An Analog Neural Network for Wideband Predistortion of Pico-cell Power Amplifiers

by Melin Ngwar

A thesis
presented to the Department of Electronics
in fulfillment of the
requirement for the degree of
Doctor of Philosophy

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Carleton University
Ottawa, Ontario

Copyright ©2015, Melin Ngwar

AUTHOR'S DECLARATION

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

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

Abstract

Pico-cell base-station power amplifiers (PA) generally generate less than 2W of power and operate at peak efficiency. This implies that to meet stringent wideband wireless standards, said amplifiers require linearization. The complexity and consequently power consumption of the linearizer is proportional to the power amplifiers efficiency and independent of output power. As a consequence, standard linearizers used in high PAs become unfeasible for use with pico-cell PAs as they are power intensive thus degrading the efficiency of the linearizer-PA combination. Feedback linearizers are only valid for narrowband stimuli while feedforward linearizers also suffer from the same total efficiency degradation due to the power consumption of the auxiliary amplifier. This leaves predistortion as the only viable option provided the algorithm/architecture is tailored to provide the same linearity benefit for high PAs as pico-cell PAs but with lower power consumption. The choice of neural networks as a predistortion algorithm compared to others such as Weiner, and Hammerstein stems from their ability to provide a suitable tradeoff between ACPR and EVM metrics. This thesis introduces an efficient dynamic neural network implementation which is specifically tailored for PA linearization. The focus and novelty of this work lies in the system inversion of measured PA non-linearity with a custom training algorithm as well as circuit design and hardware implementation of analog networks. Analog circuits are chosen to eliminate the power dependence of digital circuits on data rates; an effect which is most keenly felt for wideband stimuli. The implementation challenges include circuit design for large signal synaptic weights, wideband active delay elements, and an activation function. The aforementioned challenges have been tackled to yield a weight-limited algorithm which

trains a neural network predistorter to improve the ACPR and EVM of the pico-cell power amplifier by at least 13.5dB and 8.7% respectively. Furthermore, the implemented analog neural network predistorter circuits have a bandwidth and linearity of 50MHz and 5 bits respectively with suggested improvements to increase the performance to 120MHz and 7 bits respectively.

Acknowledgements

The guidance given by my supervisor, Professor Jim Wight has been invaluable to my research efforts. His aid in the acquisition of Research grants and facilities has provided me with the best mental and physical environment throughout the course of this project.

All work done during the course of this proposal has been carried out at Ericsson DU Radio and Carleton University. I will like to acknowledge Ericsson Canada for financially supporting this research and providing the equipment required for experimental evaluation. I will also like to thank Dr. Mark Wyville, of Ericsson DU Radio for his invaluable assistance in all aspects of design and implementation. Finally, I especially thank NXP for their gratuitous fabrication and packaging of all ICs related to this work.

Financial Support for this thesis was provided by the Department of Electronics at Carleton University, Faculty of Graduate Studies at Carleton University, and Ericsson Canada.

Table of Contents

AUTHOR'S DECLARATION	ii
Abstract	iii
Acknowledgements	v
Table of Contents	vi
List of Figures	viii
List of Tables	xii
List of Abbreviations	xiii
Chapter 1 Introduction	1
1.1 Description of Thesis Objectives	1
1.2 Literary review	3
1.3 Thesis Overview	4
1.4 Thesis Contributions	5
Chapter 2 Linearization Techniques in Power Amplifiers	6
2.1 Feedback Linearization	9
2.2 Feed-forward Loop Correction	12
2.3 Pre-distortion	17
2.3.1 Pre-distortion Algorithms	22
2.4 Chapter Summary	25
Chapter 3 Neural Networks	26
3.1 Behavioral modeling and training	26
3.2 Proposed training method	33
3.3 Theoretical basis for non-linear modeling	39
3.4 Chapter Summary	40
Chapter 4 System Inversion and Pre-distortion	42
4.1 Neural Network Pre-distorter Architecture	42
4.2 Measurement Setup	46
4.3 System Inversion	50

4.4 Behavioral Pre-distortion	55
4.5 Chapter Summary.....	58
Chapter 5 Analog Circuit Design of Neural Network Components	59
5.1 Basic Building Blocks.....	60
5.2 Analog tapped delay line.....	68
5.3 Analog non-linear neuron.....	80
5.3.1 Synaptic weight	80
5.3.2 Activation function circuit.....	86
5.4 Analog linear neuron.....	89
5.5 Circuit Predistortion	92
5.6 Chapter Summary.....	94
Chapter 6 Analog Circuit Measurement of Neural Network Components.....	96
6.1 Analog non-linear neuron.....	96
6.1.1 Gain-control and linearity.....	98
6.1.2 Frequency response	99
6.1.3 Activation function circuit.....	101
6.2 Analog linear neuron.....	102
6.2.1 Gain-control and linearity.....	104
6.2.2 Frequency response	106
6.3 Analog tapped delay line.....	107
6.4 Power consumption	115
6.5 Chapter summary	116
Chapter 7 Discussion, Conclusion and Future Work.....	118
7.1 Discussion	118
7.2 Conclusion.....	120
7.3 Future Work	123
Appendix A Testing Devices	130
References.....	132

List of Figures

Figure 2.1: Basic feedback linearizer (on left) and Cartesian feedback (on right) [6]	10
Figure 2.2: Base-band feedback linearization [6]	11
Figure 2.3: Magnitude envelope feedback (on left) & Phase envelope feedback (on right) [6]	11
Figure 2.4: Basic Feed-forward Loop [6]	13
Figure 2.5: Power transfer function +/-1dB compression adjustment and no AM-PM (on left), and 0.5dB compression adjustment with AM-PM (on right) [6].....	16
Figure 2.6: Efficiency against input power for various EPR values [6]	17
Figure 2.7: Pre-distorter Action	19
Figure 2.8: IM3 component with and without pre-distortion [6].....	19
Figure 2.9: IM3 for different third order Volterra phase offsets [6].....	21
Figure 2.10: Block Diagram of adaptive pre-distorter [6]	22
Figure 2.11: Memory polynomial, Wiener and Hammerstein models	23
Figure 3.1: Block Diagram of a neuron [15].....	27
Figure 3.2: Single layer recurrent neural network (on left) and two layer feed-forward network (on right) [15].....	28
Figure 3.3: Block diagram showing system identification [15].....	29
Figure 3.4: Flow chart describing Levenberg-Marquardt algorithm	38
Figure 4.1: Transceiver chain with in-phase/quadrature pre-distorter.....	43
Figure 4.2: Block diagram of FFDNN (on left) and composition of 'LN' or 'NN' (on right) .	44
Figure 4.3: Transfer function and derivative of activation function.....	46
Figure 4.4: Measurement setup for characterizing the DUT	47
Figure 4.5: AM-AM and AM-PM plots of non-linear DUT.....	50
Figure 4.6: AM-AM and AM-PM plots of linearized DUT	56
Figure 4.7: Measured spectrum of 4-carrier WCDMA signal	57
Figure 5.1: Early-Voltage simulation of HBT transistor	61
Figure 5.2: Transconductance simulation of HBT transistor.....	62

Figure 5.3: Output impedance simulation of PMOS transistor.....	63
Figure 5.4: Transconductance of PMOS transistor.....	63
Figure 5.5: Single-Ended folded cascode operational amplifier.....	65
Figure 5.6: Fully differential folded cascode operational amplifier	65
Figure 5.7: Frequency and phase response for single-ended op-amp	67
Figure 5.8: Frequency and phase response for fully differential op-amp	67
Figure 5.9: Capacitance and quality factor variation of varactor with tuning voltage	68
Figure 5.10: Magnitude and Group Delay response of 2nd order Pade approximation	70
Figure 5.11: Second order filter with Pi network	71
Figure 5.12: Second order filter with T network	71
Figure 5.13: Schematic of single delay element	72
Figure 5.14: Layout of four cascaded delay elements	74
Figure 5.15: Transfer function gain of delay line at $V_{\text{tune}} = 2.2\text{V}$	75
Figure 5.16: Transfer function gain of delay line at $V_{\text{tune}} = 1.25\text{V}$	75
Figure 5.17: Transfer function gain of delay line at $V_{\text{tune}} = 0.3\text{V}$	76
Figure 5.18: Transfer function of group delay for delay line at $V_{\text{tune}} = 2.2\text{V}$	77
Figure 5.19: Transfer function of group delay for delay line at $V_{\text{tune}} = 1.25\text{V}$	77
Figure 5.20: Transfer function of group delay for delay line at $V_{\text{tune}} = 0.3\text{V}$	78
Figure 5.21: Schematic of synaptic weight.....	82
Figure 5.22: Schematic of multipliers and adder combination in non-linear neuron	84
Figure 5.23: Simulated output voltage of NN for control voltage (V_y) ranging from $\pm 1\text{V}$..	85
Figure 5.24: Simulated frequency response of NN multiplier for V_y ranging from 0-1V.	85
Figure 5.25: Schematic of activation function circuit.....	87
Figure 5.26: Transfer function and derivative of analog activation function.	88
Figure 5.27: Schematic of multipliers and adder combination in linear neuron.....	90
Figure 5.28: Simulated frequency response of LN multiplier for V_y ranging from 0-2V.....	91
Figure 5.29: Simulated output voltage of LN for control voltage (V_y) ranging from $\pm 2\text{V}$	92
Figure 5.30: Measured spectrum of 4-carrier WCDMA signal with circuit predistortion.	94
Figure 6.1: Integrated circuit layout of non-linear neuron.....	97

Figure 6.2: PCB used to measure non-linear neuron.	97
Figure 6.3: Measured and simulated gain against control voltage for non-linear neuron	98
Figure 6.4: Measured and simulated gain-error against control voltage for non-linear neuron.	99
Figure 6.5: Loaded measured and simulated frequency response.	100
Figure 6.6: Measured and simulated activation function.	102
Figure 6.7: Measured and simulated loaded frequency response of activation circuit.	102
Figure 6.8: Integrated circuit layout of linear neuron.	103
Figure 6.9: PCB used to measure linear neuron.	104
Figure 6.10: Measured and simulated gain against control voltage for linear neuron.	105
Figure 6.11: Measured and simulated gain-error against control voltage for linear neuron.	105
Figure 6.12: Loaded measured and simulated frequency response of linear neuron.	106
Figure 6.13: Single-Ended folded cascode operational amplifier with buffer.	108
Figure 6.14: Magnitude and phase response of buffered op-amp for various capacitive loads	110
Figure 6.15: Revised PCB used to measure delay elements.	111
Figure 6.16: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 2.3\text{V}$	111
Figure 6.17: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 1.25\text{V}$	112
Figure 6.18: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 0.3\text{V}$	112
Figure 6.19: Measured group delay at $V_{\text{tune}} = 2.3\text{V}$	113
Figure 6.20: Measured group delay at $V_{\text{tune}} = 1.25\text{V}$	114
Figure 6.21: Measured group delay at $V_{\text{tune}} = 0.3\text{V}$	114
Figure 7.1: Four input folded cascode operational amplifier.	123
Figure 7.2: Modified single delay element	124
Figure 7.3: Transimpedance amplifier as an adder	125
Figure 7.4: Simulated output voltage for control voltages ranging from $\pm 1\text{V}$	126
Figure 7.5: Frequency response of multiplier for V_y ranging from 0-1V.	126
Figure 7.6: Simulated voltage gain and its percentage error against control voltage	127
Figure 7.7: Activation functions for both fabricated and modified circuits	128

Figure A1: RF devices for linearization.....	130
Figure A2: Initial PCB used to measure non-linear neuron IC.....	131
Figure A3: Initial PCB used to measure delay IC.....	131

List of Tables

Table 1.1: Specifications and Results	2
Table 1.2: Review of ACLR improvement due to neural network pre-distorters	3
Table 2.1: Two Tone third order amplitudes with frequency [5]	7
Table 4.1: Unrestricted training performance for various delay and hidden layer configurations.	53
Table 4.2: Weight limited training performance for various maximum weights.	54
Table 4.3: Measured behavioral ACPR and EVM results of WCDMA signal	57
Table 5.1: Parameters of trained neural network	60
Table 5.2: Gain variation for analog tapped delay line.....	79
Table 5.3: Measured circuit ACPR and EVM results of WCDMA signal	93
Table 6.1: Performance comparison with other neurons	116

List of Abbreviations

3GPP	3 rd Generation Partnership Project
4C-WCDMA	4 Carrier Wideband Code Division Multiple Access
ACPR	Adjacent Channel Power Ratio
ADS	Agilent Design Software
AM-AM	Amplitude Modulation to Amplitude Modulation
AM-PM	Amplitude Modulation to Phase Modulation
ANN	Artificial Neural Network
CFR	Crest Factor Reduction
CMFB	Common Mode Feedback
CMOS	Complementary Metal Oxide Semiconductor
DNN	Dynamic Neural Network
DPD	Digital Pre-distortion
DUT	Device Under Test
EPR	Error Power Ratio
ESG	Vector Signal Generator
EVM	Error Vector Magnitude
FET	Field Effect Transistor
FFDNN	Feed-forward Dynamic Neural Network
GaN	Gallium Nitride
HBT	Hetero-junction Bipolar Transistor
HEMT	High Electron Mobility Transistor
IM3	3 rd Order Intermodulation

L-M Levenberg-Marquardt

LDMOS Laterally Diffused Metal Oxide Semiconductor

LN Linear Neuron

MLP Multilayer Perceptron

MSE Mean Square Error

NMSE Normalized Mean Square Error

NN Non-linear Neuron

N-W Nguyen-Widrow

PA Power Amplifier

PAE Power Added Efficiency

PAPR Peak to Average Power Ratio

PMOS P-channel Metal Oxide Semiconductor

PSA Power Spectrum Analyzer

RBF Radial Basis Functions

RMSE Root Mean Square Error

RF Radio Frequency

SSE Sum Squared Error

WCDMA Wideband Code Division Multiple Access

Chapter 1

Introduction

1.1 Description of Thesis Objectives

An ideal power amplifier will produce the maximum output power while consuming minimal power and exhibit the least amount of distortion. The ratio of consumed power to output power is termed “efficiency” while the distortion level of the amplifier is portrayed with the “linearity” term. Linearity and efficiency always act to oppose each other such that they cannot be jointly optimized. The solution is to design an efficient amplifier and use a linearization technique to combat the non-linearity. This thesis suggests a linearization technique (pre-distortion) which uses neural networks to combat the PA non-linearity. The proposed neural network pre-distorter is optimized through iterative behavioral simulations in order to ensure a maximally efficient pre-distorter/PA combination without compromising the PA’s linearity. It is also worthwhile to note that no wideband real-time neural network pre-distorter has been realized yet in the literature. Based on parameters from an optimized behavioral network, the analog neuron and active delay elements are designed and fabricated in a BiCMOS process.

The specifications in Table 1.1 are based on 3GPP standards for multi-carrier WCDMA modulation [1]. As the focus of this thesis is on the implementation of a reconfigurable PA linearizer and not the PA itself, an off the shelf class AB PA (with a 1dB compression of 2W) made by PHOENIX is used to demonstrate the pre-distorter’s efficacy.

Table 1.1: Specifications and Results

	Specifications	Without Pre-Distortion	Offline Pre-Distortion
Frequency/GHz	1.9 [1]	1.9	1.9
Max. P_{out} (4dB comp.)/W	--	2.2	--
Max PAE/%	--	29.63	25
Linear Gain/dB	--	22	--
Bandwidth/MHz	20 [1]	20	20
ACPR @ 5MHz	>45dB [1]	36.9	51.4
ACPR @ 10MHz	>50dB [1]	39	52.5
EVM/%	<17.5 [1]	11.8	3.1

The linearity metrics which have been targeted are the ACPRs at 5MHz and 10MHz, and the EVM as both reflect the ability of the predistorter to eliminate in-band and out of band products. The 3GPP EVM standard is 17.5%, however, in order to get appreciable distortion levels in the DUT, crest factor reduction (CFR) was used to reduce the PAPR to 8.5dB and consequently increase the EVM of the source signal to 2.9%. Intricate CFR techniques exist which reduce the PAPR while meeting the ACPR specification and keeping EVM below 17.5% but their implementation lies beyond the scope of this work. Our objective is to observe the ability of the neural linearizer to improve ACPR and EVM ratios. From Table 1.1, the ACPR specifications at 5MHz and 10MHz are met with no EVM degradation for the behavioral neural linearizer. Of more importance are the worst case improvements in ACPR and EVM respectively of 13.5dB (at a 10MHz offset) and 8.7% respectively which are realized by the behavioral linearizer.

1.2 Literary review

Table 1.2 shows the ACPR improvement as a result of neural network pre-distorters to date. Note that all references in Table 1.2 are based on behavioral models. Neural hardware, whether digital or analog, has not been properly investigated for wideband linearization because of the inherent challenges associated with migration from mathematical functionality to electronic circuitry. On average, previous neural network pre-distorters exhibit 11dB and 14dB ACPR improvement at 5MHz and 10MHz respectively as shown in Table 1.2. The proposed neural network pre-distorter (also shown in Table 1.2) exhibits an ACPR improvement of 15dB and 13.5dB at 5MHz and 10MHz respectively. The proposed behavioral model improves the ACPR by about 2dB (at a 5MHz offset) compared to previous models with a fewer number of input delays. The ACPR improvements in the proposed network compared to previous networks is not due to any structural differences, but to vigorous optimization as its parameters are subsequently used for circuit implementation.

Table 1.2: Review of ACLR improvement due to neural network pre-distorters

Frequency (MHz)	Modulation Type	No. of Neurons	Input Delays, Output delays	Power Amplifier	Δ ACLR @ 5MHz	Δ ACLR @ 10MHz	Compressed Gain
2140 [2]	2C WCDMA	30	0, 4	LDMOS Doherty	--	10dB	3dB
1960 [3]	2C WCDMA	24	3, 3	Class AB	10.4dB	18.45dB	3dB
2140 [4]	4C WCDMA	15	8, 0	LDMOS Doherty	13.16dB	--	3dB
1900	4C WCDMA	11	4, 0	Class AB	15dB	13.5dB	1.5dB

The second part of this Thesis is the implementation of analog circuits which make up the neural network predistorter. The complete predistorter is divided into three distinct components namely the analog delay line, the non-linear neuron and the linear neuron. Variable active analog delay lines provide group delays ranging from 7.5ns to 12.5ns over 10MHz of bandwidth. Non-linear and linear neurons contain 10 and 11 synaptic weights (or multipliers) respectively with each synaptic weight exhibiting 7bits of linearity and a 3dB-bandwidth of 120MHz. Each non-linear neuron also has an activation circuit which realizes an activation function that is bounded, differentiable and continuous according to training requirements.

1.3 Thesis Overview

Chapter 1 provides a description of the thesis objectives and shows the linearity results obtained to support the proposed pre-distortion scheme. Chapter 1 also compares the obtained linearity results with recently published linearity values to emphasize the validity of the proposed scheme. Chapter 2 examines the pros and cons of different linearizers to validate the choice of predistortion as appropriate for pico-cell PAs. Chapter 2 also compares different predistortion algorithms to justify the further investigation of neural networks as linearizers and finally, Chapter 3 provides a brief introduction to neural networks as well as a detailed examination of the training algorithms used to change an arbitrary neural network into a function approximation. Chapter 4 is an examination of the system identification, system inversion and offline predistortion of the class-AB PA using a specific neural network configuration. Chapter 5 describes the analog circuit design and simulated performance of the components which make up the optimized neural network obtained in Chapter 4. Chapter

6 describes the measurement results obtained from fabricated components described in Chapter 5. Finally, Chapter 7 describes improvements made to the analog circuits since fabrication as well as a feasibility study for complete predistorter implementation using the realized analog circuits based on the simulated and measured results.

1.4 Thesis Contributions

Contributions made in this Thesis involve the modification of the standard L-M training algorithm to account for finite weights with minimal degradation in accuracy and a reduction in processing time. This contribution resulted in a journal submission to the IEEE Neural Network Society [5]. Secondly, the system identification, inversion and optimization of the DUT using behavioral neural networks resulted in a publication at an IEEE Signal Processing conference [6]. Finally, the utilization of neural networks in PA linearization is a fairly new concept and no hardware implementations currently exist which utilize neural networks to pre-distort the dynamic non-linearity of a PA. The realized analog circuits in this Thesis are a significant portion of such an implementation and they resulted in another publication at an IEEE Neural Network conference [7].

Chapter 2

Linearization Techniques in Power Amplifiers

Mathematically, a memoryless non-linear transfer function can be written by the following equation:

$$v_o = k_o + k_1 v_i + k_2 v_i^2 + k_3 v_i^3 + \dots + k_n v_i^n \quad (2.1)$$

Taking a two tone test where $v_i = v_1 \cos \omega_1 t + v_2 \cos \omega_2 t$, and considering only terms up to the third order in (2.1), the output component amplitude against frequency is shown in Table

2.1. The output amplitude and phase at ω_1 is distorted by $k_3 v_1 (\frac{3}{4} v_1^2 + \frac{3}{2} v_2^2)$, and the output amplitude and phase at ω_2 is distorted in a similar manner. In addition to amplitude distortion at the carrier frequencies, the non-linear transfer function generates additional frequencies at harmonics and inter-modulation products of the carrier frequencies.

Harmonics are easy to eliminate by filtering but inter-modulation products resulting from odd-order terms are close to the carrier and cannot be removed by a filter. The culmination of close in inter-modulation products, and amplitude and phase distortion represent the memoryless non-linearity in a power amplifier.

Power amplifiers also exhibit memory effects as a result of the matching networks as well as lumped components present in a transistor. The best PA from an RF perspective can be detrimental to the system performance. This is because RF PA designers aim for the highest gain and widest bandwidth. To achieve said RF performance, high quality matching networks with cascaded gain stages are used.

Table 2.1: Two Tone third order amplitudes with frequency [8]

Frequency	Component Amplitude
dc	$k_0 + \frac{k_2}{2}(v_1^2 + v_2^2)$
ω_1	$k_1 v_1 + k_3 v_1 \left(\frac{3}{4} v_1^2 + \frac{3}{2} v_2^2 \right)$
ω_2	$k_1 v_2 + k_3 v_2 \left(\frac{3}{4} v_2^2 + \frac{3}{2} v_1^2 \right)$
$2\omega_1$	$\frac{k_2 v_1^2}{2}$
$2\omega_2$	$\frac{k_2 v_2^2}{2}$
$\omega_1 \pm \omega_2$	$k_2 v_1 v_2$
$\omega_2 \pm \omega_1$	$k_2 v_1 v_2$
$3\omega_1$	$\frac{k_3 v_1^3}{4}$
$3\omega_2$	$\frac{k_3 v_2^3}{4}$
$2\omega_1 \pm \omega_2$	$\frac{3}{4} k_3 v_1^2 v_2$
$2\omega_2 \pm \omega_1$	$\frac{3}{4} k_3 v_1 v_2^2$

The result is an increase in RF bandwidth of the absolute gain and a corresponding increase in phase variation with frequency. Phase variation with frequency is directly proportional to group delay which is a direct reflection of the PA's memory. The consequence of memory or group delay is a performance limitation for various linearizers as described in the operation of feedforward, feedback and predistortion linearizers in subsequent sections.

Time invariant convergent non-linearities with memory can be modeled with a Volterra series. Equation (2.1) can be rewritten as shown in (2.2) to incorporate memory effects

$$v_o = k_o + \sum_{n=1}^{\infty} \int \dots \int h_n(\tau_1, \dots, \tau_n) u(t - \tau_1) \dots u(t - \tau_n) d\tau_1 \dots d\tau_n \quad (2.2)$$

Expanding equation 2.2 for better clarity yields (2.3) below

$$v_o = k_o + \int h_1(\tau_1) u(t - \tau_1) d\tau_1 + \iint h_2(\tau_1, \tau_2) u(t - \tau_1) u(t - \tau_2) d\tau_1 d\tau_2 + \sum_{n=3}^{\infty} \int \dots \int h_n(\tau_1, \dots, \tau_n) u(t - \tau_1) \dots u(t - \tau_n) d\tau_1 \dots d\tau_n \quad (2.3)$$

The first integral in (2.3) corresponds to a linear system with memory. The second term incorporates the memory into the non-linearity by taking the product of the input at every possible delay combination and multiplying by the two dimensional kernel ($h_2(\tau_1, \tau_2)$).

The first problem with modeling a power amplifier using the Volterra series is the time invariance assumption. Since the carrier traps or charge distribution in the device change with time, the amplifier non-linearity is not time invariant. As a result, the Volterra kernels have to be recomputed in real time to accurately model the power amplifier. Secondly, the dimension of the volterra kernels increases with the degree of non-linearity. A power amplifier model can show reasonable accuracy with a 5th order non-linearity implying a Volterra kernel with 5 dimensions is required for the fifth order term. Such kernels are very difficult to measure for a time invariant non-linearity and almost impossible to measure in real time especially when the input signal is not discretely multi-tone (which is the case for almost every digitally modulated signal).

The non-linear model suggested in this thesis is an approximation of the Volterra series based on neural networks with back-propagation algorithms used to recalculate “Volterra kernels”.

A more detailed explanation is given in Chapter 3.

2.1 Feedback Linearization

Linearization in power amplifiers is most commonly achieved by feedback, adaptive pre-distortion, and feed-forward. A basic block diagram of feedback linearization is shown in Figure 2.1 along with a possible implementation of Cartesian feedback linearization. The expression for feedback gain is shown in (2.4) below.

$$\frac{v_o}{v_i} = \frac{Ae^{j\theta}}{1 + \beta(Ae^{j\theta})} \quad (2.4)$$

Where β is the feedback gain, and $Ae^{j\theta}$ is the complex open loop gain.

Based on (2.4), any linearity improvement in the inter-modulation or adjacent channel products is equal to $1/\beta$. The implication of this requirement is that the open loop gain has to be much greater than $1/\beta$ for $\beta A \gg 1$. This is a difficult requirement to achieve at radio and microwave frequencies. Also, for the feedback to be stable, θ has to be 0 degrees over the entire signal bandwidth (with an appropriate margin for adjacent channels). Deviations in θ of over 45 degrees cause significant losses in the linearity improvement or closed loop gain. RF amplifiers with high gain are observed to have rapid in-band phase variations versus frequency due to high quality factor matching networks existing in cascaded structures thus causing the feedback loop to be unstable for wideband applications [9]. An example of Cartesian feedback implementation is [10] which realizes an ACPR of 21dB for a 50kHz bandwidth, 16-QAM signal.

Figure 2.1: Basic feedback linearizer (on left) and Cartesian feedback (on right) [9]

Figure 2.2 shows envelope feedback which is much easier to implement compared to RF feedback because relative processing in the feedback path, which causes phase delays, is much smaller at baseband compared to RF. Again, even at baseband, care should be taken to ensure that phase inversion in the feedback path occurs at all in-band frequencies. Figure 2.3 expatiates upon Figure 2.2 by showing the individual compositions of the magnitude and phase feedback loops. The feedback loop in Figure 2.3 is also less sensitive to phase variations in its open loop gain compared to Cartesian feedback as demonstrated by the closed loop gain expression in (2.5).

$$\frac{v_o}{v_i} = ((\alpha_o \cdot v_i - v_o)G + \alpha_o)G_{PA} \quad (2.5)$$

Figure 2.2: Base-band feedback linearization [9]

Figure 2.3: Magnitude envelope feedback (on left) & Phase envelope feedback (on right) [9]

Based on (2.5), we can observe that any variation in the phase of G (Video amplifier's gain) does not significantly affect the closed loop gain provided $(\alpha_o \cdot v_i - v_o)$ is small.

Furthermore, the video amplifier operates at baseband and has a much smaller phase-frequency variation compared to its RF counterpart. Finally, any rapid variations in the phase-frequency response of the PA's gain leaves the closed loop gain unchanged thus cancelling the AM-AM effect of the PA.

AM-PM compensation requires separate circuitry also shown in Figure 2.3. The mixer acts as a phase detector whose amplified phase error is used to correct the signal's phase at the PA's

input. The delay equalizer ensures that the input and output signals of the PA have the same nominal phase and the 90° phase shifter ensures that the phase detector has a $\sin(\cdot)$ transfer function thus being directly proportional to the phase error for small angles. Equations (2.6) to (2.8) depict AM-PM operation.

$$v_{in}(t) = r(t)\cos(\omega t + \psi(t)) \quad (2.6)$$

$$v_o(t) = A[r(t)]\cos(\omega t + \psi(t) + \phi[r(t)]) \quad (2.7)$$

$$v_e(t) = 0.5 * r(t) * A[r(t)]\sin(\phi[r(t)]) \quad (2.8)$$

Where $v_{in}(t)$ and $v_o(t)$ are the RF input and output to the PA respectively, and $v_e(t)$ is the low pass filtered phase detector output. The AM-AM compensation circuit is essentially a phase-locked loop and detailed loop analysis is required to definitively determine the effect of AM-AM effects in the phase detector characteristics on the loop's response. Loop analysis will also provide a good estimate of acquisition or synchronization time which in turn sets an upper limit for the modulation bandwidth. Envelope feedback has been successfully realized with wider bandwidths compared to Cartesian, as shown in [11] and [12] which linearize single carrier WCDMA signals to obtain ACPR improvement ratios of 15dB.

2.2 Feed-forward Loop Correction

The basic feed-forward loop operation is shown in Figure 2.4. Equations representing the functionality of the simple feed-forward loop in Figure 2.4 are shown in (2.9) to (2.12).

Let $v_{in} = v \cos \omega t$,

$$v_{pa} = a_1(v \cos \omega t) - a_3(v \cos \omega t)^3 \quad (2.9)$$

Where v_{pa} is the output of the power amplifier and v_{in} is the input signal. The output of the power amplifier is coupled by a factor of α resulting in the input to the error power amplifier given by:

$$v_e = v_{in} - \alpha v_{pa} = v \cos \omega t - \alpha a_1(v \cos \omega t) + \alpha a_3(v \cos \omega t)^3 \quad (2.10)$$

If $\alpha = \frac{1}{a_1}$,

$$v_e = \alpha a_3(v \cos \omega t)^3 \quad (2.11)$$

If the error power amplifier has a gain of $\frac{1}{\alpha}$ ($= a_1$), the feed-forward loop output is given by:

$$v_{out} = a_1(v \cos \omega t) - a_3(v \cos \omega t)^3 + \left(\frac{1}{\alpha}\right)\alpha a_3(v \cos \omega t)^3 \quad (2.12)$$

Figure 2.4: Basic Feed-forward Loop [6]

With the exception of the main and error amplifiers, all other components in the feed-forward loop are passive and can be made to operate in an almost ideal manner with careful design. The active components are thus the bottlenecks to the loop's linearity performance. The Error amplifier also exhibits a non-linear response as given by the following equations.

$$v_{epa} = b_1 v_e - b_3 v_e^3 = b_1 (\alpha a_3 (v \cos \omega t)^3) - b_3 (\alpha a_3 (v \cos \omega t)^3)^3 \quad (2.13)$$

Equation (2.13) shows that the error amplifier in Figure 2.8 provides a distorted component at the fundamental frequency which is proportional to a_3 of the main amplifier and has a 9:1 slope with respect to the input power. Equations (2.9) – (2.13) assume no AM-PM but the following equations examine the effect of AM-PM distortion on the error amplifier requirements.

$$v_{pa} = a_1 (v \cos \omega t) - a_3 (v \cos (\omega t + \varphi_3))^3,$$

where v_{pa} is the main amplifier output with a third order volterra phase angle.

Using coefficients from Table 2.1, the component of v_{pa} at ω is:

$$v_{pa_w} = a_1 (v \cos \omega t) + \frac{3}{4} a_3 v^3 \cos (\omega t + \varphi_3) \quad (2.14)$$

(2.14) can also be represented by $v_{pa_w} = \gamma v \cos (\omega t + \phi)$ where

$$\gamma a_1 v \cos (\phi) = a_1 v - \frac{3}{4} a_3 v^3 \cos (\varphi_3), \quad \gamma a_1 v \sin (\phi) = \frac{3}{4} a_3 v^3 \sin (\varphi_3) \quad (2.15) [9]$$

γ and ϕ represent AM-AM and AM-PM compression respectively. If $a_1 = 1$, (2.15) can be further reduced to the following expressions.

$$a_3 = \frac{4}{3}\sqrt{1 + \gamma^2 - 2\gamma\cos\phi}, \tan\varphi_3 = \frac{\gamma\sin\phi}{1-\gamma\cos\phi} \quad (2.16) [9]$$

If $\phi > 0$, then AM-PM compression exists and a_3 is higher compared to the case when $\phi = 0$ as seen from (2.16). From (2.14), the required signal from the error amplifier is shown in (2.17).

$$v_{ea_w} = -\frac{3}{4}a_3v^3\cos(\omega t + \varphi_3) \quad (2.17)$$

Equation (2.17) implies that if a_3 is high as is observed in the case with AM-PM compression of the main amplifier, the error amplifier is required to produce more power compared to a case with no AM-PM. A high a_3 drives the error amplifier towards saturation causing the error amplifier to exhibit non-linearities which are not compensated for by the loop.

Further linearity constraints due to feed-forward linearization occur from drift errors and tracking errors due to the input coupler and adder. Drift errors require adaptation of the input and output delays to ensure accurate coupling of the error signal and linearized output. Tracking errors are summarized by the input coupling coefficient “ α ”. Any deviation from the “ $\alpha = \frac{1}{a_1}$ ” requirement will result in a reduction in the IM3 slope from 9:1 to 3:1.

However, such deviations can also yield desirable nulls in the IM3 product at specific power levels as shown in Figure 2.5 for $\alpha = \pm 1dB$. Figure 2.5 also shows the power transfer characteristic for a feed-forward linearizer with $\alpha = 0.5$ and a PA which exhibits AM-PM characteristics. The inclusion of AM-PM in the PA characteristic has reduced the IM3

improvement due to linearization at a 2dB input backoff from 50dB to 20dB. IM3

improvement ratios are indicators of ACPR improvement due to linearization for wideband signals thus they are a worthwhile observation. A more realistic measure of the feed-forward loop's efficacy in suppressing intermodulation is obtained using wideband signals such as WCDMA. A feed-forward loop realized in [13] shows 15dB of ACPR improvement with a single carrier WCDMA stimulus.

Figure 2.5: Power transfer function +/-1dB compression adjustment and no AM-PM (on left), and 0.5dB compression adjustment with AM-PM (on right) [9]

The linearizer's efficiency can be estimated by (2.18) where the main amplifier is assumed to be a class AB and the error amplifier is a class A.

$$\eta = \frac{P_{max}}{\frac{P_{max}}{0.65 * \left(\frac{P}{P_{max}}\right)^{0.5}} + \frac{P_{max}}{0.4 * EPR}} \quad (2.18)$$

Where P_{max} is the main PA's 1dB compression, P is the output power level and EPR is the ratio between the 1dB compression points of the main and error power amplifiers. Figure 2.6

shows the efficiency variation with input power for various EPR values. An EPR of 10dB yields the best efficiency/linearity trade-off and for wideband inputs with approximately 10dB of peak to average power ratios, the average efficiency reduces from 21% to 14% due to the inclusion of feed-forward linearization. This is equivalent to a 50% increase in main amplifier's DC power without linearization which makes the feed-forward loop fairly inefficient compared to feedback and pre-distortion.

Figure 2.6: Efficiency against input power for various EPR values [9]

In conclusion, feed-forward linearization is fast and can yield adequate ACPR improvement. However, it is inefficient which makes it a poor choice for linearization of pico-cell amplifiers.

2.3 Pre-distortion

The final linearization scheme examined in this proposal is predistortion as depicted in its basic form by Figure 2.7, which shows how an input signal with a magnitude of V_{in} will yield

a compressed power amplifier output of V_s . To obtain the desired linear magnitude of V_o , the pre-distorter increases the input drive level from V_{in} to V_p . Considering a memoryless PA model consisting of first and third degree non-linearity terms only, the PA and pre-distorter equations are shown below.

$$v_o = a_1 v_p - a_3 v_p^3 = a_1 v_{in} \Rightarrow v_p = v_{in} + \frac{a_3}{a_1} v_p^3.$$

Setting $a_1 = 1$ and solving the expression for v_p iteratively yields an infinite series for v_p in terms of v_{in} as given in (2.19)

$$v_p = v_{in} + a_3 v_{in}^3 + 3a_3^2 v_{in}^5 + 12a_3^3 v_{in}^7 + 37a_3^4 v_{in}^9 + \dots \quad (2.19) [9]$$

The conclusion we can derive from (2.19) is that it is impossible to find a finite pre-distortion expression which can remove non-linearities for even the simplest PA model. We can however, find expressions which reduce undesirable non-linearities such as third-order inter-modulation distortion products (IM3) which are close to the carrier.

Let $v_p = v_{in} + b_3 v_{in}^3 \Rightarrow$

$$v_o = a_1 v_{in} + (a_1 b_3 - a_3) v_{in}^3 - 3a_3 b_3 v_{in}^5 - 3a_3 b_3 v_{in}^7 - a_3 b_3^3 v_{in}^9 \quad (2.20)[9]$$

If $b_3 = \frac{a_3}{a_1}$, the third order term in (2.20) is eliminated. While the elimination of the third order term does not remove IM3 products, it reduces them. This is because IM3 products are now a result of fifth order terms and higher causing an increase in the IM3 slope of the power transfer function from 3 to 5 as shown in Figure 2.8.

Figure 2.7: Pre-distorter Action

Figure 2.8: IM3 component with and without pre-distortion [9]

Reference [9] uses the transfer function in (2.21) to incorporate AM-PM effects in the non-linear model of the PA by weighing each order of non-linearity with a magnitude and

corresponding phase. Such an expression is a very simplistic approximation of a Volterra series but an analytical evaluation of (2.21) can yield valuable insight into pre-distorter limitations. The problem with (2.21) is that many combinations of magnitude and phase coefficients can yield the same AM-AM and AM-PM effects which leads to ambiguity when computing the predistorter coefficients.

$$V_o = a_1 V_i(wt) + a_3 [V_i(wt + \phi_3)]^3 + a_5 [V_i(wt + \phi_5)]^5 + \dots \quad (2.21)$$

Where $V_i(wt) = V_i(\tau)\cos(wt)$, $V_i(\tau)$ is the envelope of the input signal.

To cancel inter-modulation products, both the magnitude and phase coefficients at each order for the pre-distorter need to be found in terms of the PA magnitude and phase coefficients. This has been done in detail in [9] for a PA model up to the 7th order. Figure 2.9 shows the deviation in IM3 with and without pre-distortion for variation in the third order phase coefficient of a 5th order PA model. The degradation of IM3 with variation in Volterra kernels is an indication of the long term memory effect of the power amplifier. To maintain the expected linear response to pre-distortion, the Volterra coefficients have to be updated in real time when the PA model varies. This implementation is called adaptive pre-distortion as shown in Figure 2.10. The power amplifier response changes with environmental and temperature conditions. Even if the power amplifier is properly modeled with non-linearity and memory effects for a given set of environmental conditions, a versatile pre-distortion scheme should be reconfigurable with varying environmental conditions. An adaptive pre-distortion mechanism as shown in Figure 2.10 consists of a pre-distortion model with memory based on Volterra kernels which are in turn based on adaptive parameters. The

adaptive parameters are usually obtained from coupling of the PA output signal. Stability is not usually a concern in adaptive feedback since the time constants associated with changing environmental conditions are much larger compared to practical modulation time constants (seconds to us). If the power amplifier is operating under conditions with quickly changing time constants, the environmental effect has to be incorporated in the pre-distortion model to maintain the open loop response of the predistorter-PA combination. Linearizer adaptation and its algorithms lie beyond the scope of this work as the efficacy of the proposed pre-distorter implementation is only demonstrated over short term memory ($< 10\text{ms}$), however, said implementation is also compatible for adaptation.

Figure 2.9: IM3 for different third order Volterra phase offsets [9]

Figure 2.10: Block Diagram of adaptive pre-distorter [9]

To summarize what has been observed so far, feedback linearizers are only appropriate for narrowband signals and feedforward linearizers may be less efficient compared to pre-distorters. The power consumption of pre-distorters generally depends on the complexity of the chosen algorithm and intricately depends on circuit design and implementation. As a result, more investigation is needed before a definitive statement can be made about predistorter efficiency.

2.3.1 Pre-distortion Algorithms

There are a significant number of algorithms which have been investigated for the purpose of non-linear compensation. This section examines the more prevalent ones which have been successful to some degree as pre-distorters. Figure 2.11 shows the basic configurations of a memory polynomial model, Wiener and Hammerstein models.

Figure 2.11: Memory polynomial, Wiener and Hammerstein models

Memory polynomials can be described by (2.22) where b_{mk} represents non-linear coefficients corresponding to a particular delay given by m , p_m is the polynomial order at the m th delay and M is the order of memory required for equalization of the PA's impulse response.

$$y(n) = \sum_{m=0}^{m=M} x_{n-m} \sum_{k=1}^{k=p_m} b_{mk} (x(n-m))^{k-1} \quad (2.22) [14]$$

An implementation of predistortion using memory polynomials was examined in [14]. ACPR improvements of 12dB and 14dB were reported for single carrier and 3 carrier WCDMA modulation respectively. A disadvantage with the memory polynomial as reported by [14] is the inability of the algorithm to converge for memory depths which are greater than one. This limits the ability of the memory polynomial to linearize wideband signals and makes it unfavorable for usage as an adaptive algorithm.

The Wiener model is described by (2.23) where $h(n)$ is the impulse response representing the PA's inverse memory function, $\otimes$ is the convolution operator and a_k is the set of coefficients for a polynomial of the p th order.

$$y(n) = \sum_{k=0}^{k=p} a_k (x(n) \otimes h(n))^p \quad (2.23)$$

A predistorter based on the Wiener model is realized in [15] for an ACPR improvement ratio of about 14dB

The Hammerstein model as shown in (2.24), applies a polynomial before filtering as opposed to the Wiener model which performs the same functions in reverse order.

$$y(n) = [\sum_{k=0}^{k=p} a_k (x(n))^k] \otimes h(n) \quad (2.24)$$

Again predistortion based on the Hammerstein model according to [16] also yields about 15dB of improvement in ACPR. All three structures yield very similar improvements in ACPR. Hammerstein structures have slightly better ACPR because filtering occurs at the predistorter's output hence there is greater control over equalization of the PA's frequency response. This comes at a cost of lower EVM compared to neural networks as described in [17]. Neural network structures do not drastically vary from Wiener or Hammerstein models and as such significantly improved gains in ACPR or EVM are not expected as observed from [2]-[4]. This implies that any performance improvements in terms of power consumption and linearity are implementation specific which is the focus of this work. A detailed description of neural network operation is presented in the next chapter.

2.4 Chapter Summary

This chapter first characterizes a non-linear power amplifier for both dynamic and memoryless operation. Next, it examines different ways to combat non-linearity such as feedback, feed-forward linearization, and pre-distortion. Pre-distortion is the chosen linearization scheme for implementation because it can accommodate wider bandwidth than the feedback linearizer and is more efficient compared to the feed-forward linearizer. Finally, this Chapter also looks at different pre-distortion algorithms in order to determine which or if any algorithm has proven its superiority over the others and the conclusion is that they all yield similar performances from a behavioral standpoint.

Chapter 3

Neural Networks

3.1 Behavioral modeling and training

The electrical model of a neuron is shown in Figure 3.1. Each input to the neuron is scaled by a weight and the sum of all scaled inputs is shifted by a bias. The biased sum of weighted inputs is put through an activation function. The activation function is any bounded and differentiable non-linear function. The neural network operation of the behavioral diagram in Figure 3.1 is represented by (3.1) which shows the neuron output (y_k) as a function of its $\varphi(\cdot)$ activation function and inputs (x_j). A layer consists of a specified number of neurons and the outputs of each layer are equal to the outputs of the neurons in the layer. In this way, a hierarchical structure of multiple layers can be obtained as shown in Figure 3.1.

$$v_k = \sum_{j=1}^m w_{kj} x_j, y_k = \varphi(v_k + b_k) \quad (3.1)$$

Where w_{kj} is the synaptic weight to the k th neuron at the j th input index, m is the number of weights per neuron, b_k is the bias of the neuron, and v_k is the input to the activation function.

The activation function denoted by $\varphi(x)$ causes non-linearity in the neural network and as mentioned before, it must be bounded, continuous and differentiable. Common examples of activation functions are shown in (3.2).

$$\varphi(x) = \frac{1}{1+e^{-x}}, \quad \frac{2}{1+e^{-2x}} - 1, \quad e^{-x^2}, \quad \frac{1}{1+x^2} \quad (3.2)$$

The expressions in (3.2) are continuous, converge to either 1,0 or -1 as x approaches $\pm\infty$, and are differentiable with respect to x .

Figure 3.2 shows a two layer network (the input layer is not considered) with 4 neurons in the hidden layer and 2 neurons in the output layer. The neurons in the hidden layer only accept external inputs while the neurons in the output layer only accept as inputs, the outputs of the hidden layer. Each neuron has the structure of Figure 3.1 and the architecture of Figure 3.2 is called a feed-forward neural network. Non-linear systems with memory are modeled using dynamic neural networks. Dynamic networks could either be feed-forward or recurrent. The feed-forward dynamic networks use tapped delay lines at the input layer and the output of each tap is treated as a separate input to the hidden layer. Recurrent networks use feedback at discrete delay intervals to model non-linear systems with both transient and steady state memory.

Figure 3.1: Block Diagram of a neuron [18]

Figure 3.2 gives an example of a recurrent network where the unit-delayed outputs of all 4 neurons are fed back into the input layer. Recurrent networks are more difficult to implement compared to feed-forward networks because the former requires stability considerations when computing the feedback delay coefficients.

Figure 3.2: Single layer recurrent neural network (on left) and two layer feed-forward network (on right) [18]

For any neural network to accurately model a non-linear system, the weights and biases associated with each neuron have to be adjusted. This process is called network training. The input and output data of the non-linear device for the desired operating range is collected and said data is used to train the neural network. Training is done by an iterative process whereby, the error function at the network output is computed for a given set of weights and

biases; the network weights and biases are then updated before the next iteration according to an optimization algorithm such that the desired output error is reduced.

Figure 3.3: Block diagram showing system identification [18]

Figure 3.3 is a basic depiction of the system identification of a non-linear device. The “Unknown System” in Figure 3.3 represents the non-linear device to be modeled and the error (e_n) between the unknown device (d_n) and the neural network model (y_n) for a given input is used to adjust the model’s weights and biases towards a minimum error.

The choice of optimization algorithm is very important as it can determine how fast the system identification takes place. The purpose of optimization is the minimization of the error function or more specifically, the sum square error function (SSE) given by (3.3). Optimizing (3.3) ensures that the optimum weights minimize the error over the entire sample space instead of the error at a specific time index as shown in (3.4).

$$E(\mathbf{w}) = 0.5 * \sum_{i=1}^M (d(\mathbf{x}_n) - y(\mathbf{x}_n, \mathbf{w}))^2 \quad (3.3)$$

$$e(\mathbf{x}_n, \mathbf{w}) = d(\mathbf{x}_n) - y(\mathbf{x}_n, \mathbf{w}) \quad (3.4)$$

Where $\mathbf{x}_n$ refers to the input(s) vector(s) at time index n , E is the SSE, e is the error at the j th iteration, y is the neural network function and d is the nonlinear device to be identified. $\mathbf{w}$ is a vector which represents all network weights and is given by: $\mathbf{w} = [w_1, w_2, \dots, w_c]^T$ where c is the number of weights in the neural network. Note that e occurs at a particular time index while E is the error-sum at M successive time indices. $\mathbf{x}_n$ is a vector whose length is equal to the memory depth for a single input or the product of the memory depth and the number of inputs for multiple inputs. The most commonly used optimization algorithm is the Levenberg-Marquardt algorithm based on the method of steepest descent and the Gauss-Newton method. The Levenberg-Marquardt algorithm is so efficient because it exploits the fast convergence of the steepest descent and the Gauss-Newton methods for steep and gradual gradients respectively. The weight update per iteration ($\mathbf{w}_{old} \rightarrow \mathbf{w}_{new}$) for the steepest descent method as shown in (3.6) depends on first order derivatives which constitute the Jacobian matrix shown in (3.5).

$$J(\mathbf{w}) = \begin{bmatrix} \frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_2} & \dots & \frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_c} \\ \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_2} & \dots & \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_c} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_2} & \dots & \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_c} \end{bmatrix} \quad (3.5)$$

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \eta (J^T(\mathbf{w}_{old}) \cdot \mathbf{e}(\tilde{\mathbf{x}}, \mathbf{w}_{old})) \quad (3.6)$$

Where $\tilde{\mathbf{x}} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$, and $\mathbf{e}(\tilde{\mathbf{x}}, \mathbf{w}) = [e(\mathbf{x}_1, \mathbf{w}), e(\mathbf{x}_2, \mathbf{w}), \dots, e(\mathbf{x}_M, \mathbf{w})]$. η is initially, arbitrarily chosen as a small value to ensure $E_{new} < E_{old}$ but can be increased in subsequent iterations to ensure faster convergence of $E(\mathbf{w})$ to a minimum. (3.6) is a weight solution to the linear approximation of the SSE function which becomes increasingly invalid as the error

function approaches a minimum point. As a result, η needs to be very small for the linear approximation to remain valid or for $E_{new} < E_{old}$ implying slow convergence for the steepest gradient algorithm. Newton's method approximates a quadratic $E(\mathbf{w})$ function which becomes increasingly valid as $E(\mathbf{w})$ approaches its minimum and less valid the further $E(\mathbf{w})$ is from its minimum. The implication is fast convergence of the Newton method around the minimum point and slow convergence otherwise. Equation (3.7) shows the expression for the Hessian matrix which is used to compute the Newton weight update as shown in (3.8).

$$\mathbf{H}(\mathbf{w}) = \begin{bmatrix} \frac{\partial^2 E(\mathbf{w})}{\partial w_1^2} & \frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_c} \\ \frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_2} & \frac{\partial^2 E(\mathbf{w})}{\partial w_2^2} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_2 \partial w_c} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E(\mathbf{w})}{\partial w_c \partial w_1} & \frac{\partial^2 E(\mathbf{w})}{\partial w_c \partial w_2} & \cdots & \frac{\partial^2 E(\mathbf{w})}{\partial w_c^2} \end{bmatrix} \quad (3.7)$$

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{H}(\mathbf{w}_{old})^{-1} \cdot (\mathbf{J}^T \mathbf{e}(\tilde{\mathbf{x}}, \mathbf{w}_{old})) \quad (3.8)$$

Equation (3.8) is only valid if the Hessian matrix is positive definite. This is because the Hessian matrix is the two dimensional equivalent of the second order differential of the error function, and for a quadratic (second order) Taylor series approximation, a positive second order differential implies the function is at its minimum. It is impossible to guarantee a positive definite Hessian matrix for all input values making the Newton optimization method slightly impractical. In addition, computation of the Hessian matrix can be quite intensive as shown in (3.9) by the expansion of a single element in $\mathbf{H}(\mathbf{w})$. Equation (3.9) contains a second order derivative term which requires computation of a derivative matrix with a size of

$M \times c^2$; the size of said matrix becomes increasingly unmanageable for large networks with wideband inputs.

$$\frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_2} = \sum_{i=1}^M \frac{\partial e(\mathbf{x}_i, \mathbf{w})}{\partial w_1} \cdot \frac{\partial e(\mathbf{x}_i, \mathbf{w})}{\partial w_2} + e(\mathbf{x}_i, \mathbf{w}) \cdot \frac{\partial^2 e(\mathbf{x}_i, \mathbf{w})}{\partial w_1 \partial w_2} \quad (3.9)$$

The Gauss-Newton method resolves this issue in Hessian matrix computation by approximating (3.9) with (3.10) [19]-[20]. The resulting Hessian matrix approximation obtained by combining (3.10) with (3.7) is given by (3.11) and the Gauss-Newton weight update is computed in (3.12).

$$\frac{\partial^2 E(\mathbf{w})}{\partial w_1 \partial w_2} \approx \sum_{n=1}^M \frac{\partial e(\mathbf{x}_n, \mathbf{w})}{\partial w_1} \cdot \frac{\partial e(\mathbf{x}_n, \mathbf{w})}{\partial w_2} \quad (3.10)$$

$$\mathbf{S}(\mathbf{w}) = \mathbf{J}^T \mathbf{J} \approx \mathbf{H}(\mathbf{w}) \quad (3.11)$$

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \mathbf{S}(\mathbf{w}_{old})^{-1} \cdot (\mathbf{J}^T(\mathbf{w}_{old}) \cdot \mathbf{e}(\tilde{\mathbf{x}}, \mathbf{w}_{old})) \quad (3.12) [18]$$

Equation (3.12) minimizes the cost function or SSE given by (3.3). The Gauss-Newton method is better in terms of speed than the Newton method for minimizing the error over a large number of samples; this effect is quite beneficial for modeling non-linear systems which accept wide-band inputs as wide-band signals have a large sample space.

For the Gauss-Newton algorithm to be valid, $\mathbf{J}^T \mathbf{J}$ has to be non-singular. For $\mathbf{J}^T \mathbf{J}$ to be non-singular, $\mathbf{J}$ must have row-rank n . This condition might not always hold, and to guard against that possibility, the updates in (3.12) are made to the Gauss-Newton algorithm in (3.12) yielding the Levenberg-Marquardt algorithm [18].

$$\mathbf{w}_{new} = \mathbf{w}_{old} + (\mathbf{S}(\mathbf{w}_{old}) + \lambda \mathbf{I})^{-1} (\mathbf{J}^T(\mathbf{w}_{old}) \cdot \mathbf{e}(\tilde{\mathbf{x}}, \mathbf{w}_{old})) \quad (3.13) [18]$$

where $\mathbf{I}$ is the identity matrix and λ is an arbitrary scaling factor. $\lambda\mathbf{I}$ ensures that $\mathbf{S}(\mathbf{w}_{old}) + \lambda\mathbf{I}$ is non-singular and a positive λ ensures that $\mathbf{S}(\mathbf{w}_{old}) + \lambda\mathbf{I}$ is always positive definite. λ can also be varied after each iteration to achieve fast convergence as obtained by the steepest gradient algorithm when the cost function shown in (3.14) is far from its minimum. Around the minimum value of the cost function, for small λ , the quadratic approximation of the SSE ensures that the cost function given by (3.14) quickly converges.

$$F(\mathbf{w}) = 0.5 * \lambda \|\mathbf{w} - \mathbf{w}_o\|^2 + E(\mathbf{w}) \quad (3.14)[18]$$

Here, $F(\mathbf{w})$ is the cost function which is minimized by Levenberg-Marquardt optimization and $\mathbf{w}_o$ is the vector of initial weights. The cost function resulting from Levenberg-Marquardt (L-M) optimization is identical to the SSE except for the first term in (3.14) which introduces additional error due to the λ term and the initial weight vector. Weight initialization possesses a randomness which manifests in the cost function. Said randomness is limited by using the Nguyen-Widrow (N-W) initialization algorithm which utilizes a uniform random distribution to spread the bias and weight vectors of all neurons evenly across the desired output range [21]. As λ ensures that the Hessian approximation is non-singular and positive definite, the minimum λ depends on the input signal and neural network architecture.

3.2 Proposed training method

The standard L-M method does not impose any restrictions on the weight vector thus $|\mathbf{w}|$ can go to infinity or become quite large depending on the function to be approximated. This property is not problematic for behavioral modeling. However, in practical implementations

where $|\mathbf{w}|$ is limited by circuit constraints such as supply rails in analog devices or number of bits in digital, an optimization method is desired which can constrain $|\mathbf{w}|$ to a hard limit with negligible performance degradation. This section describes such a modification which is a novel contribution of this thesis. The proposed alteration to the standard L-M optimization algorithm is in the modification of $\mathbf{w}$, $e(\mathbf{x}_i, \mathbf{w})$ and $\mathbf{J}(\mathbf{w})$ for a maximum $\mathbf{w}$. Equation (3.4) assumes that $|\mathbf{w}| \leq \infty$ as is the case with any behavioral neural implementation. Practical implementations however have a maximum physical value based on circuit restrictions (w_{max}) which the behavioral synaptic weight can exceed for a given application. By limiting the behavioral synaptic weights to $|w_q| \leq w_{max}$ (with $q = 1, 2, \dots, c$) so as to mimic a practical implementation, the Jacobian matrix becomes undefined for $|w_q| \geq w_{max}$. The proposed method specifies a restriction scheme for $|w_q| \geq w_{max}$ and redefines the Jacobian matrix for $|\mathbf{w}| \leq \infty$ despite the $|w_q| \leq w_{max}$ restriction. The new error function as shown in (3.15) is identical to the previous in (3.4) for $|\mathbf{w}| \leq \mathbf{w}_{max}$ where $\mathbf{w}_{max} = [w_{1_{max}}, w_{2_{max}}, \dots, w_{c_{max}}]$. The vector $\mathbf{w}_{max}$ is a collection of the maximum physical values of each synaptic weight in the neural network. However, we can assume that the same implementation is used to realize all synaptic weights in the network hence, $w_{1_{max}} = w_{2_{max}} = \dots = w_{c_{max}} = w_{max}$. For $|\mathbf{w}| > \mathbf{w}_{max}$, $\mathbf{w}$ is replaced with $\mathbf{w}_{clip}$ which is identical to $\mathbf{w}$ except for indices (q) where w_q exceeds $w_{q_{max}}$. Equation (3.16) is an example of $\mathbf{w}_{clip}$ where $|w_c| \geq w_{c_{max}}$. As more synaptic weights exceed their corresponding maximum values, $\mathbf{w}_{clip}$ is updated accordingly.

$$e(\mathbf{x}_n, \mathbf{w}) = d(\mathbf{x}_n) - y(\mathbf{x}_n, \mathbf{w}), |\mathbf{w}| \leq \mathbf{w}_{max}$$

$$e(\mathbf{x}_n, \mathbf{w}) = d(\mathbf{x}_n) - y(\mathbf{x}_n, \mathbf{w}_{clip}), |\mathbf{w}| > \mathbf{w}_{max} \quad (3.15)$$

$$\mathbf{w}_{clip} = [w_1, w_2, \dots, \text{sgn}(w_c) \cdot w_{c_max}] \quad (3.16)$$

Based on the example in (3.16), where $|w_c| \geq w_{c_max}$, the error for $|\mathbf{w}| > \mathbf{w}_{max}$ as obtained from (3.15) is constant with respect to w_c implying that:

$$\frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_c} = \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_c} = \dots = \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_c} = 0 \text{ for } |w_c| \geq w_{c_max}. \quad (3.17)$$

The new Jacobian matrix becomes:

$$J(\mathbf{w}) = \begin{bmatrix} \frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_1, \mathbf{w})}{\partial w_2} & \dots & 0 \\ \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_2, \mathbf{w})}{\partial w_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_1} & \frac{\partial e(\mathbf{x}_M, \mathbf{w})}{\partial w_2} & \dots & 0 \end{bmatrix} \quad (3.18)$$

Again, the above analysis is only subject to a single weight (w_c) exceeding its maximum value. As more weights reach their respective thresholds, the Jacobian matrix needs to be updated dynamically.

The rate of convergence to a final solution determines the training speed of the algorithm and this section examines the convergence properties of the L-M method to determine if any benefits can be achieved from the proposed modification. For the sake of simplicity, we assume that the SSE in (3.4) requires a scalar weight (w) as opposed to a vector of weights ($\mathbf{w}$). This reduces the Newton weight update expression in (3.8) to a single dimension as shown in (3.19), where E' and E'' are first and second order derivatives respectively with respect to w .

$$w_{new} = w_{old} - \frac{E'(w_{old})}{E''(w_{old})} \quad (3.19)$$

Convergence is expressed as a reduction in the error between the weight at which the SSE is minimum (w_{opt}) and the weights at successive iterations. The errors at successive iterations are given in (3.20) and the resulting convergence expression in (3.21) is obtained by substituting (3.19) in (3.20) and executing a second order Taylor series expansion.

$$\epsilon_{new} = w_{new} - w_{opt}, \epsilon_{old} = w_{old} - w_{opt}. \quad (3.20)$$

$$\epsilon_{new} = \epsilon_{old} - \frac{E'(w_{opt} + \epsilon_{old})}{E''(w_{opt} + \epsilon_{old})} \approx \epsilon_{old} - \frac{E'(w_{opt}) + \epsilon_{old} \cdot E''(w_{opt}) + \epsilon_{old}^2 \cdot \frac{E'''(w_{opt})}{2}}{E''(w_{opt}) + \epsilon_{old} \cdot E'''(w_{opt}) + \epsilon_{old}^2 \cdot \frac{E''''(w_{opt})}{2}} \quad (3.21)$$

Given that $E'(w_{opt}) = 0$ for an assumed quadratic SSE function about w_{opt} , (3.21) reduces to (3.22).

$$\epsilon_{new} \approx \frac{\frac{\epsilon_{old}^2}{2} (E'''(w_{opt}) + \epsilon_{old} \cdot E''''(w_{opt}))}{E''(w_{opt}) + \epsilon_{old} \cdot E'''(w_{opt}) + \epsilon_{old}^2 \cdot \frac{E''''(w_{opt})}{2}} \quad (3.22)$$

As $\epsilon_{old} \rightarrow 0$, $w_{opt} \rightarrow w_{old}$, and ϵ_{new} reduces to the expression in (3.23).

$$\epsilon_{new} \approx \frac{\epsilon_{old}^2}{2} \cdot \frac{E'''(w_{old})}{E''(w_{old})} \quad (3.23)$$

From (3.23), the quadratic rate of convergence is $\frac{1}{2} \cdot \frac{E'''(w_{old})}{E''(w_{old})}$ implying that a smaller

$E''(w_{old})$ increases the rate of convergence. For multiple dimensions where $\mathbf{w}_{old}$ is a vector representing the weights at a given iteration, $E''(w_{old})$ becomes the Hessian matrix in (3.7) or its L-M approximation ($\mathbf{S}$) in (3.11). Our proposed modification of the Jacobian matrix in (3.18) reduces the determinant of $\mathbf{S} = \mathbf{J}^T \mathbf{J}$ for $|\mathbf{w}| > \mathbf{w}_{max}$ and consequently increases the

rate of convergence. We thus expect to see convergence times with an underlying proportionality to w_{max} because as w_{max} reduces, the probability of weight clipping increases leading to a lower determinant of the Hessian matrix approximation and consequently a higher rate of convergence.

Figure 3.4 is a flow chart depicting the basic description of our Levenberg-Marquardt implementation. The proposed modification to the norm is highlighted in yellow shade.

As the N-W initialization algorithm (described in [21]) has a random component, different initial weights are obtained each time the algorithm is called. This is undesirable for demonstration of the proposed algorithm's efficacy as both speed of convergence and training accuracy depend on the initial set of weights. The randomness in the N-W algorithm per training sequence is removed by using the same seed for random number generation at the beginning of each training sequence. This ensures that the same set of initial weights are obtained for a given neural network configuration.

The smaller loop in Figure 3.4 which executes the gradient descent portion of the Levenberg-Marquardt method iterates until the mean squared error (MSE) decreases or until λ reaches its maximum value. The larger loop in Figure 3.4, which involves calculation and modification of the cost function and Jacobian matrices respectively, is performed for a fixed number of iterations or until there is a divergence between the mean-squared-error of the training and validation data. The latter stopping criterion is obtained by dividing the available data sequence into training and validation datasets. The training dataset is used to compute weight updates thus its mean-squared-error (MSE) always decreases with consecutive

iterations. The MSE of the validation data is based on weight updates from the training data. As a result the validation MSE decreases if the weight update improves the neural network model of the desired non-linearity and increases if the weight update improves the neural network model-fit to noise fluctuations in the training data. The optimum set of weights is obtained when the respective MSEs for the training and validation data diverge over 6 consecutive iterations thus signaling a break to the outer loop.

Figure 3.4: Flow chart describing Levenberg-Marquardt algorithm

Previously investigated methods of restricted network training with finite gain weights are [22]-[24]. Reference [22] normalizes the network weights to a predetermined maximum

value after each iteration, this leads to convergence issues based on the choice of maximum weight. In [23], analog circuits are designed to yield a gain of 5 times the activation function range, a method which becomes invalid for a dynamic network with multiple neurons as gain variation increases with non-linearity and bandwidth. Finally, [24] optimizes the neural network one weight at a time which is an inefficient training method for large networks.

3.3 Theoretical basis for non-linear modeling

A neural network is a very powerful tool to use for non-linear device modeling because of its inherent non-linearity as a result of its activation function. The additional ability of the network to use tapped delay lines as inputs also gives the neural model the ability to incorporate the mixed memory and non-linear effects exhibited by a power amplifier. With sufficient degrees of freedom (number of weights), a neural network can approximate a Volterra series or any convergent non-linearity with memory to a high degree of accuracy. This is based on the proof in [25] which shows that any non-linearity can be modeled by a linear combination of other specific non-linearities.

Reference [25] proves mathematically that standard multilayer feed-forward networks (input layer, one hidden layer and an output layer) can approximate any measurable function to any desired degree of accuracy. In addition, radial basis function (RBF) networks are subject to the Universal Approximation Theorem. Radial basis function networks are feed-forward networks with radial basis functions as activation functions. They are identical to the dynamic networks previously examined with the added specification of a radial basis function for activation.

The Universal Approximation Theorem as stated by [18] proves that any RBF network of the form $F(x) = \sum_{i=1}^{m_1} w_i G(\frac{x-t_i}{\sigma})$ can successfully approximate continuous functions provided the RBF function is of the format $G(x) = e^{-\frac{\|x-t_i\|^2}{2\sigma^2}}$. Where t_i is the center (or bias) of the i th neuron, m_1 is the number of neurons in the network, w_i is the synaptic weight corresponding to the i th neuron, and σ is the width of each neuron.

The radial basis functions used in [18] for activation are of the form: $G(x) = e^{-\frac{\|x-t_i\|^2}{2\sigma^2}}$, but other useful radial basis functions exist which are not exponentials such as all the activation functions stated in (3.2). Also, [18] requires that σ should be the same for all neurons (σ is constant across $F(x)$) which is an unnecessary constraint to the universal approximation theorem. Finally, the universal approximation theorem stated by [18] requires that

$\int_{t_i-a}^{t_i+a} G(x)dx \neq 0$ which is not valid for the sigmoid function given by: $\varphi(x) = \frac{2}{1+e^{-2x}} - 1$.

Alternatively, the universal approximation theorem as proven by George Cybenko in [26] removes the aforementioned constraints by proving universal approximation for a sigmoid activation function with $\int_{-a}^a \varphi(x)dx = 0$. Based on the universal approximation proofs in [18] and [26] for Gaussian and sigmoid functions, universal approximation with neural networks can be extended to all activation functions which are bounded, continuous, and differentiable.

3.4 Chapter Summary

This chapter describes the structure of a neuron which is the basic building block for multilayer perceptron neural networks. Next, basic neural network architectures are

examined as well as key optimization algorithms for function approximation or system identification. Consequently, a proposed optimization algorithm for weight restricted network training (a key contribution in this thesis) based on the Levenberg-Marquardt method is described. The proposed method incorporates weight limitations into the standard L-M equations thus ensuring a minimal effect on the accuracy of the trained neural network. Weight limitation is also theoretically shown to cause a reduction in convergence or training time. Finally, this chapter provides a theoretical justification for using neural networks as function approximators based on the universal approximation theory.

Chapter 4

System Inversion and Pre-distortion

4.1 Neural Network Pre-distorter Architecture

So far, we have shown that any memoryless non-linearity can be modeled with a two layer feed-forward RBF network to any desired finite error. Dynamic non-linearities such as those exhibited by power amplifiers with wideband stimuli require dynamic neural networks for accurate modeling. The accuracy of static non-linear modeling depends on the number of hidden layer neurons while dynamic non-linear characteristics depend on a combination of memory depth and hidden layer neurons. Neural network memory is obtained by placing tapped delay elements at the network inputs or outputs. With more than two hidden layers, the position of delay elements in the architecture becomes a lot more flexible. This thesis only examines two layer structures as they have a proven theoretical basis for function approximation and provide a good tradeoff between complexity and performance. Dynamic non-linear modeling with neural networks thus requires empirical determination of two parameters – memory depth and number of hidden layer neurons. For a memoryless non-linear model, there is only one degree of freedom – the number of hidden neurons required to achieve a minimum error. This makes a dynamic neural network significantly more difficult to train and accounts for additional error compared to static modeling. Another reason for error in the dynamic neural network is that the impulse response of the neural network filter is generated using tapped delay lines (discrete memory) while the power amplifier exhibits continuous memory. With a significant number of delay taps and a high sampling rate, the

discrete impulse response could be made to closely approximate the continuous impulse response but there will always be a residual error which cannot be mitigated by model training.

A pre-distorter can be incorporated into the receiver chain as shown by Figure 4.1. The pre-distorter in Figure 4.1 uses the quadrature signals from a complex baseband modulation scheme as inputs to generate the pre-distorted quadrature signals given by $I_p(t)$ and $Q_p(t)$.

Figure 4.1: Transceiver chain with in-phase/quadrature pre-distorter

The pre-distorter in Figure 4.1 also dynamically updates its weights according to an optimization algorithm to compensate for long term memory effects in the PA thus realizing adaptive pre-distortion. The DSP portion of the adaptive dynamic neural network (DNN) computes and updates the new weights using the Levenberg-Marquardt optimization algorithm while the analog portion of the adaptive DNN realizes the mathematical pre-

distortion function with short-term memory. Short term memory compensation is realized in the pre-distorter by using a feed-forward dynamic neural network (FFDNN) for linearization. A two-layer FFDNN is shown in Figure 4.2 and the dynamic linearization capability of the FFDNN is obtained by the inclusion of tapped delay elements in the input layer. The input layer of the FFDNN accepts source data as quadrature inputs and applies the instantaneous and delayed versions of each input to the hidden layer. The hidden layer is comprised of nonlinear neurons which are labeled as ‘NN’ with a detailed functional depiction which is also shown in Figure 4.2.

Figure 4.2: Block diagram of FFDNN (on left) and composition of 'LN' or 'NN' (on right)

Each nonlinear node performs a linear combination of its inputs before application to an activation function. Finally, the output layer is comprised of linear neurons ‘LN’ which perform linear combinations of all non-linear neuron outputs. ‘LN’ has the same composition as ‘NN’ in Figure 4.2 minus the activation function. The input layer outputs the vector $\mathbf{x}_n$ in (4.1) where N is the memory depth of external inputs given by I_{in} and Q_{in} , and n is the sampling index.

$$\mathbf{x}_n = [I_{in}(n), I_{in}(n-1), \dots, I_{in}(n-N), Q_{in}(n), Q_{in}(n-1), \dots, Q_{in}(n-N)]^T \quad (4.1)$$

The instantaneous output of the k th neuron in the hidden layer is shown in (4.2) where $f(.)$ is the activation function in (4.3), x_{ni} is a scalar representing the value of $\mathbf{x}_n$ at index i , w_{ki} as shown in Figure 4.2 is the weight value at the i th input to the k th neuron, M is the memory depth of the input layer, and b_k is the nodal bias.

$$v_k(n) = f\left(\sum_{i=1}^{2(N+1)} w_{ki} x_{ni} + b_k\right) \quad (4.2)$$

$$f(x) = \frac{-0.144}{1+e^{(-12.2x+1.31)}} - \frac{0.713}{1+e^{(12.6x)}} - \frac{0.144}{1+e^{(-12.2x-1.31)}} + 0.5 \quad (4.3)$$

The transfer function and gradient of the activation function in (4.3) are show in Figure 4.3 to meet all the requirements for an activation function which are symmetry, non-linearity and differentiability. The expression in (4.3) is obtained from an activation circuit designed in Cadence with a $0.25\mu m$ BiCMOS process and its derivation is described in the next chapter. The activation function ranges from -0.2 to 0.2 which is also the range of the external inputs and outputs of the FFDNN (I_{in} , Q_{in} , I_p and Q_p). The output layer's expression is given by (4.4) where P is the number of nodes in the hidden layer.

$$I_p(n) = \sum_{k=1}^P w_{(P+1)k} v_k(n) + b_{P+1}, Q_p(n) = \sum_{k=1}^P w_{(P+2)k} v_k(n) + b_{P+2}. \quad (4.4)$$

Figure 4.3: Transfer function and derivative of activation function.

4.2 Measurement Setup

This subsection describes the measurement apparatus used to characterize the device under test (DUT). In order to test the efficacy of the neural network as a pre-distorter as well as the validity of the proposed training algorithm, we measure the dynamic non-linear characteristics of a class-AB power amplifier (PA) and use it to generate a hardware compatible neural network pre-distorter model. The neural network pre-distorter is obtained by executing the following sequenced steps with the measurement apparatus shown in Figure 4.4.

1. Generate a wideband modulation signal to trigger the memory effects of the device under test (DUT)

2. Obtain the baseband non-linear data from the output of the DUT- system identification.
3. Use the input and output data from steps one and two respectively to train a two-layer FFDNN to mimic the inverse non-linearity of the DUT – system inversion.
4. Use the trained FFDNN to compute pre-distorted data before transmission to the DUT – pre-distortion.

Figure 4.4: Measurement setup for characterizing the DUT

The chosen modulation signal is a 4-carrier wideband code division multiple access (4C-WCDMA) signal which possesses sufficient bandwidth (20MHz) to stimulate dynamic non-linearity in the DUT. Details on the WCDMA composition can be viewed in [27]. The 4C-

WCDMA source data (I_s, Q_s in Figure 4.4) is generated in ADS using the testbench described in [28] in accordance to 3GPP specifications [1]. The source data is then transferred to Matlab for subsequent processing.

The next step involves characterization of the DUT which is realized in Figure 4.4 by connecting both switches to terminal (a). The source data (I_s, Q_s) is then transferred to the memory of a vector signal generator (ESG, Agilent E4438C) using Matlab's instrument control toolbox. The ESG is a transmitter which samples digital data from its memory, converts the digital signal to an analog, and up-converts the baseband analog signal to a radio frequency of 1.9 GHz. The resulting RF signal is transmitted to the DUT through a pre-amplifier for power amplification. On the receive side, the power spectrum analyzer (PSA) down-converts, digitizes and samples the attenuated output of the DUT into its memory. The amplified non-linear quadrature signals (I_{a1}, Q_{a1}) are transferred to Matlab to train the FFDNN, obtain the AM-AM and AM-PM functions (as shown in Fig. 4.5), and calculate the normalized mean squared error (NMSE) and adjacent channel power ratio (ACPR). Both the PSA and ESG operate with the same reference clock for carrier and clock synchronization of the transmitter and receiver. The trigger input is set on the first sample from the ESG and it signals the PSA to begin capturing data.

During pre-distortion, both switches are connected to terminal (b) where the pre-distorted signals (I_p, Q_p) are computed from the source data using the trained FFDNN. The received data (I_a, Q_a) is expected to be linear for successful pre-distortion.

The DUT is a class-AB PA made by PHOENIX (PA1109C) with a 1dB output compression point of 33dBm and 22dB of gain at 1.9GHz. The supply voltage is 10V with a DC bias current of 0.7A. The peak power added efficiency (PAE) is 30% at 3dB gain compression and 25% at the 1.5dB compression point. The chosen operating compression point is 1.5dB to yield a good trade-off between pre-distorter complexity and performance.

In order to generate a linear drive signal which compresses the class-AB PA by 1.5dB, we include a pre-amplifier after the ESG. The pre-amplifier linearly increases the DUT's input power to acceptable levels for distortion. The pre-amplifier is a class-A PA made by Mini-Circuits (ZRL-3500+) with a 1dB output compression point of 24.6dBm and 21dB of gain at 1.9GHz. The supply voltage is 12V with a DC bias current of 0.46A. The required power level of the pre-amplifier in order to drive the DUT into 1.5dB of gain compression is 12.5dBm at the output or -8.5dBm at the input. This value is 11.5dB below its 1dB output compression point implying that the pre-amplifier does not contribute any significant amount of static distortion to the transceiver chain.

The output power from the DUT exceeds the maximum input power specification of the power spectrum analyzer (PSA, Agilent E4440A) hence the need for the attenuator shown in Fig. 4.4. A picture of the pre-amplifier, power amplifier and attenuator combination is shown in the Appendix under Fig. A1.

Fig. 4.5 shows the measured AM-AM and AM-PM responses respectively of the DUT for one slot of a WCDMA frame (666.7 us) sampled at 100MHz. The scatter points in Figure 4.5

show a 40dB variation in gain compression due to memory effects and an underlying 1.5dB gain compression due to non-linearity.

Figure 4.5: AM-AM and AM-PM plots of non-linear DUT

The AM-PM scatter plot shows over 200 degrees of phase variation due to memory effects and the static non-linearity of the phase error in Figure 6 is overshadowed by its memory-caused variation.

4.3 System Inversion

This section describes the training procedure and results obtained from the generation of the inverse PA FFDNN model using both the standard and proposed L-M algorithms. The efficacy of the proposed training method is verified by modeling the inverse non-linearity of the DUT with a FFDNN. The performance metrics for training accuracy of the FFDNN are

the ACPR and NMSE. If we assume that the dynamic non-linearity of the class-AB PA is invertible, the inverse non-linearity can be obtained by applying the measured non-linear output of the PA (I_{a1}, Q_{a1}) to the input of the FFDNN and iterating according to the standard and modified L-M methods to obtain the desired linear signal (I_s, Q_s) at the output of the FFDNN. The mathematical definitions of ACPR and NMSE are given in (4.5) and (4.6) respectively.

$$\text{NMSE}(d, y) = \frac{\sum_{n=1}^M (d(x_n) - y(x_n, \mathbf{w}))^2}{\sum_{n=1}^M (d(x_n))^2} \quad (4.5)$$

$$\text{ACPR}(Y) = \frac{\int_{f_o - BW}^{f_o + BW} |Y(f)|^2 df}{\int_{BW}^{BW} |Y(f)|^2 df} \quad (4.6)$$

In (4.6), f_o is the carrier offset, and BW is the single sided integration bandwidth. We now have all the information required for training namely: the FFDNN architecture, the measured data obtained from the DUT, and the activation function given by (4.3). The training dataset consists of 67K samples (one WCDMA frame at 100Mz sampling) all of which are used to compute the neural weight update per iteration. This type of training which utilizes all the training samples to compute new weights at each iteration is called Batch training and further details on its operation can be found in [29]-[32].

The training inputs to the L-M algorithm consists of the complex input signal given by $I_{in}(n) + jQ_{in}(n)$ in Figure 4.2, and the desired output signal given by $d(\mathbf{x}_n)$. The L-M training output is the output of the FFDNN in Figure 4.2 given by $y(\mathbf{x}_n, \mathbf{w}) = I_p(n) + jQ_p(n)$. When training the FFDNN to model the inverse non-linearity (system inversion) of

the DUT, the training inputs are obtained from the measurement setup in Fig. 4.4 where

$$d(\mathbf{x}_n) = I_s(n) + jQ_s(n) \text{ and } I_{in}(n) + jQ_{in}(n) = I_{a1}(n) + jQ_{a1}(n).$$

The first step involves using unrestricted learning (standard L-M method) to find the optimum FFDNN configuration (number of input delays, number of hidden neurons). This is done by varying the number of delays in the input layer and the number of hidden neurons to achieve the best performance. The optimum configuration is then trained using the proposed weight-restricted method and the performance is compared with the unrestricted algorithm. The performance criteria are the NMSE, ACPR, power consumption (or number of hidden neurons) and number of iterations required for convergence.

Table 4.1 shows the unrestricted training performance of the network for all combinations of 0, 2 and 4 input delays with 11, 13 and 15 neurons in the hidden layer. The ACPR and NMSE in Tables 4.1 and 4.2 are not obtained from pre-distortion but from the inverse model of the PA as approximated by a FFDNN. The NMSE column in Tables 4.1-4.2 showing the system inversion model of the PA is obtained from (4.5). The ACPR columns in Tables 4.1-4.2 are obtained from (4.6) where $Y = \mathcal{F}\{y(\mathbf{x}_n, \mathbf{w})\}$ with $\mathcal{F}\{\cdot\}$ representing a Fourier transform. The “# of iterations” column in Table 4.1 is the iteration index for which the respective NMSEs for the training and validation datasets diverge over 6 consecutive iterations. The “# of iterations” column is an indication of the speed with which the unrestricted L-M algorithm converges to a final solution. The comparison in Table 4.1 is used to choose the configuration which yields the best tradeoff between NMSE, ACPR, “# of iterations”, and complexity or power consumption.

The optimum configuration which optimizes both NMSE and ACPR is a network with 4 input delays and 15 hidden neurons (“4-15”). However, said network is slow to converge and consumes the most power.

Table 4.1: Unrestricted training performance for various delay and hidden layer configurations.

# of Delays	# of Hidden Neurons	NMSE (dB)	ACPR@ ± 5 MHz offset (dBc)		ACPR@ ± 10 MHz offset (dBc)		# of Iterations
			Low	High	Low	High	
0	11	-19.2	48.3	47.3	49.0	49.0	>2000
2	11	-41.0	47.8	47.2	48.7	48.9	386
4	11	-42.6	49.3	49.2	50.5	50.6	701
0	13	-19.2	44.7	44.1	45.6	45.4	33
2	13	-41.9	48.9	48.4	49.5	49.6	654
4	13	-42.9	50.2	50.1	50.8	51.1	439
0	15	-19.2	49	48.2	49.7	49.9	294
2	15	-42.9	50.5	50.4	50.7	51.4	551
4	15	-43.3	50.5	50.4	51.3	51.6	2047

The network with 4 input delays and 11 hidden neurons (“4-11”) converges 2.9 times faster than the former for a worst case degradation in ACPR and NMSE of 1.2dB and 0.7dB respectively. Its power consumption is also 27% lower thus making it the ideal choice for pre-distortion.

Table II shows the performance variation with the maximum weight parameter (w_{max}) using the proposed training method for both the “4-11” and “4-15” neural configurations. In the “4-11” configuration, with the exception of $w_{max} = 7.5$, convergence time increases with the maximum weight as expected. Performance degradation for both NMSE and ACPR is <0.2 dB for $w_{max} \geq 7.5$. This property is especially useful in analog synapses because of

their constant gain-bandwidth product. Reductions in gain by a factor of 2 can effectively double the bandwidth with negligible loss in performance due to the proposed method.

Convergence time reduction is not substantial (10%) for $w_{max} \geq 7.5$ in the “4-11” configuration.

The “4-15” configuration reduces convergence time by 57% with $<0.4\text{dB}$ reduction in ACPR and no degradation in NMSE based on a comparison between $w_{max} = 1.6$ and $w_{max} \geq 4.7$ as shown in Table 4.2.

Table 4.2: Weight limited training performance for various maximum weights.

Max. Weight (w_{max})	# of Delays	# of Hidden Neurons	NMSE (dB)	ACPR@ $\pm 5\text{MHz}$ offset (dBc)		ACPR@ $\pm 10\text{MHz}$ offset (dBc)		# of Iterations
				Low	High	Low	High	
2.0	4	11	-41.4	48.1	47.8	49.3	49.4	457
3.0	4	11	-42.0	48.5	48.4	49.7	49.9	578
5.0	4	11	-42.3	49.0	48.8	50.1	50.3	648
7.5	4	11	-42.4	49.2	49.0	50.3	50.5	632
10.0	4	11	-42.5	49.3	49.1	50.4	50.6	703
≥ 15.1	4	11	-42.6	49.3	49.2	50.5	50.6	701
1.6	4	15	-43.3	50.5	50.1	51.3	51.2	884
2.0	4	15	-42.2	49.1	48.7	50.1	49.8	877
2.5	4	15	-42.6	49.2	48.7	50.4	50.1	1686
3.0	4	15	-42.2	49.2	48.7	50.3	50.1	1728
3.5	4	15	-42.6	49.3	48.8	50.4	50.2	1776
4.1	4	15	-43.3	50.5	50.4	51.3	51.6	2040
≥ 4.7	4	15	-43.3	50.5	50.4	51.3	51.6	2047

The aforementioned observations lead us to conclude that speed improvement due to the proposed method is more pronounced in larger networks. While the improved speed due to the proposed method is a favorable by-product, it is not the primary objective which is to

ensure the minimum performance degradation for variations in w_{max} . Said objective has been achieved for the chosen “4-11” configuration by observing a worst case reduction in ACPR and NMSE of 1.4dB and 1.2dB respectively for maximum weights (w_{max}) ranging from 2 to 15.

4.4 Behavioral Pre-distortion

This subsection examines the ability of the inverse PA model obtained from neural network training to pre-distort the DUT. The optimum FFDNN in terms of power consumption, ACPR, NMSE and convergence time as derived from training is the “4-11” configuration with $w_{max} = 7.5$. This optimum FFDNN is used to compute the pre-distorted data (I_p, Q_p) from the source data (I_s, Q_s) before application to the DUT as shown in the measurement setup of Figure 4.4. The baseband output of the DUT (I_a, Q_a) is then used to compute the ACPR and error vector magnitude (EVM) thus verifying the efficacy of the weight limited FFDNN as a pre-distorter.

The linearized AM-AM and AM-PM plots are shown in Figure 4.6. The linearized gain variation for mid-range to large signal reduces from 10dB (Figure 4.5) to 0.5dB (Figure 4.6) due to linearization. Phase error variation for large signals also reduces from 80 degrees (Figure 4.5) to 3 degrees (Fig. 4.6).

EVM, as given by (4.7), is used in addition to ACPR as a linearization performance metric.

Table 4.3 shows the EVM performance of the ideal, non-linear and linearized DUT.

$$EVM_{\%}(y) = \sqrt{NMSE(d, y)} * 100 \quad (4.7)$$

$$PAPR = \left(\frac{\max_n |d(x_n)|}{\frac{1}{M} \sum_{n=1}^M |d(x_n)|} \right)^2 \quad (4.8)$$

Figure 4.6: AM-AM and AM-PM plots of linearized DUT

Crest factor reduction (CFR) is used to reduce the peak to average power ratio (PAPR), given by (4.8), of the 4-carrier WCDMA signal from 12.5dB to 8.5dB while keeping the ACPR levels larger than 59.4dB as shown in the first row of Table 4.3. Further details on CFR techniques can be viewed in [33]-[35]. The PA's non-linearity increases the EVM by 8.9% while pre-distortion increases the EVM by only 0.2% as shown in Table 4.3. Table 4.3 also shows the ACPR values for non-linear and linearized signals at carrier offsets (f_o) of ± 5 MHz and ± 10 MHz. Based on Table 4.3, the best and worst case ACPR improvement ratios due to linearization at the 5MHz offset are 15.6dB and 14.5dB respectively. The 10MHz offset shows best and worst case ACPR improvement ratios of 15dB and 13.5dB respectively.

Table 4.3: Measured behavioral ACPR and EVM results of WCDMA signal

	ACPR@ ± 5 MHz offset (dBc)		ACPR@ ± 10 MHz offset (dBc)		EVM (%)
	Low	High	Low	High	
Ideal	59.6	59.5	59.4	59.5	2.9
Non-Linear	37.2	36.9	39.5	39.0	11.8
Linearized	52.8	51.4	54.5	52.5	3.1

A depiction of the ideal, non-linear and linearized spectra as obtained from the power spectrum analyzer (PSA) is shown in Figure 4.7 to demonstrate the linearization improvement in adjacent bands.

Figure 4.7: Measured spectrum of 4-carrier WCDMA signal

4.5 Chapter Summary

This chapter verifies the learning and convergence properties of the proposed algorithm which addresses the weight restrictions found in practical synaptic weight implementations. It does so by measuring the dynamic non-linear characteristics of a class-AB PA and uses the measured data to train an inverse PA model with a FFDNN architecture. The optimum FFDNN configuration as obtained by standard L-M training yields the best trade-off between power consumption, accuracy and speed. The proposed L-M training method ensures that the chosen FFDNN weights do not exceed a prescribed limit while maintaining the training accuracy. The optimized network obtained using the proposed algorithm compared to the standard has been shown to converge in 90% of the time with negligible performance degradation. Convergence time is reduced even further to 47% using the proposed method for larger neural networks. Furthermore, the maximum synaptic weight is shown to be an additional parameter for optimizing performance in terms of convergence time, ACPR and NMSE. Finally the trained FFDNN obtained from the proposed method is used to pre-distort the class-AB PA over a 20MHz bandwidth. The resulting improvements in the ACPR range from 13.5dB to 15.6dB and the EVM reduces by 8.7%.

Chapter 5

Analog Circuit Design of Neural Network Components

Our thesis objective is to implement efficient pre-distorter components for low power amplifiers using neural networks. As neural network ASIC implementations are still uncommon, it is difficult to find a common basis for a comparison between digital and analog implementations especially in terms of power consumptions. For instance the work carried out in [36] compares an analog neuron to an FPGA digital implementation to show a power reduction by a factor of 2.6 in the analog neuron. However, the analog neuron is realized in $0.35\mu\text{m}$ CMOS while the FPGA is realized in 90nm CMOS. In addition the comparison of an ASIC to an FPGA neglects the power benefit that will be realized by the implementation of a digital ASIC. However, the fact that an analog neuron realization in a technology which is 3.9 times larger than the digital implementation can still achieve less power is a compelling enough reason to indicate that an analog neural network pre-distorter could be more efficient compared to a digital. After the implementation of our analog neural network components, we will revisit the power comparison with a digital ASIC neuron implemented by Siemens while taking into account the process technology and accuracy (number of bits).

The FFDNN architecture is comprised of three building blocks namely the delay elements, the non-linear neuron and the linear neuron. The non-linear neuron is further comprised of synaptic weights, an adder, and an activation function. The linear neuron possesses the same components as the non-linear neuron except for the activation function. All three building

blocks will be implemented in this Chapter. Due to size constraints, the entire pre-distorter cannot be realized on the same IC thus a tapped delay line, the non-linear neuron, and linear neuron designs are realized on separate ICs to demonstrate their functionality.

The properties of the trained neural network obtained in Chapter 4 are used to obtain design specifications for the analog circuits described in this Chapter. Table 5.1 is a summary of the neural network properties which are relevant to circuit design.

Table 5.1: Parameters of trained neural network

Neural Network	Two-layer MLP, Neurons: 11-2, Memory depth:4
Input data rate	10Mbps
Unit delay	10ns
Num. of inputs per non-linear/linear neuron (NN/LN)	10/11
Abs. max. synaptic weight	2-15 V/V
Bias range	+/- 0.2 V
Input range	+/- 0.2 V
Output range	+/- 0.2 V

5.1 Basic Building Blocks

This sub-section describes the properties of the fundamental components used to realize the delay elements and neurons. All designs are carried out in Cadence using a $0.25\ \mu\text{m}$ BiCMOS provided by NXP. Figure 5.1 shows the Early-voltage derivation for an npn-HBT transistor with a width of $0.4\mu\text{m}$, length of $1.5\mu\text{m}$, and 4 emitters ($0.4 \times 1.5 \times 4$). The Early voltage which is the x-intercept in a collector-current (I_C) vs collector-emitter voltage (V_{CE}) dc simulation can be determined by the ratio of the y-intercept to the slope. Figure 5.1 shows the Early voltage computed in the aforementioned manner at each bias current to yield an

average Early voltage (V_A) of 164.85V. The resulting output impedance (r_o) for a 0.5mA bias is 329.7k Ω from $r_o = \frac{V_A}{I_C}$.

The transconductance (g_{mh}) of the same HBT transistor is shown in Figure 5.2 with the plot of $\frac{g_{mh}}{I_C}$ against collector-current (I_C). It is also worthwhile to note that $\frac{g_{mh}}{I_C} = \frac{1}{V_T}$ where V_T is the thermal voltage. From Figure 5.2, the g_{mh} at a bias current of 0.5mA is 17.61mA/V.

Figure 5.1: Early-Voltage simulation of HBT transistor

Figure 5.2: Transconductance simulation of HBT transistor

HBT transistors are preferred to MOS transistors in this process because of their smaller output capacitances and higher current density. As a consequence they enable operation at much higher frequencies compared to MOS transistors. Unfortunately, no pnp-HBT transistors are available in this process thus necessitating the use of PMOS transistors for current sourcing. The PMOS transistor used in subsequent designs has a width of 10 μ m, length of 0.25 μ m, and multiplicity of 10 ($10 \times 0.25 \times 10$). This is the largest available PMOS transistor which can support the currents drawn by the HBT transistors. Figure 5.3 shows the output impedance derivation of the aforementioned PMOS transistor by taking the slope of the curve in a drain current (I_D) vs drain-source voltage (V_{DS}) simulation. The output impedance (r_{DS}) at 0.5mA in the saturation region is 10k Ω .

Figure 5.3: Output impedance simulation of PMOS transistor

Figure 5.4 shows the transconductance (g_{mp}) of the same PMOS transistor with the plot of g_{mp} against drain-current (I_D).

Figure 5.4: Transconductance of PMOS transistor

From Figure 5.4, the g_{mp} at a bias current of 0.5mA is 4.14mA/V which is lower than that of the HBT by a factor of 4.25.

The next building block which uses the characterized transistors is the operational amplifier which has been implemented in both single ended and fully differential configurations as shown in Figure 5.5 and 5.6 respectively. Both operational amplifiers use a folded cascode configuration whose properties are examined in detail in [37]. The common mode feedback block (CMFB) in Fig. 5.6 ensures a constant reconfigurable bias at its differential input terminals. The CMFB provides a constant bias by comparing the average voltage of its differential inputs to a fixed bias and adjusting the current sources attached to the corresponding terminals to compensate for any deviation. A detailed description of its operation is shown in [37]. The frequency response of both operational amplifiers can be assumed to have the first order transfer function given by (5.1).

$$\frac{V_o}{V_i} = \frac{sC_L g_{mh} R_c R_o + g_{mh} R_o}{sC_L (R_o + R_c) + 1} \quad (5.1)$$

Where $R_o \approx \frac{g_{mp} r_{DS}^2}{2} || r_o$, $C_L = C_{s1}$ or $2 * C_s$, and $R_c = R_{s1}$ or R_s . Note that: $V_i = V_{i+} - V_{i-}$, $V_o = V_{o+} - V_{o-}$ in subsequent diagrams and expressions.

The upper limit of V_{b1} is $V_{DD} - (V_{GS} + V_{eff}) = 1.55$ where V_{GS} is the gate source voltage which is 0.71V at 0.5mA, and $V_{eff} = V_{GS} - V_t = 2 * \frac{I_D}{g_m} = 0.24$ [37]. The ratio $\frac{I_D}{g_m}$ can be obtained from Figure 5.4 and V_t is the threshold voltage of the PMOS transistor.

The small signal gain is given by $g_{mh} R_o = 67.2dB$ from (5.1) where R_o is on the order of:

$$\frac{g_{mp} r_{DS}^2}{2} || r_o = 215.4k || 329.7k = 130.3k\Omega.$$

Single-End Folded Cascode Op-amp (Gs)

Figure 5.5: Single-Ended folded cascode operational amplifier

Figure 5.6: Fully differential folded cascode operational amplifier

The desired outcome of the frequency response given by (5.1) is to have the pole

$(\frac{1}{C_L(R_o+R_c)} \approx \frac{1}{C_LR_o})$ dominate the transfer function until just after the unity gain frequency

given by $w_t = \frac{g_m h}{C_L}$. At $1.2 \cdot w_t$, the zero given by $\frac{1}{C_LR_c}$ becomes active thus improving the

phase margin by cancelling any second order poles in the amplifier which seek to decrease

the phase of the transfer function to 180° around w_t and make the device unstable. If

$$C_L = 2pF, \text{ then } f_t = \frac{w_t}{2\pi} = 1.4GHz \text{ and } R_c = \frac{1}{1.2 \cdot w_t C_L} = 47.3\Omega.$$

The calculated values of R_c and V_{b1} are optimized in the layout to yield the following parameters corresponding to the schematics shown in Figures 5.5 and 5.6.

$$V_{DD} = 2.5V, V_{SS} = 0, V_{b1} = 1.1V, V_c = 1.25V, I_b = 2 \cdot I_B = 1mA, V_c = 1.25V, R_{s1} = 58.8\Omega, C_{s1} = 2pF, R_s = 88.9\Omega, C_s = 1.1pF.$$

Figures 5.7 and 5.8 show the ac-simulations of the single-ended and fully differential operational amplifiers (op-amp) respectively for their extracted layouts. The single-ended and differential op-amps show small signal gains of 64.5dB and 68.4dB respectively which vary by less than 2.7dB from the calculated value. Also the unity gain frequencies of 1.4GHz and 1.7GHz for the single-ended and differential op-amps respectively are quite close to the calculated value. Finally, the phase margins which are 93.8° for the single-ended and 79.6° for the differential op-amp show that both op-amps will be stable.

The final component characterization required in the design of the delay elements is the varactor. The BiCMOS process has two differential varactors labelled as “vacapDNMOSNW” and “vacapdb” both of which are reverse biased differential NMOS diodes except that “vacapdb” has a SiGe polysilicon contact at the anode. As a consequence “vacapdb” has a steeper tuning curve compared to “vacapDNMOSNW” thus making “vacapDNMOSNW” more suitable for our purposes. The chosen “vacapDNMOSNW” device has a width of $3.525\mu m$, length of $1.0\mu m$, and a multiplicity of 8 ($3.525 \times 1.0 \times 8$). Its capacitance and quality factor as a function of control voltage are shown in Figure 5.9.

Figure 5.7: Frequency and phase response for single-ended op-amp

Figure 5.8: Frequency and phase response for fully differential op-amp

Figure 5.9: Capacitance and quality factor variation of varactor with tuning voltage

The capacitance of the varactor varies from 87.6fF to 195.7fF with a nominal value of 155.2fF. The varactor also has a finite quality factor which ranges from 1125 to 2400. The finite quality factor indicates a parasitic series resistance of 149Ω at a resonance frequency of 5MHz given a nominal control voltage.

5.2 Analog tapped delay line

Since data in the vector signal generator is sampled at 100MHz, the delay element is required to provide 10ns of group delay over the 10MHz single-sided base-bandwidth. From the neural network properties in Table 5.1, the delay element should also have a linear unity gain over an input ranging from -0.2V to 0.2V. Due to the wideband nature of the signal and baseband operation of the pre-distorter, active delay elements are used to realize the group

delay such that IC real estate is minimized. The transfer function of a time delay and its rational approximation as determined by Pade are shown in (5.2) [38].

$$H(s) = e^{-\tau s} \approx \frac{1 - k_1 s + k_2 s^2 + \dots \pm k_n s^n}{1 + k_1 s + k_2 s^2 + \dots + k_n s^n} \quad (5.2)$$

Where τ is the desired group delay.

Based on (5.2) and as observed in Figure 5.10, a minimum of two orders are required to achieve the required group delay over a 10MHz bandwidth. The transfer function coefficients of a 2nd order approximation of (5.2) are $k_1 = \frac{\tau}{2}$ and $k_2 = \frac{\tau^2}{12}$ leading to the expression shown in (5.3). Equation (5.3) is an all-pass transfer function implying a gain of 0dB throughout its entire frequency range. Figure 5.10 shows the transfer function of the group delay based on (5.3) for $\tau = 10ns$. The group delay reduces by less than 10% of its initial value within a 33MHz bandwidth. This implies that a second order Pade approximation is only useful for frequencies below $\frac{1}{3\tau}$ and higher order approximations are required for wider bandwidths.

$$H(s) = \frac{1 - \frac{\tau}{2}s + \frac{\tau^2}{12}s^2}{1 + \frac{\tau}{2}s + \frac{\tau^2}{12}s^2} \quad (5.3)$$

The transfer function given by (5.3) can be realized with a single operational amplifier configuration shown in Figure 5.11. Analysis of the circuit in Figure 5.11 can be done by converting the Pi network consisting of R_1 , R_2 , and C_V to the T network in Figure 5.12 with impedances Z_a , Z_b , and Z_{in} . The impedance relationships of the T network to the Pi network are given in (5.4).

Figure 5.10: Magnitude and Group Delay response of 2nd order Pade approximation

$$Z_a = Z_b = \frac{R_2}{sR_2C_v + 2}, Z_{in} = \frac{s^2R_1R_2C_v^2 + s2R_1C_v + 1}{s^2R_2C_v^2 + s2C_v} \quad (5.4)$$

Performing nodal analysis at the T junction of Figure 5.3 yields the transfer function shown in (5.5).

$$\frac{V_o}{V_i} = \frac{R_4}{R_3 + R_4} \cdot \left(1 + \frac{Z_a}{Z_{in}}\right) - \frac{Z_a}{Z_{in}} \quad (5.5)$$

Substituting the actual components given by (5.4) into (5.5) yields the transfer function in (5.6).

$$\frac{V_o}{V_i} = \frac{\frac{R_4}{R_3 + R_4} \left(s^2R_1R_2C_v^2 + s \left(2R_1 - \frac{R_2R_3}{R_4} \right) C_v + 1 \right)}{s^2R_1R_2C_v^2 + s2R_1C_v + 1} \quad (5.6)$$

Figure 5.11: Second order filter with Pi network

Figure 5.12: Second order filter with T network

In order for (5.6) and (5.3) to be equivalent, the following conditions must hold:

$R_3 = 3 \cdot R_4$ and $R_1 = 0.75 \cdot R_2$. The result is the filter response given in (5.7)

$$\frac{V_o}{V_i} = \frac{0.25 \cdot (s^2 0.75 R_2^2 C_v^2 - s \cdot 1.5 R_2 C_v + 1)}{s^2 0.75 R_2^2 C_v^2 + s \cdot 1.5 R_2 C_v + 1} \quad (5.7)$$

Equation (5.7) is now a second order approximation of a delay element. The only problem with (5.7) is an attenuation factor of 0.25 by each delay element. This problem is further exacerbated when the delay cells are cascaded to yield a tapped delay line. To eliminate the attenuation and isolate the output from any load impedance, a buffered gain block is placed at the output of the delay filter as shown in Figure 5.13. A further consequence of $R_3 = 3 \cdot R_4$ is that the DC bias at the positive terminals of amplifiers Gs1 and Gs2 are 4 times smaller than the common mode voltage of Ga1 provided terminal V_b is at ground. This problem is fixed by setting V_b to the common mode voltage of Ga1 which is also the bias required for op-amp operation in Gs1 and Gs2.

Figure 5.13: Schematic of single delay element

The capacitor given by C_v is a varactor in order to allow for tunable delays. This way, process variations during fabrication which influence the group delay can be compensated for. In addition, tunable delays allow the resulting pre-distorter to be compatible with different modulation formats which operate at different sampling frequencies. The varactors

supported by the $0.25\mu m$ BiCMOS process available to us have their capacitive control at the input terminal V_i shown in Figure 5.11. As a result, the input bias voltage controls the group delay. For cascaded delay elements, said input bias will be the output from previous stages as well as the input to synaptic weights in the non-linear neuron. As such, it needs to be kept constant or isolated from varactor tuning. This is achieved by another feedback-operational amplifier at the input of the delay element as shown in Figure 5.13. This way, the varactor can be tuned by varying the common mode voltage of the preceding differential amplifier (V_{tune}) from 0.3V to 2.2V without affecting the bias at the differential input terminals - V_i in Fig. 5.13. The complete differential delay element (as shown in Figure 5.13) uses two single ended delay elements (shown in Figure 5.11) to realize differential operation for better linearity by cancellation of second order non-linear products. A fully differential delay element also provides an output signal which is compatible with the differential input of the synaptic weight in the non-linear node.

By comparing (5.5) and (5.6), $\tau = 3R_2C_v$ implying $R_2 = 21.48k\Omega \rightarrow R_1 = 0.75 * R_2 = 16.1k\Omega$ for $\tau = 10ns$ at the nominal C_v of 155.2fF as shown in Figure 5.9. The resistors R_4 and R_i can be arbitrarily set to $5k\Omega$ implying $R_3 = 3 * R_4 = 15k\Omega$ and $R_f = 4 * R_i = 20k\Omega$. Finally the common-mode voltage of Gal is half the supply voltage of 2.5V implying $V_b = 1.25V$.

The number of delay elements required for pre-distortion as derived from Chapter 4 is four. Figure 5.14 shows the layout of four cascaded active delay cells where the output of each delay element is available to yield an analog tapped delay line. The calculated components

assume an open-circuit load which is not the case for the first three delay elements. In addition, they do not incorporate parasitics encountered in real circuit implementation such as the finite quality factor of the varactors. For the aforementioned reasons, the calculated circuit components in Figure 5.13 were tuned in the layout to the following values:

$R_3 = 15k\Omega, R_1 = 12.5k\Omega \rightarrow R_2 = 16.67k\Omega, R_{41} = 4.16k\Omega, R_{42} = 3.89k\Omega, R_{43} = R_{44} = 4.19k\Omega, R_i = 5k\Omega, R_{f1} = 24.89k\Omega, R_{f2} = 25.81k\Omega, R_{f3} = 24.45k\Omega, R_{f4} = 24.25k\Omega$ where $R_{41}, R_{42}, R_{43}, R_{44}$ correspond to the resistor R_4 shown in the schematic for delay elements 1, 2, 3 and 4 respectively. The resistors R_{f1} to R_{f4} also correspond to the resistor R_f in the schematic for delay elements 1 to 4 respectively.

Figure 5.14: Layout of four cascaded delay elements

Design verification for the layout in Figure 5.14 is done by simulating the group delay, gain and linearity at each tap for the extracted layout circuit. The magnitude-frequency response of the delay line at the nominal and tuning range limits for the varactor are shown in Figures 5.15-5.17.

Figure 5.15: Transfer function gain of delay line at $V_{\text{tune}} = 2.2\text{V}$

Figure 5.16: Transfer function gain of delay line at $V_{\text{tune}} = 1.25\text{V}$

Figure 5.17: Transfer function gain of delay line at $V_{tune} = 0.3V$

From Figures 5.16, where V_{tune} is nominal, the gain is unity up to about 30MHz after which peaking occurs until the -3dB frequency of 316.2MHz. This is also identical to the simulated gain-bandwidth of an ideal second order Pade transfer function as shown in Figure 5.10. At the tuning limits, the gain is unity up to about 10MHz after which it ripples until the -3dB frequency of about 330MHz. Consequently, to maintain a unity gain over all tuning values, we must limit the usable bandwidth to 10MHz. Figures 5.18-5.20 show the group delay transfer function of all four taps at the nominal and tuning limits. The group delay of the delay line is calculated by taking the derivative of the phase of its transfer function with respect to angular frequency. Figures 5.18-5.20 show the expected relationship between successive delays which is that the 2nd, 3rd, and 4th delays are integer multiples of the first delay. Also, varactor tuning allows for delays ranging from 7.5ns to 12.2ns and consequently sampling rates ranging from 82MHz to 133MHz.

Figure 5.18: Transfer function of group delay for delay line at $V_{\text{tune}} = 2.2\text{V}$

Figure 5.19: Transfer function of group delay for delay line at $V_{\text{tune}} = 1.25\text{V}$

Figure 5.20: Transfer function of group delay for delay line at $V_{\text{tune}} = 0.3\text{V}$

The simulated group delay variation at the nominal V_{tune} is 0.15ns which is higher than the group-delay variation of 0.02ns given by the ideal transfer function. The aforementioned discrepancy occurs due to non-idealities in circuit implementation such as a finite quality factor in the tuning element, finite gain in the operational amplifier, and parasitic components (capacitors, inductors) in on-chip resistors especially those connected to the tuning element. The greatest delay variation at any given tuning voltage occurs at the 4th delay because its transfer function has the highest order. Consequently, worst case delay variations at tuning voltages of 2.3V , 1.25V , and 0.3V are 0.3% , 1.1% , and 2.4% respectively of their respective delays. Delay variation is thus inversely proportional to the quality factor of the varactor as can be observed from Figure 5.9.

The final performance metric for the analog delay line is its linearity which is shown in Table 5.2 for all four delay elements. Data in the gain error column is calculated using

(5.8). Since V_{tune} is equivalent to the common mode bias or operating point of the differential signal, a nominal V_{tune} which corresponds to an operating point which is midway between the supply and ground should yield the optimum linearity.

$$G_e = \frac{G(V_i=0) - \max(G(V_i=\pm 0.2))}{G(V_i=0)} * 100 \quad (5.8)$$

As the supply limits are approached, the PMOS and HBT transistors enter the triode and saturation regions respectively thereby deteriorating the linearity. The data in Table 5.2 agrees with the linearity prediction as the gain error is minimum at the nominal V_{tune} of 1.25V and worsens as the tuning limits are approached.

Table 5.2: Gain variation for analog tapped delay line

$V_{\text{tune}}(\text{V})$	Delay 1		Delay 2		Delay 3		Delay 4	
	Gain (V/V)	Gain Err. (%)	Gain (V/V)	Gain Err. (%)	Gain (V/V)	Gain Err. (%)	Gain (V/V)	Gain Err. (%)
2.2	1.0	0.28	0.98	0.6	0.97	0.87	0.96	1.1
1.25	1.01	0.021	1.01	0.027	1.01	0.036	1.01	0.048
0.3	1.01	0.34	1.01	0.87	1.01	1.37	1.01	1.66

Alternate analog delay structures existing in the literature are given in [39]-[40]. Both references contain circuits that exhibit second order transfer functions which do not follow Pade's approximation. As a result, the circuit-bandwidths relative to their group delays are not optimal. For instance, [39] describes a tunable delay circuit with a group delay of 3.75ns and a constant gain-bandwidth of 30MHz. A Pade approximated circuit such as the implementation described in this Thesis has a constant gain-bandwidth of $\approx 1/(3\tau)$ which equals 89MHz for the same group delay. The circuit in [40] describes a fixed delay circuit

with a group delay of 80ps and a constant gain-bandwidth of 1GHz. The Pade implementation would achieve up to 4.2GHz of bandwidth.

5.3 Analog non-linear neuron

The analog neuron is comprised of synaptic weights, an activation circuit and an adder. This section describes transistor level circuits corresponding to the synaptic weight and activation function. We also derive transfer functions describing performance based on the aforementioned circuitry.

5.3.1 Synaptic weight

Synaptic weights have been realized with memresistors, CMOS Gilbert cells, and low power CMOS current mode multipliers as described in [41]-[46], [47], and [48]-[49] respectively. Reliable memristor structures are still being investigated and established large signal models are nonexistent at this time. Low power CMOS current mode multipliers have low output currents (order of nA) thus requiring loads in the order of $M\Omega$ to yield the high output voltage necessary to drive the power amplifier. Such high load impedances are detrimental to the bandwidth. The high non-linearity of CMOS Gilbert cell multipliers is apparent in [36] and [47] from the 25% non-linearity at 0.3V/V of weight gain for an input ranging from -6% to +6% of the supply rails.

The synaptic weight implementation of choice is based on a bipolar Gilbert cell multiplier which is realized in Cadence with a 0.25 μ m BiCMOS process from NXP. Bipolar Gilbert cell transconductors have a linear large signal multiplication relationship as shown in (1) thus making them well suited for synaptic weights over a wide range of control values. We do not

incorporate memory (or weight storage) in our multipliers as the proposed algorithm is demonstrated using off-line training to yield control values which can then be applied off-chip using external memory on a printed circuit board (PCB).

The design specifications as obtained from Table 5.1 requires the synaptic weight to have a maximum linear gain between 2V/V and 15V/V for an input ranging from -0.2V to 0.2V.

The rule of thumb for predistortion is a bandwidth which is 5 times the input data-rate of 10Msps hence the desired bandwidth of 50MHz for the synaptic weight.

A typical gain-cell transconductor as described in [37] has the transfer function in (1) where I_1 and I_2 are bias currents to the input and output stages respectively. V_{in} is the differential input voltage, R_E is the emitter degeneration resistor and I_o is the differential output current. Based on (5.9), a typical gain-cell transconductor can only achieve two quadrant multiplication as both I_1 and I_2 are restricted to positive values. In addition, given that I_2 is the output bias current, low values of I_2 will give a smaller bandwidth compared to higher I_2 . To combat the aforementioned effects, neural weighting is done using the fully differential four quadrant voltage multiplier in Fig. 5.21.

$$I_o = \frac{V_{in}I_2}{R_E I_1}. \quad (5.9)$$

Using the Ebers-Moll model to approximate the current-voltage relationship of a bipolar transistor, the large signal transfer function of the schematic in Fig. 5.21 is derived in (5.10)-(5.15) with the restrictive assumption that $\max(V_x) < 0.5 * I_{b1} * R_{e1}$ where variables in (5.10)-(5.15) are shown in Fig. 5.21. Note that:

$V_x = V_{x+} - V_{x-}, V_y = V_{y+} - V_{y-}, V_o = V_{o+} - V_{o-}$ in successive expressions. Equation (5.10) is the differential output voltage of the input stage from voltage terminals V_{e1} and V_{e2} , where V_T is the thermal voltage of the bipolar transistor, I_{b1} is the bias current to the input stage and V_x is the differential input.

$$\Delta v = V_{e2} - V_{e1} = V_T \ln \left(\frac{I_{b1} + V_x/R_{e1}}{I_{b1} - V_x/R_{e1}} \right). \quad (5.10)$$

Figure 5.21: Schematic of synaptic weight

The collector current expressions of all output transistors connected to the load (R_L) as a function of the differential input V_x are listed in (5.11)-(5.12).

$$I_{c3} = \frac{I_2}{1 + e^{(\Delta v/V_T)}} = I_2 + \frac{V_x I_2}{R_{e1} I_{b1}}. \quad (5.11)$$

$$I_{c4} = I_2 - \frac{V_x I_2}{R_{e1} I_{b1}}, I_{c5} = I_3 + \frac{V_x I_3}{R_{e1} I_{b1}}, I_{c6} = I_3 - \frac{V_x I_3}{R_{e1} I_{b1}}. \quad (5.12)$$

From (5.11)-(5.12), it is apparent that output current is dependent on control currents I_2 and I_3 as shown in Fig. 5.21. In order to ensure voltage multiplication at the output, I_2 and I_3 are obtained from a separate transconductor with V_y as its input. The output expression of the second transconductor is given by (5.13) where R_{e2} is its emitter degeneration resistor.

Equation (5.13) allows the control currents I_2 and I_3 to be expressed as a function of a separate differential input voltage V_y .

$$I_2 - I_3 = \frac{V_y}{R_{e2}}. \quad (5.13)$$

The final output expression as derived in (5.14) calculates the differential current (I_a) at the load (R_L). Equation (5.14) is initially expressed in terms of the collector currents connected to it before substitution of (5.13) into the resulting expression. Equation (5.15) calculates the differential output voltage (V_o) by multiplying I_a by the load resistor.

$$I_a = (I_{c3} + I_{c6}) - (I_{c4} + I_{c5}) = \frac{V_x(I_2 - I_3)}{R_{e1}I_{b1}} = \frac{V_x V_y}{R_{e1}R_{e2}I_{b1}}. \quad (5.14)$$

$$V_a = I_a * R_L = \frac{V_x V_y R_L}{R_{e1}R_{e2}I_{b1}}. \quad (5.15)$$

A comparison of (5.14) and (5.9) shows that the differential inputs in (5.15), V_x and V_y , can have both positive and negative values allowing for 4 quadrant multiplication whereas the scaling factor given by I_2 in (5.9) can only have positive values thus limiting the performance of a simple gain cell to two quadrant multiplication. From (5.15), the proposed circuit in Fig. 5.21 will yield 4 quadrant multiplication for large signal inputs provided $|V_x| < 0.5 * I_{b1} * R_{e1}$. Gain-control and bandwidth for Fig. 5.21 are shown in Fig. 5.22-5.23 given the following design parameters: $R_L = 4 \text{ k}\Omega$, $I_{b2} = 0.37 \text{ mA}$, $I_{b1} = 0.32 \text{ mA}$, $R_{e2} = 2R_{e1} = 5 \text{ k}\Omega$, $V_{DD} = 2.5 \text{ V}$, and $V_{SS} = 0 \text{ V}$. The maximum input voltage range given by $|V_x| < 0.5 * I_{b1} * R_{e1} = 0.4 \text{ V}$ is twice the desired value of 0.2 V .

The adder combines the outputs from all ten multipliers (given by Fig. 5.21) as well as an external differential offset. From Table 5.1, the desired offset applied through the adder should range from -0.2V to 0.2V. The adder circuit consists of a simple feedback operational amplifier configuration as shown in Fig. 5.22 where $R_i = 5\text{ k}\Omega$, $R_f = 17.5\text{ k}\Omega$ and V_b is the external differential offset. The complete equation describing the relationship between the inputs to the multiplier (V_x, V_y) and the output of the adder (V_o) is given in (5.16).

$$V_o = \frac{V_{x1}V_{y1}R_L R_f}{R_{e1}R_{e2}I_{b1}R_i} + \dots + \frac{V_{x10}V_{y10}R_L R_f}{R_{e1}R_{e2}I_{b1}R_i} + V_b \cdot \frac{R_f}{R_i} \quad (5.16)$$

Figure 5.22: Schematic of multipliers and adder combination in non-linear neuron

Subsequent simulations for gain, linearity and bandwidth of the synaptic weights incorporate the effect of the adder by observing the response at the output of the adder (V_o) to a stimulus at the input of a single multiplier. Consequently, further statements about the gain, linearity and bandwidth of the synaptic weights actually correspond to the weight and adder combination. Fig. 5.23 shows the simulated output (V_o) against input (V_x) voltage for various control voltages (V_y). The slope or gain increases with control voltage with the maximum

slope occurring at $V_y = \pm 1$. Fig. 5.24 shows the simulated unloaded frequency response of the multiplier for various values of V_y . The frequency response of the synaptic weight shows a 3dB-bandwidth of approximately 50MHz across all control values with negligible gain variation below 12.5MHz.

Figure 5.23: Simulated output voltage of NN for control voltage (V_y) ranging from $\pm 1V$.

Figure 5.24: Simulated frequency response of NN multiplier for V_y ranging from 0-1V.

Summarily, the analog multiplier design in Fig. 5.21 is a synaptic weight with 50MHz of bandwidth and gain values which range from -3.3V/V to 3.3V/V for inputs within $\pm 0.2V$. An examination of its linearity and gain with respect to control voltage (V_y) is done in the next Chapter.

5.3.2 Activation function circuit

A precise activation function was required in Chapter 4 to determine the design specifications for other neural components such as the synaptic weight range and output voltage range. As a result the activation circuit was realized prior to training in Chapter 4 and the neural network properties in Table 5.1 are a consequence of the circuit configuration in Fig. 5.25.

The activation function circuit is customarily implemented with a differential pair. A differential pair of bipolar transistors is optimal for obtaining an activation function due to the exponential relationship which exists between base voltage and collector current in bipolar transistors. However, simple differential pairs in the 0.25 μm BiCMOS process have steep gradients and in order to regulate the slope of the activation function, a transconductor was applied to the input. The complete activation circuit as shown in Fig. 5.25 consists of a cascaded non-linear transconductor and a differential pair. The non-linearity of the transconductor is regulated by the emitter degeneration resistor given by R_e and a voltage gain transfer function is obtained by applying the load resistor R_{L1} to the output of the transconductor.

The transconductor used in Fig. 5.25 is a standard configuration and a detailed derivation of its transconductance is shown in [36] to yield the linear voltage gain approximation in (5.17). Note that: $V_i = V_{i+} - V_{i-}$, V_T is the thermal voltage of bipolar transistors in the process, and I_{e1} and I_{e2} are emitter currents shown in Fig. 5.25.

$$G_v = \frac{R_{L1}}{R_e + r_{e1} + r_{e2}}, \text{ where } r_{e1} = \frac{V_T}{I_{e1}}, r_{e2} = \frac{V_T}{I_{e2}} \quad (5.17)$$

Figure 5.25: Schematic of activation function circuit

The gain expression in (5.17) is non-linear as it depends on the emitter currents of the input differential pair. The desired non-linearity can be iteratively determined by varying the bias current (I_{b1}) or R_e for a fixed input voltage range. The output voltage of the transconductor is $G_v \times V_i$ which is then substituted in the large signal expression of a simple differential pair to yield the expression in (5.18). Where R_L is the output load and I_{b2} is the bias current of the differential pair.

$$V_o = \frac{I_{b2} R_L}{2} \left(\frac{1}{e^{\frac{G_v V_i}{V_T} + 1}} - \frac{1}{e^{\frac{-G_v V_i}{V_T} + 1}} \right) \quad (5.18)$$

The expression in (5.18) meets all the requirements for an activation function which are symmetry, non-linearity and differentiability. However, since the BiCMOS process uses

Hetero-Bipolar Junction transistors (HBT), the Ebers-Moll transistor model is not accurate enough for our purposes. Hence the need for a finer approximation as given in (5.19).

$$V_o = f(V_i) = \frac{-0.144}{1+e^{(-12.2V_i+1.31)}} - \frac{0.713}{1+e^{(12.6V_i)}} - \frac{0.144}{1+e^{(-12.2V_i-1.31)}} + 0.5 \quad (5.19)$$

Equation (5.19) is a function approximation of the transfer function shown in Fig. 5.26, where Fig. 5.26 is obtained from the simulated DC sweep in Cadence for the activation circuit in Fig. 5.25 given the following parameters: $R_L = 200 \Omega$, $R_{L1} = 2.5 k\Omega$, $R_e = 5 k\Omega$, $I_{b1} = 0.2 \text{ mA}$, $V_T = 26 \text{ mV}$, and $I_{b2} = 2 \text{ mA}$. Function approximation is done by deducing from (5.18) that V_o is a linear combination of non-linear expressions of the form: $\frac{a}{1+e^{(b \times V_i + c)}}$, where a , b , and c are unknown variables. The final approximation in (5.19) is iteratively determined by using Levenberg-Marquardt optimization to calculate the unknown variables as well as the number of non-linear expressions required for a fine approximation.

Figure 5.26: Transfer function and derivative of analog activation function.

According to Fig. 5.26, the output of the activation function is limited to $\pm 0.2V$ which is the same as the linear input range of the synaptic weight circuit. This property is important because it allows the same synaptic weight circuit to be used for both input and output neurons. Fig. 5.26 also shows the gradient of the activation function whose slope gradually decreases from 1.6 V/V to 0.05 V/V for an input ranging from 0 V to 0.3V. The activation input range (-0.3 V to 0.3 V) is the range for which the gradient decreases to approximately zero. This value cannot be theoretically determined prior to training as the input of the activation function is the sum of scaled delayed quadrature inputs (I_{in} and Q_{in}). Thus, we empirically observed that an activation range choice of $\pm 0.3V$ kept the maximum trained behavioral synaptic weight within an order of magnitude to that obtained from the maximum analog synaptic weight (3.3 V/V). We also found through multiple iterations that transfer functions with abrupt or steep slopes and consequently high gradients increase the ratio between the maximum and minimum synaptic weights of a network during training. On the other hand, gradual slopes reduce the performance of the network due to insufficient non-linearity. Consequently, a gradient which gradually changes within the chosen activation input range (as shown in Fig. 5.26) allows a good tradeoff between synaptic weight range and network performance. The simulated transfer function of the activation circuit has an unloaded 3dB-bandwidth of 340MHz with $<0.1\text{dB}$ gain variation below 60MHz as shown in Fig. 5.24.

5.4 Analog linear neuron

The linear neuron as shown in Fig. 4.2 is a linear combination of all non-linear neuron outputs. Given that our pre-distortion application requires 11 non-linear neurons, each linear

neuron contains 11 synaptic weights. Also, the output of the linear neuron is applied to a mixer with an input impedance of 600Ω hence the need for a buffer at the output of the adder to isolate the gain of the operational amplifier from its load. Other design parameters such as the synaptic weight range, input range and bandwidth are identical to those of the non-linear neuron. The synaptic weight circuit is implemented using the same schematic as the non-linear neuron which is shown in Fig. 5.21 given the following parameters: $R_L = 4\text{ k}\Omega$, $I_{b2} = 0.37\text{ mA}$, $I_{b1} = 0.32\text{ mA}$, $R_{e2} = 10\text{ k}\Omega$, $R_{e1} = 7.7\text{ k}\Omega$, $V_{DD} = 2.5\text{ V}$, and $V_{SS} = 0\text{ V}$. The schematic of the adder in the linear neuron implementations contains a few differences compared to the non-linear neuron as shown in Fig. 5.27.

Figure 5.27: Schematic of multipliers and adder combination in linear neuron

The operational amplifier used in negative feedback is now single ended as per the input requirement of the mixer and the adder parameters are as follows: $R_i = 5\text{ k}\Omega$, $R_f = 53.7\text{ k}\Omega$ and $I_b = 1\text{ mA}$. The higher feedback resistor - R_f ($53.7\text{ k}\Omega$ to $17.5\text{ k}\Omega$) in the linear neuron compared to the non-linear is expected to reduce the bandwidth of the linear neuron by a

factor of 3.07 ($\frac{57.3}{17.5}$). The simulated unloaded frequency response of a synaptic weight in the linear neuron is shown in Fig. 5.28 for various control voltages. The 3dB-bandwidth is approximately 12.6MHz across all control values and it is less than the 3dB-bandwidth of the non-linear neuron by a factor of 3.97. The difference in the realized and expected ratios is due to the higher R_e (7.7 k Ω to 2.5 k Ω) in the linear neuron compared to the non-linear. Fig. 5.29 shows the simulated output (V_o) against input (V_x) voltage for various control voltages (V_y). The slope or gain increases with control voltage and the maximum slope occurs at $V_y = \pm 2$.

Summarily, the synaptic weights in linear neurons have 12.6MHz of bandwidth and gain values which range from -3.4V/V to 3.4V/V for inputs within ± 0.2 V. An examination of their linearity and gain with respect to control voltage - V_y is done in the next Chapter.

Figure 5.28: Simulated frequency response of LN multiplier for V_y ranging from 0-2V.

Figure 5.29: Simulated output voltage of LN for control voltage (V_y) ranging from $\pm 2V$.

5.5 Circuit Predistortion

The aforementioned circuit implementation of analog delays, synaptic weights, and an activation function have introduced non-idealities such as bandwidth limitations in magnitude and group delay, non-linearity, and input offsets respectively. These non-idealities have an adverse effect on predistortion and this section quantifies the reduction in predistorter efficacy compared to a completely behavioral predistorter. The circuit predistorter model also gives a better approximation of predistortion gains to be expected from a hardware implementation. Section 4.4 describes the hardware linearization of the DUT using the behavioral models (in Matlab) of the neural network predistorter. This section also achieves hardware predistortion by using the aforementioned circuit models (in Cadence) of a complete neural network. The same measurement sequence used for

behavioral linearization (in Figure 4.4) is used for circuit linearization except that the predistorter is realized in Cadence instead of Matlab.

The ACPR and EVM results for the Ideal, Non-Linear and Linearized cases which are replicated in Table 5.3 have already been examined in Section 4.4. The last row shows the predistortion results based on the described circuit models in Section 5.1-5.4 of this Chapter. The EVM remains unchanged compared to the behavioral predistorter (3.1%) while the best and worst case ACPR improvement ratios at the 5MHz offset are 10.3dB and 10dB respectively. The 10MHz offset shows best and worst case ACPR improvement ratios of 12.2dB and 11.2dB respectively.

Table 5.3: Measured circuit ACPR and EVM results of WCDMA signal

	ACPR@ ± 5 MHz offset (dBc)		ACPR@ ± 10 MHz offset (dBc)		EVM (%)
	Low	High	Low	High	
Ideal	59.6	59.5	59.4	59.5	2.9
Non-Linear	37.2	36.9	39.5	39.0	11.8
Linearized	52.8	51.4	54.5	52.5	3.1
Linear. Cir.	47.5	46.8	51.7	50.2	3.1

A depiction of the ideal, non-linear, behavioral, and circuit linearized spectra as obtained from the power spectrum analyzer (PSA) is shown in Figure 5.30 to demonstrate the linearization improvement in adjacent bands.

Figure 5.30: Measured spectrum of 4-carrier WCDMA signal with circuit predistortion.

5.6 Chapter Summary

This chapter describes the analog circuits used in the realization of the dynamic neural network whose training was described in Chapter 4. The neural network components consisting of active delay lines, linear neurons, and non-linear neurons are implemented at a transistor level using a $0.25\mu\text{m}$ BiCMOS process in Cadence. The active delay lines are realized using operational amplifiers in a feedback configuration which yields a second order transfer function for a unit delay according to the Pade approximation. The resulting circuit is a 4-tap delay line with a tunable unit delay which ranges from 7.5ns to 12.3ns over a 10MHz bandwidth. Furthermore, the achieved gain per delay element is unity with a worst case variation of 1.37% over the input range of $\pm 0.2\text{V}$. Both linear and non-linear neurons comprise of synaptic weights and an adder. The synaptic weights are realized using a 4-

quadrant bipolar Gilbert cell multiplier for optimum linearity while the adders are implemented using op-amps with negative feedback. The synaptic weights and adder combination have tunable gains which range from -3.4V/V to $+3.4\text{V/V}$ and 3dB-bandwidths of 12.6MHz and 50MHz for linear and non-linear neurons respectively. The non-linear neuron also contains an activation circuit whose transfer function is anti-symmetric and differentiable over a 340MHz bandwidth. A linearity examination of synaptic weights in both linear and non-linear neurons is done in the next chapter.

Finally, this chapter quantifies the effect of circuit non-idealities on predistortion to get a better approximation of projected hardware performance based on the chosen circuit implementation. Based on the circuit model, the ACPR has a worst case reduction of 5dB compared to the behavioral model while the EVM remains the same.

Chapter 6

Analog Circuit Measurement of Neural Network Components

This Chapter describes the results obtained from the measurement of the linear and non-linear neuron circuits as well as the active delay line. The measurement apparatus consists of a signal generator (Agilent's E4438C), the device under test, an oscilloscope (Agilent's DSO8104A), and a network analyzer (Agilent's E5071B).

6.1 Analog non-linear neuron

The integrated circuit layout of the non-linear neuron consisting of 10 multipliers, an adder and an activation circuit is shown in Fig. 6.1. The pad-limited silicon occupancy is 2.5mm by 2.5mm. The pad-limitation is due to off-chip application of control voltages to the synaptic weights on a PCB using external memory or resistor dividers for non-reconfigurable applications. The first PCB used to test the fabricated non-linear neuron is shown in Fig. A2 in the Appendix, however the circuit oscillates due to inappropriate terminations at the input and output terminals. The final PCB revision is shown in Fig. 6.2. Its composition is fairly standard given the use of resistors to set bias voltages/currents. The devices on the PCB which had not been previously anticipated prior to testing are the 10pF capacitors at the output terminals which improve stability. The PCB tracts which are connected to the output of the op-amp adder introduce parasitics which reduce the phase margin of the op-amp thus providing conditions which are favorable for oscillation during feedback. The output capacitors reduce the open-loop gain to $<0\text{dB}$ before the phase shift is close to 180° . A detailed description of stability during measurement is given in section 6.3.

Figure 6.1: Integrated circuit layout of non-linear neuron.

Figure 6.2: PCB used to measure non-linear neuron.

6.1.1 Gain-control and linearity

This sub-section examines the linearity and gain against control voltage of the synaptic weight and adder combination in the non-linear neuron. Fig. 6.3 shows a comparison between measured and simulated differential gain $\left(G = \left|\frac{V_o}{V_x}\right|\right)$ against control voltages. Both curves closely track each other and increase linearly for $|V_y| \leq 0.8V$. This implies linear large signal multiplication for $|V_x| \leq 0.2V$ and $|V_y| \leq 0.8V$ which is a beneficial property during real-time weight adaptation. The measured gain response has a -40mV input offset compared to the simulated and its maximum value at $V_y = 1V$ is 3.8V/V compared to 3.4V/V for the simulated.

Figure 6.3: Measured and simulated gain against control voltage for non-linear neuron

Fig. 6.4 shows the gain-error, which is calculated by $\left(G_e = \frac{G(V_x=0) - \max(G(V_x=\pm 0.2))}{G(V_x=0)} * 100\right)$, with respect to the control voltage. For both simulated and measured curves, the gain-

error peaks when the control voltage approaches the noise floor of the circuit thus resulting in a lower limit on the dynamic range of the circuit. Ignoring the effect of noise at low control voltages, the maximum simulated and measured gain errors are 5.06% and 2.94% respectively. The maximum measured gain error of 2.94% implies an equivalent digital precision of greater than 5 bits as calculated by $\frac{1}{2^5} * 100 = 3.12\%$.

Figure 6.4: Measured and simulated gain-error against control voltage for non-linear neuron.

6.1.2 Frequency response

An unloaded frequency response of the PCB in Fig. 6.2 cannot be measured as the oscilloscope, cable connectors, and the PCB tracts provide parasitic capacitive loads at the output terminals. As a consequence the measured 3dB-bandwidth is considerably lower than the simulated unloaded bandwidth. Fig. 6.5 shows a comparison of both measured and simulated frequency responses for the analog neuron with a parasitic load of 35pF. The measured frequency responses are consistent in terms of bandwidth over all control voltages.

Figure 6.5: Loaded measured and simulated frequency response.

Both measured and simulated loaded 3dB-bandwidths are about 800kHz. The simulated and measured curves only differ slightly in their absolute gain values for a specific control voltage which is consistent with the Gain- V_y comparison in Fig. 6.3. The dynamic range of the oscilloscope restricts the gain measurements to frequencies below 32MHz, above which the received signals fall below the noise floor and are unmeasurable. It is important to note that non-linear neurons are only connected to other neurons implying they would not be subject to parasitic loads in a neural network implementation. Consequently, the unloaded bandwidth is the true measure of the maximum input rate that the neuron can support. While the unloaded frequency response cannot be measured, its properties can still be glimpsed from the measured loaded frequency response. For frequencies below 800 kHz, both unloaded and loaded frequency responses are the same. Above 800 kHz, the first pole due to the parasitic load kicks in and the gain drops off like a first order filter. The first order pole from the unloaded response now becomes a second order pole in the loaded transfer function

and its value is reduced based on the capacitive load. The pole which triggers a second order slope occurs at about 25MHz and 20MHz for measurement and simulation respectively, and it is an indication of the unloaded bandwidth. As a consequence, based on the limitation of our instrumentation, the most we can tell about the unloaded bandwidth of the fabricated circuit is that it is greater than 32MHz but less than 50MHz.

6.1.3 Activation function circuit

The measured and simulated transfer functions of the activation circuit are shown in Fig. 6.6. Due to IC process variations, the load resistance given by R_L in Fig. 5.25 has reduced from $200\ \Omega$ to $171\ \Omega$ leading to a different simulated transfer function from the previously shown curve in Fig. 5.26. The simulated and measured activation functions in Fig. 6.6 are closely correlated in their respective shapes, ranges and slopes due to a NMSE of -26dB. The measured activation function is not as smooth as its simulated counterpart due to noise fluctuations in the oscilloscope. Furthermore, there is an 11mV input-offset between the simulated and measured transfer functions.

The measured and simulated frequency responses of the activation circuit with a parasitic load are shown in Fig. 6.7. Both measured and simulated loaded bandwidths are about 8MHz and 9MHz respectively. Again, the 3-dB bandwidth is much smaller than the unloaded bandwidth of 340MHz as shown in Fig. 5.24 due to the 70pF parasitic load due to PCB tracts and instrumentation. The measured gain is within 1dB of the simulated in the pass-band and said gain variations can be due to input mismatches and noise fluctuations. In the cut-off region, both measured and simulated curves exhibit similar roll-off gradients.

Figure 6.6: Measured and simulated activation function.

Figure 6.7: Measured and simulated loaded frequency response of activation circuit.

6.2 Analog linear neuron

The integrated circuit layout of the linear neuron consisting of 11 multipliers and an adder is shown in Fig. 6.8. The pad-limited silicon occupancy is 2.5mm by 1.5mm. The PCB used to

test the fabricated non-linear neuron is shown in Fig. 6.9. Its composition is fairly standard given the use of resistors to set bias voltages/currents. The linear neuron chip contains a buffer at the output of the adder which maintains the open circuit gain of the op-amp for resistive loads above 500Ω . The unfortunate side-effect of the buffer is a reduction in the phase margin at the unity gain frequency given a parasitic load (due to the PCB tracts and instrumentation) at the output of the buffer. Increasing the capacitance at the output of the buffer does not improve stability as was the case for the non-linear neuron. Instead, the device becomes even more unstable as the phase margin reduces even further with greater capacitive loads. A more detailed description of op-amp stability in the presence of a buffer is done in Section 6.3. Also, ac signals from the power supply due to switching generate a 10.2MHz signal with a root mean squared voltage of 12mV at the linear neuron's output on the PCB shown in Fig. 6.9.

Figure 6.8: Integrated circuit layout of linear neuron.

Figure 6.9: PCB used to measure linear neuron.

6.2.1 Gain-control and linearity

This sub-section examines the linearity and gain tuning of the synaptic weight-adder combination in the linear neuron against control voltage. Fig. 6.10 shows a comparison between measured and simulated differential gain $\left(G = \left|\frac{V_o}{V_x}\right|\right)$ against control voltages. Both curves closely track each other and increase linearly for $|V_y| \leq 1.0V$. This implies linear large signal multiplication for $|V_x| \leq 0.2 V$ and $|V_y| \leq 1.0 V$. The measured gain response has a -80mV offset compared to the simulated and its maximum value at $V_y = +2V$ is 3.93V/V compared to 3.46V/V for the simulated.

Figure 6.10: Measured and simulated gain against control voltage for linear neuron.

Fig. 6.11 shows the gain-error with respect to the control voltage. The simulated gain-error reduces with control voltage implying large signal clipping due to transistor thresholds is the predominant cause of non-linearity. The measured gain-error has an underlying decreasing trend to a minimum at the -80mV control offset.

Figure 6.11: Measured and simulated gain-error against control voltage for linear neuron.

The underlying trend is superimposed by random variations which could be due to the 10MHz signal from the power supply as well as noise variations for small signals. The maximum simulated and measured gain errors are 5% and 6.5% respectively which are equivalent to a minimum digital precision of 4 bits.

6.2.2 Frequency response

Unlike the non-linear neuron, the loaded and unloaded 3dB-bandwidths of the linear neuron are expected to be comparable due to the presence of an on-chip buffer which isolates the chip from the PCB and measurement apparatus by having a low output impedance compared to the parasitic impedance. Fig. 6.12 shows a comparison of both measured and simulated frequency responses with a parasitic load for the linear neuron over various control voltages. The measured frequency responses are parallel to each other thus giving them the same bandwidth across control voltages.

Figure 6.12: Loaded measured and simulated frequency response of linear neuron.

The measured and simulated 3dB-bandwidths with a parasitic load are 10MHz and 12.6MHz respectively which, as expected, are close to the unloaded 3dB-bandwidth of 12.6MHz. The simulated and measured curves differ in their absolute gain values for a specific control voltage in accordance to the Gain- V_y comparison in Fig. 6.10. The transfer functions of the linear neuron are not as smooth as its non-linear counterpart within a decade above or below 10MHz. This is due to the small (10MHz) signal from power supply switching which adversely affects (reduces) the gain as the ac-sweep approaches the oscillating frequency.

6.3 Analog tapped delay line

The integrated circuit layout of the delay chip consisting of 4 cascaded delay elements was previously shown in Fig. 5.14. Its pad-limited silicon occupancy is 2.5mm by 1.0mm and the PCBs used to measure its performance are shown in Fig. 6.15 and Fig. A3 (in the Appendix). The PCB in Fig. A3 contains resistors which are used to set bias voltages/currents on the chip. The expectation is that the on-chip buffer would isolate the op-amp in feedback from parasitics on the PCB and instrumentation. Alas, that is not the case and the circuit oscillates at 500MHz with a root mean squared voltage of 0.17V. The addition of capacitors ranging from 10pF-10nF at the buffer output reduce oscillations to $\approx 12mV$. However, the capacitors does not enable normal circuit operation either implying that while the capacitors attenuate the signal at the buffer output, they are isolated from the terminal that improves stability - the buffer input. The aforementioned observations lead to the subsequent analysis of op-amp stability in the presence of a buffer.

Fig. 6.13 shows the schematic of a single ended op-amp with a buffered output. The parameters g_{mh} and g_{mp} refer to transistor tranconductances of the input differential pair and output buffer respectively.

Figure 6.13: Single-Ended folded cascode operational amplifier with buffer

The small signal gain of the buffered op-amp is approximated by (6.1) provided the output impedance of the op-amp without a buffer (which is small due to a large C_c) is much smaller than the input impedance of the buffer (which is large due to a small C_π – input capacitance of PMOS transistor).

$$\frac{V_o}{V_i} \approx \frac{V_{o1}}{V_i} \times \frac{V_o}{V_{o1}} \quad (6.1)$$

The expression $\frac{V_{o1}}{V_i}$ has already been computed in (5.1) while $\frac{V_o}{V_{o1}}$ is derived in [37] and shown in (6.2) below.

$$\frac{V_o}{V_{o1}} \approx \frac{g_{mp}}{g_{mp} + G_o} \text{ where } G_o = \frac{1}{sL_o + \frac{1}{sC_o}} \quad (6.2)$$

Substituting (5.1) and (6.2) in (6.1) yields the final small signal gain approximation for the buffered op-amp in (6.3) with a further initial assumption of $L_o \approx 0 \rightarrow G_o \approx sC_o$.

$$\frac{V_o}{V_i} \approx \frac{\frac{g_{mh}g_{mp}R_c}{C_o}(s+\frac{1}{R_cC_c})}{(s+\frac{1}{R_oC_c})(s+\frac{g_{mp}}{C_o})} \quad (6.3)$$

Where R_o is the impedance at terminal V_{o1} and is calculated in section 5.1 to be in the order of 130k Ω . The remaining parameters were also derived in section 5.1 with $I_b = 1mA$, $R_c = 58.8\Omega$, and $C_c = 2pF$. The resulting transconductances - g_{mh} and g_{mp} are 17.6mA/V and 4.14mA/V respectively as obtained from Fig. 5.3-5.4. The parasitic capacitance (C_o) due to the board and instrumentation is 35pF.

Based on (5.1) and (6.3) the inclusion of a buffer in the op-amp adds an additional pole to the open loop transfer function given by $\frac{g_{mp}}{C_o} = 18.8MHz$. This pole occurs before the first order unity gain frequency of 1.4GHz thus degrading the phase margin significantly to approximately 0° at the unity gain frequency of this (now) second order system. In the absence of any parasitic load, $C_o = 0F$ and the pole introduced by the buffer would be infinite thus the buffer would have no effect on the phase margin. As a consequence, the only viable option for measuring the delay element involves using chip probes and an oscilloscope whose combined capacitance is less than 0.5pF. Such an option is currently unavailable. The schematic in Fig. 6.13 is simulated in Cadence to verify the above conclusions on stability to yield the plot in Fig. 6.14. Fig. 6.14 shows the open-loop gain and phase response to frequency for parasitic loads (C_o) ranging from 1.6pF to 70pF. The phase margins are 43.8°, 19.9°, 11°, and 8° for load capacitances of 1.6pF, 10pF, 35pF and 70pF respectively.

Figure 6.14: Magnitude and phase response of buffered op-amp for various capacitive loads

The phase margin decreases with capacitive load in accordance with (6.3) thus increasing the circuit's instability. The only way to improve stability is by further reducing the first pole ($\frac{1}{R_o C_c}$) through the compensation capacitor (C_c) at the expense of bandwidth or by adding a resistor in series with the load capacitance (C_o). A sufficiently large resistor in series with the load capacitor creates a zero which cancels out the pole introduced by the load capacitance.

The revised PCB in Fig. 6.15 eliminates oscillation by including a $5k\Omega$ resistance in series with the load. It also includes low pass filters at the power and varactor-tuning terminals to attenuate the ac signals from the switching power supplies. The measured magnitude and group delay responses as obtained from a network analyzer are shown in Fig. 6.16-6.22. The gain plots at tuning voltages of 2.3V, 1.25V, and 0.3V are shown in Figures 6.16, 6.17, and 6.18 respectively. Each trace in the gain plots corresponds to an output in the 4-tap delay line.

Figure 6.15: Revised PCB used to measure delay elements

Figure 6.16: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 2.3\text{V}$

Figure 6.17: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 1.25\text{V}$

Figure 6.18: Measured transfer function magnitude of delay line at $V_{\text{tune}} = 0.3\text{V}$

The circuit exhibits a low pass response (in agreement with simulation results) up to about 20MHz after which resonance occurs. The measured resonance frequency is also about 100MHz as predicted by simulation.

The measured group-delay plots at tuning voltages of 2.3V, 1.25V, and 0.3V are shown in Figures 6.19, 6.20, and 6.21 respectively. The group delay exhibits the expected low-pass characteristic (as shown in Fig. 5.10) with a nominal value of 20ns and a tuning range from 17.6ns to 22ns. The nominal group delay is twice the expected value which consequently reduces the measured bandwidth to about 5MHz which is half the simulated value.

Figure 6.19: Measured group delay at $V_{\text{tune}} = 2.3\text{V}$

The difference in the expected and measured nominal delay is most likely due to the on-chip varactor whose characteristic should have been separately measured as it is based on the non-linear reverse biased capacitance of a diode.

Figure 6.20: Measured group delay at $V_{\text{tune}} = 1.25\text{V}$

Figure 6.21: Measured group delay at $V_{\text{tune}} = 0.3\text{V}$

Furthermore, the noise from switching supplies varies the varactor's capacitance leading to variations in the group delay's frequency response. The PCB contains a low pass filter to eliminate switching noise but further attenuation is required for a more stable response.

6.4 Power consumption

This subsection is an examination of the power consumed by the implemented analog circuits and the extrapolated power consumption of a complete predistorter. Each delay element consumes 17mA of current and consequently 42.5mW ($17\text{mA} \times 2.5\text{V}$) of power. According to Table 5.1, the neural network predistorter requires 4 delay elements per quadrature input and two quadrature inputs implying a total power consumption of 340mW ($2 \times 4 \times 42.5\text{mW}$) in the input layer. In the non-linear neuron, the synaptic weight, adder and activation circuit consume 3.2mA, 6mA and 10.8mA of current respectively. The total power consumed per non-linear neuron is 122mW as obtained from 10 synaptic weights, an adder, and an activation circuit. The power consumed by the hidden layer is thus 1.34W due to 11 non-linear neurons in the hidden layer. Finally, the components in the linear neuron namely the synaptic weights and an adder use 3.2mA and 5mA respectively. Note that the adder in the linear neuron consumes less current than its non-linear counterpart because it is single ended thus it does not need common mode feedback circuitry. The power consumed per linear neuron is thus 100.5mW as obtained from 11 synaptic weights and an adder. The resulting power utilized in the output layer of the predistorter is 201mW as 2 linear neurons are required.

The total power consumed by the predistorter from the input, hidden and output layers is thus 1.88W. This number is comparable to other digital predistorters (not neural networks) in

[50]-[51] with similar bandwidths and much smaller devices (28nm) which consume 2W-4W of power. A further comparison of this neuron implementation to other neural ASICs is done in Table 6.1. Other neuron implementations exist but the examples in Table 6.1 are of realized multi-layer perceptron (MLP) structures which are suitable for use as function approximators. The MA-16 in Table 6.1 is a digital ASIC synapse by Siemens used in the Synapse I-II neural networks which uses 19 times more power than the proposed analog implementation. When the higher resolution and gate length of the digital process is compensated for, said ratio decreases from 19 to 1.74.

Table 6.1: Performance comparison with other neurons

Device	Bandwidth	Linearity	Power consumption	IC Process
This work	50MHz	5 bits	0.12W	0.25 μ m BiCMOS
MA-16 [52]	50MHz	16 bits	2.6W	1 μ m CMOS
ETANN [53]-[54]	334kHz	4-6 bits	0.15mW per synapse	1 μ m CMOS
[36],[44]	---	< 3bits	0.094W (4.7mW per synapse)	0.35 μ m CMOS

The ETANN is an analog neuron which consumes ten times less power but its bandwidth is 145 times smaller. Custom neurons in [36] and [44] have comparable power consumption with our implementation.

6.5 Chapter summary

This chapter describes the measurement results obtained for all the fabricated analog circuits namely: the active delays, and the linear and non-linear neurons. The active delay has a gain

bandwidth of 10MHz and a group delay which ranges from 17.5ns to 22ns over a 5MHz bandwidth. The group delay was designed for a nominal value of 10ns and the discrepancy is likely due to the on-chip varactor which is inaccessible; hence its characteristics cannot be verified.

The linear neuron has tunable gain from -3.4V/V to 3.9V/V over a 3dB-bandwidth of 10MHz which is close to the simulated bandwidth of 12.5MHz. It also exhibits a worst case non-linearity of 4bits for small signal inputs which is due to noise from switching power supplies which manifest at the output.

The non-linear neuron has tunable gain from -3.6V/V to 3.8V/V. Due to the absence of an output buffer, its measured bandwidth is reduced by parasitic loads (from the PCB and oscilloscope) to 0.8MHz which corresponds to an unloaded bandwidth of about 50MHz. It also exhibits a worst case linearity of 5bits for small signal inputs approaching the noise floor of the oscilloscope. The measured and simulated activation functions are closely correlated as verified by their NMSE of -26dB. The measured loaded bandwidth (due to parasitics) of the activation function is 8MHz which is close to the simulated value of 10MHz.

Finally, based on the power consumed by the individual analog components, the projected total power consumed by the complete predistorter is 1.88W which is comparable to existing DPD implementations. It is worthwhile to note that the said DPD structures ([50]-[51]) are realized in a 28nm process which is about 10 times smaller than the MOS transistors in our process. The projected power savings resulting from a smaller process are discussed in the next Chapter. The realized analog neuron also consumes less power than existing digital neuron implementations for MLP structures.

Chapter 7

Discussion, Conclusion and Future Work

7.1 Discussion

Based on the real estate cost of IC fabrication, the initial presumption was that the neural network components could be fabricated on an IC and then assembled on a PCB for full predistorter realization. However, based on individual component tests, we have noticed that the long tracts and lossy substrate of a PCB will create parasitics which can degrade the bandwidth as well as create stability issues. As such, for optimum performance, the complete realization should be on-chip. Obviously such an implementation would require significant chip area - less than 7.5mm x 7.5mm based on the current implementation. An addendum to on-chip implementation is the need for on-chip weight storage. The proposed predistorter implementation requires 132 synaptic weights which in turn require 264 pads for external control to differential inputs. On chip weight storage thus become a necessity otherwise the device becomes pad limited and increases in size not to mention being very difficult to manage off-chip. On-chip memory can be achieved by using a process which has floating gate transistors. Excepting that, efficient digital memory would have to be implemented on chip; this is a feasible research task considering that synaptic weight updates only occur every few seconds, hence very low digital sampling frequencies.

The size of npn transistors used at the output terminals of all circuits in this Thesis is $0.4 \times 1.5 \times 4$ with a corresponding collector-source capacitance of 65.3pF. Due to the unavailability of pnp transistors, the npn transistors use pmos transistors as active loads with

size and corresponding drain-source capacitance equal to $10 \times 0.25 \times 10$ and 322.6fF respectively. For the same bias current, the pmos transistors are thus 4.94 times slower than the bipolar. Consequently, full bipolar functionality with pnp transistors would increase the achieved bandwidths in the implemented circuits by a factor of 3 to 360MHz. A newer version of the $0.25\mu\text{m}$ BiCMOS design kit from NXP has recently been released with full bipolar functionality thus the aforementioned bandwidth improvement can be realized in the near future.

Subsequent power reduction can be achieved in the same process at the expense of the output voltage/power by reducing the current and transistor sizes by a factor of 2 thus decreasing the power consumption to 0.76W . As these are now the smallest transistors in the process, further power reduction requires a step-down in the technology or process. As $0.13\mu\text{m}$ and 65nm both require 1.2V power supplies, 65nm would be the preferable choice as the bandwidth may increase by a factor of 4 (to 1.44GHz) for the same current density provided bipolar transistor sizes scale down in proportion to their CMOS counterparts. The resulting power benefit from a step down in technology would be a reduction in power consumption to 0.38W . Using a smaller process than the 65nm would require a redesign of all circuits as the transistor breakdown or power supply voltage decreases and devices begin to operate in the sub-threshold region.

Summarily, this Thesis shows that neural networks are a viable option for pre-distortion by taking significant steps towards their realization. A few implementation issues still need to be tackled before a full neural network predistorter can be realized. Furthermore, the reduction in power consumption by using an analog implementation compared to a digital

has been verified with more suggestions on how to improve power savings and bandwidth. As this implementation is still relatively new, full predistorter functionality with the suggested power savings might require a long design cycle. However, an extrapolated analog implementation in a 65nm process could consume 0.4W of power with a bandwidth of 1.4GHz. Such performance has not yet been realized in any DPD structures thus making the complete analog implementation worthwhile.

7.2 Conclusion

The main objective of this Thesis is to realize a predistorter implementation based on neural networks with a potential power benefit compared to existing pre-distorters. Power optimization is done by designing the neural network with an analog implementation in mind as well as utilizing the minimum number of neurons to achieve the required model accuracy. This work shows that efficient predistortion with neural networks is certainly feasible by first characterizing an RF-PA for a wideband stimulus before designing, fabricating and testing individual components which make up the neural network predistorter.

Chapter 2 initially analyzes the dynamic and memoryless characteristics of a non-linear power amplifier before looking at linearization schemes such as feedback, feed-forward linearization, and pre-distortion. Pre-distortion can accommodate a wider bandwidth than the feedback linearizer and is more efficient compared to the feed-forward linearizer. Finally, Chapter 2 examines different pre-distortion architectures including neural networks thus showing that they all yield similar performances.

Chapter 3 provides background on neural networks including key optimization algorithms for function approximation or system identification. Based on the Levenberg-Marquardt method,

a proposed optimization algorithm for weight-restricted network training is described. The proposed method incorporates the finite dynamic range of analog synaptic weights into the proposed algorithm with a minimum loss in accuracy and a reduction in convergence time. Chapter 4 verifies the proposed algorithm's ability to train an inverse PA model using the measured dynamic non-linear characteristics of a class-AB PA. The proposed L-M training method ensures that the trained weights do not exceed a prescribed limit with negligible degradation in accuracy. The proposed method was shown to reduce convergence time by 10% and 53% for small and larger neural networks respectively. Finally, predistortion of the DUT using the trained neural network over a 20MHz bandwidth results in ACPR improvements which range from 13.5dB to 15.6dB and an EVM which reduces by 8.7%. Chapter 5 describes the design of analog circuits used in the realization of a neural network predistorter. The first component is an active 4-tap delay line with a tunable unit delay which ranges from 7.5ns to 12.3ns over a 10MHz bandwidth. The remaining components are linear and non-linear neurons which in turn comprise of synaptic weights and an adder. The synaptic weights and adder combination have tunable gains which range from -3.4V/V to +3.4V/V and 3dB-bandwidths of 12.6MHz and 50MHz for linear and non-linear neurons respectively. The non-linear neuron also contains an activation circuit whose transfer function is anti-symmetric and differentiable over a 340MHz bandwidth. Finally, a circuit simulation of the complete predistorter in Cadence shows the ACPR has a worst case reduction of 5dB compared to the behavioral model while the EVM remains the same. Chapter 6 describes the measurement results obtained for the active delays, and the linear and non-linear neurons. The active delay has a gain bandwidth of 10MHz and a group delay

which ranges from 17.5ns to 22ns over a 5MHz bandwidth. Discrepancies between the simulated and measured delays are likely due to process variations in the on-chip varactor. The linear neuron has tunable gain from -3.4V/V to 3.9V/V over a 3dB-bandwidth of 10MHz and a worst case non-linearity of 4bits at small signal outputs due to switching noise fluctuations. The non-linear neuron has tunable gain from -3.6V/V to 3.8V/V, a (parasitic) loaded bandwidth of 0.8MHz, and a worst case (or small signal) non-linearity of 5bits. The NMSE between the measured and simulated activation functions is -26dB implying a close correlation; and the measured (loaded) bandwidth of the activation function is 8MHz. Finally, the projected total power consumed by the complete predistorter is 1.88W which is comparable to existing DPD implementations and less than its digital neural network counterparts. Improvements to the delay and activation circuits have been suggested in Section 7.3 in order to decrease the projected power consumption of the complete predistorter from 1.88W to 1.53W. Section 7.3 also suggests improvements to the adders in the linear and non-linear neurons which can improve the linearity from 5 bits to 7 bits and the bandwidth from 50MHz to 120MHz.

Contributions made in this Thesis involve the modification of the standard L-M training algorithm to account for finite weights with minimal degradation in accuracy and a reduction in processing time. Also, the utilization of neural networks in PA linearization is a fairly new concept and no hardware implementations currently exist which utilize neural networks to pre-distort the dynamic non-linearity of a PA. The realized analog circuits in this Thesis is a significant portion of such an implementation and further work would include

perfecting the existing design, integrating the individual components to achieve full functionality, and incorporating memory for adaptive control.

7.3 Future Work

The fabricated neural network components have since been improved for linearity, bandwidth, and power consumption and their measured performance can be verified as a future exercise. Starting with the delay element whose schematic is shown in Fig. 5.13, its power consumption is reduced by a third when the delay circuit is made fully differential as opposed to two single ended delay circuits. To achieve this single to differential conversion, we use a quadrilateral folded cascode op-amp whose schematic is shown in Fig. 7.1. The op-amp in Fig. 7.1 has 2 differential pairs at its input to provide a high impedance terminal at each differential input of the delay element as shown in Fig. 7.2. An op-amp with a single differential pair at its input would have a low input impedance, which is approximately equal to R_4 , in its corresponding delay implementation.

Figure 7.1: Four input folded cascode operational amplifier

Figure 7.2: Modified single delay element

This causes the transfer function to significantly deviate from a 2nd order Pade approximation. The benefit of the modified implementation in Fig. 7.2 aside from reduced real estate on the chip is a reduction in total current from 17mA to 12mA implying a power consumption of 30mW per delay element.

The next component whose performance has been improved is the adder in both the non-linear and linear neurons. The improved adder as shown in Fig. 7.3 is a transimpedance amplifier which serves to improve the linearity and bandwidth of all synaptic weights connected to it.

The fabricated synaptic weight implementation in Fig. 5.21 is a multiplier whose output is a differential current which is converted to a voltage by the resistor $R_L || R_i$ where R_i is the input resistance of the adder. When $R_L || R_i$ is high, the time constant seen by the multiplier is large implying a smaller bandwidth. Secondly, a large $R_L || R_i$ increases the voltage at the output terminals of the multiplier thus increasing its non-linear contribution to the signal.

Figure 7.3: Transimpedance amplifier as an adder

Utilizing a smaller R_L at the expense of larger gain at the adder does not improve the bandwidth as the multiplier and adder combination is now limited by the constant gain-bandwidth product of the adder. The transimpedance adder in Fig.7.3 is a solution to both bandwidth and linearity issues since the effective load seen by the multiplier is now R_f/G_a where G_a is the open circuit gain of the op-amp. The voltage at the multiplier terminals is now small signal hence linear. The bandwidth is also improved by the lower effective load as seen by the multiplier. The feedback resistor (R_f) and capacitor (C_f) in Fig. 7.3 are $9.45\text{k}\Omega$ and 120pF respectively. The feedback capacitor removes peaking in the magnitude response while the feedback resistor sets the gain of the multiplier and adder combination. Fig. 7.4 shows the simulated output (V_o) against input (V_x) voltage for various control voltages (V_y). The slope or gain increases with control voltage with the maximum slope occurring at $V_y = \pm 1$. Fig. 7.5 shows the simulated unloaded frequency response of the multiplier for various values of V_y .

Figure 7.4: Simulated output voltage for control voltages ranging from $\pm 1V$.

The frequency response of the synaptic weight shows a 3dB-bandwidth of approximately 120MHz across all control values with negligible gain variation below 70MHz. The 3dB-bandwidth using the transimpedance amplifier adder has been increased by a factor of 2.4 compared to the fabricated resistive feedback op-amp adder.

Figure 7.5: Frequency response of multiplier for V_y ranging from 0-1V.

Fig. 7.6 shows the gain control and linearity (gain percentage error) of synaptic weights when loaded with the transimpedance adder. The linear gain variation is the same as before with the input and control voltages within the following respective ranges: $|V_x| \leq 0.2 V$ and $|V_y| \leq 0.8 V$. The maximum gain error, however, as obtained from schematic simulations, decreases from 3.3% to 0.76% which is equivalent to 7 bits of digital precision.

Figure 7.6: Simulated voltage gain and its percentage error against control voltage

Summarily, the transimpedance adder increases the 3dB-bandwidth and linearity of synaptic weights from 50MHz to 120MHz and 5bits to 7bits respectively.

The final circuit revision is the activation circuit in the non-linear neuron whose power consumption has been reduced by a factor of 6.75. As shown in Fig. 5.25, the activation circuit consists of a non-linear transconductor in series with a differential pair. The activation function is thus a product of the transfer functions due to the transconductor and differential pair. Tuning the respective non-linearities of both transfer functions yields the desired activation function. The transfer function of the differential pair depends on its bias current while that of the transconductor is determined by its emitter degeneration resistor

(R_e). The transfer function of the transconductor in the fabricated circuit is highly non-linear due to a relatively low R_e ($5\text{k}\Omega$) while that of the differential pair is less non-linear due to a high bias current (2mA). Consequently, the predominant power consumption occurs in the differential pair and its common mode feedback circuitry as the transconductor is biased at only 0.2mA . The total power consumption can be significantly reduced without affecting the activation function by increasing the non-linearity of the differential pair (thereby reducing its bias current) and increasing the linearity of the transconductor. The bias current of the differential pair is reduced from 2mA to 0.2mA thus reducing the total current utilized by the activation circuit from 10.8mA to 1.6mA . The bias current of the transconductor remains unchanged but its R_e is increased from $5\text{k}\Omega$ to $6\text{k}\Omega$ thus maintaining the activation function of the previous activation circuit as shown in Fig. 7.7.

Figure 7.7: Activation functions for both fabricated and modified circuits

Fig. 7.7 shows that the activation functions of both fabricated and modified circuits are almost identical.

Based on the aforementioned circuit improvements, the total power consumed by the complete neural network predistorter reduces from 1.88W to 1.53W and the linearity increases from 5bits to 7bits. From Table 5.3, which is based on the analog circuit models, the worst case ACPR improvement ratios are 10dB and 11.2dB at the 5MHz and 10MHz offsets respectively. Applying the new circuit models with 7bits of weight linearity to predistortion improves the original (5bit) ACPR by at least 2.3dB and 4.1dB at 5MHz and 10MHz offsets respectively. Again, the EVM remains unchanged compared to the behavioral predistorter (3.1%). As previously mentioned, the above improvements could be verified with measurements as a future exercise in addition to integrating the individual components to achieve full predistorter functionality, and incorporating memory for adaptive control.

Appendix A

Testing Devices

Figure A1: RF devices for linearization

Figure A2: Initial PCB used to measure non-linear neuron IC

Figure A3: Initial PCB used to measure delay IC

References

- [1] 3GPP TS 25.141 V4.1.0 (2001-06), 3rd Generation Partnership Project; Technical Specification Group Radio Access Networks; Base station conformance testing (FDD) (Release 4).
- [2] F. Mkadem, S. Boumaiza, “Physically Inspired Neural Network Model for RF Power Amplifier Behavioral Modeling and Digital Predistortion”, IEEE Transactions on Microwave Theory and Techniques, Vol. 59, No. 4, April 2011, pp. 913 –923
- [3] M. Rawat, K. Rawat, and F. M. Ghannouchi, “Adaptive Digital Predistortion of Wireless Power Amplifiers/Transmitters Using Dynamic Real-Valued Focused Time-Delay Line Neural Networks”, IEEE Transactions on Microwave Theory and Techniques, Vol. 58, No. 1, January 2010, pp. 95 –104
- [4] S. Boumaiza, F. Mkadem, “Wideband RF Power Amplifier Predistortion using Real-Valued Time-Delay Neural Networks”, 39th European Microwave Conference, 2009, pp.1449 –1452
- [5] M. Ngwar, J.S. Wight, “Weight-Restricted Training of Neural Networks with the Levenberg-Marquardt Algorithm,” IEEE Trans. Neural Netw., *submitted*
- [6] M. Ngwar, J. Wight, “An Artificial Neural Network for Wideband Pre-distortion of Efficient Pico-cell Power Amplifiers,” IEEE ISCCSP, May 2014, pp. 562-565
- [7] M. Ngwar, J. Wight, “A Fully Integrated Analog Neuron for Dynamic Multi-layer Perceptron Networks,” IEEE IJCNN, Jul. 2015, *accepted*
- [8] J. W.M. Rogers, C. Plett, *Radio Frequency Integrated Circuit Design*, 1st ed., Artech House, Inc., 2003, pp 23 – 29
- [9] S. C. Cripps, *Advanced techniques in RF PA Design*, 1st ed., Artech House, Inc., 2002, pp.111 –255

- [10] C. Mikkell, W. Høyerby, M. A. E. Andersen, “TEDS Base-Station Power Amplifier Using Low-Noise Envelope Tracking Power Supply”, IEEE Transactions on Microwave Theory and Techniques, Vol. 57, No. 7, July 2009, pp. 1687 – 1693
- [11] Y. Y. Woo, J. Kim, J. Yi, S. Hong, I. Kim, J. Moon, B. Kim, “Adaptive Digital Feedback Predistortion Technique for Linearizing Power Amplifiers”, IEEE Transactions on Microwave Theory and Techniques, Vol. 55, No. 7, May 2007, pp. 932 – 940
- [12] C. D. Presti, D. F. Kimball, P. M. Asbeck, “Closed-Loop Digital Predistortion System With Fast Real-Time Adaptation Applied to a Handset WCDMA PA Module”, IEEE Transactions on Microwave Theory and Techniques, Vol. 60, No. 3, March 2012, pp. 604 – 618
- [13] J. Presa, J. Legarda, H. Solar, J. Melendez, A. Munoz, A. Garcia-Alonso, “An Adaptive Feedforward Power Amplifier for UMTS Transmitters”, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 4, No. 15, Sept. 2004, pp. 2175 – 2719
- [14] J. Kim, K. Konstantinou, “Digital predistortion of wideband signals based on power amplifier model with memory”, IEEE Electronics Letters, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.
- [15] F. Mkadem, D. Y. Wu, and S. Boumaiza “Wiener G-Functionals for Nonlinear Power Amplifier Digital Predistortion”, IEEE MTT-S International Microwave Symposium Digest, pp. 1-3, June 2012
- [16] G. Xu, T. Liu, Y. Ye, T. Xu, “FPGA Implementation of Augmented Hammerstein Predistorters for RF Power Amplifier Linearization”, IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, pp. 481-484, Oct. 2009
- [17] M. Isaksson, D. Wisell, D. Rönnow, “Wide-Band Dynamic Modeling of Power Amplifiers Using Radial-Basis Function Neural Networks”, IEEE Transactions on Microwave Theory and Techniques, Vol. 53, No. 11, Nov. 2005, pp. 3422 – 3428

- [18] S. Haykin, *Neural Networks A Comprehensive Foundation*, 2nd ed., Pearson Education, Inc., 1999, pp 11 – 25, 68 - 70, 121 – 133
- [19] B. M. Wilamowski, Y. Hao, “Improved Computation for Levenberg–Marquardt Training,” *IEEE Trans. Neural Networks*, vol. 25, no. 6, pp. 930–937, Jun. 2010
- [20] J. X. Peng, K. Li, G.W. Irwin, “A new Jacobian matrix for optimal learning of single-layer neural networks,” *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 119–129, Jan. 2008
- [21] M. T. Hagan, M. Menhaj, “Training feedforward networks with the Marquardt algorithm”, *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, November 1994, pp. 989 – 993
- [22] K. P. Lam, S. T. Mak, “An FPGA-based eigenfilter using fast Hebbian learning,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 765-768, April 2003.
- [23] B. K. Dolenko, H. C. Card, “Tolerance to analog hardware of on-chip learning in backpropagation networks,” *IEEE Trans. on Neural Networks*, vol. 6, no. 5, pp. 1045-1052, Sept. 1995.
- [24] S. Bai-qing, Z. Can-xin, C. Xiao-dong, “The Subjected SPDS Algorithm of Forward Neural Network,” *International Conference on Management Science and Engineering*, pp. 405-409, Aug. 2007.
- [25] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*. *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989
- [26] G. Cybenko, *Approximations by superpositions of sigmoidal functions*, *Mathematics of Control, Signals, and Systems*, 2 (4), 1989, pp 303-314
- [27] E. Dahlman, P. Beming, J. Knutsson, F. Ovesjo, M. Persson, C. Roobol, “WCDMA-the radio interface for future mobile multimedia communications,” *IEEE Trans. Vehicular Technology*, vol. 47, pp. 1105–1118, Nov. 1998

- [28] F. Fernez, "An ADS bench for generating multi-carrier 3GPP WCDMA ACLR test signals," *High Frequency Electronics*, pp. 34-42 November 2002.
- [29] E. Barnard, "Optimization for training neural nets," *IEEE Trans. Neural Networks*, vol. 3, pp. 232–240, Mar. 1992
- [30] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, pp. 989–993, Nov. 1994
- [31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 32, no. 9, pp. 533–536, Oct. 1986.
- [32] P. P. van der Smagt, "Minimization methods for training feedforward neural networks," *Neural Networks*, vol. 7, no. 1, pp. 1–11, 1994.
- [33] N. Lashkarian, "Crest Factor Reduction in Multi-carrier WCDMA Transmitters," *IEEE PIMRC*, vol. 1, pp. 321-325, Sept. 2005.
- [34] P.F.G. da Silva, E.G. de Lima, "Design of crest factor reduction techniques based on clipping and filtering for wireless communications systems," *IEEE SBCCI*, pp. 1-5, Sept. 2013
- [35] K. Yadavelli, D. Efstathiou, M. Manglani, "Crest factor reduction engine for multicarrier WCDMA transmitted signals," *IEEE PIMRC*, vol. 3, pp. 2207-2211, Sept. 2004
- [36] L. Gatet, H. Tap-Beteille, and F. Bony, "Comparison Between Analog and Digital Neural Network Implementations for Range-Finding Applications," *IEEE T. on Neural Networks.*, vol. 20, no. 3, pp. 460-470, March 2009.
- [37] D. A. Johns, K. Martins, *Analog Integrated Circuit Design*, 1st ed., J. Wiley & Sons, 1997, pp. 16-56, 256-273, 287-291, 584-597.
- [38] C. Brezenski, "Extrapolation algorithms and Pade approximations", *Applied Numerical Mathematics*, vol. 20, pp. 299–318, 1996

- [39] G. Gurun, A. Sisman, J. Zahorian, S. Satir, M. Karaman, P. Hasler, and L. Degertekin, "A tunable analog delay element for high-frequency dynamic beamforming," IEEE IUS, pp. 345-345, Sept. 2009.
- [40] L. Xiaofeng, L. Jin, L. Hoi, L. Hao, "A 2.5- to 3.5-Gb/s Adaptive FIR Equalizer With Continuous-Time Wide-Bandwidth Delay Line in 0.25- μ m CMOS," IEEE Jour. of Solid-State Circuits, no. 8, vol. 41, pp. 1908-1919, Aug. 2006.
- [41] S. P. Adhikari, C. Yang, H. Kim, L. O. Chua, "Memristor Bridge Synapse-Based Neural Network and Its Learning," IEEE T. on Neural Networks., vol. 23, no. 9, pp. 1426-1435, Sept. 2012.
- [42] Y. Kim, Y. Zhang, P. Li, "A Digital Neuromorphic VLSI Architecture with Memristor Crossbar Synaptic Array for Machine Learning," IEEE SOCC, pp. 328-333, Sept. 2012.
- [43] F. Merrikh-Bayat, S. B. Shouraki, "Memristive Neuro-Fuzzy System," IEEE T. on Cybernetics., vol. 43, no. 1, pp. 269-285, Feb. 2013.
- [44] Z. Biolek, D. Biolek, and B. Biolková, "Spice model of memristor with nonlinear dopant drift", Radio Eng., vol. 18, no. 2, pp. 210-214, 2009.
- [45] F. Corinto and A. Ascoli, "A boundary condition-based approach to the modeling of memristive nano-structures", IEEE Trans. Circuits Syst. I, vol. 59, no. 11, pp. 2713-2726, Nov. 2012.
- [46] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM - ThrEshold Adaptive Memristor Model", IEEE Transactions on Circuits and Systems -- I, vol. 60, no. 1, pp. 211-221, 2013.
- [47] L. Gatet, H. Tap-Beteille, and M. Descure, "Analog Neural Network Implementation for a Real-Time Surface Classification Application," IEEE J. of Sensors., vol. 8, no. 8, pp. 1413-1421, August 2008.
- [48] E. P. Santana, R. C. S. Freire, A. I. A. Cunha, "A CMOS analog four-quadrant multiplier for CNN synapses," IEEE ICCDCS, pp. 1-4, Mar. 2012.

- [49] F. Diotalevi, M. Valle, G. M. Bo, E. Biglieri, "Analog CMOS current mode neural primitives," IEEE ISCAS, vol. 2, pp. 717-720, May 2000.
- [50] M. Ouzillou, (Aug. 12, 2011). Linearize Power Amps with RF Predistortion. Microwaves&RF. Retrieved from <http://mwrf.com/components/linearize-power-amps-rf-predistortion>.
- [51] Altera Corporation, (Jan. 2012). Designing Polyphase DPD Solutions with 28-nm FPGAs (White Paper). Retrieved from <http://www.altera.com/literature/wp/wp-01171-polyphase-dpd.pdf>.
- [52] U. Ramacher, J. Beichter, N. Bruls, "A General-Purpose Signal Processor Architecture for Neurocomputing and Preprocessing Applications," Journal of VLSI Signal Processing, vol. 6, pp. 45-56, 1993.
- [53] M. Holler, T. Simon, H. Castro, R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 'floating gate' synapses," IEEE IJCNN, vol. 2, pp. 191-196, Jun. 1989.
- [54] S. M. Fakhraie, K. C. Smith, VLSI-compatible implementations for artificial neural networks, vol. #382, Springer Boston, Dec. 1996, pp. 165-170.