial values of c_l, a_{kp} and b_q are found by the least-squares approach. After the a_{kp} and b_q are obtained, the c_l are updated by Newton's method. The adaptation continues until all coefficients converge.

Least-Squares Method. Assuming the c_l coefficients in the previous iteration are known, denoted by $c_l^{(i)}$, we can derive the least-square solutions of $a_{kp}^{(i+1)}$ and $b_q^{(i+1)}$ by solving (79) and (80) together. Rearranging (79) and (80) in matrix form, we have

$$\left([\mathbf{U}\ \mathbf{V}^{(i)}]^H[\mathbf{U}\ \mathbf{V}^{(i)}]\right)\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix} = \left[\mathbf{U}\ \mathbf{V}^{(i)}\right]^H \mathbf{z}, \tag{85}$$

Petitioners Exhibit 1005
Page 52 of 117

where $(\cdot)^H$ denotes complex conjugate transpose,

$$\begin{aligned}
\mathbf{U} &= [\mathbf{u}_{10}, \cdots, \mathbf{u}_{K0}, \cdots, \mathbf{u}_{1P}, \cdots, \mathbf{u}_{KP}], \\
\mathbf{V}^{(i)} &= [\mathbf{v}_2^{(i)}, \cdots, \mathbf{v}_Q^{(i)}] \\
\mathbf{a} &= [a_{10}, \cdots, a_{K0}, \cdots, a_{1Q}, \cdots, a_{KQ}]^T \\
\mathbf{b} &= [b_2, \cdots, b_Q]^T \\
\mathbf{z} &= [z(0), \cdots, z(N-1)]^T
\end{aligned}$$

with the vectors $\mathbf{u}_{kp}$ and $\mathbf{v}_q^{(i)}$ defined as

$$\begin{aligned}
\mathbf{u}_{kp} &= [u_{kp}(0), \cdots, u_{kp}(N-1)]^T \\
\mathbf{v}_q^{(i)} &= [v_q^{(i)}(0), \cdots, v_q^{(i)}(N-1)]^T.
\end{aligned}$$

Note that $v_q^{(i)}(n)$ in the i-th iteration is generated by plugging $c_l^{(i)}$ into (83).

It follows from (85) that the least-squares estimates of $\mathbf{a}$ and $\mathbf{b}$ at the $(i+1)$-th iteration are

$$\begin{bmatrix} \hat{\mathbf{a}}^{(i+1)} \\ \hat{\mathbf{b}}^{(i+1)} \end{bmatrix} = \left([\mathbf{U}\ \mathbf{V}^{(i)}]^H [\mathbf{U}\ \mathbf{V}^{(i)}] \right)^{-1} [\mathbf{U}\ \mathbf{V}^{(i)}]^H \mathbf{z}. \tag{86}$$

Newton's Method. Once $\mathbf{a}^{(i+1)}$ and $\mathbf{b}^{(i+1)}$ are obtained, Newton's method can be used to update the c_l coefficients. Rewriting $c_l^{(i)}$ in vector form as $\mathbf{c}^{(i)} = [c_0^{(i)}, c_1^{(i)}, \cdots, c_{L-1}^{(i)}]^T$, the new $\mathbf{c}^{(i+1)}$ is found by Newton's method [34, pp. 632-637] as

$$\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} - \left[\nabla_{\mathbf{c}}^2 J(\mathbf{c}^{(i)}) \right]^{-1} \nabla_{\mathbf{c}} J(\mathbf{c}^{(i)}) \tag{87}$$

where $\nabla_{\mathbf{c}} J(\mathbf{c}^{(i)})$ is the column gradient vector of J with the l-th element

$$\left[\nabla_{\mathbf{c}} J(\mathbf{c}^{(i)}) \right]_l = \left. \frac{\partial J}{\partial c_l} \right|_{\mathbf{c}^{(i)}} \tag{88}$$

and $\nabla_{\mathbf{c}}^2 J(\mathbf{c}^{(i)})$ is the Hessian of J with the (m, l) entry

$$\left[\nabla_{\mathbf{c}}^2 J(\mathbf{c}^{(i)}) \right]_{lm} = \left. \frac{\partial^2 J}{\partial c_l \partial c_m} \right|_{\mathbf{c}^{(i)}}. \tag{89}$$

We have already derived $\dfrac{\partial J}{\partial c_l}$ in (81). By taking the derivative of both sides of (81) with respect to c_m, we obtain

$$\frac{\partial^2 J}{\partial c_l \partial c_m} = \sum_{n=1}^{N} 2\ \mathrm{Re}\left[-e(n) z_{lm}^*(n) + s_m(n) s_l^*(n) \right], \tag{90}$$

Petitioners Exhibit 1005
Page 53 of 117

where

$$\begin{aligned} z_{lm}(n) &= \frac{\partial s_l(n)}{\partial c_m} \\ &= \sum_{q=2}^{Q} b_q\, y_c(n)\,(q-1)\,(q-2)\left(\sum_{l_1=0}^{L-1} c_{l_1}\,|y_c(n-l_1)|\right)^{q-3} \\ &\quad \times |x(n-l)|\,|x(n-m)|. \end{aligned} \tag{91}$$

Similar to the previous section, we can form column vectors $\mathbf{s}_l^{(i)}$, $\mathbf{z}_{lm}^{(i)}$, and $\mathbf{e}^{(i)}$ from the sequences $s_l(n)$, $z_{lm}(n)$, and $e(n)$ after plugging in $\mathbf{a}^{(i+1)}$, $\mathbf{b}^{(i+1)}$, and $\mathbf{c}^{(i)}$. Then the gradient and Hessian vectors can be re-written in matrix form as follows,

$$\nabla_{\mathbf{c}} J(\mathbf{c}^{(i)}) = -2\ \mathrm{Re}\left\{[\mathbf{S}^{(i)}]^H \mathbf{e}^{(i)}\right\}, \tag{92}$$

$$\nabla^2_{\mathbf{c}} J(\mathbf{c}^{(i)}) = 2\ \mathrm{Re}\left\{[\mathbf{S}^{(i)}]^H \mathbf{S}^{(i)} - [\mathbf{Z}^{(i)}]^H \mathbf{E}^{(i)}\right\}, \tag{93}$$

where

$$\begin{aligned} \mathbf{S}^{(i)} &= [\mathbf{s}_0^{(i)}, \cdots, \mathbf{s}_{L-1}^{(i)}], \\ \mathbf{Z}^{(i)} &= \begin{bmatrix} \mathbf{z}_{00}^{(i)} & \cdots & \mathbf{z}_{0,L-1}^{(i)} \\ \mathbf{z}_{10}^{(i)} & \cdots & \mathbf{z}_{1,L-1}^{(i)} \\ \vdots & & \vdots \\ \mathbf{z}_{L-1,0}^{(i)} & \cdots & \mathbf{z}_{L-1,L-1}^{(i)} \end{bmatrix}, \\ \mathbf{E}^{(i)} &= \begin{bmatrix} \mathbf{e}^{(i)} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{e}^{(i)} \end{bmatrix}. \end{aligned} \tag{94}$$

As a result, the final update equation for $\mathbf{c}$ is

$$\hat{\mathbf{c}}^{(i+1)} = \hat{\mathbf{c}}^{(i)} + \left[\mathrm{Re}\left\{[\mathbf{S}^{(i)}]^H \mathbf{S}^{(i)} - [\mathbf{Z}^{(i)}]^H \mathbf{E}^{(i)}\right\}\right]^{-1} \mathrm{Re}\left\{[\mathbf{S}^{(i)}]^H \mathbf{e}^{(i)}\right\}. \tag{95}$$

3.3.3 Effects of Noise and Initial Estimates

In this section, we evaluate the effects of noise and initial estimate on the performance of the least-squares/Newton algorithm through computer simulations.

Petitioners Exhibit 1005
Page 54 of 117

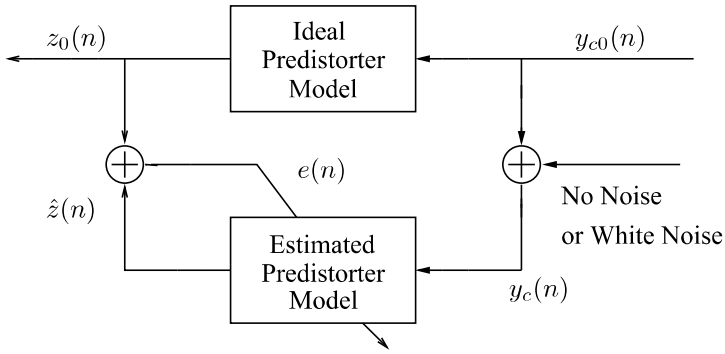


Figure 23: Block diagram of the predistorter model simulation.

The simulation setup is shown in Fig. 23. To obtain an ideal predistorter model, we first acquired the output data (sampled at 153.6 MHz IF, complex demodulated, and downsampled to 76.8 MHz) of an actual power amplifier from an experimental wideband predistortion testbed for a 15 MHz 3-carrier WCDMA input signal. The power amplifier was a LDMOS Class AB power amplifier from Xemod. We then obtained the parameters of the ideal predistorter model using the least-squares/Newton algorithm by treating the power amplifier input as the desired output of the algorithm and the power amplifier output as the input of the algorithm. Therefore, the ideal model captures the essential behavior of a predistorter, such as expanding nonlinearities. The order of the ideal model is represented by a vector $\mathbf{m} = [7, 4, 7, 6]$, where $\mathbf{m}$ is defined as

$$\mathbf{m} \quad = \quad \begin{bmatrix} P & K & Q & L \end{bmatrix} , \tag{96}$$

with P, K, Q, and L given in (76).

In the following simulations, the input to the ideal model, $y_{c0}(n)$, is a noiseless WCDMA 3-carrier baseband signal with 15 MHz bandwidth. The output of the ideal model with the noiseless input is denoted by $z_0(n)$. In the noiseless setting, $y_{c0}(n)$ and $z_0(n)$ were used, respectively, as the input and desired output of the algorithm. In the noisy setting, noise was added to $y_{c0}(n)$, which generated $y_c(n)$ with a signal-to-noise ratio of 45 dB. Then, $y_c(n)$ and $z_0(n)$ were used, respectively, as the input and desired output of the algorithm. This noisy setting agrees with the indirect learning architecture shown in Fig. 22, where the noise of the transmitter and the feedback loop is actually contained in $y_c(n)$ (the input

Petitioners Exhibit 1005
Page 55 of 117

to the predistorter training block).

To evaluate the effectiveness of the least-squares/Newton algorithm, we compare the difference between the outputs of the ideal model and the estimated model when the same noiseless input $y_{c0}(n)$ is used for both, and denote this by $e_0(n)$. Note that $e_0(n)$ is not the error $e(n)$ that the algorithm minimizes to determine the estimated model. When noise is present in $y_c(n)$, $e(n)$ is dominated by the noise and cannot show the real difference between the estimated model and the ideal model.

In the simulations, the following default options were used unless mentioned specifically.

- The order of the estimated model is [7 4 7 6], which is the same as the ideal model.
- The input and output data comprise 16k samples.
- $\mathbf{c}^{(0)} = [\,1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1\,]^T/\sqrt{6}$.

Simulation 3.1. We study the effects of noise on the performance of the algorithm. The convergency behavior of the algorithm for the noiseless setting and the noisy setting are shown in Fig. 24 and Fig. 25, respectively. The mean squared error (MSE) in both of the figures is defined as

$$\text{MSE(dB)} = 10\,\log_{10}\left(\frac{\sum_{n=1}^{N}|z_0(n)-\hat{z}(n)|^2}{\sum_{n=1}^{N}|z_0(n)|^2}\right). \tag{97}$$

In these figures, we first observe that the MSE decreases monotonically with the number of iterations. The large drops of the MSE are within the first several steps. Secondly, the estimated model coefficients show smooth changes from one step to another. Moreover, these coefficients converge very fast, usually within 10 iterations.

The power spectra of $e_0(n)$ for both cases are shown in Fig. 26. For comparison, the power spectrum of $z_0(n)$ is also shown in the figure. From the figure, we can see that the algorithm correctly (in terms of reducing $e_0(n)$) estimated the model in the noiseless case and constructed a fairly accurate model in the noisy case. The relative large in-band error may be due to the fact that the noise was added in the input of the algorithm. More in-depth analysis of this noise problem and several remedial techniques can be found in [35].

Petitioners Exhibit 1005
Page 56 of 117

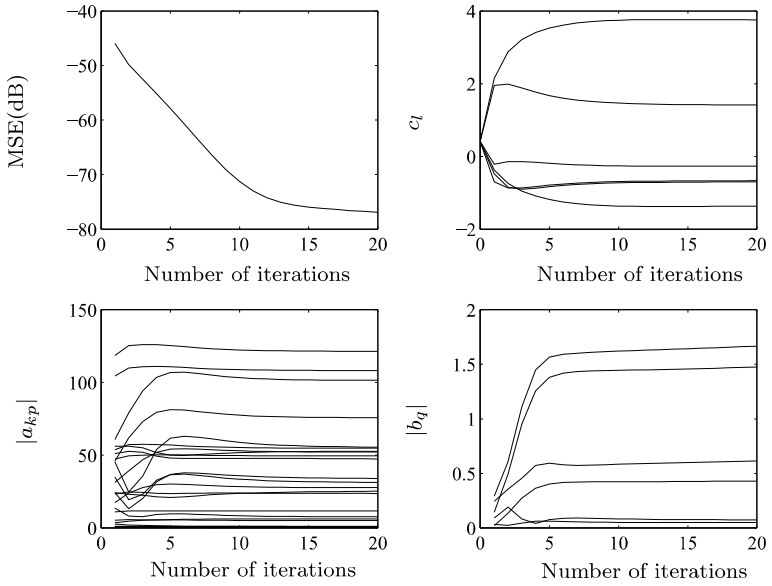


Figure 24: Trajectories of (a) mean squared error, (b) c_l coefficients, (c) amplitudes of a_{kp} coefficients, and (d) amplitudes of b_q coefficients vs. number of iterations in the noiseless setting.

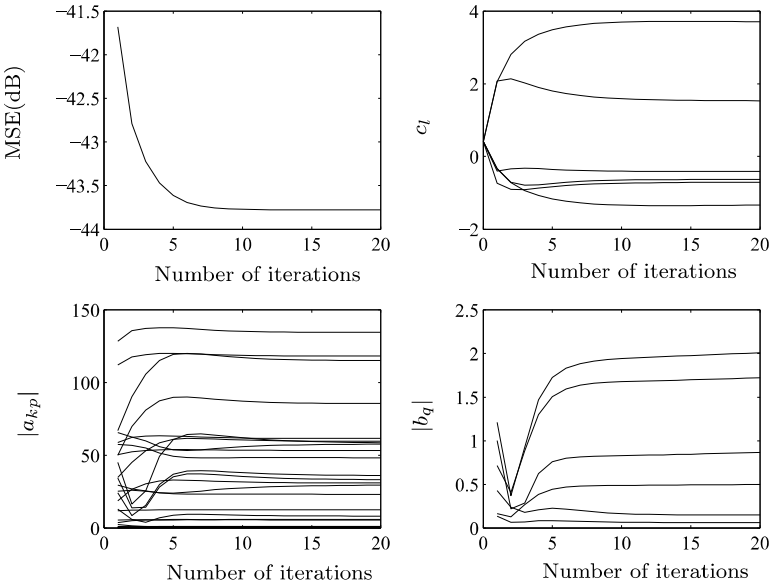


Figure 25: Trajectories of (a) mean squared error, (b) c_l coefficients, (c) amplitudes of a_{kp} coefficients, and (d) amplitudes of b_q coefficients vs. number of iterations in the noisy setting.

Petitioners Exhibit 1005
Page 57 of 117

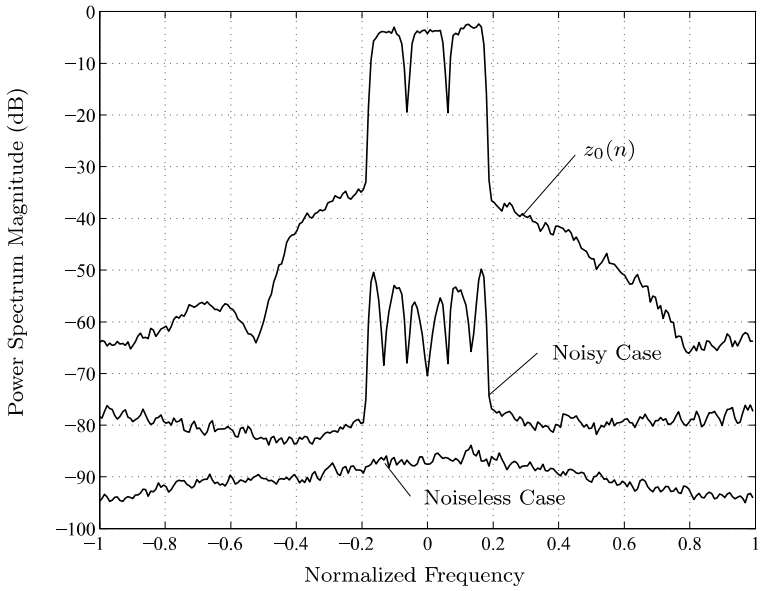


Figure 26: Comparison of the power spectrum of the noiseless predistorter model output $z_0(n)$ with the power spectra of noiseless residue $e_0(n)$ in noiseless and noisy settings.

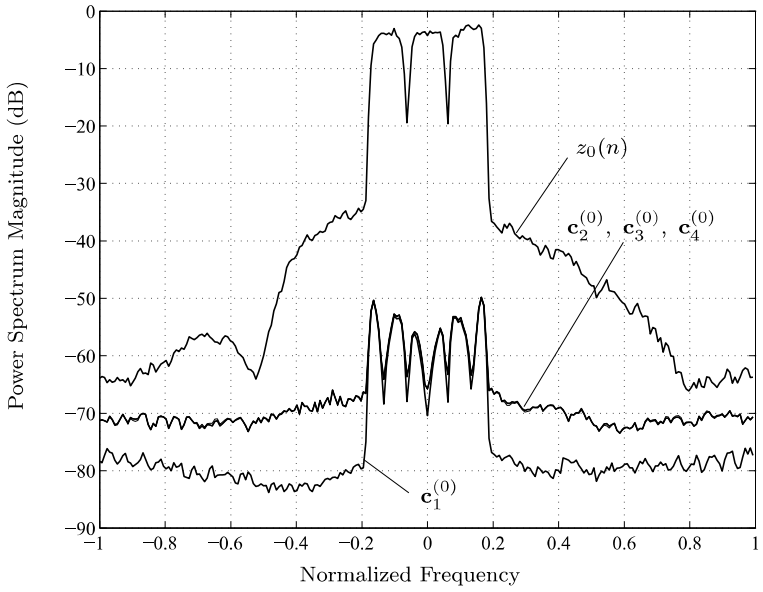


Figure 27: Comparison of the power spectrum of the noiseless predistorter model output $z_0(n)$ with the power spectra of noiseless residue $e_0(n)$ for different initialization $\mathbf{c}$'s in the noisy setting.

Petitioners Exhibit 1005
Page 58 of 117

Simulation 3.2. Since the least-squares/Newton algorithm proposed in the previous section is iterative, a different initialization may lead to a different solution. In this simulation, we started the algorithm with different initializations of the $\mathbf{c}$ coefficients:

$$\begin{aligned}
\mathbf{c}_1^{(0)} &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T / \sqrt{6} \ , \\
\mathbf{c}_2^{(0)} &= \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}^T / \sqrt{2} \ , \\
\mathbf{c}_3^{(0)} &= \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T / \sqrt{2} \ , \\
\mathbf{c}_4^{(0)} &= \begin{bmatrix} -1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix}^T / \sqrt{6} \ .
\end{aligned}$$

Here, $\mathbf{c}_1^{(0)}$ is a simple lowpass filter, $\mathbf{c}_2^{(0)}$ and $\mathbf{c}_3^{(0)}$ are highpass filters, and $\mathbf{c}_4^{(0)}$ is a random vector made up of 1's and –1's. We ran the algorithm in the noisy setting and compared the accuracy of the estimated model for each initialization. The power spectra of $e_0(n)$ for different initializations are shown in Fig. 27. The power spectrum of $z_0(n)$ is also displayed for comparison. We can see that initialization with $\mathbf{c}_1^{(0)}$ gives better performance while initialization with the others are almost the same. This simulation shows that the initialization of the algorithm does affect the ability of the algorithm to estimate the model parameters. However, initialization with $\mathbf{c}_1^{(0)}$ gives a relatively good starting point.

3.3.4 Combined Predistorter Performance

In this section, we show overall system test results using the least-squares/Newton algorithm on the wideband predistortion testbed mentioned in the previous section. The baseband input data was again a 15 MHz 3-carrier WCDMA signal. The power spectra of the output signals from the power amplifier are shown in Fig. 28. The results show that the new predistorter can effectively linearize the power amplifier. Moreover, as shown in this figure, the predistorter needs to have a 35-tap envelope filter in order to fully suppress the spectral regrowth of the power amplifier, which indicates that the power amplifier has a relatively long memory effect.

Petitioners Exhibit 1005
Page 59 of 117

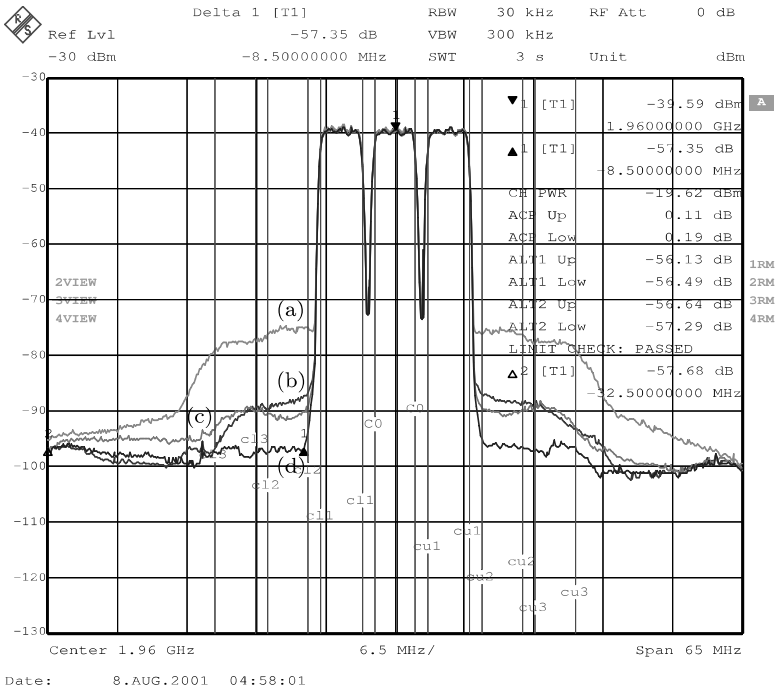


Figure 28: Overall system performance of the least-squares/Newton algorithm on the predistortion testbed, showing the power spectra of the output signal (a) without predistortion, (b) with predistorter ($\mathbf{m} = [7\ 4\ 0\ 0]$), (c) with predistorter ($\mathbf{m} = [7\ 4\ 7\ 3]$), and (d) with predistorter ($\mathbf{m} = [7\ 4\ 7\ 35]$).

Petitioners Exhibit 1005
Page 60 of 117

CHAPTER 4

EFFECTS OF EVEN-ORDER NONLINEAR TERMS

In Chapter 3, the predistorter models include both even- and odd-order nonlinear terms. However, in the literature, most of the power amplifier and predistorter models only considers the odd-order terms. In this chapter, we show that it is beneficial to include even-order nonlinear terms in both the baseband power amplifier and predistorter models [15, 16]. By including these even-order nonlinear terms, we have a richer basis set, which offers appreciable improvement.

4.1 Passband and Baseband Nonlinearities

4.1.1 Memoryless Case

Choice of the baseband predistorter is often dictated by the characteristics of the power amplifier. For low power amplifiers and/or narrowband input, the power amplifier can be regarded as memoryless or quasi-memoryless [39]. Polynomial models have been used extensively for memoryless nonlinear power amplifiers ([3, p. 69]), only odd-order polynomial terms in the power amplifier nonlinearity impact a bandpass communication signal. Suppose that a physical power amplifier obeys the following input/output relationship in the passband:

$$\tilde{y}(t) = \sum_{k=1}^{K} \tilde{b}_k \; \tilde{z}^k(t), \tag{98}$$

where $\tilde{z}(t)$ is the passband power amplifier input and $\tilde{y}(t)$ is the passband power amplifier output. The baseband counter-part of (98) is [39], [3, p. 69]

$$y(t) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{K} b_k \; z(t)|z(t)|^{k-1}, \tag{99}$$

where $b_k = 2^{1-k}\binom{k}{\frac{k-1}{2}}\tilde{b}_k$.

Although the above arguments for odd-order terms pertain to the power amplifier, when

Petitioners Exhibit 1005
Page 61 of 117

constructing a polynomial-based *predistorter*, many published papers consider only odd-order nonlinear terms as well; see e.g., [42, 43, 49]. This means that the output of the predistorter (also input to the power amplifier), $z(t)$, is expressed in terms of the input signal $x(t)$ as

$$z(t) = \sum_{\substack{l=1 \\ l \text{ odd}}}^{L} a_l \; x(t) \; |x(t)|^{l-1}. \tag{100}$$

In [49], the rationale for including only odd-order terms in the predistorter is stated: "Since even-order powers in the power series representing the power amplifier does not reflect into the first harmonic zone, concern only has to be taken to the odd powers when implementing the predistortion polynomial." In [42], it is said "We can model the predistorter and power amplifier as two truncated complex power series ... We restrict the power series to odd order terms, since even order components have no power spill over in the adjacent channel."

Here, we will show that it is beneficial to include even-order nonlinear terms in both the baseband power amplifier and predistorter models; i.e., we allow even k values in (99) and even l values in (100). By including these even-order nonlinear terms, we have a richer basis set, which offers appreciable improvement.

4.1.2 Quasi-memoryless Case

In Section 4.1.1, we observed that if the power amplifier is strictly memoryless and obeys the polynomial model (98) in the passband, then the baseband power amplifier input/output relationship is described by an odd-order polynomial (99) with real-valued coefficients. In this Section, we will show that if the power amplifier obeys a Volterra model in the passband and the input signal is narrowband, then the baseband power amplifier input/output relationship is also described by (99) but with complex-valued coefficients; such a power amplifier is said to be quasi-memoryless.

For weakly nonlinear power amplifiers, for example, the Class AB power amplifiers commonly employed in wireless handsets and base stations, the Volterra series model can be used; see e.g., [33]. In the passband, the Volterra series can be expressed as:

$$\tilde{y}(t) = \sum_{k} \int \cdots \int \tilde{h}_k(\boldsymbol{\tau}_k) \prod_{i=1}^{k} \tilde{z}(t - \tau_i) \; d\boldsymbol{\tau}_k, \tag{101}$$

Petitioners Exhibit 1005
Page 62 of 117

where $\tilde{z}(t)$ is the passband power amplifier input, $\tilde{y}(t)$ is the passband power amplifier output, $\boldsymbol{\tau}_k = [\tau_1, \ldots, \tau_k]^T$, $\tilde{h}_k(\cdot)$ is the kth-order Volterra kernel, and $d\boldsymbol{\tau}_k = d\tau_1 d\tau_2 \cdots d\tau_k$. We note the following special cases.

Special case #1: When the power amplifier is strictly memoryless; i.e., $\tilde{h}_k(\boldsymbol{\tau}_k) = \tilde{b}_k\ \delta(\boldsymbol{\tau}_k)$, we have the polynomial model (98).

Special case #2: When $\tilde{h}_k(\boldsymbol{\tau}_k)$ is non-zero only along the diagonal slice; i.e., $\tilde{h}_k(\boldsymbol{\tau}_k)$ $= \bar{h}_k(\tau)$ if $\tau_1 = \ldots = \tau_k = \tau$, and $\tilde{h}_k(\boldsymbol{\tau}_k) = 0$ otherwise, we have

$$\tilde{y}(t) = \sum_k \int \bar{h}_k(\tau) \tilde{z}^k(t-\tau)\ d\tau, \tag{102}$$

which we refer to as the memory polynomial model [17].

Assuming that $\tilde{z}(t)$ is band-limited with bandwidth much less than the carrier frequency f_o, it can be shown that the corresponding baseband relationship is [39], [3]:

$$y(t) = \sum_k \int \cdots \int h_{2k+1}(\boldsymbol{\tau}_{2k+1}) \prod_{i=1}^{k+1} z(t-\tau_i) \prod_{i=k+2}^{2k+1} z^*(t-\tau_i)\ d\boldsymbol{\tau}_{2k+1}, \tag{103}$$

where

$$h_{2k+1}(\boldsymbol{\tau}_{2k+1}) = \frac{1}{2^{2k}} \binom{2k+1}{k} \times\ \tilde{h}_{2k+1}(\boldsymbol{\tau}_{2k+1}) \times e^{-j2\pi f_o \left(\sum_{i=1}^{k+1} \tau_i - \sum_{i=k+2}^{2k+1} \tau_i\right)}, \tag{104}$$

$j = \sqrt{-1}$ and $(\cdot)^*$ denotes complex conjugation. Note from (104) that the baseband Volterra kernel $h(\cdot)$ is generally complex valued even though the passband Volterra kernel $\tilde{h}(\cdot)$ is real-valued.

If the input is narrowband such that $z(t-\tau_i) \approx z(t)$ over the support of each kernel, we can replace $z(t-\tau_i)$ by $z(t)$ in (103) to obtain [39]

$$\begin{aligned} y(t) &= \sum_k \int \cdots \int h_{2k+1}(\boldsymbol{\tau}_{2k+1})\ d\boldsymbol{\tau}_{2k+1}\ |z(t)|^{2k} z(t) \\ &= \sum_k b_{2k+1}\ |z(t)|^{2k} z(t), \end{aligned} \tag{105}$$

where

$$b_{2k+1} = \int \cdots \int h_{2k+1}(\boldsymbol{\tau}_{2k+1})\ d\boldsymbol{\tau}_{2k+1},$$

Petitioners Exhibit 1005
Page 63 of 117

and is generally complex valued according to (104).

Eq. (105) is equivalent to eq. (99). It is interesting to note that whether we start from the strictly memoryless model (98) or the Volterra model (101) with a narrowband input (the quasi-memoryless case), we both end up with the same baseband relationship (99). The difference is that in the strictly memoryless case, the b_k coefficients are real-valued and hence the power amplifier exhibits no AM/PM conversion (i.e., $\angle y(t) - \angle z(t)$ is zero or π). However, in the quasi-memoryless case, the b_k coefficients are generally complex-valued, thus creating AM/PM conversion.

In the next two sections, we will focus on the strictly memoryless and quasi-memoryless cases which have the same baseband representation (99). We will return to the discussion of nonlinear power amplifiers and predistorters with memory in Section 4.4.

4.2 Even-Order Terms in the Baseband Power Amplifier Model

If a physical power amplifier is modeled *exactly* by the polynomial nonlinearity (98) or the Volterra model (101), then indeed, we do not need to concern ourselves with even-order nonlinear terms. However, in reality, both (98) and (101) are only approximations. To the best of the authors' knowledge, Kim and Konstantinou [30] were the first to consider including even-order terms in power amplifier models. This means that instead of (99), one considers modeling the power amplifier as

$$y(t) = \sum_{k \in \mathcal{K}} b_k \; z(t)|z(t)|^{k-1}, \tag{106}$$

where $\mathcal{K}$ is a set containing all polynomial orders selected, which can be even or odd. Expressing $z(t) = |z(t)|e^{j\phi_z(t)}$, we can rewrite (106) as

$$y(t) = e^{j\phi_z(t)} F(|z(t)|), \tag{107}$$

where

$$F(|z(t)|) = \sum_{k \in \mathcal{K}} b_k |z(t)|^k. \tag{108}$$

Petitioners Exhibit 1005
Page 64 of 117

From (107) and (108), it follows that $|y(t)| = |F(|z(t)|)|$, $\angle y(t) - \angle z(t) = \angle F(|z(t)|)$; i.e., $|F(\cdot)|$ is the AM/AM response, $A(|z(t)|)$; and $\angle F(\cdot)$ is the AM/PM response, $\phi_A(|z(t)|)$. In other words,

$$F(|z(t)|) = A(|z(t)|)e^{j\phi_A(|z(t)|)}. \tag{109}$$

Given the measured AM/AM and AM/PM characteristics, we can calculate (109). We can then use (108) to solve for the b_k coefficients using linear least-squares. The following example compares the modeling accuracy when different choices of $\mathcal{K}$ are used in (108).

Example 4.1 We first measured AM/AM and AM/PM characteristics of an actual Class AB power amplifier and then constructed a look up table (LUT) for $F(\cdot)$ based on (109). We then fitted $F(|z(t)|)$ according to (108) using three sets of polynomial orders: $\mathcal{K}_1 = \{1,\ 3,\ 5\}$, $\mathcal{K}_2 = \{1,\ 3,\ 5,\ 7,\ 9\}$, and $\mathcal{K}_3 = \{1,\ 2,\ 3,\ 4,\ 5\}$. Note that $\mathcal{K}_1$ and $\mathcal{K}_3$ have the same maximum nonlinearity order whereas $\mathcal{K}_2$ and $\mathcal{K}_3$ contain the same number of terms. The polynomial coefficient estimates $\{\hat{b}_k\}$ were obtained via least-squares by regressing the measured $F(|z(t)|)$ over $|z(t)|^k$. We would like the resulting $\hat{F}(|z(t)|) = \sum_{k\in\mathcal{K}} \hat{b}_k|z(t)|^k$ to approximate $F(|z(t)|)$ well. In Figs. 29(a) and 29(b), the real and imaginary parts of the measured $F(|z(t)|)$ and the fitted $\hat{F}(|z(t)|)$ are shown. The real and imaginary parts of the fitting errors; i.e., $\hat{F}(|z(t)|) - F(|z(t)|)$, are plotted in Figs. 29(c) and 29(d), respectively, to allow easier evaluation of the goodness of fit. From these figures, we see that by including the even-order terms; i.e., including $k = 2, 4$ in (108), modeling accuracy was improved (compare the curves corresponding to $\mathcal{K}_1$ and $\mathcal{K}_3$). To give a quantitative measure of the improvement, we define a ratio, named the normalized mean square error (NMSE), as

$$\text{NMSE (dB)} = 10\log_{10}\left[\frac{\sum_{n=0}^{N-1}|F(|z(n)|) - \hat{F}(|z(n)|)|^2}{\sum_{n=0}^{N-1}|F(|z(n)|)|^2}\right], \tag{110}$$

where N is the total number of points in the LUT of $F(|z(n)|)$. The NMSEs that correspond to order sets $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$ are given in Table 1. It is seen that the choice of the nonlinearity orders $\mathcal{K}_3$ yielded the best result. Note here that although $\mathcal{K}_2$ and $\mathcal{K}_3$ contain the same number of terms, $\mathcal{K}_2$ involves higher order nonlinear terms than $\mathcal{K}_3$. In general, we would

Petitioners Exhibit 1005
Page 65 of 117

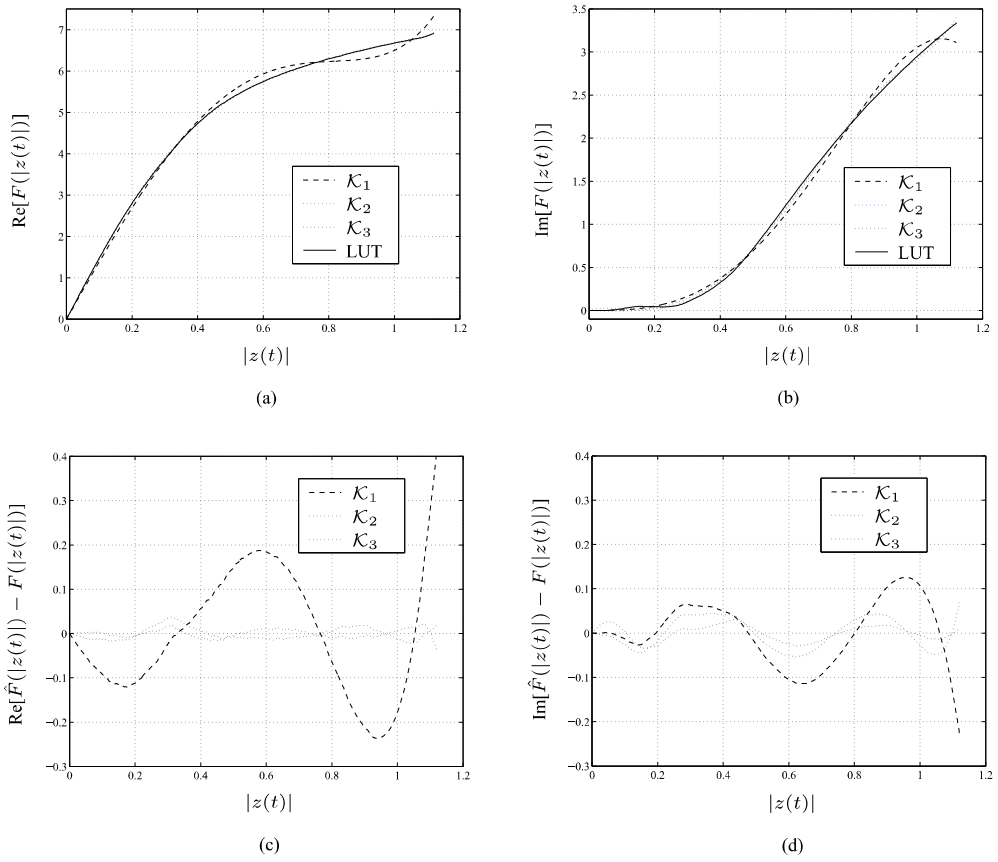


Figure 29: Fitting $F(|z(t)|)$ using polynomials of different orders. (a) and (b) show the real and imaginary parts of the measured $F(|z(t)|)$ and its polynomial approximations $\hat{F}(|z(t)|)$. (c) and (d) show the real and imaginary parts of the fitting error $\hat{F}(|z(t)|) - F(|z(t)|)$. In all figures, LUT refers to the measured power amplifier data, and the polynomial order sets $\mathcal{K}_1 = \{1,\ 3,\ 5\}$, $\mathcal{K}_2 = \{1,\ 3,\ 5,\ 7,\ 9\}$, $\mathcal{K}_3 = \{1,\ 2,\ 3,\ 4,\ 5\}$.

Table 1: NMSE for $F(|z(t)|)$

	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$
NMSE (dB)	-30.33	-41.49	-45.37

Petitioners Exhibit 1005
Page 66 of 117

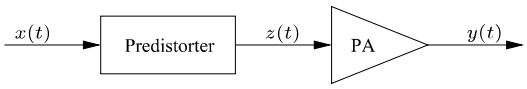


Figure 30: The predistorter precedes the power amplifier, and the objective is to have $y(t) \approx Cx(t)$, where C is a constant.

like to avoid high order polynomials because they do not extrapolate well outside of the interval of fit and the corresponding correlation matrix of $|z(t)|^k$ tends to be ill-conditioned.

We emphasize that the even k terms in the baseband power amplifier model (106) did not come from any even-order term in the passband power amplifier model (98). The fact that even k terms in (106) have any impact on power amplifier modeling is an indication that the polynomial model (98) is not precise. By including even-order terms in the baseband model, modeling error can be reduced. If we view $\{|z(t)|^k\}$ as the basis set, the set is obviously richer if even k values are allowed. To obtain a better fit to the measured power amplifier characteristics, one can choose between including even k terms and keeping the maximum order low, or allowing odd k values only and going for high orders. The first option is preferred, since low-order polynomials generally enjoy better numerical properties.

4.3 Even-Order Terms in the Baseband Predistorter Model

Similar to what we have seen for power amplifier modeling in the last section, including even-order nonlinear terms in a baseband predistorter model can improve linearization performance as well. In this section, we will first construct an ideal predistorter for a given power amplifier using a LUT, and then try to fit the LUT with a polynomial and assess the contributions from various nonlinear terms.

A diagram of the predistorter-power amplifier concatenation is shown in Fig. 30: $x(t)$ is the baseband predistorter input, $z(t)$ is the baseband predistorter output and input to the power amplifier, and $y(t)$ is the baseband power amplifier output. To find the ideal predistorter LUT values, we combine (107) - (109) to write

$$y(t) = A(|z(t)|)e^{j[\phi_A(|z(t)|)+\phi_z(t)]}. \tag{111}$$

Petitioners Exhibit 1005
Page 67 of 117

The predistorter may be expressed similarly as

$$z(t) = B(|x(t)|)e^{j[\phi_B(|x(t)|)+\phi_x(t)]}, \tag{112}$$

where $B(|z(t)|)$ and $\phi_B(|z(t)|)$ are the AM/AM and AM/PM responses of the predistorter, respectively. In other words,

$$\begin{aligned} |z(t)| &= B(|x(t)|) \\ \phi_z(t) &= \phi_B(|x(t)|) + \phi_x(t). \end{aligned} \tag{113}$$

Substituting (113) into (111), we obtain

$$y(t) = A(B(|x(t)|))e^{j[\phi_A(B(|x(t)|))+\phi_B(|x(t)|)+\phi_x(t)]}. \tag{114}$$

This way, we have expressed the power amplifier output $y(t)$ in terms of the predistorter input $x(t)$. Since the goal is to make the overall predistorter-power amplifier concatenation linear with a gain C; i.e., $y(t) = Cx(t)$, we need

$$\begin{aligned} A(B(|x(t)|)) &= C|x(t)| \\ \phi_A(B(|x(t)|)) &= -\phi_B(|x(t)|), \end{aligned} \tag{115}$$

where without loss of generality, we have assumed $C > 0$ is real valued. Therefore, the predistorter can be constructed from the AM/AM and AM/PM characteristics of the power amplifier as follows,

$$\begin{aligned} B(|x(t)|) &= A^{-1}\left(C|x(t)|\right) \\ \phi_B(|x(t)|) &= -\phi_A(B(|x(t)|)). \end{aligned} \tag{116}$$

This of course requires that A^{-1} exists, which usually is not a problem since most power amplifier's AM/AM conversion is monotonic. Similar to (108), we define a complex function $G(|x(t)|)$ as

$$G(|x(t)|) = B(|x(t)|)e^{j\phi_B(|x(t)|)}, \tag{117}$$

Petitioners Exhibit 1005
Page 68 of 117

and the predistorter input/output relationship can be written as

$$z(t) = \sum_{l\in\mathcal{L}} a_l \; x(t)|x(t)|^{l-1} \tag{118}$$

$$= e^{j\phi_x(t)} \sum_{l\in\mathcal{L}} a_l \; |x(t)|^l$$

$$= e^{j\phi_x(t)} G(|x(t)|), \tag{119}$$

where $\mathcal{L}$ is a set containing all polynomial orders selected for the predistorter. From equations (116) and (117), we see that given a memoryless power amplifier's AM/AM and AM/PM characteristics $A(\cdot)$ and $\phi_A(\cdot)$, we can calculate the predistorter $G(\cdot)$ function values. We can then obtain the $\hat{a}_l$ estimates by regressing $G(|x(t)|)$ over $|x(t)|^l$. Note that if the power amplifier is static and its AM/AM and AM/PM characteristics are known, then a predistorter constructed using the LUT is sufficient. However, in the absence of direct and exact power amplifier measurements, a parametric form of the predistorter is often desirable.

In the next example, we will assess whether the inclusion of even l terms in the predistorter (119) improves predistortion performance.

Example 4.2 The power amplifier is the same as in Example 4.1, whose AM/AM and AM/PM characteristics are known. We first obtained the predistorter LUT $G(\cdot)$ values according to (116) and (117). Next, we estimated the $\hat{a}_l$ coefficients that correspond to the order sets; i.e., $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$, $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$, and $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$. The resulting fitted $\hat{G}(|x(t)|) = \sum_{l\in\mathcal{L}} \hat{a}_l \; |x(t)|^l$. The real and imaginary parts of the fitting errors; i.e., $\hat{G}(|x(t)|) - G(|x(t)|)$, are shown in Figs. 31(a) and 31(b), respectively, and the NMSEs resulted from the fittings are given in Table 2. In Fig. 32, we compare the performance of the predistorters on an IS-95 CDMA signal, where the power spectra of the power amplifier outputs before and after predistortion are plotted. Comparing Tables 2 and 1,

Table 2: NMSE for $G(|x(t)|)$

	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$
NMSE (dB)	-39.24	-48.84	-48.32

we observe that while Set $\mathcal{K}_3$ gave around 4 dB of NMSE improvement over Set $\mathcal{K}_2$, Sets

Petitioners Exhibit 1005
Page 69 of 117

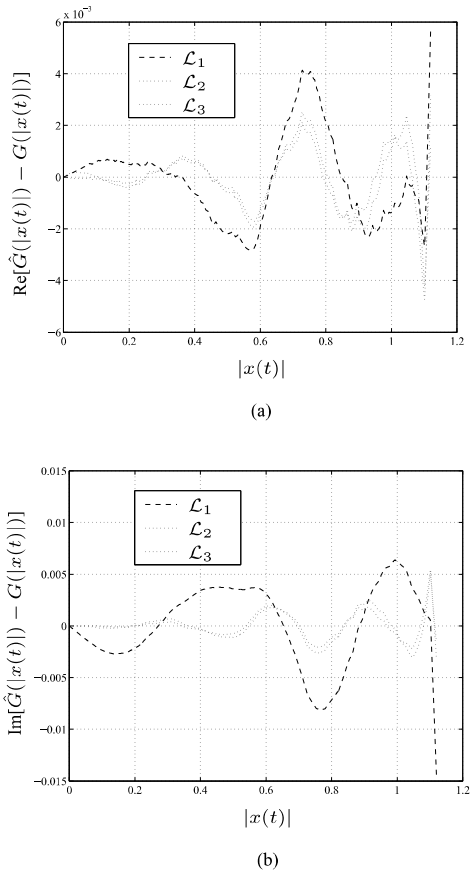


Figure 31: The errors $\hat{G}(|x(t)|) - G(|x(t)|)$ for three sets of polynomial orders $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$, $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$, and $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$. The real part of the error is shown in Figure (a), and the imaginary part of the error is shown in Figure (b). $\mathcal{L}_2$ and $\mathcal{L}_3$ resulted in comparable amount of error and both outperform $\mathcal{L}_1$.

Petitioners Exhibit 1005
Page 70 of 117

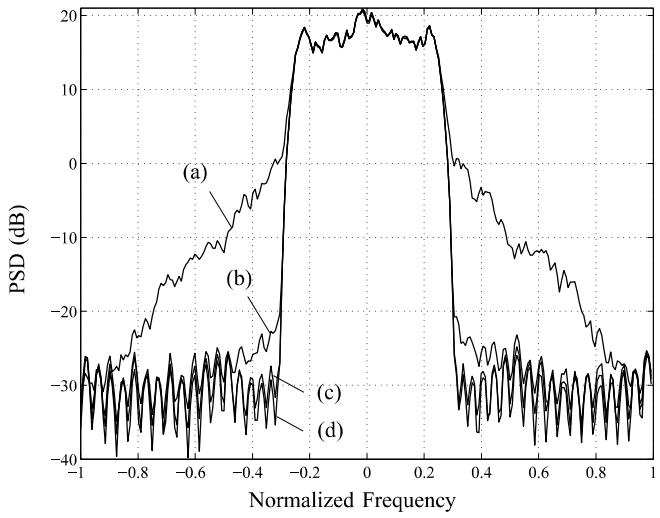


Figure 32: Predistortion linearization performance in terms of spectral regrowth suppression. power amplifier output power spectral density (PSD) is shown for the following cases: (a) there is no predistorter; (b) predistorter (119) with $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$ is used; (c) predistorter (119) with $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$ or predistorter (119) with $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ is used (the two lines coincide). (d) PSD of the original input.

$\mathcal{L}_3$ and $\mathcal{L}_2$ resulted in similar NMSEs. This is because the predistorter generally has an expanding nonlinearity, which can be fitted well by high order polynomials. On the other hand, the power amplifier usually has a compressing characteristic, for which the inclusion of high order polynomial terms cannot give as much improvement as for power amplifier modeling. Nonetheless, in both power amplifier and predistorter models, the benefit of including even-order terms is clear. By including even-order terms, modeling accuracy is improved and generally lower order polynomials can be used.

The rationale for including even-order terms in the predistorter is similar to that for the power amplifier. Given a nonlinear power amplifier and assuming that it is invertible, there exists an ideal predistorter. By allowing even l terms in the predistorter model, we have at hand a richer basis set to fit the predistorter and hence potential gains in predistorter performance.

Petitioners Exhibit 1005
Page 71 of 117

4.4 Extensions to Power Amplifiers and Predistorters with Memory

The same argument regarding the benefits of even-order nonlinear terms carry over to power amplifiers and predistorters with memory. As an example, let us consider the memory polynomial power amplifier model recently proposed by Kim and Konstantinou [30], which has the following form,

$$y(n) = \sum_{k \in \mathcal{K}} \sum_{p=0}^{P-1} b_{kp}\ z(n-p)|z(n-p)|^{k-1}, \tag{120}$$

where $\mathcal{K}$ is a set containing the polynomial orders, and $P-1$ is the maximum delay. The next example shows that if we keep the same maximum nonlinear order but include the even-order nonlinear terms as well, power amplifier modeling accuracy can be improved.

Example 4.3 The baseband power amplifier input $z(n)$ is a 15 MHz 3-carrier WCDMA signal. The wideband power amplifier has memory effects. The memory polynomial coefficients estimates $\{\hat{b}_{kp}\}$ were obtained from measured $z(n)$, $y(n)$ data and equation (120) by least-squares. We compared three different memory polynomials, which all have $P = 3$ but different order sets: $\mathcal{K}_1 = \{1,\ 3,\ 5\}$, $\mathcal{K}_2 = \{1,\ 3,\ 5,\ 7,\ 9\}$, and $\mathcal{K}_3 = \{1,\ 2,\ 3,\ 4,\ 5\}$. To give a quantitative measure of the approximation accuracy, we define a normalized mean square error similar to (110); i.e.,

$$\text{NMSE (dB)} = 10 \log_{10} \left[\frac{\sum_{n=0}^{N-1} |y(n) - \hat{y}(n)|^2}{\sum_{n=0}^{N-1} |y(n)|^2} \right], \tag{121}$$

where

$$\hat{y}(n) = \sum_{k \in \mathcal{K}} \sum_{p=0}^{P-1} \hat{b}_{kp}\ z(n-p)|z(n-p)|^{k-1}. \tag{122}$$

The NMSEs that correspond to the three different memory polynomial power amplifier models are given in Table 3.

Table 3: NMSE when the power amplifier is modeled by memory polynomial.

	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$
NMSE (dB)	-36.40	-37.38	-37.44

Petitioners Exhibit 1005
Page 72 of 117

The memory polynomial is also shown to be a robust predistorter model with memory [17]:

$$z(n) = \sum_{l \in \mathcal{L}} \sum_{q=0}^{Q-1} a_{lq}\ x(n-q)|x(n-q)|^{l-1}, \tag{123}$$

where $\mathcal{L}$ is a set containing the polynomial orders, and $Q-1$ is the maximum delay. The memory polynomial predistorter in (123) cannot be obtained directly similar to memoryless predistorter construction. We shall use the indirect learning structure [21] to estimate the predistorter coefficients for power amplifiers exhibiting memory effects.

Example 4.4 In this example, we used the memory polynomial power amplifier model described in Example 3.7, which were extracted from an actual power amplifier with a 15 MHz WCDMA input. The predistorters all have the same $Q = 4$ but different nonlinear orders: $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$, $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$, and $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$. In Fig. 33, we show the predistortion performance in terms of spectral regrowth suppression, where a 3-carrier WCDMA signal with 15 MHz bandwidth is used. The difference between Sets $\mathcal{L}_1$ and $\mathcal{L}_3$ is that even-order nonlinear terms are included in $\mathcal{L}_3$, which gave about 5 dB of additional spectral regrowth suppression for this particular example. The order set $\mathcal{L}_2$ has comparable performance as $\mathcal{L}_3$, but the maximum nonlinear order was 13 instead of 7.

These examples show that the benefits of including even-order nonlinear terms apply to power amplifiers and predistorters with memory effects as well.

Petitioners Exhibit 1005
Page 73 of 117

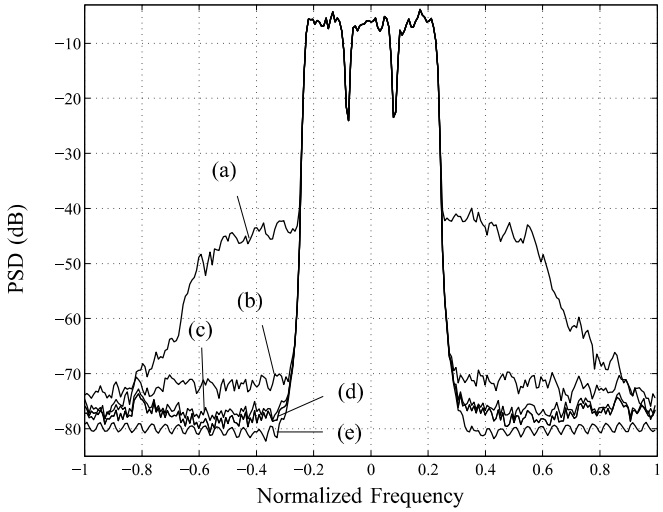


Figure 33: Predistortion linearization performance in terms of spectral regrowth suppression. Power amplifier output power spectral density (PSD) is shown for the following cases: (a) there is no predistorter; (b) predistorter (123) with $\mathcal{L}_1 = \{1,\ 3,\ 5,\ 7\}$ and $Q = 4$ is used; (c) predistorter (123) with $\mathcal{L}_2 = \{1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13\}$ and $Q = 4$ is used; (d) predistorter (123) with $\mathcal{L}_3 = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\}$ and $Q = 4$ is used. (e) PSD of the original input.

Petitioners Exhibit 1005
Page 74 of 117

CHAPTER 5

ANALOG IMPERFECTION COMPENSATION

The ideal performance of digital predistortion relies on robust predistorters that can completely compensate for the nonlinearities of the power amplifier. In reality, however, the performance can also be affected by the analog imperfections in the transmitter. In this chapter, we study modeling and compensation of analog imperfections in two types of transmitters: one uses two-stage upconversion, and the other uses direct upconversion.

Here, we assume that the analog imperfections in the feedback path are negligible, which is usually true with carefully designed downconverter (e.g., filters with relatively flat frequency response, such as LC bandpass and lowpass filters, and digital demodulation, which is free of any demodulation errors).

5.1 Two-Stage Upconversion Transmitter

In the upconverter, another configuration is to use digital modulation and two-stage upconversion, i.e., the baseband signal is first converted to IF and then to RF. Because of the stringent image rejection requirements of the transmitter, a SAW filter is often used in the IF stage for this configuration (see Fig. 34). However, the SAW filter tends to have relatively large magnitude and phase variations over frequency, thereby distorting the predistorted waveform. Ma *et al.* [31] proposed to use an equalizer after the predistorter to compensate for the frequency response of the SAW filter and provided an empirical approach to construct the equalizer. In this memorandum, we present a frequency-domain least-squares approach to design the equalizer. This approach yields equalization over a desired frequency band while maintaining low energy outside of the desired band.

5.1.1 System Setup

Figure 35 shows the overall diagram of a predistortion linearization system. The baseband input $u(n)$ first goes through the predistorter, whose output $z(n)$ is then equalized

Petitioners Exhibit 1005
Page 75 of 117

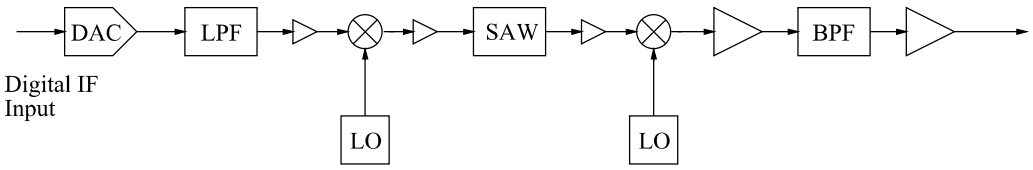


Figure 34: Two-stage upconversion transmitter.

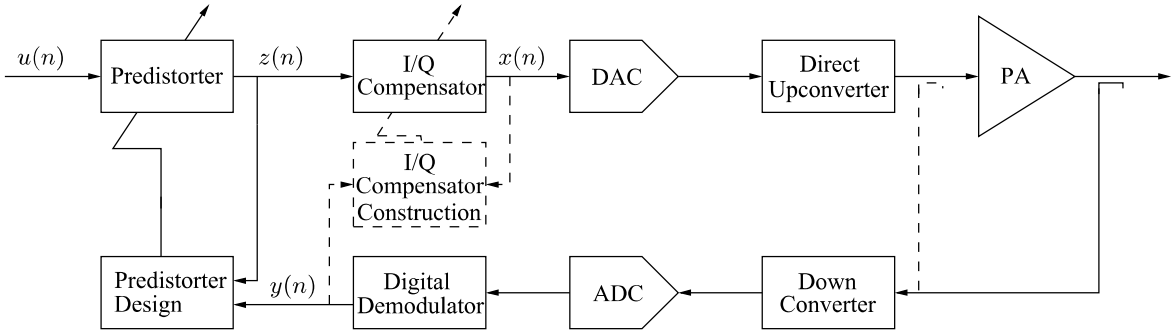


Figure 35: Block diagram of the baseband predistortion system with equalization. The dashed lines refer to the feedback path for equalizer training.

to generate $x(n)$. After digital modulation and digital-to-analog conversion, the signal is upconverted to the carrier frequency and amplified by the power amplifier. The feedback path with the solid lines is used for predistorter training, while the dashed lines show the feedback loop for the equalizer construction. Because of the frequency response of the upconverter, the equalizer needs to be trained first to compensate for the upconverter before the predistorter design. Thus, in the training phase, we first let $x(n) = z(n) = u(n)$ and obtain the feedback $y(n)$ using the dashed-line loop. Based on $x(n)$ and $y(n)$, we design the equalizer and plug it into the transmitter. Then the solid-line loop is activated, and the training for the predistorter starts. Note that in the equalizer training phase, the input signal $u(n)$ should be able to excite the whole frequency range that we aim to equalize.

5.1.2 Channel Estimation

Given $x(n)$ and $y(n)$, our goal is to design an equalizer that can equalize the channel in the band of interest and has low energy outside of the band. In this section, we first present a least-squares method to estimate the channel. Then we provide a frequency-domain least-squares approach to design an equalizer for the channel in the next section.

Petitioners Exhibit 1005
Page 76 of 117

For a linear channel with impulse response $h(k)$ and input $x(n)$, the output $y(n)$ of the channel is given by

$$y(n) = \sum_{k=0}^{K-1} x(n-k)h(k), \tag{124}$$

where K is the length of the channel. For a block of N data samples, $x(n)$ and $y(n)$, $0 \leq n \leq N-1$, (124) can be written in vector form, i.e.,

$$\mathbf{y} = \mathbf{X}\mathbf{h}, \tag{125}$$

where $\mathbf{y} = [y(K-L-1) \quad y(K-L) \quad \dots \quad y(N-L-1)]^T$ with L a selectable delay, $\mathbf{h} = [h(0) \quad h(1) \quad \dots \quad h(K-1)]^T$, and

$$\mathbf{X} = \begin{bmatrix} x(K-1) & x(K-2) & x(K-3) & \cdots & x(0) \\ x(K) & x(K-1) & x(K-2) & \cdots & x(1) \\ \vdots & \vdots & \vdots & & \vdots \\ x(N-1) & x(N-2) & x(N-3) & \cdots & x(N-K) \end{bmatrix}. \tag{126}$$

In this formulation, not all N samples are used in order to avoid the boundary effect. Moreover, we choose the delay L to be

$$L = \lfloor (K-1)/2 \rfloor, \tag{127}$$

where $\lfloor x \rfloor$ is the largest integer that is less than or equal to x, such that the tap with the maximum magnitude is approximately located at the center of the channel's impulse response. The delay introduced in $\mathbf{y}$ means that the sampled channel response is noncausal; for example, $y(K-L-1)$ depends not only on the past input $x(n)$, $0 \leq n \leq K-L-1$, but also on the future input $x(n)$, $K-L-1 < n \leq K-1$. It is true that the underlying real physical system is always causal. However, the delay caused by the analog system can fall between sampling intervals and make the system appear noncausal.

For example, suppose a continuous-time signal $s(t)$ is constructed from a discrete-time signal $s(n)$. Then we know the following relationship holds [37, p. 88]:

$$s(t) = \sum_{n=-\infty}^{\infty} s(n)\ \mathrm{sinc}[(t-nT)/T], \tag{128}$$

Petitioners Exhibit 1005
Page 77 of 117

where

$$\text{sinc}(x) \quad = \quad \frac{\sin(\pi x)}{\pi x}. \tag{129}$$

When $s(t)$ passes through a continuous-time system with impulse response $\delta(t-\tau_0)$, where $0 < \tau_0 < T$, the sampled output of the system, $\tilde{s}(m)$, is given by

$$\tilde{s}(m) \quad = \quad s(mT) \quad = \quad \sum_{n=-\infty}^{\infty} s(n)\ \text{sinc}[(\tau_0 + mT - nT)/T]. \tag{130}$$

Defining $k = m - n$ and substituting into (130), we have

$$\tilde{s}(m) \quad = \quad \sum_{k=-\infty}^{\infty} s(m-k)\ \text{sinc}\left[\frac{\tau_0}{T} + k\right]. \tag{131}$$

From (131), it follows that the impulse response of the discrete-time system is

$$h(k) = \text{sinc}\left[\frac{\tau_0}{T} + k\right], \quad k = -\infty, \ldots, \infty. \tag{132}$$

Therefore, the equivalent discrete-time system has a noncausal response although the underlying continuous-time system is causal. It is easy to show that the discrete-time Fourier transform (DTFT) of $h(k)$ is

$$H(e^{j\omega}) = e^{j\omega\tau_0/T}, \tag{133}$$

which indicates that the discrete-time system only introduces a non-integer delay to the input signal. If we assume that the input $s(n)$ is uncorrelated and the discrete-time system is causal, the minimum mean-square error estimate of the system impulse response $h_c(k)$ can be shown to be

$$h_c(k) = \text{sinc}\left[\frac{\tau_0}{T} + k\right], \quad k = 0, 1, \ldots, \infty. \tag{134}$$

An explicit expression of $H_c(e^{j\omega})$, the DTFT of $h_c(k)$, is difficult to obtain. In Fig. 36, we illustrate the difference between $H_c(e^{j\omega})$ and $H(e^{j\omega})$ through numerical simulations with $\tau_0 = T/16$, $-200 \le k \le 200$ for $h(k)$, and $0 \le k \le 200$ for $h_c(k)$. We can see that the noncausal system preserves the characteristics of the underlying continuous-time system while the causal system distorts the magnitude response.

Petitioners Exhibit 1005
Page 78 of 117

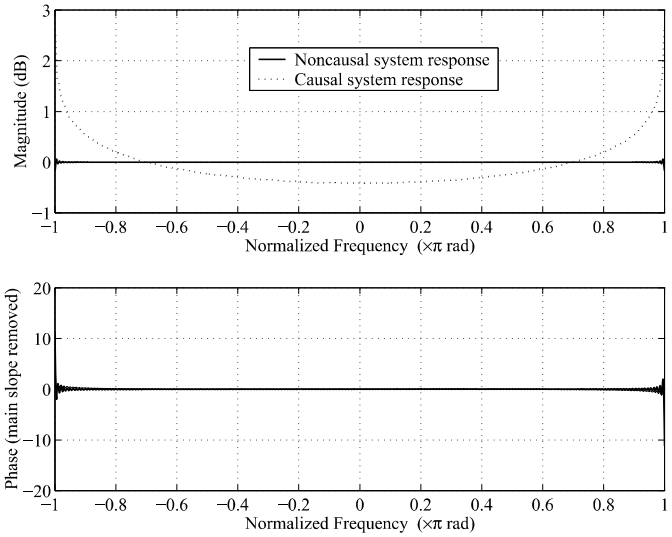


Figure 36: Comparison between noncausal system response $H(e^{j\omega})$ (solid line) and causal system response $H_{\rm c}(e^{j\omega})$ (dotted-line). Note that the mean slopes of the phase responses have been removed. The phase responses of $H(e^{j\omega})$ and $H_{\rm c}(e^{j\omega})$ coincide.

In conclusion, if the synchronization between $x(n)$ and $y(n)$ is not perfect, i.e., non-integer delay exists, it is better to model the underlying continuous time system as a non-causal discrete-time system.

To estimate the channel filter $\mathbf{h}$ in (125), we define the least-squares cost function as

$$J = (\mathbf{y} - \mathbf{X}\mathbf{h})^H(\mathbf{y} - \mathbf{X}\mathbf{h}), \tag{135}$$

where $(\cdot)^H$ denotes complex conjugate transpose, $\mathbf{y}$ is from the real measurements, and $\mathbf{X}\mathbf{h}$ is the modeled output. The least-squares estimate of $\mathbf{h}$, which minimizes J, can then be easily obtained as

$$\hat{\mathbf{h}} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}. \tag{136}$$

5.1.3 Equalizer Design

Since the upconverter channel has bandpass characteristics, a direct inverse of the channel would cause the equalizer to have a bandstop type of response, i.e., very large out-of-band response. To avoid this, the equalizer in our design is bandlimited; i.e., it equalizes the channel within the band of interest and has low out-of-band response.

Let $a(k)$, $k = 0, 1, \cdots, K-1$, denote the equalizer. Then the convolved response of the

Petitioners Exhibit 1005
Page 79 of 117

channel and the equalizer, denoted by $c(k)$, can be written as

$$c(k) \;=\; \sum_{l=0}^{K_a-1} h(k-l)a(l), \quad k = 0, 1, \cdots, K_a + K - 2, \tag{137}$$

which has length $K_c = K_a + K - 1$. To design the bandlimited equalizer, we first define the in-band cost function as:

$$J_1 \;=\; \int_{-\omega_\mathrm{p}}^{\omega_\mathrm{p}} \left[\sum_{k_1=0}^{K_c-1} c(k_1)e^{-j\omega k_1} - e^{-j\omega n_0}\right] \left[\sum_{k_2=0}^{K_c-1} c(k_2)e^{-j\omega k_2} - e^{-j\omega n_0}\right]^* \mathrm{d}\omega, \tag{138}$$

where $e^{-j\omega n_0}$ is the desired frequency response of the equalizer-channel cascade within the passband $[-\omega_\mathrm{p}, \omega_\mathrm{p}]$. To minimize the out-of-band energy of the equalizer, the out-of-band cost function is defined as

$$\begin{aligned} J_2 \;=\; & \int_{-\pi}^{-\omega_\mathrm{p}} \left[\sum_{l=0}^{K_a} a(l)e^{-j\omega l}\right] \left[\sum_{k=0}^{K_a} a(k)e^{-j\omega k}\right]^* \mathrm{d}\omega \\ & + \int_{\omega_\mathrm{p}}^{\pi} \left[\sum_{l=0}^{K_a} a(l)e^{-j\omega l}\right] \left[\sum_{k=0}^{K_a} a(k)e^{-j\omega k}\right]^* \mathrm{d}\omega. \end{aligned} \tag{139}$$

Therefore, the overall cost function can be written as

$$J = J_1 + w\, J_2, \tag{140}$$

where w is a weighting factor. The optimal equalizer minimizes J and can be found by taking the partial derivative of J with respect to $a^*(k)$ [pretending $a(k)$ is constant [5]] and setting it to zero, i.e.,

$$\frac{\partial J}{\partial a^*(k)} \;=\; \frac{\partial J_1}{\partial a^*(k)} + w\,\frac{\partial J_2}{\partial a^*(k)} \;=\; 0, \quad k \;=\; 0,\; \ldots,\; K_a - 1. \tag{141}$$

To find $\dfrac{\partial J_1}{\partial a^*(k)}$, we substitute (137) into (138) and take the derivative, which yields

$$\begin{aligned} \frac{\partial J_1}{\partial a^*(k)} \;=\; & \int_{-\omega_\mathrm{p}}^{\omega_\mathrm{p}} \left[\sum_{k_1=0}^{K_c-1} \sum_{l=0}^{K_a-1} h(k_1-l)a(l)e^{-j\omega k_1} - e^{-j\omega n_0}\right] \\ & \times \sum_{k_2=0}^{K_c-1} h^*(k_2-k)e^{j\omega k_2}\, \mathrm{d}\omega. \end{aligned} \tag{142}$$

Rearranging (142) and carrying out the integration, we obtain

$$\begin{aligned} \frac{\partial J_1}{\partial a^*(k)} \;=\; & \sum_{l=0}^{K_a-1} a(l) \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} h(k_1-l)\; h^*(k_2-k)\; 2\omega_\mathrm{p}\; \mathrm{sinc}\left[\frac{\omega_\mathrm{p}(k_1-k_2)}{\pi}\right] \\ & - \sum_{k_2=0}^{K_c-1} h^*(k_2-k)\; 2\omega_\mathrm{p}\; \mathrm{sinc}\left[\frac{\omega_\mathrm{p}(n_0-k_2)}{\pi}\right]. \end{aligned} \tag{143}$$

Petitioners Exhibit 1005
Page 80 of 117

Similarly,

$$\frac{\partial J_2}{\partial a^*(k)} = \sum_{l=0}^{K_a-1} a(l) \left\{ 2\pi \,\mathrm{sinc}(l-k) - 2\omega_\mathrm{p} \,\mathrm{sinc}\left[\frac{\omega_\mathrm{p}(l-k)}{\pi}\right] \right\}. \tag{144}$$

Substituting (143) and (144) into (141) , rearranging the result, and rewriting it in matrix form, we have

$$(\mathbf{R}_1 + w\,\mathbf{R}_2)\ \mathbf{a} = \mathbf{h}_\mathrm{p}, \tag{145}$$

where $\mathbf{a} = [a(0)\ \ldots\ a(K_a-1)]^T$, and the elements of matrix $\mathbf{R}_1$, $\mathbf{R}_2$ and column vector $\mathbf{h}_\mathrm{p}$ are defined as

$$\mathbf{R}_1(l,k) = \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} 2\omega_\mathrm{p}\, h(k_1-l)\ h^*(k_2-k)\ \mathrm{sinc}\left[\frac{\omega_\mathrm{p}(k_1-k_2)}{\pi}\right] \tag{146}$$

$$\mathbf{R}_2(l,k) = 2\pi \,\mathrm{sinc}(l-k) - 2\omega_\mathrm{p} \,\mathrm{sinc}\left[\frac{\omega_\mathrm{p}(l-k)}{\pi}\right] \tag{147}$$

$$\mathbf{h}_\mathrm{p}(k) = \sum_{k_2=0}^{K_c-1} 2\omega_\mathrm{p}\, h^*(k_2-k)\ \mathrm{sinc}\left[\frac{\omega_\mathrm{p}(n_0-k_2)}{\pi}\right]. \tag{148}$$

Note that $\mathbf{R}_1$ and $\mathbf{R}_2$ are Toeplitz matrices. Thus, only the first row and column of the matrices need to be calculated. From (145), we can see that the optimal $\mathbf{a}$, which minimizes J, is given by

$$\hat{\mathbf{a}} = (\mathbf{R}_1 + w\,\mathbf{R}_2)^{-1}\ \mathbf{h}_\mathrm{p}. \tag{149}$$

5.1.4 Experimental Results

In this section, we present experimental results from our digital predistortion test bed. The configuration of the test bed is shown in Fig. 37. The digital data play, record, and computations are done by a Celerity system with 150 MSPS maximum I/O rate. It sends out 16-bit I/Q data streams to the DAC evaluation board continuously and acquires 12-bit digital IF data samples from the ADC evaluation board when needed. The DAC and ADC used here are, respectively, AD9777 and AD9430 from Analog Devices. The digital modulation can be done either in the Celerity system or the DAC, while the digital demodulation is done in the Celerity system. The power amplifier under test is a KLAM.

The equalizer is designed to equalize the upconverter channel for the predistorted waveform. Therefore, the training signal for the equalizer needs to cover the signal band of

Petitioners Exhibit 1005
Page 81 of 117

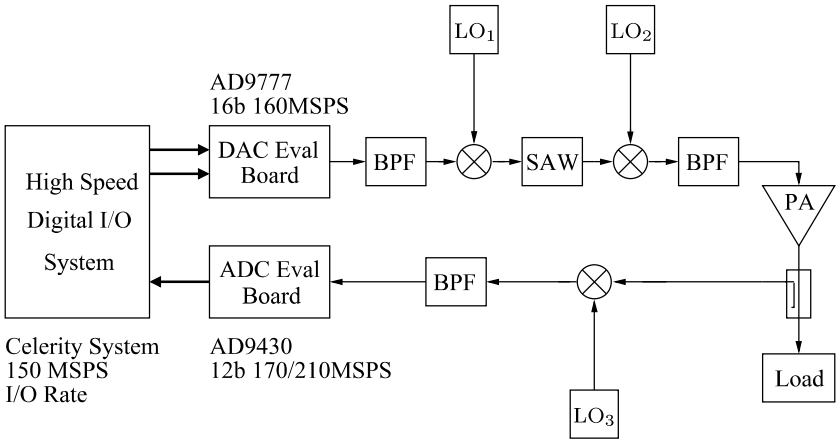


Figure 37: Block diagram of the test bed.

the predistorted signal. We can design a dedicated training signal beforehand to cover the desired band. In practice, we may also turn off the equalizer first, construct a temporary predistorter, and use its output as a training signal. This signal covers most of the desired band and is easily available. An example of such a training signal is shown in Fig. 38.

With the training signal, the dashed loop in Fig. 35 was used for collecting the input and output data of the upconverter channel. Fig. 39 shows the frequency responses of the estimated channel, the equalizer, and the cascade of the two. Here, the estimated channel has 51 taps. More taps help to reduce the windowing effects of the estimated channel filter, thereby providing a finer resolution of the channel. The equalizer is assumed to be 14 taps. The relatively short length of the equalizer helps to reduce the implementation cost. The equalizer is designed with $\omega_{\rm p} = 0.7\pi$ and $w = 10^{-3}$. Recall that $[-\omega_{\rm p}, \omega_{\rm p}]$ is the frequency band of equalization and w is the weighting factor in the cost function (140). Increasing w reduces the out-of-band response of the equalizer. However, this is at the expense of decreasing the accuracy of the equalizer in-band. Thus, there is tradeoff when selecting the right weighting factor w.

The performance of the equalizer in a predistortion system is shown in Fig. 40. The equalizer was designed with $\omega_{\rm p} = 0.7\pi$ and $w = 10^{-3}$. We see that the equalizer significantly improves the linearization performance of the predistortion system.

Petitioners Exhibit 1005
Page 82 of 117

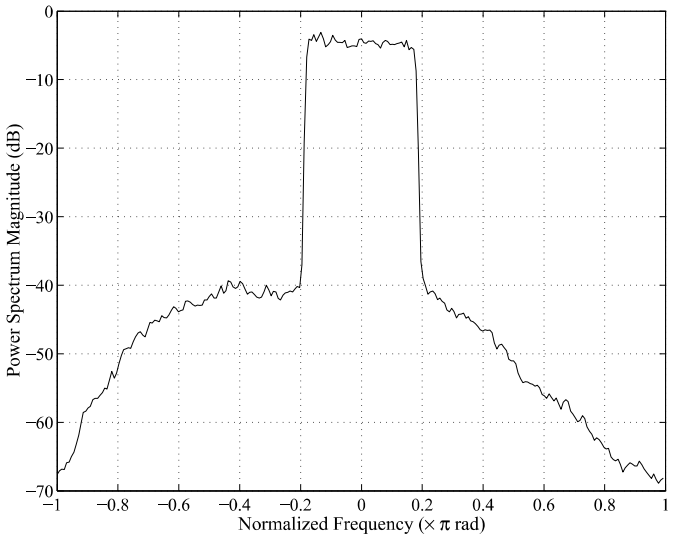


Figure 38: Example of a training signal from predistorted waveform with the equalizer turned off.

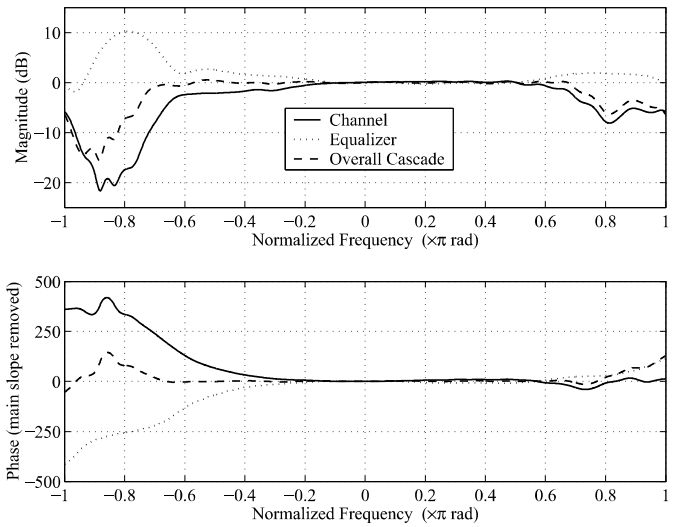


Figure 39: Magnitude and phase responses of the estimated channel (solid line), the equalizer (dotted line), and the overall cascade of the channel and the equalizer (dashed line). The estimated channel has 51 taps. The equalizer has 14 taps and is designed with $\omega_{\mathrm{p}}= 0.7\pi$, $w = 10^{-3}$.

Petitioners Exhibit 1005
Page 83 of 117

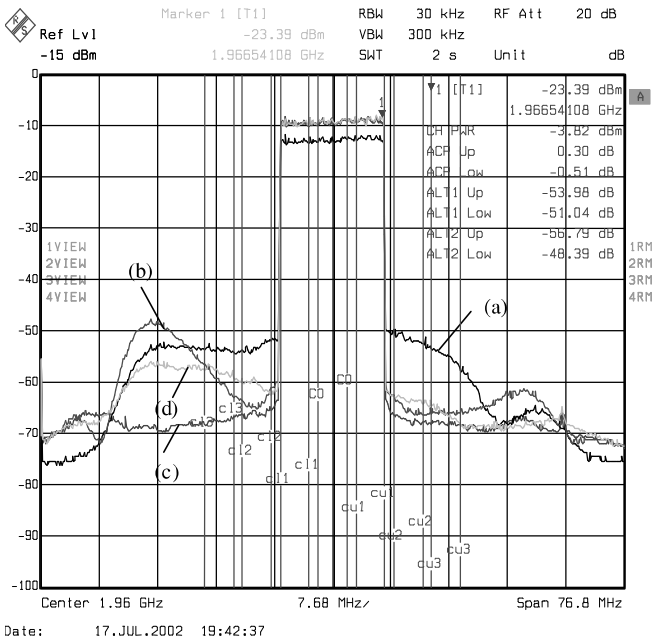


Figure 40: Comparison of power spectral densities (PSDs). (a) Training signal; (b) Power Amplifier output with predistortion but no equalizer; (c) PA output with predistortion and equalizer; (d) PA output with the same predistorter as in (c) but no equalizer.

Petitioners Exhibit 1005
Page 84 of 117

5.2 Direct Upconversion Transmitter

In the direct upconversion transmitter, the I/Q data streams are directly modulated to RF. This structure enables the upconverter to be easily reconfigured to generate RF signals in different frequency bands. It also uses fewer components and is easier to integrate than two-stage upconversion. However, in practice, the quadrature carriers in the analog modulator do not have exactly the same amplitudes and an exact phase difference of 90 degrees. These effects are called gain/phase imbalance and can cause cross-talk between the I and Q channels. Note that the asymmetry between the analog reconstruction filters in the I and Q branches before the modulator also contributes to the imbalance. In addition, leakage of the carrier to the transmitted signal manifests itself in the demodulated received signal as a dc offset.

For narrowband inputs, the gain/phase imbalance can be considered as frequency-independent. Its impacts on predistortion have been analyzed (e.g. [7]), and various compensation techniques have been proposed (e.g. [9], [8]). For wideband inputs, the gain/phase imbalance exhibits frequency-dependent behavior, which may be due to both the reconstruction filters and analog modulator. A compensation technique for the reconstruction filters is proposed in [45]. In [46], the modeling and compensation of both the reconstruction filters and analog modulator in wideband receivers are considered, but the modulator is assumed to be frequency-independent. Moreover, since the image in [46] is caused by adjacent channel interference, which is not available at the receiver, the compensation techniques in [46] are "blind". In a predistortion system, a feedback path is often present. Therefore, the output of the direct upconverter is usually available. In this section [12], we propose a general model that describes the effects of both the reconstruction filters and the frequency-dependent analog modulator, which we refer to as frequency-dependent gain/phase imbalance and dc offset. Based on the input and output of the direct upconverter, we then develop techniques to extract the model parameters and construct compensators.

Petitioners Exhibit 1005
Page 85 of 117

5.2.1 Channel Models

Figure 35 illustrates the general structure of the baseband predistortion system considered here. As mentioned in the previous section, we use direct upconversion in the transmitter chain, and a one-stage downconverter and digital demodulator in the feedback path. Additional baseband processing for the input signal $u(n)$ in the transmitter includes predistortion and I/Q compensation, whose outputs are denoted by $z(n)$ and $x(n)$, respectively. During the initialization phase, the predistorter and I/Q compensator are bypassed: i.e., $x(n) = z(n) = u(n)$. We also bypass the power amplifier and acquire the direct upconverter output in baseband, i.e., $y(n)$. Based on $x(n)$ and $y(n)$, we can estimate the parameters of the channel model and construct the I/Q compensator (shown by the dashed loop). After the I/Q compensator is activated, we put the power amplifier back in the loop and start training the predistorter (shown by the solid loop).

A detailed view of the channel, from $x(n)$ to $y(n)$, is shown in Fig. 42, where $\mathrm{Re}\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$ denote the real and imaginary parts of a complex number, respectively. We also use subscripts i (in-phase) and q (quadrature) to denote the real and imaginary parts of a complex sequence; for example, in Fig. 42, we have $x(n) = x_{\mathrm{i}}(n) + jx_{\mathrm{q}}(n)$ and $y(n) = y_{\mathrm{i}}(n) + jy_{\mathrm{q}}(n)$.

Real I/Q Channel Model

The frequency-dependent gain/phase imbalance comes from the frequency-dependent behavior of the analog components on the I and Q paths. To model the I and Q channels and the cross coupling channels between them, we use four real filters, $\mathbf{h}_{11}, \mathbf{h}_{12}, \mathbf{h}_{21}$, and $\mathbf{h}_{22}$ (see Fig. 43(a)). The channel output $y(n)$ can then be written as

$$\begin{aligned} y(n) = & \sum_{k=0}^{K-1} \{[x_{\mathrm{i}}(n-k)h_{11}(k) + x_{\mathrm{q}}(n-k)h_{12}(k)] + j[x_{\mathrm{q}}(n-k)h_{22}(k) + x_{\mathrm{i}}(n-k)h_{21}(k)]\} \\ & + d_{\mathrm{i}} + w_{\mathrm{i}}(n) + j[d_{\mathrm{q}} + w_{\mathrm{q}}(n)], \end{aligned} \tag{150}$$

where $x(n) = x_{\mathrm{i}}(n) + jx_{\mathrm{q}}(n)$ is the baseband input, $w(n) = w_{\mathrm{i}}(n) + jw_{\mathrm{q}}(n)$ is the additive white noise, and $d = d_{\mathrm{i}} + jd_{\mathrm{q}}$ is the dc offset. In (150), we assume that all four filters have the same length K, which helps to simplify the derivations in the following sections. However, the same methodology can still be applied if the $\mathbf{h}$ filters have different lengths.

Petitioners Exhibit 1005
Page 86 of 117

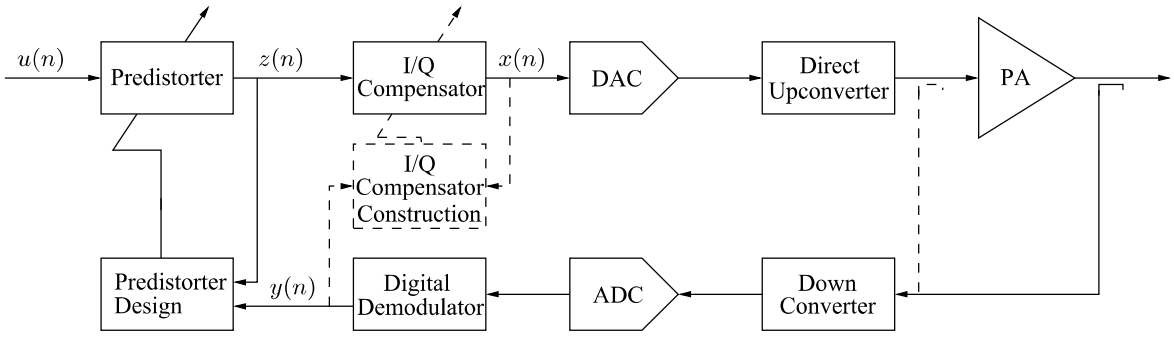


Figure 41: Block diagram of the baseband predistortion system. The dashed lines refer to the feedback loop for I/Q compensator training.

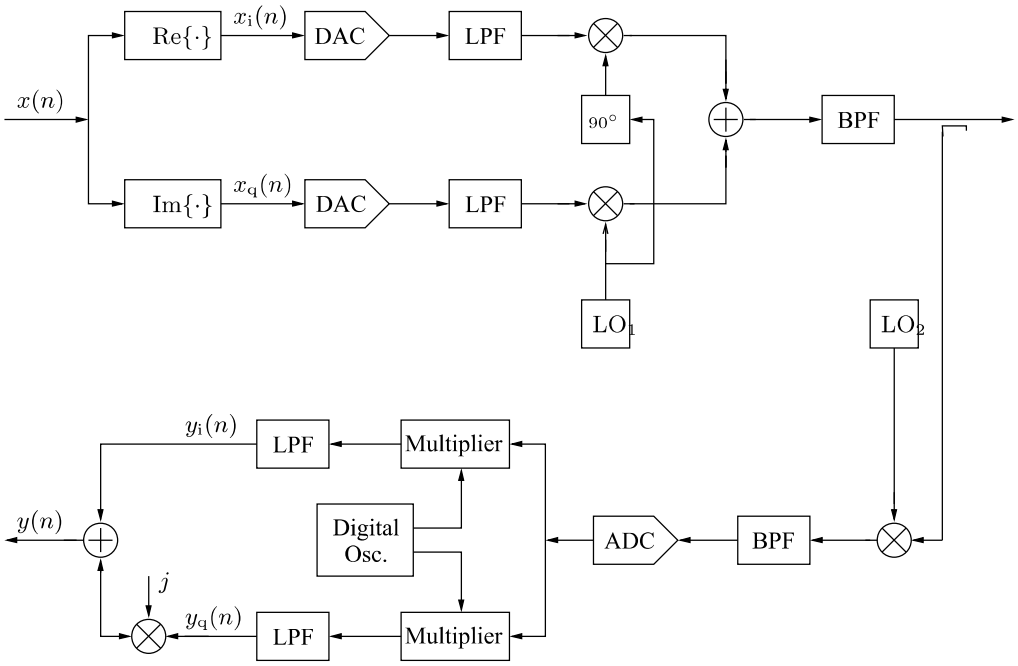


Figure 42: Detailed block diagram of the path from the DAC input $x(n)$ to the baseband feedback data samples $y(n)$.

Petitioners Exhibit 1005
Page 87 of 117

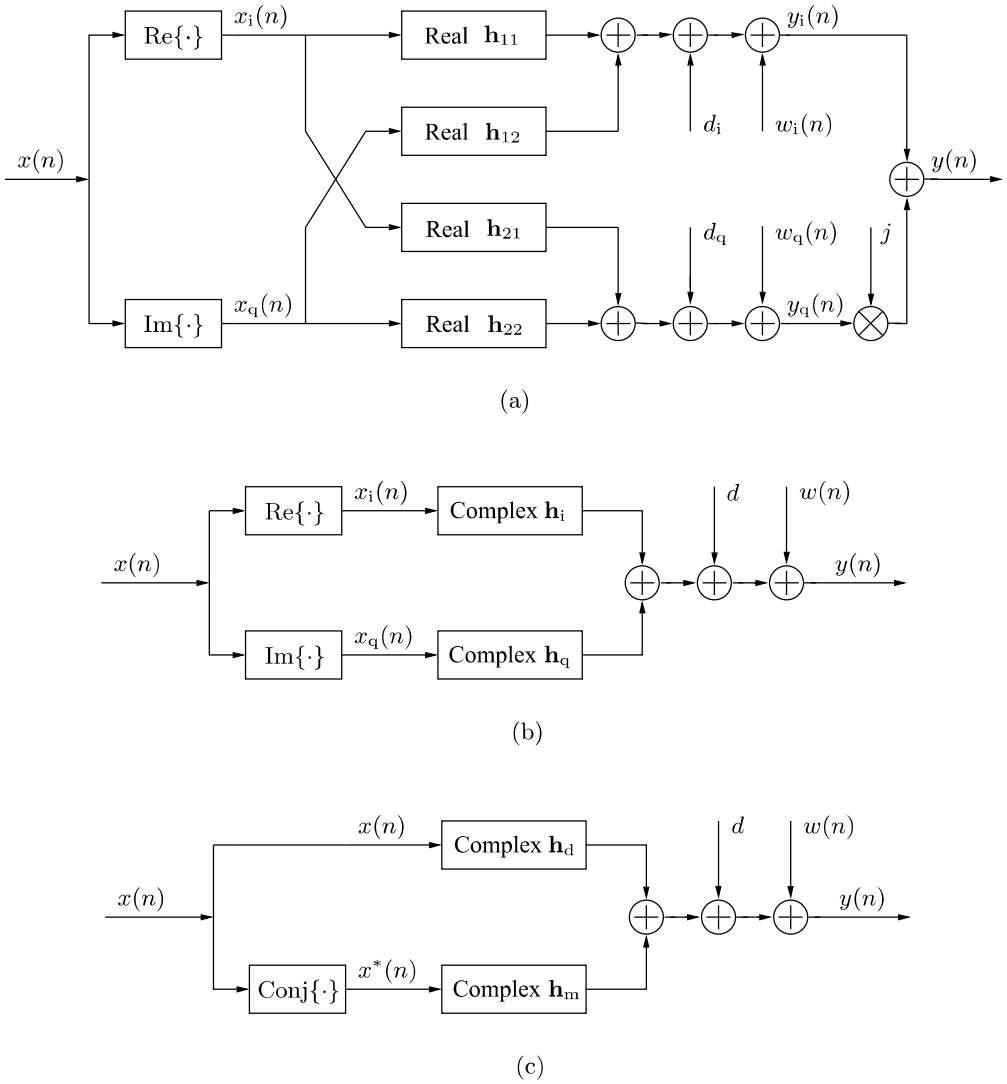


Figure 43: Block diagrams of three channel models: (a) real I/Q model; (b) complex I/Q model; (c) direct/image model.

Petitioners Exhibit 1005
Page 88 of 117

Complex I/Q Channel Model

In (150), we can combine the terms that have $x_{\rm i}(n-k)$ or $x_{\rm q}(n-k)$ and rewrite it in a more compact form; i.e.,

$$y(n) \;=\; \sum_{k=0}^{K-1} [x_{\rm i}(n-k)h_{\rm i}(k) + x_{\rm q}(n-k)h_{\rm q}(k)] + d + w(n), \tag{151}$$

where

$$h_{\rm i}(k) = h_{11}(k) + jh_{21}(k),\;\; h_{\rm q}(k) = h_{12}(k) + jh_{22}(k). \tag{152}$$

A block diagram of the complex I/Q channel model is shown in Fig. 43(b).

Direct/Image Channel Model

The complex I/Q model in (151) gives the relationship between $y(n)$ and the real and imaginary parts of $x(n)$. However, it is not clear from (151) how the modulator imbalance affects the input $x(n)$ as a whole. We know that

$$x_{\rm i}(n) \;=\; \frac{x(n) + x^*(n)}{2},\;\; x_{\rm q}(n) \;=\; \frac{x(n) - x^*(n)}{2j} \tag{153}$$

where $(\cdot)^*$ denotes complex conjugate. Substituting (153) into (151) and rearranging the result, we obtain

$$y(n) \;=\; \sum_{k=0}^{K-1} [x(n-k)h_{\rm d}(k) + x^*(n-k)h_{\rm m}(k)] + d + w(n), \tag{154}$$

where

$$h_{\rm d}(k) = \frac{h_{\rm i}(k) - jh_{\rm q}(k)}{2},\;\; h_{\rm m}(k) = \frac{h_{\rm i}(k) + jh_{\rm q}(k)}{2} \tag{155}$$

are, respectively, the *direct* and *image* transfer functions. Fig. 43(c) shows a block diagram of this channel model. Here, $x^*(n)$ is viewed as an image of $x(n)$ since when $x(n)$ has a one-sided spectrum, $x^*(n)$ shows up on the other side of the carrier. The image $x^*(n)$ is undesired and appears in the output when $h_{\rm i}(k) \neq h_{\rm q}(k)$, i.e., $h_{\rm m}(k) \neq 0$. When the input $x(n)$ has a double-sided spectrum, the image occupies the same spectrum as the input and is covered up by the original input. However, it may still degrade predistortion performance [7].

Petitioners Exhibit 1005
Page 89 of 117

The three models proposed in this section are all equivalent models. However, they reveal different aspects of the channel, and each has its own pros and cons. For implementation purposes, since all the complex operations have to be done in real numbers, the real I/Q model is the most convenient and efficient. For channel estimation purposes, the complex I/Q model is more compact than the real I/Q model. Moreover, its inputs are real sequences, which leads to a real correlation matrix in the estimation process. In contrast, with the direct/image model, we will have to deal with a complex correlation matrix. The advantage of the direct/image model is that it reveals the effects of the system on the input and its image, which cannot be directly observed from the other two models.

5.2.2 Channel Estimation

Because of the reasons stated above, we derive channel estimation algorithms based on the complex I/Q model.

Least-Squares Method

For a block of $x(n)$ and $y(n)$ data samples, (151) can be written in vector form; i.e.,

$$\mathbf{y} = \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} + \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}} + d\,\mathbf{1}_P + \mathbf{w}, \tag{156}$$

where $\mathbf{y} = [y(K-L-1),\ \ldots,\ y(N-L-1)]^T$ with L a selectable delay, $\mathbf{X}_{\mathrm{i}} = \mathrm{Re}(\mathbf{X})$ and $\mathbf{X}_{\mathrm{q}} = \mathrm{Im}(\mathbf{X})$ with

$$\mathbf{X} = \begin{bmatrix} x(K-1) & x(K-2) & \cdots & x(0) \\ x(K) & x(K-1) & \cdots & x(1) \\ \vdots & \vdots & & \vdots \\ x(N-1) & x(N-2) & \cdots & x(N-K) \end{bmatrix}, \tag{157}$$

$\mathbf{h}_{\mathrm{i}} = [h_{\mathrm{i}}(0),\ \cdots,\ h_{\mathrm{i}}(K-1)]^T$, $\mathbf{h}_{\mathrm{q}} = [h_{\mathrm{q}}(0),\ \cdots,\ h_{\mathrm{q}}(K-1)]^T$, $\mathbf{1}_P$ is a column vector of length $P = (N-K+1)$ filled with all ones, and $\mathbf{w} = [w(K-1),\ \ldots,\ w(N-1)]^T$. Note that not all N samples of $y(n)$ are used in the formulation in order to avoid the boundary effect. Here, the system output $y(n)$ has been nominally matched with the input $x(n)$, i.e., the relative delay, amplitude, and phase difference between $y(n)$ and $x(n)$ have been removed.

Petitioners Exhibit 1005
Page 90 of 117

To find the least-squares estimates of the channel coefficients, we define a cost function as follows:

$$J = ||\mathbf{y} - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}} - d\mathbf{1}_P||^2, \tag{158}$$

where $||\cdot||^2$ denotes the l_2 norm of a vector. The optimal $\mathbf{h}_{\mathrm{i}}$, $\mathbf{h}_{\mathrm{q}}$, and d that minimize the cost function can be found by setting the partial derivatives of J with respect to $\mathbf{h}_{\mathrm{i}}^*$, $\mathbf{h}_{\mathrm{q}}^*$, and d^* to zero (pretending that $\mathbf{h}_{\mathrm{i}}$, $\mathbf{h}_{\mathrm{q}}$ and d are constants [5]); i.e.,

$$\frac{\partial J}{\partial \mathbf{h}_{\mathrm{i}}^*} = \mathbf{X}_{\mathrm{i}}^H(\mathbf{y} - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}} - d\mathbf{1}_P) = \mathbf{0} \tag{159}$$

$$\frac{\partial J}{\partial \mathbf{h}_{\mathrm{q}}^*} = \mathbf{X}_{\mathrm{q}}^H(\mathbf{y} - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}} - d\mathbf{1}_P) = \mathbf{0} \tag{160}$$

$$\frac{\partial J}{\partial d^*} = \mathbf{1}_P^T(\mathbf{y} - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}} - d\mathbf{1}_P) = 0. \tag{161}$$

Since $\mathbf{X}_{\mathrm{i}}$ and $\mathbf{X}_{\mathrm{q}}$ in (159) and (160) are real matrices, the Hermitian transpose is replaced by a simple transpose. Rearranging (159) - (161) and combining, we have

$$\begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{q}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{1}_P \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{q}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{1}_P \\ \mathbf{1}_P^T\mathbf{X}_{\mathrm{i}} & \mathbf{1}_P^T\mathbf{X}_{\mathrm{q}} & \mathbf{1}_P^T\mathbf{1}_P \end{bmatrix} \begin{bmatrix} \mathbf{h}_{\mathrm{i}} \\ \mathbf{h}_{\mathrm{q}} \\ d \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{y} \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{y} \\ \mathbf{1}_P^T\mathbf{y} \end{bmatrix}. \tag{162}$$

Therefore, the least-squares estimates of the $\mathbf{h}_{\mathrm{i}}$, $\mathbf{h}_{\mathrm{q}}$, and d are

$$\begin{bmatrix} \hat{\mathbf{h}}_{\mathrm{i}} \\ \hat{\mathbf{h}}_{\mathrm{q}} \\ \hat{d} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{q}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{1}_P \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{q}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{1}_P \\ \mathbf{1}_P^T\mathbf{X}_{\mathrm{i}} & \mathbf{1}_P^T\mathbf{X}_{\mathrm{q}} & \mathbf{1}_P^T\mathbf{1}_P \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{y} \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{y} \\ \mathbf{1}_P^T\mathbf{y} \end{bmatrix}. \tag{163}$$

Note that a coarse estimate of the dc offset can be obtained as the difference between the mean of the system output and that of the system input; i.e.,

$$\hat{d} = \frac{1}{N}\sum_{n=0}^{N-1}[y(n) - x(n)]. \tag{164}$$

From the complex I/Q model shown in Fig. 43(b), we know that d is actually given by

$$d = d_y - \left[\mathrm{Re}\{d_x\}\sum_{k=0}^{K-1}\mathbf{h}_{\mathrm{i}}(k) + \mathrm{Im}\{d_x\}\sum_{k=0}^{K-1}\mathbf{h}_{\mathrm{q}}(k)\right], \tag{165}$$

Petitioners Exhibit 1005
Page 91 of 117

where d_x and d_y are, respectively, the mean values of the input $x(n)$ and the output $y(n)$. Therefore, in order for (164) to be accurate, either d_x needs to be very small or $\sum_{k=0}^{K-1} \mathbf{h}_{\mathrm{i}}(k) \approx 1$ and $\sum_{k=0}^{K-1} \mathbf{h}_{\mathrm{q}}(k) \approx j$. If either of these conditions hold, then we may use (164) to estimate the dc offset and adjust the cost function (158) to

$$J = (\mathbf{y} - \hat{d}\mathbf{1}_P - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}})^H(\mathbf{y} - \hat{d}\mathbf{1}_P - \mathbf{X}_{\mathrm{i}}\mathbf{h}_{\mathrm{i}} - \mathbf{X}_{\mathrm{q}}\mathbf{h}_{\mathrm{q}}), \tag{166}$$

Similar to the previous derivation of the least-squares solutions, the least-squares estimates of the $\mathbf{h}_{\mathrm{i}}$, $\mathbf{h}_{\mathrm{q}}$ are then given by

$$\begin{bmatrix} \hat{\mathbf{h}}_{\mathrm{i}} \\ \hat{\mathbf{h}}_{\mathrm{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{q}} \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{q}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T(\mathbf{y} - \hat{d}\mathbf{1}_P) \\ \mathbf{X}_{\mathrm{q}}^T(\mathbf{y} - \hat{d}\mathbf{1}_P) \end{bmatrix}. \tag{167}$$

Least-Squares Method with Diagonal Loading

Since the input signal $x(n)$ is usually a bandpass signal, the channel estimates are only accurate within the signal band. The out-of-band responses are somewhat arbitrary, depending on the noise floor in $x(n)$ and $y(n)$. This is not desired in predistortion applications since the baseband signal after predistortion has low level out-of-band spectral regrowth, which needs to be accurately preserved in order to compensate for the power amplifier nonlinearity. To overcome this problem, we can add a low-level white noise to both $x(n)$ and $y(n)$ and then apply (163). The white noise creates a flat out-of-band frequency response, but its level is kept low enough not to affect the in-band channel. By using a low-level white noise that is uncorrelated with $x(n)$ and $y(n)$ and zero mean, it can be shown that (163) becomes

$$\begin{bmatrix} \hat{\mathbf{h}}_{\mathrm{i}} \\ \hat{\mathbf{h}}_{\mathrm{q}} \\ \hat{d} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{i}} + \sigma^2\mathbf{I} & \mathbf{X}_{\mathrm{i}}^T\mathbf{X}_{\mathrm{q}} & \mathbf{X}_{\mathrm{i}}^T\mathbf{1}_P \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{i}} & \mathbf{X}_{\mathrm{q}}^T\mathbf{X}_{\mathrm{q}} + \sigma^2\mathbf{I} & \mathbf{X}_{\mathrm{q}}^T\mathbf{1}_P \\ \mathbf{1}_P^T\mathbf{X}_{\mathrm{i}} & \mathbf{1}_P^T\mathbf{X}_{\mathrm{q}} & \mathbf{1}_P^T\mathbf{1}_P \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_{\mathrm{i}}^T\mathbf{y} + \sigma^2\mathbf{e} \\ \mathbf{X}_{\mathrm{q}}^T\mathbf{y} + j\,\sigma^2\mathbf{e} \\ \mathbf{1}_P^T\mathbf{y} \end{bmatrix}, \tag{168}$$

where σ^2 is the variance of the artificial white noise, $\mathbf{I}$ is a $K \times K$ identity matrix, and $\mathbf{e} = [\mathbf{0}_L^T \;\; 1 \;\; \mathbf{0}_M^T]^T$, where $\mathbf{0}_L$ and $\mathbf{0}_M$ are, respectively, length L and $M = K - L - 1$ column vectors filled with all zeros. In the case where the dc offset is estimated using (164), the

Petitioners Exhibit 1005
Page 92 of 117

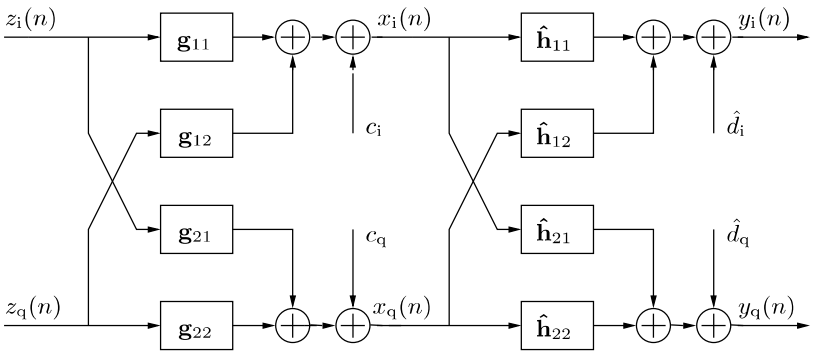


Figure 44: Cascade of the I/Q compensator and the channel.

least-squares estimates of the $\mathbf{h}_{\rm i}$, $\mathbf{h}_{\rm q}$ with diagonal loading reduce to

$$\begin{bmatrix} \hat{\mathbf{h}}_{\rm i} \\ \hat{\mathbf{h}}_{\rm q} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{\rm i}^T\mathbf{X}_{\rm i} + \sigma^2\mathbf{I} & \mathbf{X}_{\rm i}^T\mathbf{X}_{\rm q} \\ \mathbf{X}_{\rm q}^T\mathbf{X}_{\rm i} & \mathbf{X}_{\rm q}^T\mathbf{X}_{\rm q} + \sigma^2\mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_{\rm i}^T(\mathbf{y} - \hat{d}\mathbf{1}_P) + \sigma^2\mathbf{e} \\ \mathbf{X}_{\rm q}^T(\mathbf{y} - \hat{d}\mathbf{1}_P) + \sigma^2\mathbf{e} \end{bmatrix}. \quad (169)$$

The diagonal loading proposed here has the additional advantage of regularizing the solution, i.e., reducing the condition number of the correlation matrix, so that more accurate solutions can be achieved.

5.2.3 Compensator Construction

In this section, we design I/Q compensators to mitigate the frequency-dependent I/Q imbalance and dc offset. It turns out that these imperfections can be fully compensated by an I/Q compensator that has a structure similar to the channel models described in Section 5.2.1. There are two approaches to construct such a compensator. In the two-step approach, the channel is first estimated, and the compensator is constructed based on the channel estimates. In the one-step approach, the compensator is constructed directly using the system input $x(n)$ and system output $y(n)$.

Two-Step Approach

To ease the derivation, we choose the real I/Q channel model to represent both the I/Q compensator and the direct upconverter channel. The cascade of the I/Q compensator and the channel is shown in Fig. 44, where the I/Q compensator is represented by four real filters, $\mathbf{g}_{11}$, $\mathbf{g}_{12}$, $\mathbf{g}_{21}$, $\mathbf{g}_{22}$, and a dc component c.

dc Offset Compensation

Petitioners Exhibit 1005
Page 93 of 117

The dc offset $\hat{d}$ of the channel can be compensated by the dc component c in the I/Q compensator. To achieve this, c after passing through the four $\hat{\mathbf{h}}$ filters should be equal to $-\hat{d}$, i.e.,

$$\begin{aligned} c_{\rm i}\ s_{11} + c_{\rm q}\ s_{12} &= -\hat{d}_{\rm i} \\ c_{\rm i}\ s_{21} + c_{\rm q}\ s_{22} &= -\hat{d}_{\rm q}, \end{aligned} \tag{170}$$

where

$$\begin{aligned} s_{11} &= \sum_{k=0}^{K-1} \hat{h}_{11}(k),\ s_{12} = \sum_{k=0}^{K-1} \hat{h}_{12}(k) \\ s_{21} &= \sum_{k=0}^{K-1} \hat{h}_{21}(k),\ s_{22} = \sum_{k=0}^{K-1} \hat{h}_{22}(k). \end{aligned} \tag{171}$$

It is clear from (170) that the solution for $c_{\rm i}$ and $c_{\rm q}$ is

$$\begin{bmatrix} c_{\rm i} \\ c_{\rm q} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}^{-1} \begin{bmatrix} -\hat{d}_{\rm i} \\ -\hat{d}_{\rm q} \end{bmatrix}. \tag{172}$$

Note that when the inverse matrix in (172) is close to the identity matrix, we have

$$\begin{bmatrix} c_{\rm i} \\ c_{\rm q} \end{bmatrix} \approx \begin{bmatrix} -\hat{d}_{\rm i} \\ -\hat{d}_{\rm q} \end{bmatrix}. \tag{173}$$

Gain/Phase Imbalance Compensation

After passing the dc component c through the $\hat{\mathbf{h}}$ filters, the four $\mathbf{g}$ filters in the compensator and the four $\hat{\mathbf{h}}$ filters in the channel are connected directly. To achieve ideal compensation, the cascade of filters should satisfy the following relationship in the frequency domain; i.e., for $\omega \in [0,\ 2\pi)$,

$$\begin{bmatrix} \hat{H}_{11}(e^{j\omega}) & \hat{H}_{12}(e^{j\omega}) \\ \hat{H}_{21}(e^{j\omega}) & \hat{H}_{22}(e^{j\omega}) \end{bmatrix} \begin{bmatrix} G_{11}(e^{j\omega}) & G_{12}(e^{j\omega}) \\ G_{21}(e^{j\omega}) & G_{22}(e^{j\omega}) \end{bmatrix} = e^{-j\omega n_0}\, \mathbf{I}_2, \tag{174}$$

where $\mathbf{I}_2$ is a 2-by-2 identity matrix, and n_0 is the desired delay of the overall cascade.

Petitioners Exhibit 1005
Page 94 of 117

Therefore, we have

$$\begin{bmatrix} G_{11}(e^{j\omega}) & G_{12}(e^{j\omega}) \\ G_{21}(e^{j\omega}) & G_{22}(e^{j\omega}) \end{bmatrix} = \begin{bmatrix} \hat{H}_{11}(e^{j\omega}) & \hat{H}_{12}(e^{j\omega}) \\ \hat{H}_{21}(e^{j\omega}) & \hat{H}_{22}(e^{j\omega}) \end{bmatrix}^{-1}$$
$$= A(e^{j\omega}) \begin{bmatrix} \hat{H}_{22}(e^{j\omega}) & -\hat{H}_{12}(e^{j\omega}) \\ -\hat{H}_{21}(e^{j\omega}) & \hat{H}_{11}(e^{j\omega}) \end{bmatrix}, \quad (175)$$

where

$$A(e^{j\omega}) = [\hat{H}_{11}(e^{j\omega})\hat{H}_{22}(e^{j\omega}) - \hat{H}_{12}(e^{j\omega})\hat{H}_{21}(e^{j\omega})]^{-1} \quad (176)$$

is the determinate. In the time domain, (175) becomes,

$$g_{11}(k) = \hat{h}_{22}(k) * a(k), \; g_{12}(k) = -\hat{h}_{12}(k) * a(k)$$
$$g_{21}(k) = -\hat{h}_{21}(k) * a(k), \; g_{22}(k) = \hat{h}_{11}(k) * a(k), \quad (177)$$

where $*$ denotes convolution and $a(k)$ is the inverse Fourier transform of $A(e^{j\omega})$. In other words, $a(k)$ is the inverse of the filter

$$h_c(k) = \hat{h}_{11}(k) * \hat{h}_{22}(k) - \hat{h}_{12}(k) * \hat{h}_{21}(k), \quad (178)$$

which can be constructed either in the time domain or in the frequency domain. Here, we present a frequency-domain least-squares approach to design $a(k)$.

To find the optimal $a(k)$ coefficients, we adopt a frequency-domain least-squares approach. Suppose that filter $a(k)$ has K_a taps, the convolved response of $a(k)$ and $h_c(k)$ can be written as

$$c(k) = \sum_{l=0}^{K_a-1} h_c(k-l)a(l), \quad k = 0, 1, \cdots, K_a + 2K - 3, \quad (179)$$

which has length $K_c = K_a + 2K - 2$. (Note that filter $h_c(k)$ is of length $2K - 1$.) We then define a cost function as the integrated difference between the frequency response of filter $c(k)$ and the desired frequency response, $e^{j\omega n_0}$, within a given passband $[-\omega_p, \; \omega_p]$; i.e.,

$$J_c = \int_{-\omega_p}^{\omega_p} \left[\sum_{k_1=0}^{K_c-1} c(k_1)e^{-j\omega k_1} - e^{-j\omega n_0} \right] \left[\sum_{k_2=0}^{K_c-1} c(k_2)e^{-j\omega k_2} - e^{-j\omega n_0} \right]^* d\omega. \quad (180)$$

Petitioners Exhibit 1005
Page 95 of 117

Substituting (179) into (180), we have

$$J_c = \int_{-\omega_p}^{\omega_p} \left[\sum_{k_1=0}^{K_c-1} \sum_{l_1=0}^{K_a-1} h_c(k_1-l_1)a(l_1)e^{-j\omega k_1} - e^{-j\omega n_0} \right] \times \left[\sum_{k_2=0}^{K_c-1} \sum_{l_2=0}^{K_a-1} h_c(k_2-l_2)a(l_2)e^{-j\omega k_2} - e^{-j\omega n_0} \right]^* d\omega. \quad (181)$$

To find the optimal $a(k)$ that minimizes the cost function, we take the partial derivative of J_c in (181) with respect to $a^*(l)$ (pretending $a(l)$ is constant [5]) and set it to zero; i.e.,

$$\frac{\partial J_c}{\partial a^*(l)} = \int_{-\omega_p}^{\omega_p} \left[\sum_{k_1=0}^{K_c-1} \sum_{l_1=0}^{K_a-1} h_c(k_1-l_1)a(l_1)e^{-j\omega k_1} - e^{-j\omega n_0} \right] \times \sum_{k_2=0}^{K_c-1} h_c^*(k_2-l)e^{j\omega k_2} d\omega = 0; \quad l = 0, \cdots, K_a-1. \quad (182)$$

Rearrange (182) to write

$$\frac{\partial J_c}{\partial a^*(l)} = \sum_{l_1=0}^{K_a-1} a(l_1) \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} h_c(k_1-l_1)h_c^*(k_2-l) \int_{-\omega_p}^{\omega_p} e^{-j\omega(k_1-k_2)} d\omega - \sum_{k_2=0}^{K_c-1} h_c^*(k_2-l) \int_{-\omega_p}^{\omega_p} e^{-j\omega(n_0-k_2)} d\omega = 0. \quad (183)$$

In predistortion applications, since least-squares with diagonal loading is used for estimating the four h filters, filters $h_{11}(k)$ and $h_{22}(k)$ have flat responses outside of the signal band. Moreover, the responses of the cross coupling filters $h_{12}(k)$ and $h_{21}(k)$ are usually much smaller than those of the filters $h_{11}(k)$ and $h_{22}(k)$. Therefore, the out-of-band response of filter $h_c(k)$ may be considered as flat with unit gain. Thus, we can use $\omega_p = \pi$ in (183). Substituting $\omega_p = \pi$ into (183) and carrying out the integration, we obtain

$$\frac{\partial J_c}{\partial a^*(l)} = \sum_{l_1=0}^{K_a-1} a(l_1) \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} h_c(k_1-l_1)\ h_c^*(k_2-l)\ 2\pi\ \mathrm{sinc}(k_1-k_2) - \sum_{k_2=0}^{K_c-1} h_c^*(k_2-l)\ 2\pi\ \mathrm{sinc}(n_0-k_2) = 0, \quad (184)$$

where

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \quad (185)$$

Petitioners Exhibit 1005
Page 96 of 117

For l from 0 to $K_a - 1$, we have a set of equations from (184); i.e.,

$$\sum_{l_1=0}^{K_a-1} a(l_1) \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} 2\pi\ h_c(k_1 - l_1)\ h_c^*(k_2 - l_2)\ \mathrm{sinc}(k_1 - k_2)$$
$$= \sum_{k_2=0}^{K_c-1} 2\pi\ h_c^*(k_2 - l)\ \mathrm{sinc}(n_0 - k_2). \tag{186}$$

Removing 2π from both sides of (186) and rewriting it in matrix form, we have

$$\mathbf{R}_h \mathbf{a} \ = \ \mathbf{h}_a, \tag{187}$$

where $\mathbf{a} = [a_0\ \ldots\ a_{K_a-1}]^T$, and the elements of matrix $\mathbf{R}_h$ and column vector $\mathbf{h}_a$ are defined as

$$\mathbf{R}_h(l_1, l) \ = \ \sum_{k_1=0}^{K_c-1} \sum_{k_2=0}^{K_c-1} h_c(k_1 - l_1)\ h_c^*(k_2 - l)\ \mathrm{sinc}(k_1 - k_2) \tag{188}$$

$$\mathbf{h}_a(l) \ = \ \sum_{k_2=0}^{K_c-1} h_c^*(k_2 - l)\ \mathrm{sinc}(n_0 - k_2). \tag{189}$$

Note that $\mathbf{R}_h$ is a Toeplitz matrix, so we only need to calculate the first row and column of the matrix. The least-squares estimate of $\mathbf{a}$ from (187) is

$$\hat{\mathbf{a}} \ = \ \mathbf{R}_h^{-1} \mathbf{h}_a. \tag{190}$$

One-Step Approach

The I/Q compensator is the pre-inverse of the underlying channel. However, since the channel is a linear system, its pre-inverse is the same as its post-inverse, whose input and desired output are, respectively, $y(n)$ and $x(n)$. Therefore, the post-inverse, i.e., the compensator, can be obtained using (168) by treating $y(n)$ as the input and $x(n)$ as the desired output, i.e.,

$$\begin{bmatrix} \mathbf{g}_\mathrm{i} \\ \mathbf{g}_\mathrm{q} \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_\mathrm{i}^T \mathbf{Y}_\mathrm{i} + \sigma^2 \mathbf{I} & \mathbf{Y}_\mathrm{i}^T \mathbf{Y}_\mathrm{q} & \mathbf{Y}_\mathrm{i}^T \mathbf{1}_P \\ \mathbf{Y}_\mathrm{q}^T \mathbf{Y}_\mathrm{i} & \mathbf{Y}_\mathrm{q}^T \mathbf{Y}_\mathrm{q} + \sigma^2 \mathbf{I} & \mathbf{Y}_\mathrm{q}^T \mathbf{1}_P \\ \mathbf{1}_P^T \mathbf{Y}_\mathrm{i} & \mathbf{1}_P^T \mathbf{Y}_\mathrm{q} & \mathbf{1}_P^T \mathbf{1}_P \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Y}_\mathrm{i}^T \mathbf{x} + \sigma^2 \mathbf{e} \\ \mathbf{Y}_\mathrm{q}^T \mathbf{x} + j\,\sigma^2 \mathbf{e} \\ \mathbf{1}_P^T \mathbf{x} \end{bmatrix}. \tag{191}$$

The diagonal loading here is also essential to guarantee a flat frequency response outside of the signal band. The main advantage of the one-step approach is that it generates

Petitioners Exhibit 1005
Page 97 of 117

Table 4: Ideal Model Coefficients

k	$h_{\mathrm{i}}(k)$	$h_{\mathrm{q}}(k)$
1	$-0.0046 + j\,0.0016$	$-0.0017 - j\,0.0040$
2	$0.0064 - j\,0.0013$	$0.0013 + j\,0.0066$
3	$0.0054 - j\,0.0080$	$0.0088 + j\,0.0053$
4	$-0.0013 - j\,0.0063$	$0.0078 - j\,0.0003$
5	$0.9982 - j\,0.0060$	$-0.0011 + j\,1.0009$
6	$0.0052 + j\,0.0003$	$-0.0024 + j\,0.0077$
7	$0.0091 - j\,0.0052$	$0.0023 + j\,0.0092$
8	$0.0055 - j\,0.0054$	$0.0046 + j\,0.0042$
9	$0.0044 + j\,0.0038$	$-0.0027 + j\,0.0057$

the $\mathbf{g}$ filters directly. The previous approach, in contrast, constructs the $\mathbf{g}$ filters through convolutions of the estimated $\mathbf{h}$ filters and a separately designed common inverse filter $a(k)$. Therefore, the one-step approach may help to reduce the total number of taps required for the compensation filters.

5.2.4 Simulations

In this section, we evaluate the performance of the I/Q compensators designed in Section 5.2.3 through computer simulations. The input signal, $x(n)$, in the simulations is a 15 MHz 11-carrier CDMA signal shifted to the lower sideband. Given this input, we first extracted a block of 40000 $y(n)$ samples through the dashed-loop in Fig. 41 using our experimental testbed. After matching $y(n)$ with $x(n)$ (delay, amplitude, phase rotation), the coefficients of an ideal channel model were obtained using (168), which are given in Table 4 with $d = 0.0114 - j\,0.0023$.

Simulation 5.1 We simulated the dashed-loop in Fig. 41, where the DAC and direct upconverter were replaced by the ideal channel model; the downconverter, ADC, and digital demodulator were replaced by an additive white noise yielding an SNR of 45 dB. Given $x(n)$ and $y(n)$, I/Q compensators were constructed by the two approaches proposed in Section 5.2.3. Fig. 45 shows the outputs of the ideal channel model with and without compensators. Here, the 9/9 IQ compensator means that, in (177), the $\hat{\mathbf{h}}$ filters have 9 taps and $a(k)$ has 9 taps. We can see that both the 9/9 I/Q compensator from the two-step approach and the 9-tap compensator from the one-step approach were able to fully suppress the image.

Petitioners Exhibit 1005
Page 98 of 117

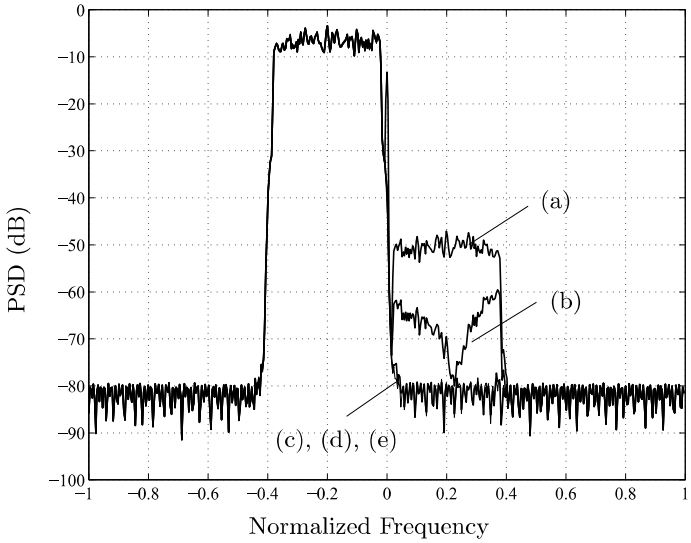


Figure 45: Comparison of the direct upconverter outputs without I/Q compensation and with different I/Q compensators. (a) Without I/Q compensation; (b) With a 1-tap I/Q compensator; (c) With a 9/9 I/Q compensator constructed using the two-step approach; (d) With a 9-tap I/Q compensator constructed using the one-step approach; (e) Original input. Here, (c), (d), and (e) coincide.

However, the 1-tap compensator could not suppress the image completely, which suggests that the gain/phase imbalance of the direct upconverter is frequency-dependent.

Simulation 5.2 We simulated the solid loop in Fig. 41 after plugging the I/Q compensators constructed in the previous simulation into the loop. The configuration of the loop is the same as in the previous simulation except that the power amplifier block was replaced by a memory polynomial power amplifier model, whose parameters are given in (Example 3.7). The predistorter also follows a memory polynomial model here, i.e.,

$$z(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} u(n-q)|u(n-q)|^{k-1}. \tag{192}$$

The simulation result is shown in Fig. 46. The predistorter used in the simulation has $Q = 3$ and $K = 7$. We can see that with either the 9/9 I/Q compensator from the two-step approach or the 9-tap I/Q compensator from the one-step approach, the predistorter is able to fully suppress the spectral regrowth. However, with the 1-tap I/Q compensator, there is a residue image after predistortion.

Petitioners Exhibit 1005
Page 99 of 117

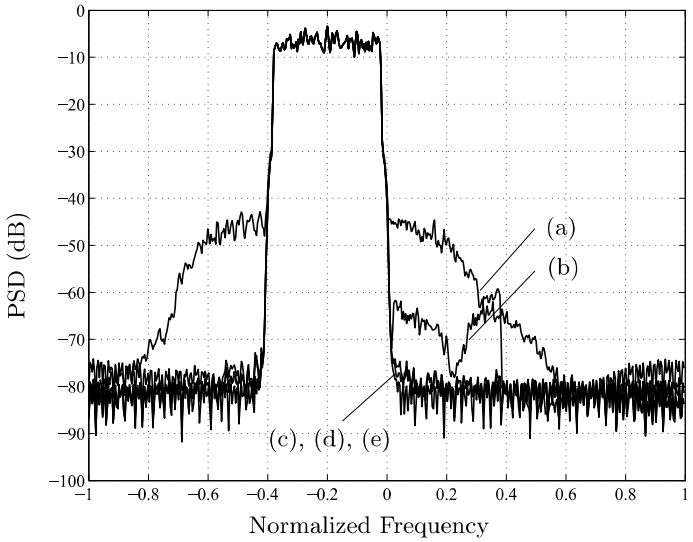


Figure 46: Comparison of the power amplifier output with different I/Q compensators and predistorters. (a) Without predistortion but with a 9-tap I/Q compensation from the one-step approach; (b) With predistortion and a 1-tap I/Q compensator; (c) With predistortion and a 9/9 I/Q compensator from the two-step approach; (d) With predistortion and a 9-tap I/Q compensator from the one-step approach; (e) Original input. Here, (c), (d), and (e) almost coincide.

Petitioners Exhibit 1005
Page 100 of 117

CHAPTER 6

REAL-TIME IMPLEMENTATIONS

In Chapter 3, we have shown that the memory polynomial predistorter is a good choice for linearizing power amplifiers with memory effects. In this chapter, we investigate real-time implementation aspects of the memory polynomial predistorter. We implement the predistorter training algorithm on a Texas Instruments TMS320C67xx processor and evaluate the performance of the trained predistorter on our wideband digital predistortion testbed [13].

6.1 Memory Polynomial Model

Here, we adopt the model of Kim and Konstantinou [30] for the predistorter,

$$z(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} a_{kq}\ x(n-q)|x(n-q)|^{k-1}, \tag{193}$$

which we call a memory polynomial. The input $x(n)$, output $z(n)$, and coefficients a_{kq} of the model are all complex valued in general. Note that if the maximum delay $Q = 0$, (193) reduces to

$$z(n) = \sum_{k=1}^{K} a_{k0}\ x(n)|x(n)|^{k-1}, \tag{194}$$

which is a conventional memoryless polynomial. A direct implementation of the predistorter model in (193) requires multiplications on the order of K^2Q. However, an efficient implementation is possible by observing that (193) is equivalent to

$$z(n) = \sum_{q=0}^{Q} x(n-q) \sum_{k=1}^{K} a_{kq}\,|x(n-q)|^{k-1} \tag{195}$$

$$= \sum_{q=0}^{Q} x(n-q)\ \mathrm{LUT}_q(|x(n-q)|), \tag{196}$$

[23], where the nonlinear polynomial for each delay q is implemented by a lookup table (LUT) indexed by $|x(n-q)|$. Therefore, only Q complex multiplications per sample are needed.

Petitioners Exhibit 1005
Page 101 of 117

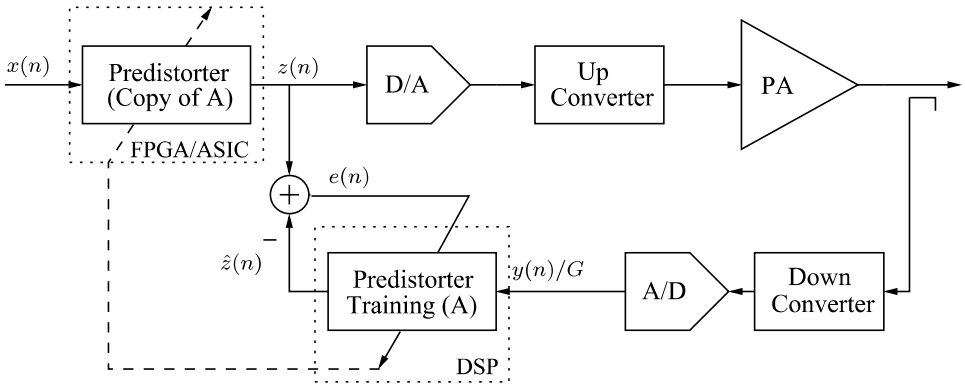


Figure 47: The indirect learning architecture for the predistorter.

6.2 Indirect Learning Architecture

The indirect learning architecture has been introduced in Chapter 2. It is shown here again in Fig. 47. In the architecture, the predistorter performs the same computation, such as (196), for every input sample at high-speed. This kind of task is well suited for field programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs). The predistorter training block, however, involves relatively complex computations, which require a powerful digital signal processor (DSP), such as the Texas Instruments (TI) TMS320C67xx. The time required to train the predistorter determines the ability of the predistorter to response to changes in power amplifier characteristics. Although these changes usually happen slowly, which may be due to temperature drift, aging, etc., a powerful DSP increases the flexibility of the overall system.

6.3 Predistorter Construction

In the context of predistorter training (see the training block of Fig. 47), (193) becomes

$$z(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} a_{kq}\ y(n-q)|y(n-q)|^{k-1}. \tag{197}$$

Since $z(n)$ is linear in the parameters a_{kq}, the latter can be estimated by a simple least-squares method. By defining a new sequence

$$r_{kq}(n) = y(n-q)|y(n-q)|^{k-1}, \tag{198}$$

Petitioners Exhibit 1005
Page 102 of 117

we can rewrite (197) in matrix form as

$$\mathbf{z} = \mathbf{R}\ \mathbf{a}, \tag{199}$$

where

$$\begin{aligned}
\mathbf{z} &= [z(0), \cdots, z(N-1)]^T, \\
\mathbf{R} &= [\mathbf{R}_0, \cdots, \mathbf{R}_Q], \\
\mathbf{R}_q &= [\mathbf{r}_{1q}, \cdots, \mathbf{r}_{Kq}], \\
\mathbf{r}_{kq} &= [r_{kq}(0), \cdots, r_{kq}(N-1)]^T, \\
\mathbf{a} &= [a_{10}, \cdots, a_{K0}, \cdots, a_{1Q}, \cdots, a_{KQ}]^T.
\end{aligned}$$

The least-squares solution for (199) is

$$\hat{\mathbf{a}} = (\mathbf{R}^H\mathbf{R})^{-1}\mathbf{R}^H\mathbf{z}, \tag{200}$$

where $(\cdot)^H$ denotes complex conjugate transpose. The accuracy and stability of the solution $\hat{\mathbf{a}}$ are directly related to the numerical condition of the matrix $\mathbf{R}^H\mathbf{R}$. A good indication of such condition is the condition number of the matrix [34, p. 258]; i.e.,

$$\kappa\left(\mathbf{R}^H\mathbf{R}\right) = \frac{\lambda_{\max}}{\lambda_{\min}}, \tag{201}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are, respectively, the largest and smallest eigenvalues of $\mathbf{R}^H\mathbf{R}$. The matrix $\mathbf{R}^H\mathbf{R}$ generally has a high condition number, which also means that there is large correlation between the columns of this matrix. There are two sources for this large correlation:

1. The nonlinear polynomials, such as y, $y|y|$, $y|y|^2$, etc., are highly correlated.

2. The data sample $y(n)$ with different time indices are correlated.

The correlation due to the first source can be greatly reduced by using the orthogonal polynomial proposed in [38]. In this formulation, (197) becomes

$$z(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} b_{kq}\ \psi_k(y(n-q)), \tag{202}$$

Petitioners Exhibit 1005
Page 103 of 117

where

$$\psi_k(y) = \sum_{l=1}^{k} U_{lk}\ y\,|y|^{l-1} \tag{203}$$

with

$$U_{lk} = \frac{(-1)^{l+k}(k+l)!}{(l-1)!(l+1)!(k-l)!}. \tag{204}$$

For a K-th order polynomial, U_{lk} forms an upper triangular matrix $\mathbf{U}$, which leads to the matrix form of (202), i.e.,

$$\mathbf{z} = \mathbf{F}\,\mathbf{b}, \tag{205}$$

where $\mathbf{F} = [\mathbf{R}_0\mathbf{U}, \cdots, \mathbf{R}_Q\mathbf{U}]$. The least-squares solution for $\mathbf{b}$ is then given by

$$\hat{\mathbf{b}} = (\mathbf{F}^H\mathbf{F})^{-1}\,\mathbf{F}^H\mathbf{z}. \tag{206}$$

The orthogonal polynomial in [38] is derived for complex random signals with amplitude uniformly distributed between 0 and 1 (but is robust for non-uniformly distributed amplitudes as well). Therefore, to fully exploit the advantage of the orthogonal polynomial, the amplitude of $y(n)$ should be scaled to the $[0, 1]$ interval first before applying the $\psi_k()$ operation.

The correlation from the second source can be alleviated by using a special training signal whose samples at different time indices are independent. However, in many cases, dedicated training is not feasible. In this case, the accuracy of the solution $\hat{\mathbf{a}}$ can be improved by using higher precision floating point numbers, such as using 64-bit double precision instead of 32-bit single precision.

Fig. 48 shows an example of the condition number of the correlation matrix with different Q values and different input signals. We see that if the input signal is random with uniformly distributed amplitude in [0,1], the condition number is not affected by the number of delay terms, and the orthogonal polynomial offers great advantages. For a three-carrier WCDMA signal, the benefit of using orthogonal polynomial decreases with the increase of the number of delay terms. However, a significant reduction of the condition number is still observed.

Petitioners Exhibit 1005
Page 104 of 117

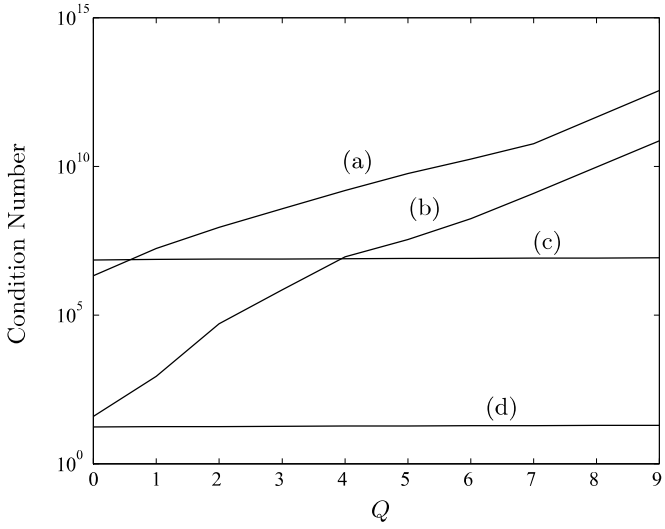


Figure 48: Condition number of the correlation matrix with different Q values and different input signals: (a) three-carrier WCDMA with $K = 5$ conventional polynomials; (b) three-carrier WCDMA with $K = 5$ orthogonal polynomials; (c) a complex random signal (amplitude uniformly distributed in [0,1]) with $K = 5$ conventional polynomials; (d) a complex random signal (amplitude uniformly distributed in [0,1]) with $K = 5$ orthogonal polynomials.

6.4 DSP Implementation

Because of the benefits of orthogonal polynomials, we focused on DSP implementation using orthogonal polynomials. To evaluate the real-time performance of the predistorter training algorithm, we selected TI TMS320C6711, which is a low-cost yet powerful floating point processor. We implemented the algorithm in C and generated the DSP-executable code with level-3 optimization provided by the TI C-compiler.

6.4.1 Implementation Details

Figure 49 shows the flowchart of the algorithm. The algorithm starts with acquiring the baseband input and output data samples of the power amplifier. The matrix $\mathbf{R}_0$ is then formed and multiplied with $\mathbf{U}$ to form the first K columns of $\mathbf{F}$. The other columns of $\mathbf{F}$ are just shifted versions of the first K columns. Next, the upper triangular portion of the coefficients of the correlation matrix $\mathbf{F}^H\mathbf{F}$ are calculated. These coefficients are sufficient to define the whole matrix since the matrix is Hermitian. To obtain the solution for (206), we adopt the Cholesky decomposition approach, which is very efficient in solving linear

Petitioners Exhibit 1005
Page 105 of 117

equations involving a Hermitian matrix [48]. Cholesky decomposition of $\mathbf{F}^H\mathbf{F}$ yields a lower-triangular matrix $\mathbf{L}$ such that

$$\mathbf{L}\,\mathbf{L}^H = \mathbf{F}^H\mathbf{F}. \tag{207}$$

Substituting (207) into (206), we see that $\hat{\mathbf{b}}$ is the solution of

$$\mathbf{L}\,\mathbf{L}^H\,\hat{\mathbf{b}} = \mathbf{F}^H\mathbf{z}, \tag{208}$$

which can be obtained easily by using forward and back substitution [48, pp. 26-30].

6.4.2 Performance Evaluation

The computation requirement of the algorithm in the previous section is determined by two factors: the calculation of the correlation matrix and Cholesky decomposition. To give a quantitative measure of the complexity, we evaluate the floating point operations (flops) required by these two steps. For example, one complex multiplication involves six flops: four real multiplications, one real addition, and one real subtraction.

It is straightforward to calculate the required number of flops once the C implementation is available. In our program, for a block of N data samples, the number of flops for obtaining $\mathbf{F}^H\mathbf{F}$ is approximately $4.5\,K^2(Q+1)^2N$. The number of flops for obtaining the Cholesky decomposition is approximately $1.5\,K^3(Q+1)^3$. Therefore, when N is large, the computations are dominated by obtaining the correlation matrix.

Table 5 shows the CPU cycles and execution time required by the C6711 starter kit to train the predistorter. We see that longer data lengths, more delay taps, and higher precision implementation all increase the computation time, although they all help to improve the predistortion performance. In practice, tradeoffs need to be made. The execution time shown here can be further reduced by (i) using new generations of TMS320C67xx processor, which are able to operate at a higher clock rate, (ii) coding the most time consuming block; i.e., the calculation of the correlation matrix, in assembly.

6.5 Testbed Measurements

In this section, we present experimental results from our digital predistortion testbed, whose configuration is shown in Fig. 50.

Petitioners Exhibit 1005
Page 106 of 117

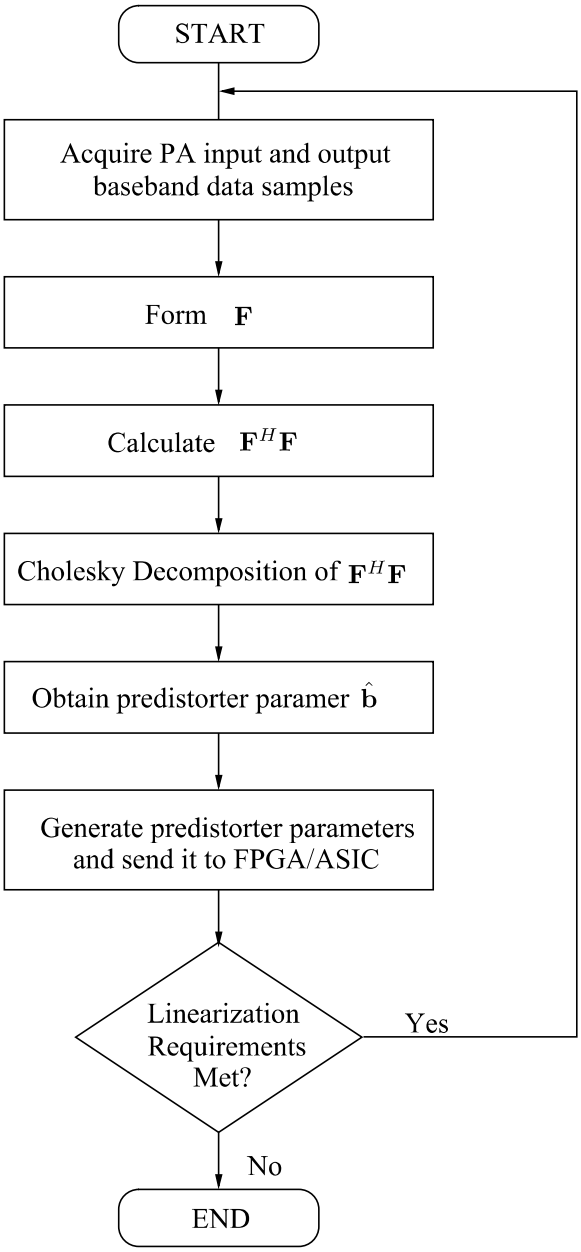


Figure 49: Flow chart of the algorithm.

Petitioners Exhibit 1005
Page 107 of 117

Table 5: Real-time performance of the predistorter training algorithm.

(a) $K = 5,\ Q = 0$

	N=5000		N= 20000	
	CPU Cycles	Execution Time (s)	CPU Cycles	Execution Time (s)
32-bit	35094948	0.2351	140036672	0.9382
64-bit	50879944	0.3409	203001588	1.3601

(b) $K = 5,\ Q = 4$

	N=5000		N= 20000	
	CPU Cycles	Execution Time (s)	CPU Cycles	Execution Time (s)
32-bit	386017219	2.5863	1542049416	10.3317
64-bit	487638321	3.2672	1969757008	13.1974

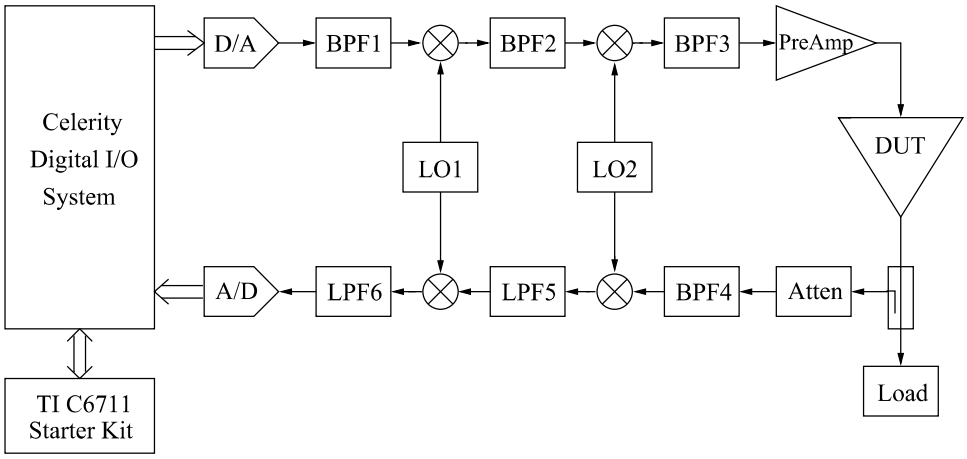


Figure 50: Block diagram of the testbed.

Petitioners Exhibit 1005
Page 108 of 117

In the testbed, the digital I/O instrument is a Celerity system with 150 MSPS 16-bit digital input and output capability. It sends out 14-bit digital IF data streams to the DAC board continuously and acquires 12-bit digital IF data samples from the ADC when needed. The DAC and ADC used here are, respectively, AD9772 and AD9430 from Analog Devices. The predistorter training algorithm is implemented on a TI C6711 starter kit, which connects with the Celerity system through a parallel port. A two-stage upconversion and downconversion chain were carefully assembled to avoid introducing extra distortions.

In the experiment, the device under test (DUT) is a Siemens CGY0819 handset power amplifier operating at the cellular band (824-849 MHz). The input to the power amplifier is a 3.6 MHz bandwidth signal centered at 836 MHz. We tested both memoryless and memory polynomial predistorters on the power amplifier. To evaluate the effects of the data length on predistortion performance, we trained each predistorter using 5,000 and 20,000 data samples. We used 64-bit implementation for both the memoryless and memory polynomial predistorters. The results are shown in Fig. 51 and Fig. 52. We see that the memory polynomial predistorter achieved more spectral regrowth suppression than the memoryless predistorter. This may be due to the memory effects in the power amplifier or the frequency response caused by the analog filters in the upconverter. Moreover, training with a longer data length helped to improve the performance of the memory polynomial predistorter. Since the memory polynomial involves more parameters ($K(Q+1)$) than the memoryless case (K), it is expected that the memory polynomial model needs more data points to estimate.

Petitioners Exhibit 1005
Page 109 of 117

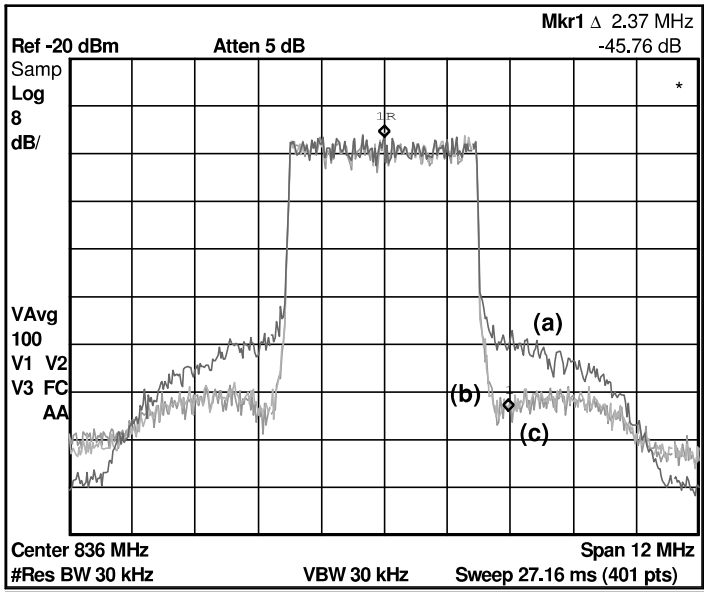


Figure 51: Measured power amplifier output PSD: (a) without predistortion; (b) with $K = 5$ memoryless predistorter trained by 5,000 data samples; (c) with $K = 5$ memoryless predistorter trained by 20,000 data samples. (b) and (c) coincide.

Petitioners Exhibit 1005
Page 110 of 117

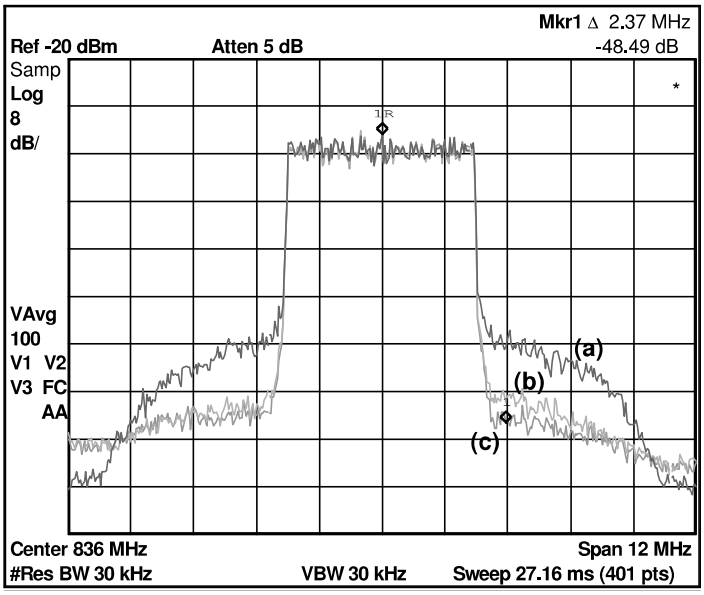


Figure 52: Measured power amplifier output PSD: (a) without predistortion; (b) with $K = 5$, $Q = 4$ memory polynomial predistorter trained by 5,000 data samples; (c) with $K = 5$, $Q = 4$ memory polynomial predistorter trained by 20,000 data samples.

Petitioners Exhibit 1005
Page 111 of 117

CHAPTER 7

CONCLUSIONS

This dissertation considered the design of digital predistortion systems to linearize power amplifiers with memory effects. By adding a digital predistorter in the baseband, the power amplifier is allowed to operate into its nonlinear region, thereby significantly increasing its efficiency. The efficiency gain translates into electricity and cooling cost savings for service providers and longer battery life for mobile terminal users. The challenge here is to address the memory effects exhibited by the higher power amplifiers or the power amplifiers for wideband signals. In addition, analog components in the transmitter have imperfections that need to be compensated as well.

7.1 Contributions

Primary contributions of this dissertation are summarized here:

- Designed novel predistorters and their parameter extraction algorithms, which include the Hammerstein predistorter, the memory polynomial predistorter, and the combined predistorter.
- Explained the benefits of including even-order terms in power amplifier modeling and predistorter design.
- Designed compensation techniques for analog imperfections in the transmitter, which include the linear frequency distortion and frequency-dependent gain/phase imbalance.
- Integrated a wideband predistortion testbed.

In addition, we implemented the memory polynomial predistorter training algorithm on a Texas Instruments C6711 Starter Kit and evaluated the real-time performance of the algorithm.

Petitioners Exhibit 1005
Page 112 of 117

7.2 Suggestions for Future Research

This dissertation can be extended in a number of directions, including:

- Designing a fast adaptive memory polynomial predistorter based on the orthogonal polynomial theory.
- Performing tests on different types of power amplifiers and establishing connections between the memory behavior of the power amplifier and the kernels of the Volterra series.
- Combining predistortion with peak-to-average ratio reduction techniques to further improve the efficiency of the power amplifier.

Petitioners Exhibit 1005
Page 113 of 117

REFERENCES

[1] Bai, E. W., "An optimal two stage identification algorithm for Hammerstein-Wiener nonlinear systems," in *Proc. American Contr. Conf.*, pp. 2756–2760, June 1998.

[2] Benedetto, S. and Biglieri, E., "Nonlinear equalization of digital satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 57–62, Jan. 1983.

[3] Benedetto, S. and Biglieri, E., *Principles of Digital Transmission with Wireless Applications*. New York, NY: Kluwer Academic/Plenum, 1999.

[4] Benedetto, S., Biglieri, E., and Daffara, R., "Modeling and performance evaulation of nonlinear satellite links – a Volterra series approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, pp. 494–507, July 1979.

[5] Brandwood, D. H., "A complex gradient operator and its application in adaptive array theory," *IEE Proc. Part F and H*, vol. 130, pp. 11–16, Feb. 1983.

[6] Cavers, J. K., "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 374–382, Nov. 1990.

[7] Cavers, J. K., "The effect of quadrature modulator and demodulator errors on adaptive digital predistorters for amplifier linearization," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 456–466, May 1997.

[8] Cavers, J. K., "New methods for adaptation of quadrature modulators and demodulators in amplifier linearization circuits," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 707–716, Aug. 1997.

[9] Cavers, J. K. and Liao, M., "Adaptive compensation for imbalance and offset losses in direct conversion transceivers," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 581–588, Nov. 1993.

[10] Chang, S. and Powers, E. J., "A simplified predistorter for compensatoin of nonlinear distortion in OFDM systems," in *Proc. IEEE Global Telecommun. Conf.*, vol. 5, pp. 3080–3084, Nov. 2001.

[11] Clark, C. J., Chrisikos, G., Muha, M. S., Moulthrop, A. A., and Silva, C. P., "Time-domain envelope measurement technique with application to wideband power amplifier modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 2531–2540, Dec. 1998.

[12] Ding, L., Ma, Z., Morgan, D. R., Zierdt, M., and Zhou, G. T., "Frequency-dependent modulator imbalance in predistortion linearization systems: modeling and compensation," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Nov. 2003.

[13] Ding, L., Qian, H., Chen, N., and Zhou, G. T., "A memory polynomial predistorter implemented using TMS320C67xx," in *Texas Instruments Developer Conf.*, Feb. 2004.

Petitioners Exhibit 1005
Page 114 of 117

[14] Ding, L., Raich, R., and Zhou, G. T., "A Hammerstein predistortion linearization design based on the indirect learning architecture," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. III, pp. 2689–2692, May 2002.

[15] Ding, L. and Zhou, G. T., "Effects of even-order nonlinear terms on predistortion," in *Proc. IEEE Digital Signal Processing Workshop*, pp. 1–6, Oct. 2002.

[16] Ding, L. and Zhou, G. T., "A robust predistorter constructed using memory polynomials," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 156–162, Jan. 2004.

[17] Ding, L., Zhou, G. T., Morgan, D. R., Ma, Z., Kenney, J. S., Kim, J., and Giardina, C. R., "Memory polynomial predistorter based on the indirect learning architecture," in *Proc. IEEE Global Telecommun. Conf.*, pp. 967–971, Nov. 2002.

[18] Ding, L., Zhou, G. T., Morgan, D. R., Ma, Z., Kenney, J. S., Kim, J., and Giardina, C. R., "A robust predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, pp. 159–165, Jan. 2004.

[19] Eskinat, E., Johnson, S. H., and Luyben, W. L., "Use of Hammerstein models in identification of nonlinear systems," *AIChE J.*, vol. 37, pp. 255–267, Feb. 1991.

[20] Eun, C. and Powers, E. J., "A predistorter design for a memory-less nonlinearity preceded by a dynamic linear system," in *Proc. IEEE Global Telecommun. Conf.*, vol. 1, pp. 152–156, Nov. 1995.

[21] Eun, C. and Powers, E. J., "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Processing*, vol. 45, pp. 223–227, Jan. 1997.

[22] Faulkner, M. and Johansson, M., "Adaptive linearization using predistortion - experimental results," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 323–332, May 1994.

[23] Giardina, C. R., Kim, J., and Konstantinou, K., "System and method for predistorting a signal using current and past signal samples," July 2001. U.S. Patent Application, Serial No. 09/915042.

[24] Graboski, J. and Davis, R. C., "An experimental M-QAM modem using amplifier linearization and baseband equalization techniques," in *Proc. IEEE Nat. Telecommun. Conf.*, pp. E3.2.1–E3.2.6, Nov. 1982.

[25] Kang, H. W., Cho, Y. S., and Youn, D. H., "On compensating nolinear distortions of an OFDM system using efficient adaptive predistorter," *IEEE Trans. Commun.*, vol. 47, pp. 522–526, Apr. 1999.

[26] Karam, G. and Sari, H., "Data predistortion techniques using intersymbol interpolation," *IEEE Trans. Commun.*, vol. 38, pp. 1716–1723, Oct. 1990.

[27] Karam, G. and Sari, H., "A data predistortion technique with memory for QAM radio systems," *IEEE Trans. Commun.*, vol. 39, pp. 336–344, Feb. 1991.

[28] Kenington, P. B., *High-Linearity RF Amplifier Design*. Boston, MA: Artech House, 2000.

Petitioners Exhibit 1005
Page 115 of 117

[29] Kenney, J. S., Woo, W., Ding, L., Raich, R., Ku, H., and Zhou, G. T., "The impact of memory effects on predistortion linearization of RF power amplifiers," in *Proc. Int. Symp. Microwave Optical Technol.*, pp. 189–193, June 2001.

[30] Kim, J. and Konstantinou, K., "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, pp. 1417–1418, Nov. 2001.

[31] Ma, Z., Zierdt, M., Dunkleberger, L., and Pastalan, J., "Memoryless power amplifier characterization for digital baseband predistortion (ii)." unpublished work, Jan. 2001.

[32] Ma, Z., Zierdt, M., and Pastalan, J., "Charaterization of power amplifier memory effect for digital baseband predistortion." unpublished work, Jan. 2001.

[33] Maas, S. A., *Nonlinear Microwave Circuits.* Piscataway, NJ: IEEE Press, 1997.

[34] Moon, T. K. and Stirling, W. C., *Mathematical Methods and Algorithms for Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1999.

[35] Morgan, D. R., Ma, Z., and Ding, L., "Reducing measurement noise effects in digital predistortion of RF power amplifiers," in *Proc. IEEE Int. Conf. Commun.*, pp. 2436–2439, May 2003.

[36] Nagata, Y., "Linear amplification technique for digital mobile communications," in *Proc. IEEE Veh. Technol. Conf.*, vol. 1, pp. 159–164, May 1989.

[37] Oppenheim, A. V. and Schafer, R. W., *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1999.

[38] Raich, R., Qian, H., and Zhou, G. T., "Digital baseband predistortion of nonlinear power amplifiers using orthogonal polynomials," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 689–692, Apr. 2003.

[39] Raich, R. and Zhou, G. T., "On the modeling of memory nonlinear effects of power amplifiers for communication applications," in *Proc. IEEE Digital Signal Processing Workshop*, Oct. 2002.

[40] Saleh, A. A. M., "Frequency-independent and frequency-dependent nolinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. 29, pp. 1715–1720, Nov. 1981.

[41] Saleh, A. A. M. and Salz, J., "Adaptive linearization of power amplifiers in digital radio systems," *Bell Syst. Technical J.*, vol. 62, pp. 1019–1033, Apr. 1983.

[42] Stapleton, S. P. and Cavers, J. K., "A new technique for adaptation of linearizing predistorters," in *Proc. IEEE Veh. Technol. Conf.*, pp. 753–758, May 1991.

[43] Stapleton, S. P. and Costescu, F. C., "An adaptive predistortion for a power amplifier based on adjacent channel emissions," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 49–56, Feb. 1992.

[44] Sundström, L., *Digital RF Power Amplifier Linearisers.* PhD thesis, Lund Univ., Lund, Sweden, 1995.

Petitioners Exhibit 1005
Page 116 of 117

[45] Tuthill, J. and Cantoni, A., "Optimum precompensation filters for iq modulation systems," *IEEE Trans. Commun.*, vol. 47, pp. 1466–1468, Oct. 1999.

[46] Valkama, M., Renfors, M., and Koivunen, V., "Compensation of frequency-selective i/q imbalances in wideband receivers: models and algorithms," in *Proc. IEEE Workshop Signal Processing Advances in Wireless Commun.*, pp. 42–45, Mar. 2001.

[47] Vuolevi, J. H. K., Rahkonen, T., and Manninen, J. P. A., "Measurement technique for characterizing memory effects in RF power amplifiers," *IEEE Trans. Microwave Theory Tech.*, vol. 49, pp. 1383–1388, Aug. 2001.

[48] Watkins, D. S., *Fundamentals of Matrix Computations*. New York, NY: John Wiley & Sons, 2 ed., 2002.

[49] Westesson, E. and Sundstrom, L., "A complex polynomial predistorter chip in CMOS for baseband or IF linearization of RF power amplifiers," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 206 –209, May 1999.

[50] Wright, A. and Nesper, O., "Multi-carrier WCDMA basestation design considerations - amplifier linearization and crest factor control," technology white paper, PMC-Sierra, Santa Clara, CA, Aug. 2002.

Petitioners Exhibit 1005
Page 117 of 117