

**Digital Assistance Design for Analog Systems:
Digital Baseband for Outphasing Power Amplifiers**

by

Yan Li

B.Eng., Electrical Engineering, University of Science and Technology of China (2004)

M.A.Sc., Electrical Engineering, McMaster University (2006)

Submitted to the

Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

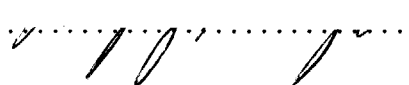
at the


MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

© Massachusetts Institute of Technology 2013. All rights reserved.

Author
Department of Electrical Engineering and Computer Science
March 7, 2013

Certified by
 Vladimir M. Stojanović
Associate Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by
 Leslie A. Kolodziej
Chair, Department Committee on Graduate Students

ARCHIVES

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 08 2013

LIBRARIES

Digital Assistance Design for Analog Systems: Digital Baseband for Outphasing Power Amplifiers

by

Yan Li

Submitted to the Department of Electrical Engineering and Computer Science
on March 7, 2013, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Digital assistance is among many aspects that can be leveraged to help analog/mixed-signal designers keep up with the technology scaling. It usually takes the form of predistorter or compensator in an analog/mixed-signal system and helps compensate the nonidealities in the system. Digital assistance takes advantage of the process scaling with faster speed and a higher level of integration. When a digital system is co-optimized with system modeling techniques, digital assistance usually becomes a key enabling block for the high performance of the overall system. This thesis presents the design of digital assistances through the digital baseband design for outphasing power amplifiers. In the digital baseband design, this thesis conveys two major points: the importance of the use of the reduced-complexity system modeling techniques, and the communications between hardware design and system modeling. These points greatly help the success in the design of the energy-efficient baseband.

The first part of the baseband design is to realize the nonlinear signal processing unit required by the modulation scheme. Conventional approaches of implementing this functionality do not scale well to meet the throughput, area and energy-efficiency targets. We propose a novel fixed-point piece-wise linear approximation technique for the nonlinear function computations involved in the signal processing unit. The new technique allows us to achieve an energy and area-efficient design with a throughput of 3.4Gsamples/s. Compared to the projected previous designs, our design shows 2x improvement in energy-efficiency and 25x in area-efficiency.

The second part of the baseband design devotes to the nonlinear compensator design, aiming to improve the linearity performance of the outphasing power amplifier. We first explore the feasibility of a working compensator by use of an off-line iterative solving scheme. With the confirmation that a compensator does exist, we analyze the structure of the nonlinear baseband-equivalent PA system and create a dynamical real-time compensator model. The resulting compensator provides the overall PA system with around 10dB improvement in ACPR and up to 2.5% in EVM.

Thesis Supervisor: Vladimir M. Stojanović
Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgments

I am fortunate enough to spend more than six years at MIT, enjoying the accompaniment of great people while striving to reach the end of my PhD journey. I have to give my first 'thank you' to my advisor Professor Vladimir Stojanović. There is a common word in Chinese describing a good teacher: teach by precept and example, and Vladimir perfectly interprets it with his self-example. As a supervisor, he plans, teaches, and guides each individual in the group; as a researcher, he works with great passion and energy, inspiring me to explore the fields that I have never thought of. I am very thankful for the opportunity to work with him for the past years and I am sure the knowledge and qualities I learnt from him will benefit me throughout my career path.

I would like to thank my thesis committee: Professors Alexander Megretski and Joel Dawson, for their patience with my thesis, as well as the research advice in various projects. I have been in collaboration with them closely in research projects for quite a long time, especially with Alex. Being a mathematician, Alex is always keen on the root of the problem and extremely helpful in providing insights to many problems.

I would like to thank Yehuda for his effort in weaving the net of people where everyone is able to find his/her comfort place and works as a team seamlessly. His talk with me from time to time always brings wisdom which probably would take me a long time to find by myself.

I would like to thank Mar Hershenson and her old team of Sabio for the opportunity to work with them as an intern. I am fortunate to be able to enjoy the spirit of entrepreneurship as well as the sunshine in California.

I would like to thank all past and current ISG folks. I am grateful for having them around throughout the tape-outs, various dry runs. Without the effort they put into our ISG's infrastructure, it would be a different story with all my tape-outs. I want to give my special thanks to my three officemates: Natasha, Michael and Jonathan. They are the best officemates you can ask for.

I would like to thank all my friends and family, who share my happiness and bitterness in my life. You are the ones I can always trust and get support from, and I am grateful to have you in my life. Lei became my husband the year I entered MIT. We worked in adjacent buildings, sometimes attended the same class, and when I stayed up for tape-outs or paper, he always prepared everything for me. I am thankful to have his encouragement, support, and the positive attitude throughout the journey, and I am sure he is as happy as I am for my graduation. Finally, I would like to thank my parents, Lixian and Anqi. No better words for you than 'I love you, dad and mom'.

Contents

1	Introduction	19
1.1	Challenges in the Analog World	19
1.2	Digitally Fix the Analog World	20
1.3	Thesis Contributions	21
1.3.1	Nonlinear Signal Processing for the Digital Baseband of AMO PA	22
1.3.2	Reduced-complexity System Modeling and PA Compensator Design	23
1.3.3	Extension of System Modeling to Hierarchical System Opti- mization	25
1.4	Thesis Overview	25
2	Nonlinear Signal Processing in a Digital Baseband Design of RF Transmitter	27
2.1	Outphasing Power Amplifier Background	28
2.1.1	LINC and AMO Systems	28
2.2	Piece-wise Linear Approximation	30
2.2.1	Algorithm	30
2.2.2	Piece-wise-linear Design Example	38
2.2.3	Comparison with General Numerical Function Generation	40
2.3	Chip Implementation	41
2.3.1	Overall Design	41
2.3.2	Blocks Design	44

2.3.3	Experimental Results	50
2.4	Summary	56
3	Reduced-complexity System Modeling in Compensator Design for an RF Transmitter	57
3.1	Digital Predistortion for PA System	58
3.1.1	Overview of Popular Digital Predistortion Techniques	58
3.1.2	Linearity Metrics	59
3.2	Digital nonlinear compensation for the LINC and AMO Systems	60
3.2.1	System Setup	60
3.2.2	Iteration-based Off-line Predistortion for LINC/AMO Systems	65
3.2.3	Analysis of Nonlinearities Throughout the System	79
3.2.4	Real-time Predistortion Model for LINC/AMO Systems	85
3.2.5	Implementations	93
3.3	Limitations of the Models	98
3.4	Summary	105
4	A Hierarchical System Design Methodology	107
4.1	Proposed Hierarchical Design Methodology	107
4.2	Design Space Reduction	108
4.2.1	Equation-based Circuit Optimization	109
4.2.2	Equation-based Robust Circuit Optimization	111
4.3	Subsystem Abstraction	113
4.4	Summary	119
5	Conclusions and Future Research Directions	121
A	Zero-avoidance Filter Design Example Using Heuristics	125
B	Robust Iterative Optimization Algorithm for Analog Circuits	129
B.1	Algorithm	129
B.1.1	Proposed robust circuit optimization framework	129

B.1.2 Sources of Variability	130
B.2 Experimental results	139
B.2.1 A two-stage op-amp example	139

List of Figures

2-1	(a) LINC, AMO SCS. (b) AMO PA system overview.	29
2-2	(a) The general concept of PWL approximation. (b) Proposed fixed-point PWL approximation.	32
2-3	(a) Micro-architecture of the PWL approximation. (b) Illustration of the computations in the PWL approximation.	35
2-4	The block diagram of the chip.	41
2-5	The hardware block diagram of the SCS system.	42
2-6	(a) The hardware block diagram of the <i>getTheta</i> block. (b) The hardware block diagram of the <i>getPhi</i> block.	45
2-7	The hardware block diagram of the <i>getAlpha</i> block.	48
2-8	Spectrum and EVM of the SCS.	51
2-9	Throughput and energy with supply scaling for AMO SCS.	52
2-10	Chip photograph.	53
2-11	(a) Power breakdown of the AMO SCS design. (b) Area breakdown of the AMO SCS design.	54
3-1	Three common nonlinear dynamical system structures. (a) Wiener model. (b) Hammerstein model. (c) Wiener-Hammerstein model.	58
3-2	An illustration of the ACPR definition.	60
3-3	PA system under compensation.	61
3-4	LINC and AMO PA architecture block diagrams.	62
3-5	Simplified schematics of the cascode class-E PA.	62
3-6	Switch network model blocks.	63

3-7	Switch network output for different values of bump inductances. 4 VDD levels are 1.1V, 1.4V, 1.8V, 2.2V. Sample duration is 0.4ns.	64
3-8	PA system under compensation.	66
3-9	Frobenius norm of the Jacobian of the function $v \rightarrow v_1$. $a = 1$	68
3-10	Frobenius norm of the Jacobian of the function $G_1(v)$. $a_1, a_2 \in [1.1, 1.4, 1.8, 2.2]$. The corresponding threshold levels for different regions are $[2.2, 2.5, 2.8, 3.2, 3.6, 4, 4.4]$	69
3-11	EVM of the uncompensated LINC system.	74
3-12	EVM of the compensated LINC system, with real-time zero-avoidance input sequence.	75
3-13	Input and output ACPR of the LINC system, with real-time zero-avoidance input sequence.	75
3-14	EVM of the uncompensated AMO system with $L_{\text{bump}}=20\text{pH}$	78
3-15	EVM of the compensated AMO system with $L_{\text{bump}}=20\text{pH}$. Input sequence is generated from offline level-avoidance filter.	78
3-16	Input and output ACPR of the AMO system with $L_{\text{bump}}=20\text{pH}$. Input sequence is generated from offline level-avoidance filter.	79
3-17	Nonlinear system of the overall transceiver signal chain.	80
3-18	Illustration of the derivation for nonzero terms in equation (3.31).	82
3-19	An example of DFT $P_1(e^{j(\frac{\Omega}{T} - \frac{2\pi r}{T})})$ with $\tau_1/T = 0.2$, $\tau_2/T = 0.3$, $r = 10$	85
3-20	Placement of the compensator in the LINC/AMO systems.	86
3-21	Compensator structure.	87
3-22	EVM of the LINC system with real-time compensator. Input sequence is generated from a real-time zero-avoidance filter.	88
3-23	ACPR of the LINC system with real-time compensator. Input sequence is generated from a real-time zero-avoidance filter.	89
3-24	EVM of the AMO system without a real-time compensator.	90
3-25	EVM of the AMO system with a real-time compensator. The input sequence is generated from an offline level-avoidance filter.	91

3-26	ACPR of the AMO system with a real-time compensator. The input sequence is generated from an offline level-avoidance filter.	92
3-27	Construction of the LTI system with discontinuities at $\pm\pi$	95
3-28	The block diagram of the compensator hardware implementation. . .	96
3-29	The block diagram of the digital baseband with AMO SCS and nonlinear compensator.	99
3-30	Overview of the integrated transmitter system with digital baseband nonlinear compensation.	100
3-31	The area breakdown of the digital baseband with nonlinear compensator.100	
3-32	The power breakdown of the digital baseband with nonlinear compensator. This is an estimation from post-layout analysis at 1GHz clock frequency.	101
3-33	Input sequence structure to test the nonlinear system's stability. . . .	101
3-34	An example of constructing output comparisons with fixed previous 4 and the current samples.	102
3-35	An example of output waveform differences with the same previous state and current input phases. The state includes four previous samples.102	
3-36	The l_2 norm error of all data sets with the same previous state and current input phase. Different curves represent the states containing different number of past samples.	103
3-37	The l_2 norm error of all data sets with the same previous state and current input phase for the body-tied PA design. Different curves represent the states containing different number of past samples.	104
4-1	Hierarchical system design.	108
4-2	The two-stage Op-amp schematic.	109
4-3	Design trade-off sets between gain and load capacitance, corresponding to grid 1.	110
4-4	An intuitive illustration of the two iterations of the algorithm.	112

4-5	The folded-cascode example with $G_m \in [0.5 \text{ mS}, 0.6 \text{ mS}]$ from optimization: Monte-Carlo check with the initial and final robust design.	114
4-6	Monte-Carlo check with the initial and final robust designs: G_m of the folded-cascode op-amp.	115
4-7	Parameter grid. Red dots are the designs used to generate parameterized model.	117
4-8	Parameterized system identification results.	117
4-9	Testing results. The x-axis represents 30 testing input patterns and for each x value, there are 121 y values, denoted as colored '+', to represent the maximal relative error of the model output compared with the spice output in time domain. The '*' represent the maximal difference of the outputs from spice among the 121 designs for a certain input pattern (some x value). The 30 input patterns have random frequencies in the ranges [DC 146MHz] for the patterns 1-10, [DC 1MHz] for the patterns 11-15, [1MHz 10MHz] for the patterns 16-20, [10MHz 50MHz] for the patterns 21-25, [50MHz 146MHz] for the patterns 26-30.	118
4-10	A coarser parameter grid 2.	119
4-11	Testing results of the doubled-range parameter model. The x-axis represents 30 testing input patterns and for each x value, there are 121 y values, denoted as colored '+', to represent the maximal relative error of the model output compared with the spice output in time domain. The '*' represent the maximal difference of the outputs from spice among the 121 designs for a certain input pattern (some x value). The 30 input patterns have random frequencies in the ranges [DC 146MHz] for the patterns 1-10, [DC 1MHz] for the patterns 11-15, [1MHz 10MHz] for the patterns 16-20, [10MHz 50MHz] for the patterns 21-25, [50MHz 146MHz] for the patterns 26-30.	120
A-1	Illustration of the zero-avoidance zone.	126
A-2	Illustration of the zero-avoidance filter design.	126

A-3	Result of a zero-avoidance filter design.	128
B-1	Flow of the iterative robust algorithm.	130
B-2	The two-stage Op-amp schematic.	131
B-3	Transistor Macro model.	133
B-4	Monotonicity of circuit performances on variation variables in a two-stage op-amp example.	134
B-5	Monotonicity of circuit performances along random lines in variation variable space in a two-stage op-amp example.	135
B-6	Constraint maximization with two variation variables.	137
B-7	Two-stage op-amp with 1-corner initial optimization design: yield improvement of gain and ω_u , power and area consumptions.	140
B-8	Two-stage op-amp five-corner optimization design: yield improvement on gain, ω_u and power, area consumptions.	141
B-9	Two-stage op-amp five-corner optimization design: DC gain and ω_u comparison of initial and final robust designs.	143

List of Tables

2.1	LINC and AMO SCS Equations.	28
2.2	Storage comparison examples between a direct LUT map approach and fixed-point piece-wise linear approximation approach.	36
2.3	Comparison between PWL, CORDIC implementations of the 16-bit input, output function $y(x) = \cos^{-1}(x)$	36
2.4	Criterion for power supply pair selection. ($A^2 = I^2 + Q^2$)	43
2.5	Summary of arithmetic operations in each functional block of the AMO SCS.	44
2.6	Summary of accuracy and LUT size of the PWL approximated function blocks.	48
2.7	Comparison with other works.	55
3.1	ACPR and EVM performances of LINC system in off-line iterations, with input sequence generated from a real-time zero-avoidance filter.	74
3.2	ACPR and EVM performance comparisons between using input sequence with and without zero-avoidance property for LINC systems. The zero-avoidance filter has a real-time implementation as shown in Appendix A.	76
3.3	Comparison of offline iteration results with real-time and offline zero-avoidance filters.	77
3.4	ACPR and EVM performances of AMO systems with different bump inductances. The input sequences are from offline level-avoidance filtering.	77

3.5	ACPR and EVM performances of AMO system with 5pH bump inductance and 15ps path mismatch between phase and amplitude paths. The input sequence is generated from offline level-avoidance filter.	79
4.1	The folded-cascode op-amp example: specifications for nominal design	113
4.2	Folded-cascode op-amp: iterations of the robust designs from optimization: $g_m \in [0.5 \text{ mS}, 0.6 \text{ mS}]$. k denotes the range of the variability. Please refer to Section B.	113
B.1	Specifications for nominal design	132
B.2	Robust two-stage op-amp designs in iterations from optimization and Hspice simulation	140
B.3	Five-corner of the two-stage op-amp initial design.	141

Chapter 1

Introduction

1.1 Challenges in the Analog World

A combination of process scaling and demands for performance scaling brings ever-increasing challenges to analog/mixed-signal system designs. As the technology scaling front keeps pushing to the 22nm and 14nm nodes [1], designers are faced with more severe and new varieties of device non-idealities. While miniaturization in size and downward scaling of supply voltage benefit the digital world with faster transistors, a higher level of integration of digital functions and lower power consumption, devices in the scaled analog world suffer from decreased transistor intrinsic gain, dynamic range and larger process variations.

On the other hand, more functionality, lower power and cost, faster speed and ease of use have been driving generations of new products. One of the most prominent examples is the cellular handset, which evolved from a simple portable phone to a personal device capable of operating in multi-band, equipped with Wi-Fi connectivity, high-resolution camera etc. Furthermore, the bars of low-power and area are not lowered considering the integration of all kinds of functionalities. Battery life in portable devices has never been satisfying and researchers from all different disciplines are focusing on more energy-efficient and area-efficient designs.

Despite all the difficulties analog circuits are facing from process and performance scaling, the need for analog/mixed-signal can never be eliminated. Interfaces between

the human, environment and digital processing always exist and the requirements on convenience, flexibility, robustness etc. become even more challenging. To tackle all these problems and keep up with scaling, innovations have to come from all different aspects of the chip design. New device process, analog architecture, system architecture, etc. have to all come together and work in an interdisciplinary way.

1.2 Digitally Fix the Analog World

One part of the interdisciplinary effort to improve the analog/mixed-signal system design is through the digital assistance. As the driving force of process scaling, digital design has enjoyed the benefits of faster speed, more integration and lower power consumption. These benefits bring more computational power to the digital system, which can be utilized to correct non-idealities in the analog systems.

Digital assistance usually takes the form of a digital predistorter or compensator. For power amplifier (PA) systems, digital predistortion has been a popular way to enhance the linearity of the system without sacrificing power efficiency by power back-off [2-4]. A digital compensator has also been used in several other communication subsystems, such as modulators [5, 6] and analog-to-digital converters (ADCs) [7, 8]. Among all these successful examples, we observe two critical factors determining the feasibility of the digital compensation: theoretical support in system design and implementation practicality. Theoretical support from digital signal processing and nonlinear system modeling lays the foundation of the digital compensator design, while the equally important implementation part determines the actual hardware architecture. To achieve the success in the two factors, it is important to bridge the two sides and optimize the algorithm and hardware efficiency simultaneously. In this thesis, the main theme is to demonstrate the effectiveness of the co-optimization of both algorithm and hardware in the context of the digital baseband design for the outphasing power amplifiers.

1.3 Thesis Contributions

There are two main contributions in this thesis: 1) the design and implementation of an energy and area-efficient signal component separator (SCS) for the multi-level asymmetric outphasing (AMO) power amplifier (PA), and 2) reduced-complexity system modeling and compensator implementation for outphasing PAs. In addition, as an extension of the application of system modeling techniques, we demonstrate several key enabling techniques for a hierarchical system optimization methodology.

The two main parts of the thesis are dedicated to the design of the digital baseband of the outphasing PA. This effort contributes to the new generations of high-throughput wireless communication systems at millimeter-wave (mm-wave) range [9–15]. The availability of large chunks of bandwidth and maturity of CMOS process technology provide the opportunity to address several large markets with bandwidth-demanding communication applications. These mm-wave applications place great challenges on the transceiver design due to factors such as PA efficiency and linearity, high loss in wireless channel and multipath, increasing parasitics for passive components, limited amplifier gain etc. Even in cellular base stations, the drive toward flexible, multi-standard radio chips, increases the need for high-precision, high-throughput and energy-efficient backend processing. The desire to best leverage the available spectrum for these high-throughput applications creates the demand for high-efficiency and high-linearity PAs. While these conflicting PA design requirements have been satisfied in the past at low system throughputs by designing smart digital back-ends, the multi-GSamples/s throughput required in new applications puts a significant challenge on digital baseband system design to perform the necessary modulation and predistortion operations at negligible power overhead.

This desire for a high-throughput energy-efficient digital baseband becomes especially prominent for the outphasing PAs designed to improve the efficiency while satisfying the high-linearity requirements for higher-order signal constellations. This is due to the complexity of the baseband digital signal processing which hinders its capability to operate efficiently in high-throughput wideband applications. In this

thesis, we present a solution to extend the applicability of the outphasing PAs to a larger range of wideband applications.

1.3.1 Nonlinear Signal Processing for the Digital Baseband of AMO PA

The complex digital signal processing task in the baseband for the outphasing PA, and especially AMO PA, is the signal component separator (SCS). The SCS decomposes an arbitrary two-dimensional vector to two vectors under certain constraints. At low throughputs (10-100MSamples/s), the outphasing PAs would rely on complex digital signal processing to generate the outphasing vectors and make it possible to use simple, high-efficiency switching PAs on each path. At high (multi-GSamples/s) throughputs, however, a radical redesign of the signal component separator (SCS) digital signal processing implementations is needed to prevent degradation in net power efficiency due to a significant increase of digital baseband power consumption.

The conventional SCS has been traditionally implemented both in analog and digital designs [16–18]. The analog versions of SCS are obviously not suitable for high-speed and high-precision applications, so we only consider the digital SCS implementations. The SCS decomposes the original sample signal into two signals as required by the LINC/AMO, and the decomposition involves the computations of several nonlinear functions. For digitally implemented SCS, a look-up-table (LUT) is the most common way to realize the nonlinear functions. Considering that the past signal separators mainly work below 100MSamples/s with low to medium precision, LUT is indeed the simplest and most energy-efficient approach. Even for the recent AMO architecture, LUT is still a preferable choice for operations under 100MSamples/s [19]. However, the traditional LUT-based function map quickly becomes infeasible when the throughput and precision requirements go up to multi-GSamples/s and more than 10-bit range. The LUT size becomes prohibitively large for on-chip implementations with penalties in both area and speed. Besides, the number of LUTs used in the AMO SCS is significantly larger than in the LINC SCS, so the LUT solutions that

can barely work for LINC render AMO implementations infeasible. On the other hand, at these high throughputs a direct nonlinear function synthesis through iterative algorithms such as CORDIC [20] or nonlinear filters [21] proves to be more area compact but with prohibitive power footprint for the overall power efficiency of the PA.

In this thesis, we present the function synthesis algorithms and a corresponding chip implementation, designed using an alternative approach to compute the nonlinear functions: it is both more area and energy-efficient than state-of-the-art methods like LUTs, CORDIC or nonlinear filters. The chip results demonstrate an AMO SCS working at 3.4GSamples/s with 12-bit accuracy and over 2x energy savings and 25x area savings compared to traditional AMO SCS implementation. The new approach is based on the piece-wise linear (PWL) approximation of a nonlinear function. The approximation consists of the computations of LUT, add, and multiply. In order to minimize the computational cost while maintaining high accuracy and throughput, we propose a novel algorithm to find the fixed-point representation of the approximation. The idea of the fixed-point version of the approximation is to use as few operations as possible and minimize the number of input bits to all the operations so as to achieve high throughput. With these considerations, we are able to achieve a fixed-point representation of typical LINC or AMO nonlinear functions, which consists of one small LUT, one adder and one multiplier. The hardware architecture derived from this special algorithm achieves a nice balance between area, energy-efficiency, throughput and computational accuracy.

1.3.2 Reduced-complexity System Modeling and PA Compensator Design

For a transmitter system in compliance with communication standards, linearity metrics in terms of error-vector-magnitude (EVM) and adjacent-channel-power-ratio (ACPR) have to be met. In order to achieve high linearity in the PA system, the input signal has to back-off from the peak power level to minimize the distortion

associated with that operation region. The situation becomes even worse for wideband signals that have a high peak-to-average-power-ratio (PAPR), such as wideband code-division multiple access (WCDMA) in the universal mobile telecommunications system (UMTS), or orthogonal frequency-division modulation (OFDM) in current 4G LTE standards. Various linearization techniques, such as feedback, analog predistortion, and feed-forward, have been adopted to enhance the PA linearity [22–30]. Digital predistortion (DPD) is another promising technique. Compared with other linearization techniques, it is free of stability issues and has the potential for significantly smaller area and faster speed. However, it does require an adequate modeling effort to achieve an efficient, reduced-complexity dynamical model of the inverse system.

In this thesis, we present a thorough analysis of the PA baseband-equivalent system as well as its inverse system. First, to establish performance bounds, we treat the problem of searching for the inverse system as a general nonlinear system-solving and employ an iterative scheme. The results from the application of the iterative scheme on the LINC/AMO PAs show nearly 15dB of possible improvement in ACPR and up to 1% of EVM. Considering that most previous treatments dealt with the bandwidth of several MHz to hundreds of MHz, and our testbench in simulation works with 2.5Gsamples/s at 45GHz carrier, this constitutes a significant improvement in linearity metrics within such wide bandwidth. With the confirmation from the successful iteration that a working compensator does exist, we move next to find the reduced-complexity inverse system model. Through the analysis of the signal-propagation through the nonlinear system, we develop an approximate model structure of the inverse system. The structure includes the concatenation of a nonlinear system with short memory and a special type of LTI system whose discrete Fourier transform has discontinuities at $\pm\pi$. The model proves its effectiveness with fitted parameters from a simulation testbench, yielding nearly 10dB improvement in ACPR and up to 2% of EVM. Finally, we build an integrated transmitter system with the digital baseband capable of SCS functionality, as well as real-time compensation, for a transmit chain with a phase modulator, a 16-way PA and its power supply switching network. The digital baseband test results are shown in this thesis, and the overall system test

process is still an ongoing work.

1.3.3 Extension of System Modeling to Hierarchical System Optimization

As an extension of the reduced-complexity modeling discussed in the previous section, we incorporate it into a hierarchical system optimization methodology. This system design methodology aims to help the system designers to allocate resources and specifications optimally for each block and sub-block of the system. This is achieved by decreasing the dimensions of the design space through the Pareto surface generation for each block, as well as the creation of the parameterized model of each block. In the thesis, we demonstrate the two enabling techniques: Pareto surface generation and the parameterized system modeling on the operational amplifier examples.

1.4 Thesis Overview

The thesis is organized as follows. Chapter 2 is devoted to the SCS design for the AMO digital baseband. It introduces the proposed piece-wise-linear algorithm to realize the nonlinear function computations. With this algorithm, we show the system and block architectures fulfilling the SCS computation in the chip design. The chip was tested and demonstrated state-of-the-art performances in energy and area-efficiency.

Chapter 3 presents the compensator design for the AMO/LINC PAs. We first demonstrate the successful off-line iteration scheme to solve for the compensated sequence for any particular input sample sequence. Next, we analyze the nonlinear system and find out the equivalent nonlinear system dynamical model structure which is used to fit the inverse model of the PA nonlinear baseband-equivalent system. We show the effectiveness of the model by applying the input sequence first to the compensator and use its output at the PA input, with the output of the PA showing improvements of 10dB in ACPR and up to 2% of EVM. Furthermore, the limitation of the modeling is investigated. Importantly, floating-body effect in the SOI process

is found the main cause for the compensation performance limits in both dynamical model and off-line iteration as verified through the comparison design for body-tied AMO PA design.

Chapter 4 discusses the hierarchical system optimization methodology and shows its dependence on the system modeling techniques. We use the proposed equation-based robust optimization method to generate the Pareto surface and employ the modeling techniques to create the parameterized dynamical model for each block. Realization of the two critical points enables an optimized system across different levels.

Chapter 5 concludes the thesis and suggests several directions for further research.

Chapter 2

Nonlinear Signal Processing in a Digital Baseband Design of RF Transmitter

The trend towards high-throughput and portability, driven mostly by the consumer market, is currently pressing the RF transmitter to have ever increasing efficiency and linearity. Even in cellular base stations, the drive toward flexible, multi-standard radio chips increases the need for high-precision, high-throughput and energy-efficient backend processing. In this chapter, we use an outphasing PA system as an example to demonstrate an energy-efficient high-throughput digital baseband design. With an improved linearity and power efficiency trade-off, outphasing PA places more challenges on its digital baseband with a complex nonlinear signal processing block. When combined with a target PA application in wideband millimeter-wave regime, the design becomes even more challenging due to the requirements from high precision, high throughput and stringent power budget. The algorithm we will show in this chapter is able to provide a solution to satisfy the requirements from all different aspects. We see that the algorithmic design success is due to our ability to take full advantage of the energy-efficient and area-efficient hardware blocks in the design process.

2.1 Outphasing Power Amplifier Background

Fundamentally, the idea of the outphasing PA is to decompose the transmitted signal from the Cartesian domain to two signals with polar modulation or its variations. By use of a more efficient and nonlinear switching-mode PA with phase modulation, the overall system's efficiency can be enhanced without sacrificing linearity. There are several popular outphasing architectures, such as linear-amplification-by-nonlinear-component (LINC) proposed by Cox [31], multi-level LINC (ML-LINC) [32], and asymmetric-multilevel-outphasing (AMO) [33–35]. In the following sections, we will use the AMO architecture as our example. However, the algorithm developed for function synthesis in the AMO can be readily applied to broader set of PA designs and RF applications.

2.1.1 LINC and AMO Systems

Both LINC and AMO PAs are outphasing PA architectures and their digital basebands perform similar computations. The LINC PA architecture is proposed with the motivation to relieve the ever existing trade-off between the power efficiency and linearity performances of the PA. By decomposing the transmitted signal to two constant-amplitude signals, high-efficiency PAs can be used to amplify the two decomposed signals without sacrificing the linearity. The AMO PA architecture, proposed in [33–35] improves the average power efficiency further by allowing the two PAs to switch among a discrete set of power supplies rather than fixing a single supply level.

Table 2.1: LINC and AMO SCS Equations.

LINC Equations	AMO Equations
$A = \sqrt{I^2 + Q^2}, \theta = \arctan(\frac{Q}{I})$ (linc1)	$A = \sqrt{I^2 + Q^2}, \theta = \arctan(\frac{Q}{I})$ (amo1)
$\alpha = \arccos(\frac{A}{2a})$ (linc2)	$\alpha_1 = \arccos(\frac{a_1^2 + A^2 - a_2^2}{2Aa_1})$ $\alpha_2 = \arccos(\frac{a_2^2 + A^2 - a_1^2}{2Aa_2})$ (amo2)
$\varphi_1 = \theta + \alpha, \varphi_2 = \theta - \alpha$ (linc3)	$\varphi_1 = \theta + \alpha_1, \varphi_2 = \theta - \alpha_2$ (amo3)
	$f(\varphi_1) = \frac{1}{1 + \tan(\varphi_1)}, f(\varphi_2) = \frac{1}{1 + \tan(\varphi_2)}$ (amo4)

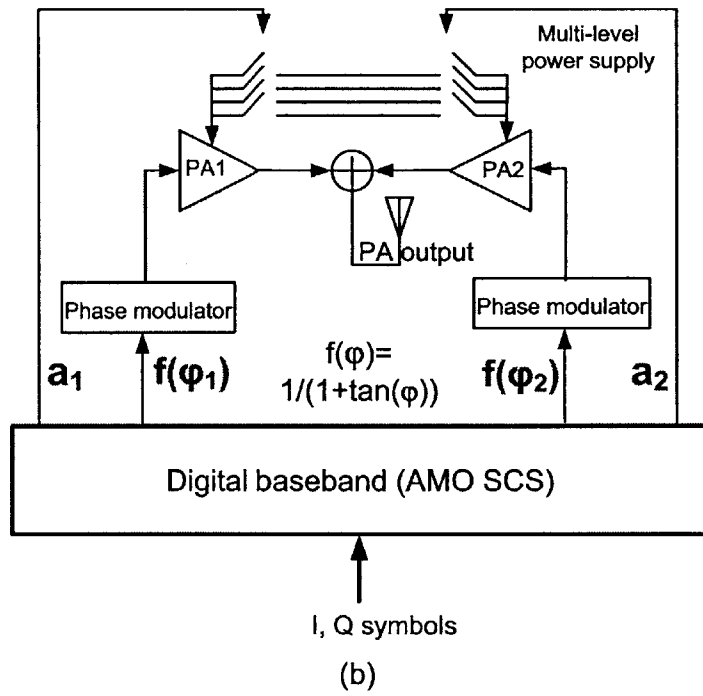
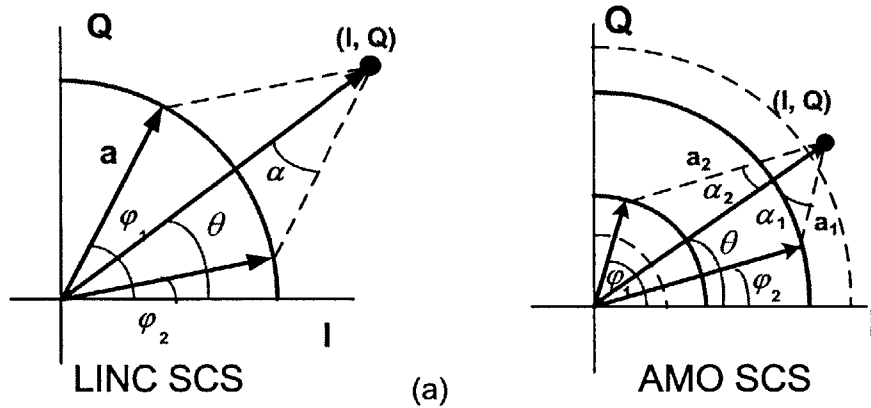


Figure 2-1: (a) LINC, AMO SCS. (b) AMO PA system overview.

Fig. 2-1(a) shows the working schemes of LINC SCS and AMO SCS for an arbitrary IQ sample (I,Q). The SCS decomposes the (I,Q) to two signals with phases of φ_1, φ_2 and amplitudes of a_1, a_2 , where for LINC $a_1 = a_2 = a$. The outphasing angles φ_1 and φ_2 for both architectures are derived from the equations summarized in Table 2.1. In AMO equations, a_1, a_2 denote the power supplies of the two PAs respectively. a_1, a_2 are restricted to the set of $\mathcal{V} = \{V_1, V_2, V_3, V_4\}$, where $V_1 \leq V_2 \leq V_3 \leq V_4$ are the four levels of supply voltages. Equations in (amo4) of Table 2.1 are in the signal decomposition process simply due to the architecture requirement from the digital-to-RF-phase-converter (DRFPC) [36], which converts the digital outputs to RF modulated signals and takes a function of the phase $f(\varphi)$ as the input. Generally, computations in (amo4) depend on the type of the modulator and may be different than what we present here.

The typical low-throughput LINC SCS and recent AMO implementations [16–19, 37] usually involve the use of coordinate rotational digital computer (CORDIC) [20] and LUT map for the nonlinear functions in Table 2.1 [18, 38]. The maturity of the CORDIC algorithm and simplicity of the LUT approach make themselves suitable for the LINC SCS applications whose throughput is below 100MSamples/s and with low to medium resolution (≤ 8 bits for example). However, the approaches become less attractive or even prohibitive for our target mm-wave wideband applications where the throughput is in the multi-GSamples/s range with high phase resolution (≥ 10 bits for example). In the next section, we show our proposed solution: using fixed-point PWL approximations on the nonlinear functions which provides a balance among accuracy, power and area.

2.2 Piece-wise Linear Approximation

2.2.1 Algorithm

The motivation for a new approach to the nonlinear function computation is to replace complex computations with simple and energy-efficient computations. For example,

table look-up with LUTs of reasonable sizes, adders and multipliers are the favorable computations to perform. We also realize that all functions involved in the SCS computations are smooth in almost the whole input range. Hence, they are suitable to be approximated by functions with simple structured basis functions, such as polynomials, splines and etc. These considerations lead us to the PWL function approximation of the nonlinear functions.

Fig. 2-2(a) shows the general application of the PWL approximation to any smooth nonlinear function. The input x is divided into several intervals, where a linear function $y_i = a_i \times x + c_i$, $x \in [x_i, x_{i+1})$ is constructed in each interval to approximate the actual function value in that range. With this approximation, the computation of the nonlinear function only consists of the linear function computation in each interval (add and multiply), plus a relatively small LUT for the linear function parameters a_i , c_i in each interval. In terms of accuracy, for any function which has a continuous second-order derivative, the approximation error is bounded by the interval length, the second-order derivative and does not depend on higher-order derivatives, as shown in [39],

$$|error| \leq \frac{1}{8}(x_{i+1} - x_i)^2 \max_{x_i \leq x \leq x_{i+1}} |y''(x)|. \quad (2.1)$$

Here, x_i , x_{i+1} are the boundaries of the i^{th} interval and y'' is the second-order derivative in x . We observe that the approximation error can be made arbitrarily small as we increase the number of approximation intervals. These initial examinations of the computational complexity and approximation accuracy of the piece-wise linear approximation make it an appealing alternative technique for the LINC and AMO SCS designs.

In order to benefit from the nice properties of the PWL approximation, we need to tailor it to be hardware-implementation friendly. Most importantly, all the arithmetic computations have to be converted to their fixed-point counterparts, and the question is whether the resulting fixed-point computations are able to operate at multi-GSamples/s throughputs with high accuracy. The most seemingly obvious solution is a direct quantization of the parameters in the floating-point representation

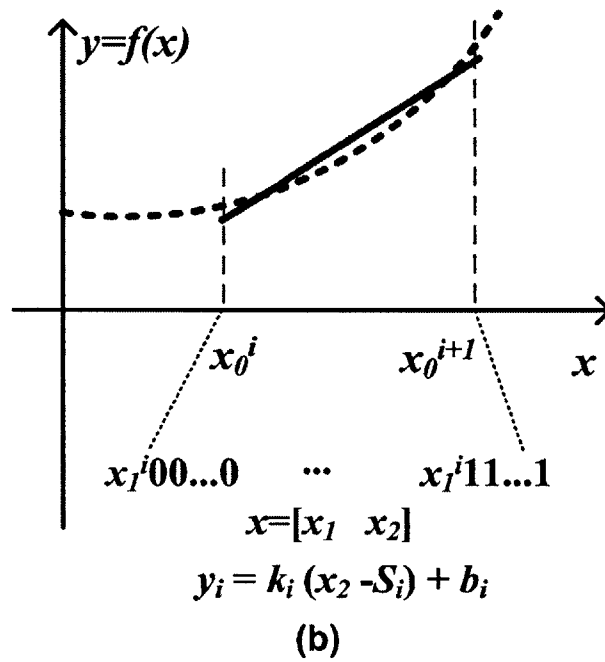
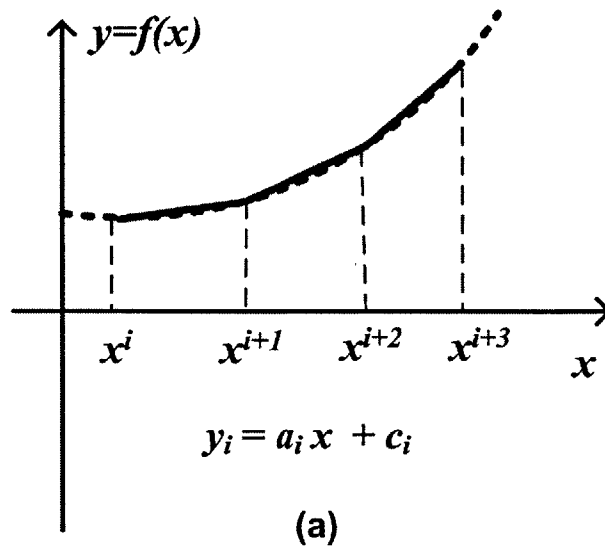


Figure 2-2: (a) The general concept of PWL approximation. (b) Proposed fixed-point PWL approximation.

of the approximation formula. However, this may not be an optimal solution if throughput is the major concern and bottleneck, because the operands of the add and multiply a_i, c_i are quantized to have the same long bits as the output, and these long-bit arithmetics are likely to be in the critical timing path. Further optimization of the long multiplication would only add complexity to the design. In what follows, we present a modified formulation of the fixed-point PWL approximation and show its capability of running at a much higher throughput than the direct quantization version of the approximation.

The setup of our problem is to compute a nonlinear function of m -bit output with m -bit input $x \in [0, 1)$, using the PWL approximation. An m -bit input x can be decomposed to x_1 and x_2 as $x = [\underbrace{x_1}_{m_1\text{-MSB bit}}, \underbrace{x_2}_{m_2\text{-LSB bit}}]$, where $m = m_1 + m_2$. Naturally, x_1 divides the input range to 2^{m_1} intervals and it is the indexing number of those intervals. Fig. 2-2(b) shows an enlargement of the i^{th} interval of the approximation, where x_1 takes its i^{th} value, and x_2 takes 2^{m_2} values, ranging from 0 to $2^{m_2} - 1$. Under this setup, we have our proposed fixed-point scheme shown in (2.2).

$$y_i = \underbrace{b_i \cdot \mathbf{1}}_{m_1\text{-MSB bit}} + \underbrace{k_i(x_2 - S_i \cdot \mathbf{1})}_{m_2\text{-LSB bit}}, \quad i = 0, 1, \dots, 2^{m_1} - 1. \quad (2.2)$$

Here, $y_i = [y([i, 0]), y([i, 1]), \dots, y([i, N_2 - 1])]^T$, $x_2 = \frac{1}{N}[0, 1, \dots, N_2 - 1]^T$, $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathbb{R}^{N_2}$, $N_1 = 2^{m_1}$, $N_2 = 2^{m_2}$, $N = 2^m$, $m = m_1 + m_2$, $k_i, S_i, b_i \in \mathbb{R}$ and they are all fixed-point numbers.

The underlying idea of this formulation is to compute the m -bit output part by part. In the linear function of each interval, we use the term b_i to represent the most significant m_1 bits of the function value, and the term $k_i \cdot (x_2 - S_i \cdot \mathbf{1})$ to achieve the lower-significant m_2 bits of accuracy. Then y_i is simply the concatenation of the two parts. The procedures to find the fixed-point representations of the three parameters k_i, S_i, b_i in (2.2) are described in the following steps.

Step 1: Obtain the floating-point version of the PWL approximation.

The optimal real coefficients of the linear function in each interval in terms of the l_2

norm can be found by least-square optimization (2.3), where the design variables are k_i^r and $b_i^r \in \mathbb{R}$. The superscripts denote that they are floating-point real numbers; x_2 and y_i are defined as in (2.2).

$$\min_{k_i^r, b_i^r} \| y_i - (k_i^r \cdot x_2 + b_i^r \cdot \mathbf{1}) \|_2, \quad \text{for } i = 0, 1, 2, \dots, N_1 - 1, \quad (2.3)$$

The approximation error bound in (2.1) shows that the error is proportional to $(x_{i+1} - x_i)^2$, which in the fixed-point input case, equals 2^{-2m_1} . Let $m_1 = \lceil m/2 \rceil$, then it is possible to realize the required output m -bit accuracy with only $2^{\lceil m/2 \rceil}$ intervals. Since the number of intervals determines the number of address bits of the LUT that stores the parameters of the linear function in each interval, this LUT ($2^{\lceil m/2 \rceil}$ entries) is considerably smaller than a direct map from input to output (2^m entries). The following steps determine the fixed-point parameter values, i.e., the content of the LUT.

Step 2: Obtain the fixed-point value b_i .

b_i can be achieved simply by quantizing the b_i^r to m_1 -bit. As we mentioned before, the m -bit output is constructed part by part with b_i as the constant term in the i^{th} interval, representing the major part of the function value in that interval. As long as the functional value increment in each interval is less than 2^{-m_1} , that is, the functional derivative $|y'(x)| < 1$, it is enough to use the m_1 -MSB of b_i to represent the m_1 -MSB of the output.

Step 3: Obtain the fixed-point value S_i .

Since Step 2 yields a b_i with a maximum quantization error of 2^{-m_1} , to compensate for the accuracy loss of $b_i^r - b_i$, an extra parameter S_i^r is introduced such that $k_i^r S_i^r = b_i^r - b_i$. Its fixed-point counterpart S_i is derived as in (2.4)

$$S_i = \text{quantize}((b_i^r - b_i)/(k_i^r)). \quad (2.4)$$

The number of bits of S_i is determined such that $k_i^r S_i$ has the accuracy of $m + 1$ bits. From our experience with the functions involved in the SCS design, S_i usually has the number of bits around or a few more (i.e. 2-4) bits than $m/2$, depending on the

derivative k_i of the function in each interval.

Step 4: Obtain the fixed-point value k_i .

The slope of the function in the i^{th} interval k_i can also be obtained by simply quantizing its floating-point counterpart from the optimization procedure in Step 1. As shown in (2.2), the term $k_i(x_2 - S_i \cdot 1)$ contributes to the second part of the output - the m_2 LSBs. Since $x_2 - S_i$ has an accuracy of at least m bits, k_i has to have at least m_2 bits to make the m_2 LSBs of the output.

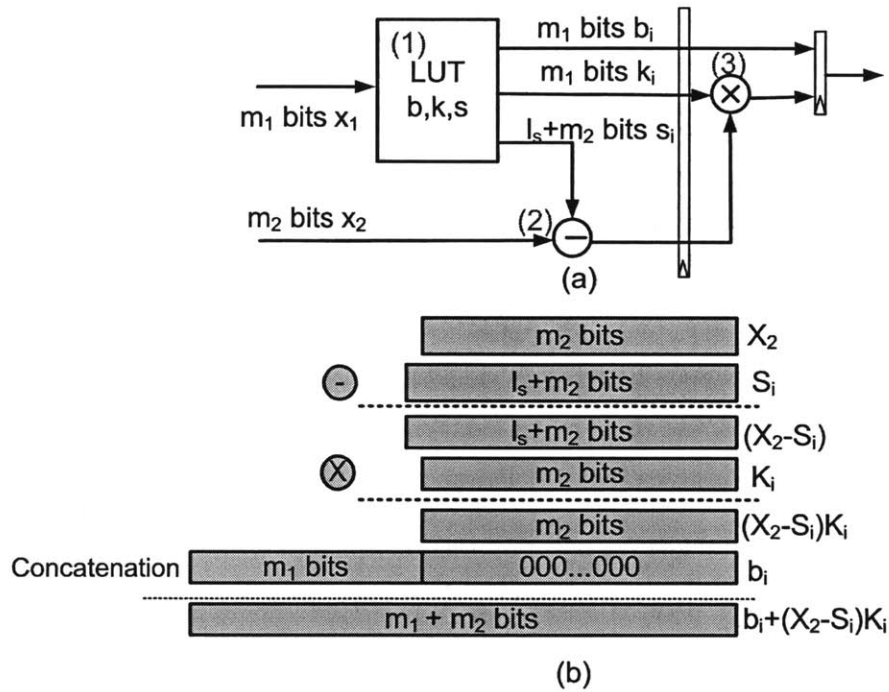


Figure 2-3: (a) Micro-architecture of the PWL approximation. (b) Illustration of the computations in the PWL approximation.

The above procedure not only provides a way to obtain the three fixed-point parameters of the linear function in each interval, but also provides benefit in the high-throughput hardware micro-architecture design. Fig. 2-3(a) shows the micro-architecture of the approximation and (b) shows more clearly how the computations are carried out. There are essentially 3 arithmetic operations involved: LUT, one adder, and one multiplier. The LUT takes the m_1 MSBs of the input as the address

and outputs the parameters b_i , k_i , S_i in the corresponding interval. Then the linear function computations follow accordingly. From Fig. 2-3(a), we notice that for all arithmetic computations, the operands have only m_1 , m_2 or $l_s + m_2$ bits, but not m bits as input. As we discussed in Step 1, it is a good choice to set $m_1 = \lceil m/2 \rceil$, hence with operands of $m/2$ bits (roughly) in all computations, we are able to achieve the m -bit output.

Table 2.2: Storage comparison examples between a direct LUT map approach and fixed-point piece-wise linear approximation approach.

m	Direct LUT size L1 (bits)	Approx. LUT size L2 (bits)	Improvement ratio(L1/L2)
10	10×2^{10}	20×2^5	2^4
12	12×2^{12}	24×2^6	2^5
14	14×2^{14}	28×2^7	2^6
16	16×2^{16}	32×2^8	2^7

Table 2.3: Comparison between PWL, CORDIC implementations of the 16-bit input, output function $y(x) = \cos^{-1}(x)$.

	Minimal clock period(ps)	Power consumption (mW) (post-extraction simulation)	Area ($\mu\text{m} \times \mu\text{m}$), Density (%)	Energy per operation (pJ/op)
Proposed PWL (hardwired LUT)	792	3.24 (at 1GHz)	80×60 , 80%	3.24
Proposed PWL (programmable LUT)	856	7.23 (at 1GHz)	250×240 , 77.5%	7.23
Unrolled radix-4 CORDIC	2600	63.1 (at 400MHz)	220×200 , 81.4%	157.75
6th order polynomial	250	42 (at 1GHz)	200×200 , 70%	42

This implies two important improvements in hardware efficiency: storage and throughput. For a direct LUT implemented function, if both the input and output have m bits, the storage required is $m \cdot 2^m$. With the proposed scheme, the storage

is $(2m_2 + l_s + m_1) \cdot 2^{m_1}$, which is approximately $1.5m \cdot 2^{m/2} \sim 2m \cdot 2^{m/2}$ assuming $m_1 = m_2 = m/2$ (when m is even) and l_s small (≤ 4). A comparison on the storage usage between the direct LUT map and the fixed-point PWL approximation approach is illustrated in Table 2.2, for practical range of m from 10 to 16. The last column of the table shows the ratio of LUT size from approximation versus the one from direct LUT map, which reflects the storage savings of 10-100x for the range of values of interest. The net area advantage of our approach versus the direct LUT will depend on the actual technology and throughput specifications, since these would dictate the type of the storage elements being used. In high-throughput applications, register-based LUTs are needed while in lower throughput conditions, SRAM-based LUTs can be used. Under both types of LUT implementations, the additional area consumption brought by one adder and one multiplier is almost negligible compared to the LUT area. For example, in 45nm SOI technology, the direct LUT implementation of a 16-bit in/out arccos function consumes an area of 19mm^2 in the register-based implementation and 0.7mm^2 in the SRAM implementation. With the PWL approximation, area consumption reduces to $46200\mu\text{m}^2$ with register implementation and $9784\mu\text{m}^2$ with SRAM. The adder and multiplier consume roughly $1280\mu\text{m}^2$ in total, which is only a small portion compared to the overall area consumption. Obviously, the PWL approximation has a large advantage in storage size and the advantage becomes more prominent as the input and output size increases. As for the throughput, because of the short operands and LUT address, the whole chain of operations: LUT, add and multiply can be easily pipelined into a few stages depending on the process and throughput requirement. For example, with a 45nm SOI process, we use two pipeline stages: table lookup, adder in the first pipeline stage and multiply in the second pipeline stage, and this structure can sustain roughly a 2-GSamples/s throughput to compute a 15-bit input and output nonlinear function.

As a side note, an alternative way to write our formulation (2.2) is

$$y_i = k_i \cdot x_2 + (-k_i S_i \cdot 1 + b_i \cdot 1) = k_i \cdot x_2 + c_i. \quad (2.5)$$

To compare the two formulations, we consider the following two aspects: storage size and arithmetic computation complexity. In terms of storage size, formulation (2.2) requires $(m_1 + m_2 + m_2 + l_s) \cdot 2^{m_1} = (2m_2 + m_1 + l_s) \cdot 2^{m_1}$ bits while (2.5) requires $(m_1 + m_2 + m_2) \cdot 2^{m_1} = (2m_2 + m_1) \cdot 2^{m_1}$ bits. Formulation (2.2) does require a little bit more storage of $l_s \cdot 2^{m_1}$ bits. However, it brings the advantage of shorter operands of the add operation. In terms of arithmetic operation complexity, formulation (2.2) requires an adder with $m_2 + l_s$ and m_2 -bit operands, multiplier with $m_2 + l_s$ and m_2 -bit operands, while (2.5) requires an m -bit full adder and m_2 -bit multiplier. As m gets large, the long adder in (2.5) may need further pipelining and complicates the design at high throughput. Furthermore, the optimization lets b_i represent the first m_1 bits while it chooses k_i and S_i in (2.2) so that $k_i(x_2 - S_i)$ exactly represent the rest of the m_2 bits, to avoid any overflow and an additional adder. Our design is more throughput rather than area-limited, therefore with the above considerations, we choose to use formulation (2.2) to achieve a higher throughput with more compact arithmetic hardware.

2.2.2 Piece-wise-linear Design Example

In this section, we show an example of computing a normalized 16-bit input, 16-bit output arccosine function $y = \arccos(x)/(2\pi)$ using the proposed PWL approximation approach. This function is one of the functions in the actual AMO SCS design.

First, we obtain a floating-point representation of the PWL approximation through the following least-square minimization:

$$\min_x \| Ax - \beta \|_2, \text{ where} \tag{2.6}$$

$$A = \begin{bmatrix} 1, & 1, & \dots, & 1 \\ \frac{0}{N^2}, & \frac{1}{N^2}, & \dots, & \frac{N-1}{N^2} \end{bmatrix}_{N \times 2}^T, x = \begin{bmatrix} b_0^r & b_1^r & \dots & b_{N-1}^r \\ k_0^r & k_1^r & \dots & k_{N-1}^r \end{bmatrix}_{2 \times N},$$

$$\beta = \begin{bmatrix} y_{0,0} & \cdots & y_{N-1,0} \\ y_{0,1} & \cdots & y_{N-1,1} \\ \vdots & \ddots & \vdots \\ y_{0,N-1} & \cdots & y_{N-1,N-1} \end{bmatrix}_{N \times N}$$

Here, $N = 8$, half of the number of input bits; $y_{i,j} = y([i, j]) = \arccos((2^N i + j)/2^{2N})/(2\pi)$, $i, j = 0, 1, \dots, N-1$, and i acts as the address for the LUT. The optimal floating-point parameters b^r, k^r yield a maximum absolute error $< 2^{-16}$ for the input range $x \in [0, 0.963]$. For input $x \in (0.963, 1]$, the PWL approximation does not behave as well because of the large derivative value when the input approaches 1. However, this case only happens when the input sample vector nearly aligns with the two decomposed vectors, namely A is approaching $a_1 + a_2$ and $\alpha_1, \alpha_2 \rightarrow 0$. One solution is to redefine the threshold values such that those samples use a set of higher level of power supplies so as to avoid the situations of $\alpha_1, \alpha_2 \rightarrow 0$.

Then, we quantize the terms b^r and k^r to 8 bits, and use equation (2.4) to obtain the offset S . It turns out that the offset parameter uses 11 bits. The resulting accuracy after all the quantization is $< 2^{-15}$ in terms of maximum absolute error.

Table 2.3 shows the place and route results of the hardware implementation with the proposed approximation approach, as well as other approaches as comparisons. There are two versions of the approximation approach shown with different ways of handling the LUT: one version has the LUT programmable and the other version has it hardwired. The approaches shown as comparisons include CORDIC and a 6th order polynomial approximation. CORDIC [40] is a general iterative approach to implement the trigonometric functions. However, due to its general purpose, it is much less energy-efficient and lower throughput compared to our PWL approximation. The polynomial approximation, as another alternative to approximate the nonlinear functions, requires many more multipliers than the PWL approximation, and hence is also less energy-efficient. To summarize, the proposed PWL approximation provides 6-20x improvement in energy-efficiency with significant area savings over the competing approaches.

2.2.3 Comparison with General Numerical Function Generation

The generation of elementary functions such as trigonometric functions, square-root, and reciprocal is crucial to many high-performance DSP applications, including the baseband signal processing in the AMO/LINC PAs discussed in previous sections. Regardless of the application, general numerical function generation is a well-defined field itself and attracts much research attention. Aside from the iterative CORDIC algorithm, which is usually too slow for high-precision applications, a variety of approximation algorithms have been proposed in the literature [41–44].

Among them, [44] proposes a similar version of the PWL approximation to generate the elementary functions. The proposed PWL algorithm used a non-uniform segmentation and the general formulation (2.5) for the computation of functional values. The non-uniform segmentation has the advantage of less segments, hence more economical in storage compared to uniform segmentation. However, it complicates the hardware of coefficients fetching, and furthermore, the address of the coefficient LUT has to be in full precision and can no longer be divided into two parts with part of the MSBs as the address. As the address becomes too long, the table look-up could potentially be the bottleneck of the hardware speed. For our application of AMO/LINC SCS, since the speed constraints limit the design more significantly than the area constraint, we may not benefit from the non-uniform segmentation. The most critical point that distinguishes our approach from the approach in [44] (as well as most other literature that uses the PWL approximation) is the way that the computation is carried out as discussed in (2.2). With our tailored formula, we maximize the hardware speed-gain with a moderate level of storage consumption. The advantage is most significant where high throughput and precision (> 12 bits) are the major design limits. In other applications where the design limits are different, other approaches may be better suited.

2.3 Chip Implementation

2.3.1 Overall Design

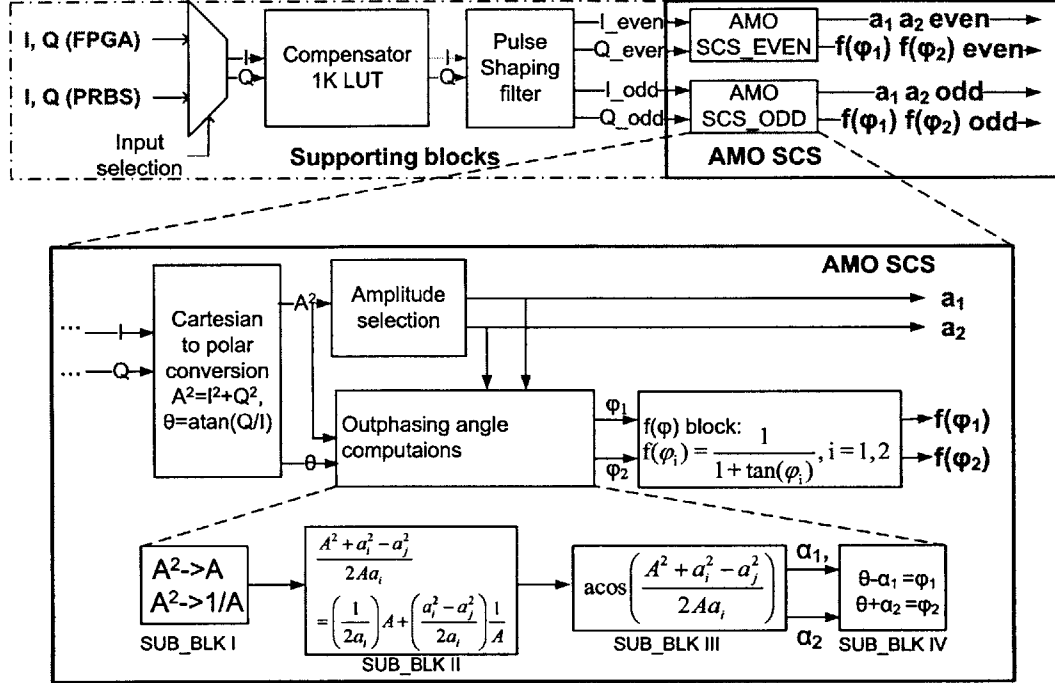


Figure 2-4: The block diagram of the chip.

The baseband design uses the 64-QAM modulation scheme and has the target symbol throughput of 1-2GSym/s. The system has an oversampling rate of 4 or 2, resulting in a system sample throughput of 4GSamples/s. The baseband needs to provide -60dB adjacent channel power ratio (ACPR). In order to meet this specification while overcoming the nonlinearity in the phase modulator DAC [36], the baseband is designed to achieve -65dB ACPR with 12-bit phase quantization.

The baseband system has a block diagram as shown in Fig. 2-4. It includes two parts of the design: supporting blocks and AMO SCS. The supporting blocks up-sample and pulse-shape the input symbol sequence from the 64-QAM constellation to appropriate sample sequences, which are then fed to the AMO SCS blocks. Shown in Fig. 2-4, the 3-bit I and Q symbols first pass through an LUT-based nonlinear

predistorter with a size of $(2^{10}) \times 24$ and produce I/Q symbols with 12-bit accuracy in each dimension. The system is not designed to have a powerful nonlinear predistorter, so this simple predistortion table is added only for preliminary symbol-space predistortion. The table size is chosen such that the predistorter has some memory while fitting in the die area. Then the 12-bit I and Q symbols pass through a pulse shaping filter which oversamples the symbols and produces 12-bit I and Q samples with shaped spectrum. Interleaving is explored here to achieve even higher throughput. The shaping filter produces one sample at the positive edge of the clock and another at the negative edge. Therefore, two copies of the AMO SCS blocks follow the even and odd outputs of the filter.

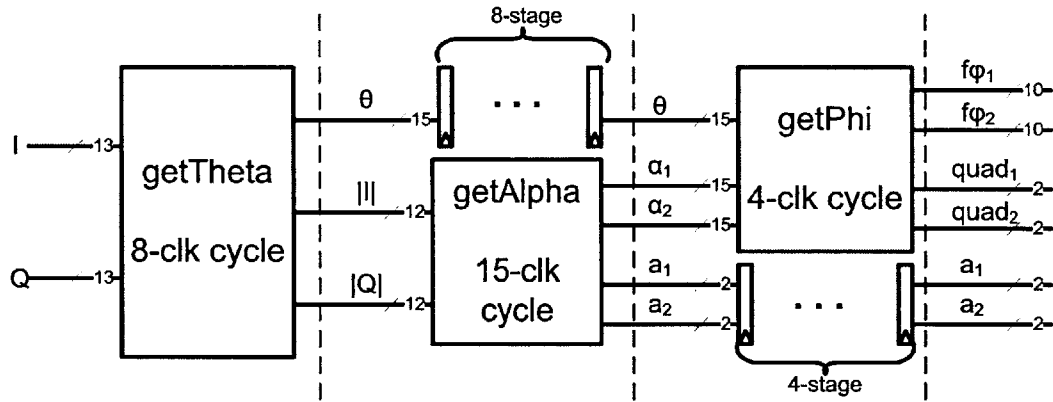


Figure 2-5: The hardware block diagram of the SCS system.

The AMO SCS part, the zoomed-in part in the bottom of Fig. 2-4, consists of four main sub-blocks: the *Cartesian-to-polar* block, *Amplitude-selection* block, *Outphasing-angle-computation* block, and the angle function $f(\varphi)$ block. The *Cartesian-to-polar* block computes the amplitude square and the angle of the I/Q samples in polar coordinates, corresponding to equation (amo1) in Table 2.1.

The *Amplitude-selection* block then takes the value of amplitude square and selects the pair of power supplies for the PAs in the two paths. Recall that the initial motivation to modify the LINC architecture to the AMO architecture is to introduce more supply levels to minimize the combiner loss especially when the outphasing angle

Table 2.4: Criterion for power supply pair selection. ($A^2 = I^2 + Q^2$)

a_1, a_2	Criterion
V_1, V_1	$A^2 \leq th_1$
V_1, V_2	$th_1 < A^2 \leq th_2$
V_2, V_2	$th_2 < A^2 \leq th_3$
V_2, V_3	$th_3 < A^2 \leq th_4$
V_3, V_3	$th_4 < A^2 \leq th_5$
V_3, V_4	$th_5 < A^2 \leq th_6$
V_4, V_4	$th_6 < A^2 \leq th_7$

is large. Therefore, the choice of the power supplies directly affects the average power efficiency. Considering the Wilkinson combiner's efficiency [33] at sample amplitude A and two PA's supply voltages a_i, a_j

$$\eta_c(A, a_i, a_j) = \left(\frac{A}{\frac{a_i + a_j}{2}} \right)^2 \left(\frac{2\left(\frac{a_i + a_j}{2}\right)^2}{a_i^2 + a_j^2} \right), \quad (2.7)$$

we design the criterion shown in Table 2.4 to select the pair of power supplies, where

$$[th_1, th_2, \dots, th_7] = [(2V_1)^2, (V_1 + V_2)^2, (2V_2)^2, (V_2 + V_3)^2, (2V_3)^2, (V_3 + V_4)^2, (2V_4)^2], \quad (2.8)$$

and $V_1 \leq V_2 \leq V_3 \leq V_4$ are the four available power supply levels. The criterion is designed to maximize the combiner's efficiency (2.7) by using the smallest pair of power supplies while the power levels are still large enough to form the transmitted sample. Obviously, there are more than the 7 levels used here that can be designed from 4 supply levels. An important factor that motivates the choice of the 7 levels is the consideration of minimizing the number of switching events for each power supply. Power supply switching is accompanied by ringing and slewing, which introduce nonlinear and memory effects into the system and cause spectrum outgrowth and degradation in the linearity performance of the overall transmitter. The rules in (2.8) make only one adjacent power supply change when the sample amplitude jumps from one region to an adjacent region. This is what happens most of the time because the pulse-shaping filter smooths the I/Q symbol transitions and limits the jumps between

I/Q samples.

The *Outphasing-angle-computation* block computes the two angles between the decomposed and transmitted vectors, corresponding to equations (amo2) and (amo3) in Table 2.1. The steps of the computations are divided into four sub-blocks in Fig. 2-4. Sub-blocks I and II compute the argument of the arccosine function $(A^2 + a_i^2 - a_j^2)/(2Aa_i)$, including square-root, inverse of square-root and summation operations. The terms $1/2a_i$ and $(a_i^2 - a_j^2)/(2a_i)$ in sub-block II are two programmable constants and selected after the determination of two supply levels. Then sub-block III computes the arccosine function and IV computes the final outphasing angles.

The last block of $f(\varphi)$ computation prepares the input signals for the phase modulator we use, which takes the form of $1/(1 + \tan(\varphi))$. The LUT used in this block can also be programmed to compensate for the static nonlinearity of the phase modulator DAC.

As a summary, Table 2.5 lists the arithmetic operations for each functional block.

Table 2.5: Summary of arithmetic operations in each functional block of the AMO SCS.

Functional block		Arithmetic operations
Cartesian-to-polar		multiply, division, arctan
Amplitude selection		Comparator
Outphasing angles	SUB_BLK I	square-root, inversion of square-root
	SUB_BLK II	multiply, add
	SUB_BLK III	arccos
	SUB_BLK IV	add
$f(\varphi)$ block		$\frac{1}{1 + \tan(\varphi)}$

2.3.2 Blocks Design

In this section, we show details of the micro-architecture of each block in the SCS system. Fig. 2-5 shows the overall pipelined hardware block diagram. It is roughly a direct translation from the conceptual block diagram in Fig. 2-4. The I/Q samples

generated by the shaping filter first pass through the *getTheta* block and produce the θ and $|I|$, $|Q|$. The following *getAlpha* block then takes $|I|$ and $|Q|$, selects the two power supplies and computes the angles α_1 and α_2 . This roughly corresponds to the *Amplitude-selection* and *Outphasing-angle-computation* blocks in Fig. 2-4. The angles α_1 and α_2 , together with θ , are inputs to the *getPhi* block, which computes the function $1/(1 + \tan(\varphi))$ on the outphasing angles φ_1 , φ_2 . This represents the $f(\varphi)$ block in Fig. 2-4. The final outputs of the SCS system are $f\varphi_1$, $f\varphi_2$, $quad_1$, $quad_2$, and a_1 , a_2 . Here, $quad_1$ and $quad_2$ are quadrant indicators of φ_1 and φ_2 , respectively; $f\varphi_1$, $f\varphi_2$ are computed with φ_1 φ_2 converted to the first quadrant; a_1 and a_2 are the digital codes that control the PA power supply switches. Next, we see how each sub-block accomplishes its tasks.

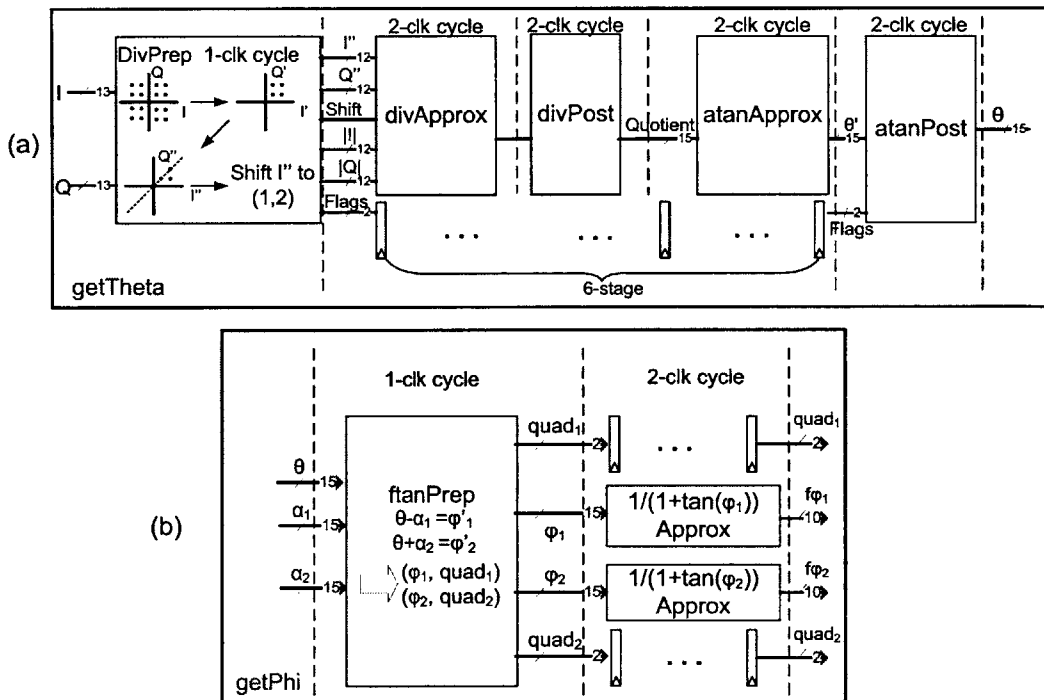


Figure 2-6: (a) The hardware block diagram of the *getTheta* block. (b) The hardware block diagram of the *getPhi* block.

getTheta block

Fig. 2-6(a) shows the micro-architecture of the *getTheta* block, which has two main operations as division and arctan. With the PWL approximation algorithm discussed in Section 2.2.1, both functions can be realized with the micro-architecture in Fig. 2-3. Before applying the approximation, it is important to carefully examine the input and output range of the function, because of the nature of the fixed-point computation. In order to have a good accuracy with the approximation, it is desirable to have an input range where the function behaves smoothly and has a nicely bounded derivative. Consider as an example the division function. The division function Q/I has two input variables, while the presented algorithm assumes a single variable function. So the computation of Q/I is divided into $1/I$, followed by $Q \times (1/I)$. The inversion function $1/I$ has a discontinuity at $I = 0$ and its derivative $-1/I^2$ becomes large as $|I|$ approaching zero. In order to use the PWL approximation with good accuracy, several preprocessing steps are necessary to massage the input before doing the approximation of the inversion function $1/I$. We implement the following treatments on the input, corresponding to the *divPrep* block in Fig. 2-6(a):

- Step (1): (I, Q) are first transformed to the first quadrant as (I', Q') where $I' = |I|$ and $Q' = |Q|$. Use a flag of two bits to indicate whether the current sample (I, Q) is actually negative or not.
- Step (2): Swap I' and Q' if $Q' > I'$, so the resulting (I'', Q'') satisfies $Q''/I'' \in (0, 1)$. The boundary values of 0 and 1 are computed as special cases separately. Again, use a flag to indicate whether the swap is performed on the current sample.
- Step (3): Shift the input I'' such that $I'' \in (1, 2)$. The shift operation is always valid because the shaping filter coefficients are programmable and can be designed such that $I, Q \in [0, 1]$. This step just means shifting the bits in I'' to the left until the MSB is 1. Record the shifted number of bits for each sample I'' .

Although it is obvious that after the transformations, Q''/I'' is different from the desired output Q/I , these preprocessing steps can be compensated. Specifically, the swap in Step (2) and the absolute operation in Step (1) are taken care of after the computation of θ ; and the shift operation in Step (3) is taken care of after the computation of $Q'' \times (1/I'')$.

- Step (1): Shift back accordingly after the computation of $Q'' \times (1/I'')$. This is an operation included in the block of *divPost*, together with the multiplication $Q'' \times (1/I'')$.
- Step (2): After the computation of θ' , for values whose flag indicating a swap operation has happened, $\theta = \pi/2 - \theta'$, otherwise $\theta = \theta'$. This is included in the *atanPost* block in Fig. 2-6(a).
- Step (3): After Step (2), we need to check further if quadrant change has happened to the current sample, and adjust the θ accordingly. This is also a part of *atanPost* block.

With properly designed preprocessing, the input of inversion function $1/x$ takes the range of $(1,2)$, and the input of function $\arctan(x)$ takes the range of $(0,1)$. In these ranges, the functions have nicely bounded derivatives, enabling them to be suitable for the fixed-point PWL approximation. The two functions approximation computations are represented by the blocks *divApprox* and *atanApprox* in Fig. 2-6(a), whose micro-architecture follows the one in Fig. 2-3(a). The overall *getTheta* block is able to achieve a throughput of 2GSamples/s in the place and route timing analysis. The look-up tables that store the b , S , and k for the two functions have sizes as summarized in the first two lines in Table 2.6. The table also gives a size comparison to the LUTs which are used directly to map the nonlinear functions. There, we can see orders of magnitude of LUT size saved by using our fixed-point PWL approximation approach. The accuracy column also shows that an output accuracy of 14 bits is achieved.

Table 2.6: Summary of accuracy and LUT size of the PWL approximated function blocks.

	max error	PWL LUT size	Direct LUT size	Improvement ratio
$1/x$	7e-5	30×2^7	15×2^{12}	4
$\arctan(x)$	6e-5	25×2^7	15×2^{15}	128
\sqrt{x}	2.3e-5	30×2^7	12×2^{19}	1638
$1/\sqrt{x}$	8.2e-5	30×2^7	12×2^{19}	1638
$\arccos(x)$	2.4e-5	30×2^7	15×2^{15}	128
$1/(1 + \tan(x))$	1.6e-5	26×2^7	10×2^{15}	100

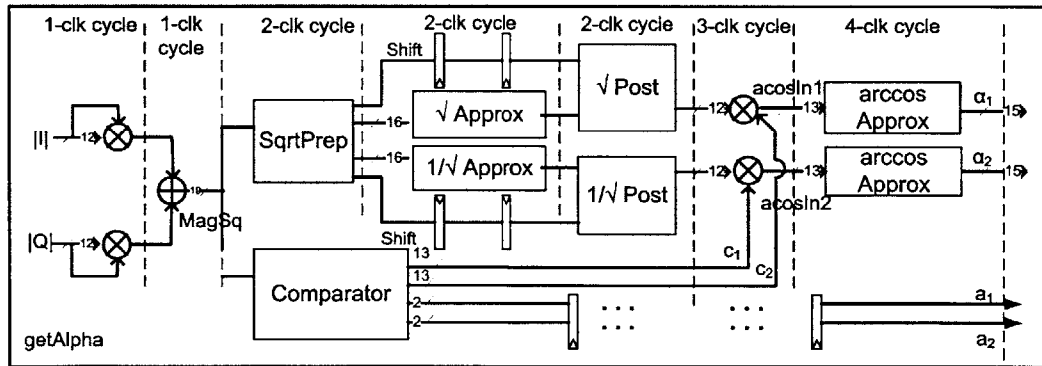


Figure 2-7: The hardware block diagram of the *getAlpha* block.

getAlpha block

Fig. 2-7 demonstrates the detailed micro-architecture of the *getAlpha* block of Fig. 2-5, also corresponding to the conceptual sub-blocks I, II and III of the *Outphasing-angle-computation* part in Fig. 2-4. The α_1 and α_2 computations include two parts: obtain the argument to the arccos function and calculate the arccos function itself. In order to obtain the argument $(a_i^2 + A^2 - a_j^2)/(2Aa_i)$, we rearrange the terms as

$$\frac{a_i^2 + A^2 - a_j^2}{2Aa_i} = c_1A + c_2\frac{1}{A}, \quad \text{and} \quad c_1 = \frac{1}{2a_i}, \quad c_2 = \frac{a_i^2 - a_j^2}{2a_i}, \quad (2.9)$$

where constants c_1 and c_2 are programmable values and are selected according to the selection of power supplies. The problem with using the original formula $(a_i^2 + A^2 - a_j^2)/(2Aa_i)$ is the long-bit division, whose inputs are on the same order of A^2 . On the other hand, (2.9) involves no computations with inputs on the order of A^2 .

The computations to obtain the terms A , $1/A$ in (2.9) include approximations of the functions \sqrt{x} and $1/\sqrt{x}$, whose inputs are the sum of $|I|^2$ and $|Q|^2$. Similarly, as we discussed for the division computation, certain input preprocessing is necessary to avoid the large derivatives near the discontinuity point at 0. The *SqrtPrep* block of Fig. 2-7 serves this purpose by scaling the input to the range of $[1/4, 1)$, namely shifting two bits at a time either to the left or right until the input fits to the range. Then the approximations to the two functions are performed and followed by the postprocessing parts that compensate for the shifting operations done to the inputs. With two more multipliers and one adder, the computations of (2.9) are now accomplished. Then the function $\arccos(x)$ takes the input arguments and obtains angles α_1 and α_2 , which is already shown in the previous example. The LUT sizes and accuracy for the three functions are summarized in Table 2.6.

getPhi block

Shown in Fig. 2-6(b) and as the final block in Fig. 2-5, *getPhi* takes the outputs α_1 , α_2 and θ from the previous *getAlpha* and *getTheta* blocks and produces the final outphasing angles $f\varphi_1$ and $f\varphi_2$. The *getPhi* block first computes the outphasing

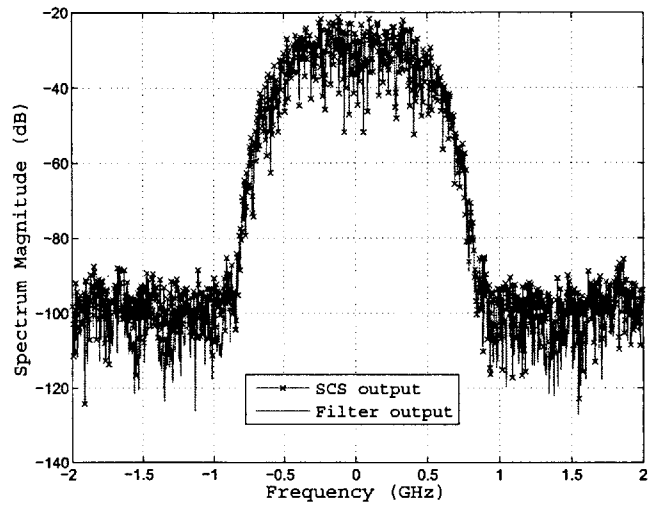
angles φ_1, φ_2 in the sub-block *ftanPrep*, then the $1/(1 + \tan(\varphi))$ block computes the final outputs. Nominally, the digital baseband SCS's tasks end after the *ftanPrep*, delivering the outphasing angles themselves. However, there may be an additional signal processing task at the interface between the digital baseband and the DRFPC phase modulator. In our case, the phase modulator we intend to use requires such a function on the outphasing angle as input.

After obtaining the outphasing angles as $\varphi_1 = \theta - \alpha_1$ and $\varphi_2 = \theta + \alpha_2$, we convert them to the first quadrants and use 2-bit flags *quad₁* and *quad₂* to indicate the quadrants. This conversion is necessary both for the sake of the phase modulator input requirement, as well as acting as a preprocessing step for the following functional approximation. By limiting the input to the first quadrant, the function $1/(1 + \tan(\varphi))$ has a nicely bounded derivative as $-1/(1 + \sin(2\varphi))$ in the range of $[0, \pi/2]$. Otherwise, the function has a discontinuity at $3\pi/4$. So it is suitable to apply the PWL approximation on this function as well. The hardware cost in terms of the LUT size is again summarized in Table 2.6.

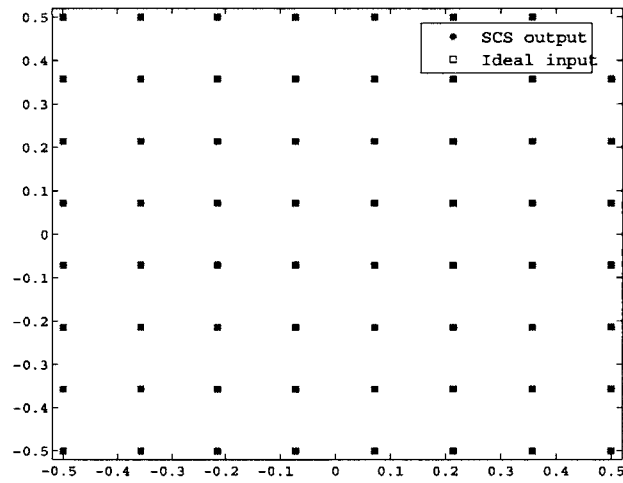
2.3.3 Experimental Results

With all nonlinear functions properly approximated and parameters quantized, the tested SCS output produces the signal spectrum as shown in Fig. 2-8a. Compared with the spectrum at the shaping filter's output, the SCS block reduces the ACPR by 2dB, from 67dB to 65dB, due to the approximation and quantization errors. Fig. 2-8b shows the 64QAM constellation diagram between SCS output and ideal input, illustrating that the SCS introduces EVM of 0.08%.

The digital AMO SCS system is fabricated in a 45nm SOI process, with 448578 gates occupying an area of 1.56mm². The chip runs up to 1.7GHz (3.4Gsample/s) at 1.1V supply. As shown in the shmoo plot of Fig. 2-9, lowering the power supply voltage decreases the dynamic power of the SCS digital system until it hits the minimum-energy point at lower throughput, where leakage energy takes over. The minimum-energy point of 58pJ per sample or 19pJ per bit in 64-QAM transmission (assuming 2x oversampling) is measured at 800MSamples/s throughput. For typical



(a) Spectrum comparison of the SCS output and shaping filter output.



(b) EVM comparison of the SCS output and ideal input.

Figure 2-8: Spectrum and EVM of the SCS.

PA efficiency of 40% and throughput of 800MSamples/s, at peak output power level of 1.8 W, the total peak PAE is affected by less than 1% ($46\text{mW}/(46\text{mW} + 1.8\text{W}/0.4)$) by this 64-QAM capable AMO SCS backend.

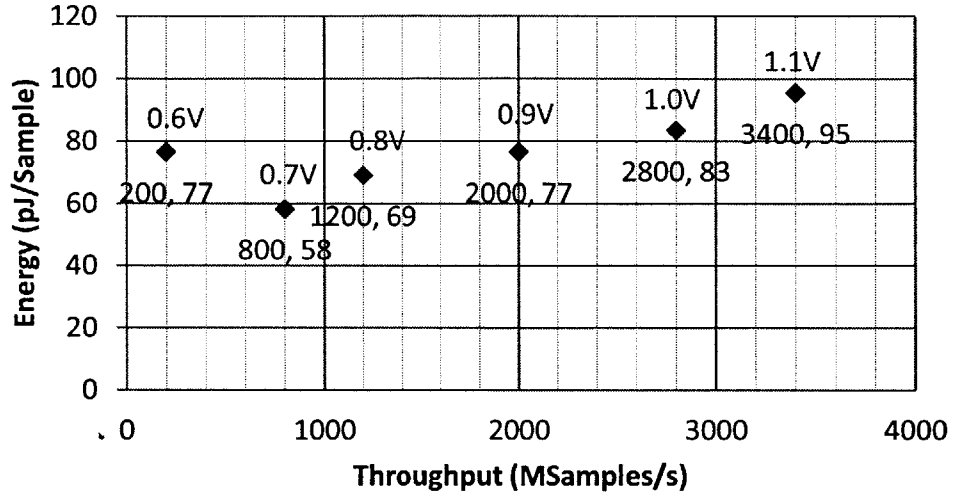


Figure 2-9: Throughput and energy with supply scaling for AMO SCS.

The chip photograph is shown in Fig. 2-10, with annotated blocks and sizes. The power breakdown of the AMO SCS is illustrated in Fig. 2-11(a). Based on the reported post-place-and-route power estimation, the contribution to the total AMO SCS power at 2GHz operation is shown. The large proportion of the clocking power is in part due to the latency-matching register stages on amplitude paths required to compensate for the depth of the phase computations, and the leakage power of the *getPhi* block is due to its programmable LUT of the $f(\varphi)$ function. The area breakdown of the AMO SCS is illustrated in Fig. 2-11(b), which shows the areas of

major functional blocks of the three main functions of the SCS. The computation of the function of $f(\varphi)$ takes over two thirds of the area due to its programmable LUTs. A comparison of our work with other digital/analog implementations of LINC/AMO SCS is summarized in the first 5 columns of Table 2.7. Our work demonstrates a design with the highest throughput and phase accuracy to date. To show a more fair comparison with other digital AMO SCS work, we scaled the design in [19] to provide the same phase accuracy, technology node and throughput. The scaled performances are summarized in the last 3 columns of the Table 2.7, and our design shows more than 2x improvement in energy-efficiency and 25x improvement in area. As a general guideline, for applications with low/medium accuracy (e.g. less than 8-bit phase resolution) requirement and low/medium throughput (e.g. up to hundreds of MSamples/s), LUT is still a good design choice because of its energy-efficiency, reasonable size and low design complexity. On the other hand, our proposed approach is more suitable for applications with high accuracy (e.g. greater than 10-bit phase resolution) and high throughput (e.g. around GSamples/s) requirements.

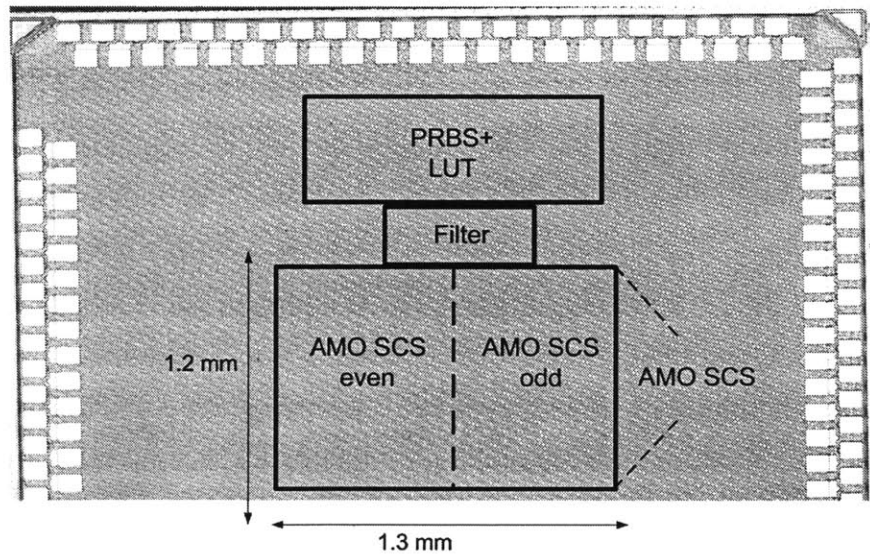


Figure 2-10: Chip photograph.

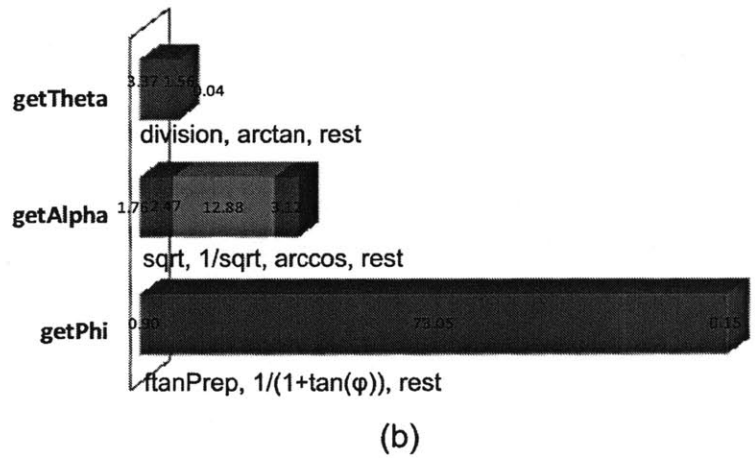
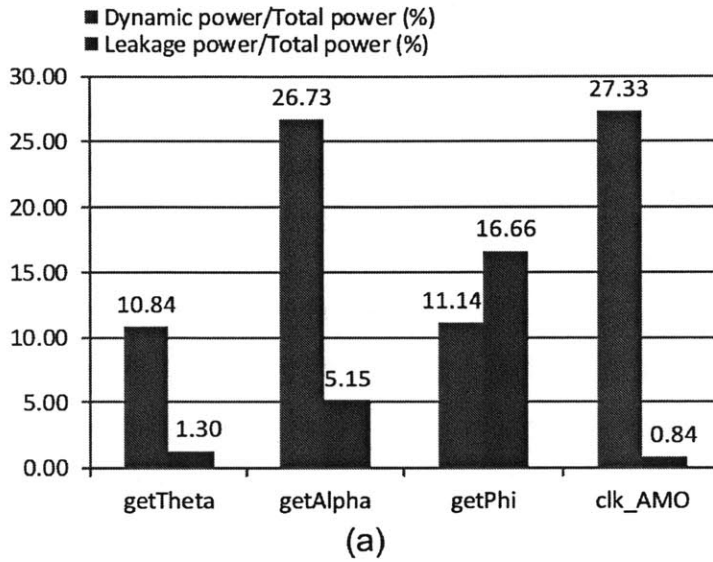


Figure 2-11: (a) Power breakdown of the AMO SCS design. (b) Area breakdown of the AMO SCS design.

Table 2.7: Comparison with other works.

	This work	[17]	[45]	[37]	[19]	[19]	[19]	[19]
Analog/Digital	Analog	Analog	Digital	Digital	Digital	Digital	Digital	Digital
Functionality	LINC	LINC	AMO	AMO	AMO	AMO	AMO	AMO
Technology	0.25 μ m CMOS	0.35 μ m CMOS	45nm SOI CMOS	90nm CMOS	90nm CMOS	90nm CMOS	Scaled to 45nm CMOS	Scaled to 45nm CMOS
Throughput (MSam/s)	3400, 800	20	1.5	50	40	40	40	Scaled to 800MSam/s
Phase Resolu- tion (bits)	12	N/A	N/A	8	8	Scaled to 12-bit	Scaled to 12-bit	Scaled to 12-bit
Power (mW)	323, 46	45	80	0.95	0.36	8.64	4.32	86.4
Energy/Sample (pJ/Sam)	95, 58	2250	5333	19	8.9	212	106	106
Area (mm ²)	1.5	0.1	0.61	0.06	0.34	8.16	2.04	40.8

2.4 Summary

In this chapter, we present the idea of designing algorithms with efficient hardware in mind in the context of a digital baseband SCS design for AMO PA architecture. The application has its own appealing characteristics such as a high average efficiency and relaxed trade-off between efficiency and linearity, is favored by the current trend towards higher throughput and the use of complex modulation schemes to have high spectral efficiency. Furthermore, its digital baseband SCS function design is challenging the traditional digital SCS implementation techniques in terms of area, power efficiency, throughput, precision requirements. These difficulties are typical problems when digital assistance is added to the analog block. In this context, we provide an example of minimizing the digital block footprint while maintaining the desired functionality.

The key to the algorithmic success is that it optimizes the usage of hardware blocks. The fixed-point piece-wise linear approximation we developed has its coefficients optimized such that it only uses the most energy-efficient hardware building blocks and uses them in an efficient way. Specifically, it uses only one LUT, one adder and one multiplier. The sizes of the LUT address, operands of the arithmetic operations are optimized to half of the input length. In this way, the AMO SCS constructed out of the algorithm demonstrates the highest throughput to date, and shows a 25x improvement in area and 2x improvement in energy-efficiency over the projected traditional designs.

Though we only demonstrate the application of the approximation algorithm with the AMO SCS, the approximations are directly applicable to LINC SCS, and enable a new class of wideband wireless mm-wave communication system designs with high energy and spectral efficiency.

Chapter 3

Reduced-complexity System Modeling in Compensator Design for an RF Transmitter

The applicability of digital assistance to the analog/mixed-signal system is not limited to delivering energy-efficient realizations of complex nonlinear signal processing blocks, as we demonstrated in Chapter 2. Digital assistance can also provide the possibility for a large range of applications to reach required high-linearity performance with analog blocks such as ADCs [46] and PAs. In high-throughput communication systems, which combine increased signalling rates with use of more complex modulation for enhanced spectral efficiency, requirements on the system linearity become more stringent. With system modeling techniques, the nonlinearity in the analog system can be compensated with a predistortion or a compensator system, whose model can usually be realized with a digital system. In this chapter, we continue to use the LINC/AMO systems as examples, to demonstrate the effectiveness of the analysis and reduced-complexity modeling in the design of the digital compensators for these LINC/AMO systems.

3.1 Digital Predistortion for PA System

3.1.1 Overview of Popular Digital Predistortion Techniques

The basic idea of a digital predistortion (DPD) technique is to create an inverse system of the nonlinear baseband-equivalent system such that when the two systems are concatenated together, the nonlinearities cancel out and the output is a linear version of the input. For systems with narrowband, static DPD is often utilized to compensate for the memoryless nonlinear behavior [47,48], and LUT is usually enough to implement the DPD. However, for wideband communication systems, memory effects will surely be present in the resulting nonlinear system model. Therefore, more advanced system modeling techniques are needed to model both the PA system itself as well as its inverse [2–4]. Most of the past work applies general nonlinear dynamical system structures in the modeling, such as Volterra series, Wiener, Hammerstein, and Wiener-Hammerstein structures, as shown in Figure 3-1. Some also have different variations of the above structures to make use of the knowledge of the system [49–52]. The choice of the models usually depend on prior knowledge of the system, and in [53], the authors show that the Wiener and Hammerstein models have different engineering interpretations and gives a guide to choose between them.

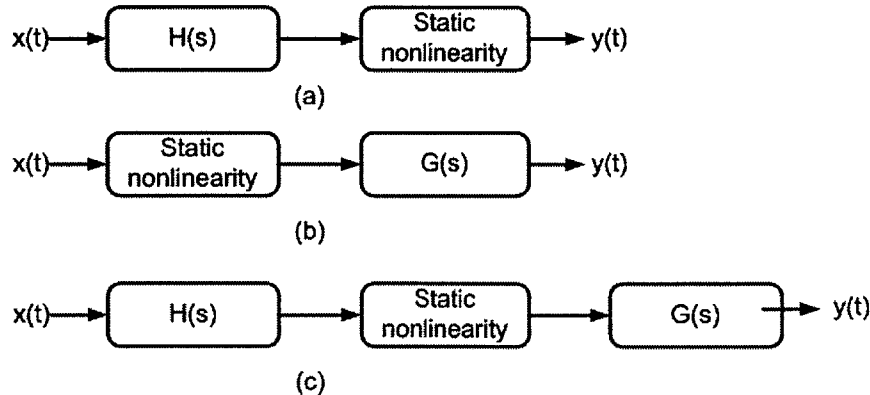


Figure 3-1: Three common nonlinear dynamical system structures. (a) Wiener model. (b) Hammerstein model. (c) Wiener-Hammerstein model.

In this chapter, we will first use off-line compensation to demonstrate the feasibility

of the compensated solution in the baseband. Then, by analyzing this data, we will show the structure of the nonlinear system, as well as its inverse. Rather than using the more general dynamical system structure, the analysis points us to a model structure with which we are able to obtain a decent compensator with parameters computed conveniently through least-square fitting.

3.1.2 Linearity Metrics

The two major metrics to evaluate the linearity performance of the PA system are: error vector magnitude (EVM) and adjacent-channel-power-ratio(ACPR). The EVM measures the ratio of root-mean-square (RMS) error of the received constellation versus the maximal magnitude of the ideal constellation, as

$$\text{EVM} = \frac{\text{Error}_{\text{rms}}}{S_{\text{max}}} \times 100\%. \quad (3.1)$$

The ACPR characterizes the spectral regrowth through a nonlinear communication chain. The nonlinearity in the system causes spurious spectrum emission to adjacent channels and ACPR measures the interference as the ratio of the average power in the adjacent channels versus in the main channel, as shown in

$$\text{ACPR}_{\text{dB}} = 10 \lg_{10} \frac{\text{Average Power}_{\text{adjacent channels}}}{\text{Average Power}_{\text{main channel}}}. \quad (3.2)$$

Figure 3-2 shows the physical meaning of the ACPR definition, where the main and adjacent channels are defined by the particular communication standard.

In the rest of the chapter, we use these two metrics to evaluate the linearity performance of the system before and after compensation.

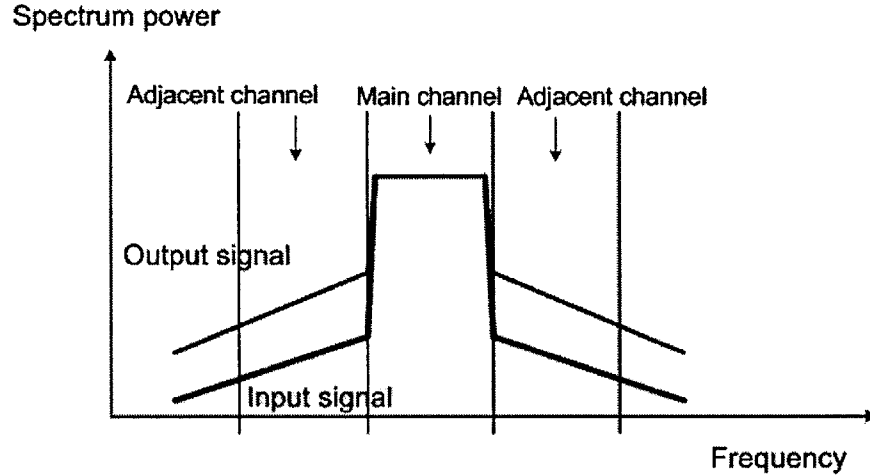


Figure 3-2: An illustration of the ACPR definition.

3.2 Digital nonlinear compensation for the LINC and AMO Systems

3.2.1 System Setup

Due to our limited access to a real LINC/AMO testing system, we confine all the digital nonlinear compensation effort to the simulation domain. Figure 3-3 shows the simulation setup for the LINC/AMO system under compensation. We use this framework to both investigate the overall system nonlinearity, as well as test our nonlinear compensator. As shown in Figure 3-3, random symbols drawn from the 64QAM constellation first pass through the shaping filter, which filters at a higher sampling rate to achieve shaped spectrum. Then the SCS decomposes the shaped samples and produces the information on the two decomposed vectors: amplitude and phase signals for AMO system and phase signals for LINC system. Amplitude and phase commands then pass to the Spectre simulator as the inputs to the PA system. The PA system consists of two switching PAs and an ideal phase modulator and ideal power combiner, which produces the final transmitted signal. To obtain the received samples, we use an "ideal demodulator" which will be explained in 'Demod-

ulation Method' subsection of Section 3.2.2 to demodulate the transmitted signal. In this setup, the blocks simulated with Spectre are the two PAs; the phase modulator is realized with a verilog-A model in Spectre. All other blocks are processed in MATLAB. As a summary, in this setup, we focus on the nonlinearity from the two outphasing PAs, shown as the shaded blocks in Figure 3-3. Path mismatch can also be added intentionally in simulation between the phase and amplitude paths. Both systems are simulated at the carrier frequency of 45GHz with 2.5GHz bandwidth and 2x symbol oversampling rate.

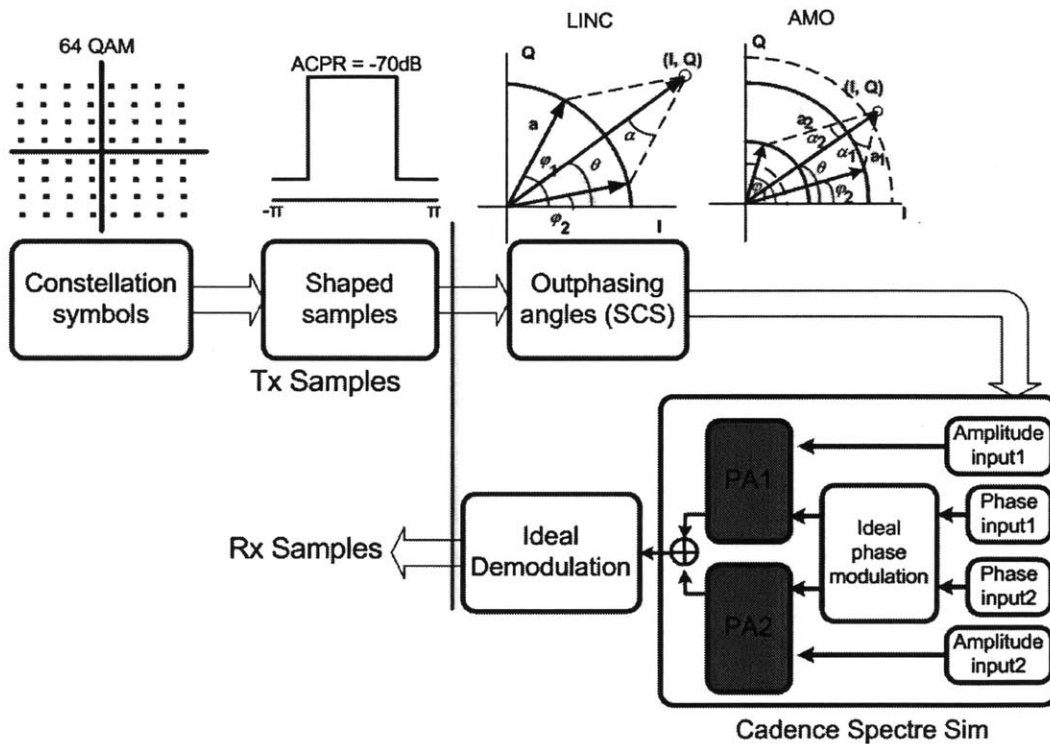


Figure 3-3: PA system under compensation.

The LINC and AMO systems under investigation have the schematic block diagrams as shown in Figure 3-4. They share the same driver and unit PA design. LINC employs one supply at 2.2V while AMO uses four supplies valued at 1.1V, 1.4V, 1.8V and 2.2V. The unit PA [54] used in these two architectures employs a cascode class-E topology whose simplified schematic is shown in Figure 3-5. It is designed to operate

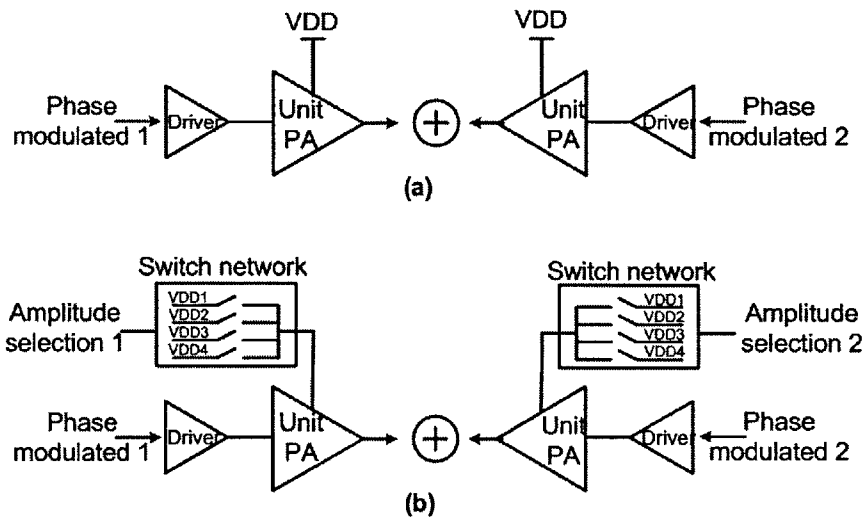


Figure 3-4: LINC and AMO PA architecture block diagrams.

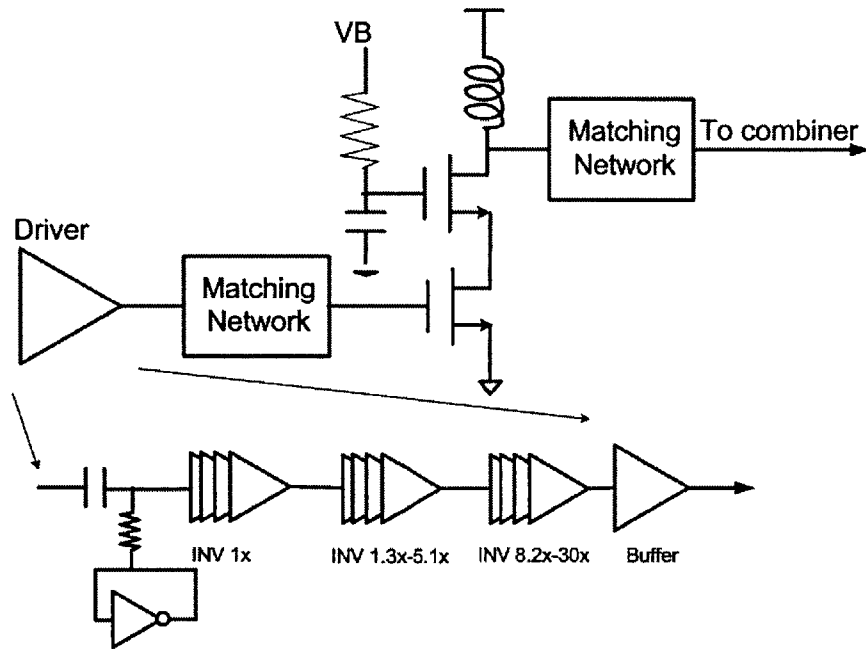


Figure 3-5: Simplified schematics of the cascode class-E PA.

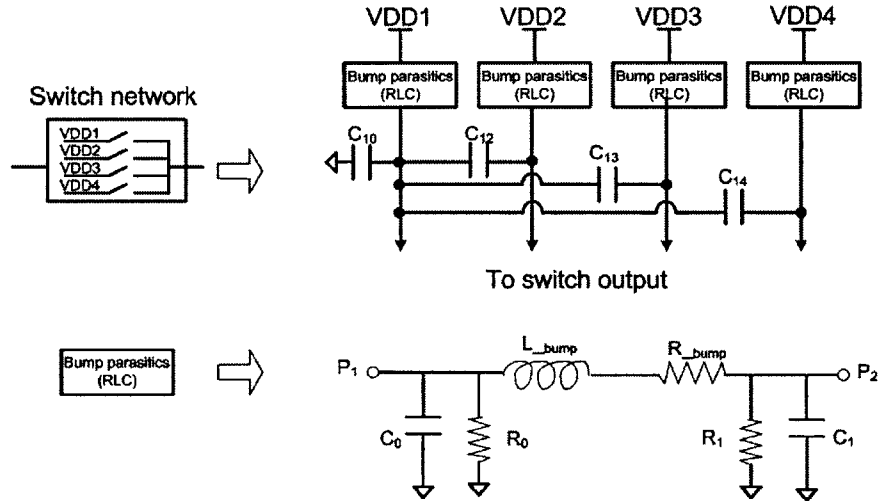


Figure 3-6: Switch network model blocks.

at 45GHz and deliver watt-level power output. The driver design uses an inverter-based topology to maintain a sufficiently large voltage swing at the class-E PA input, as shown in the enlargement part of Figure 3-5.

In terms of the sources of nonlinearity we are able to observe with this setup, the phase modulated signal will surely get distorted along the driver chain as well as the switching PA. The delay variation in the chain as well as phase to amplitude conversion can both introduce the nonlinearity into the driver chain. The requirement to pass the mm-wave carrier frequency makes it more prone to distortion. Furthermore, the two decomposed phase-modulated signals have much wider bandwidth because of the Cartesian to Polar conversion, and this leads to an imperfect cancelation when the two signals are combined together. As a result, the received samples will differ from the transmitted ones and stop-band spectrum rises, leading to the degradation in both EVM and ACPR.

For the AMO architecture, there are a few types of nonlinearity. For example, since the AMO system allows the phase-modulated signal to switch among several discrete power supplies, power supply switching becomes another important source of nonlinearity. To model the effect in simulation, we use a simplified RLC model for the switch network, as shown in Figure 3-6. The switch network models parasitics of the

bump on board, coupling capacitances from each supply to ground, as well as coupling capacitances among supplies. It also includes the RLC network of the interconnect to account for the parasitics of long wires in supply routing. Figure 3-7 shows the responses of the switch network under different bump inductance values of 5pH, 20pH and 60pH. As we can tell from the responses, it takes roughly two samples for a step response to settle in the 5pH case, five samples in the 20pH case and ten samples in the 60pH case. The resulting effect from the power supply switching is potentially an increase in compensator model complexity, in order to take into account of the memory effect.

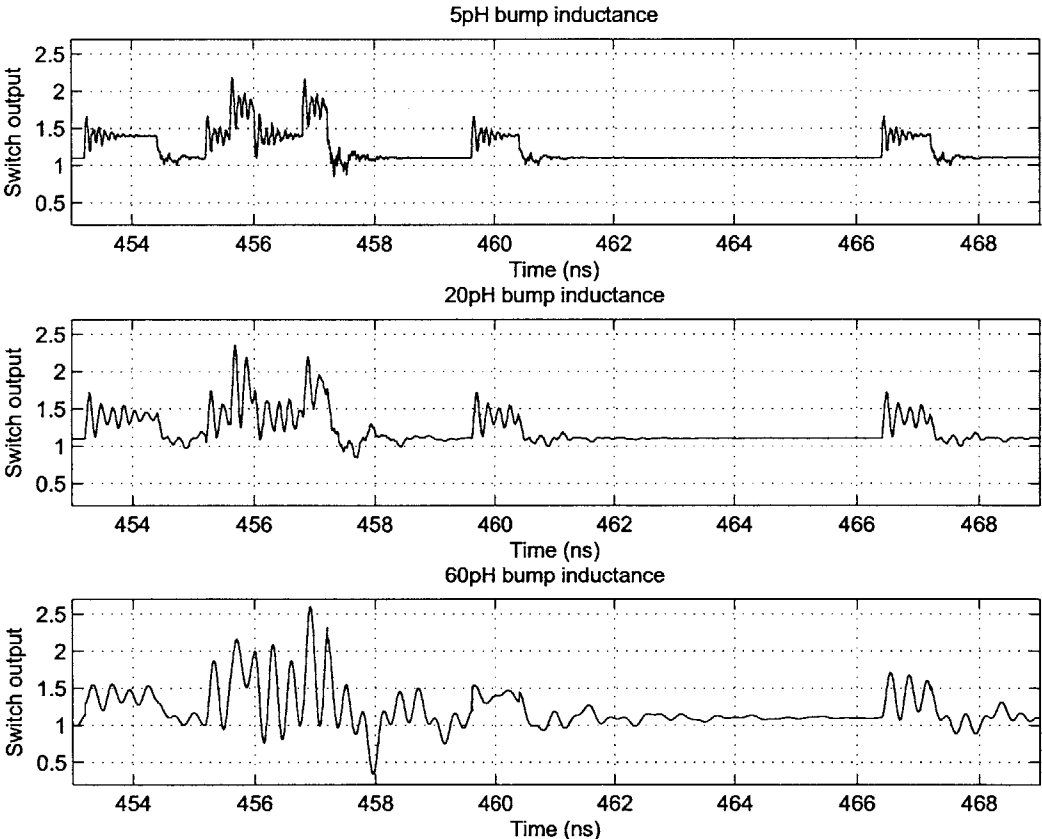


Figure 3-7: Switch network output for different values of bump inductances. 4 VDD levels are 1.1V, 1.4V, 1.8V, 2.2V. Sample duration is 0.4ns.

Another source of nonlinearity associated with the AMO architecture is the path

mismatch, referred to as the mismatch between the amplitude and phase paths for both PAs. Various factors contribute to this nonlinearity, such as mismatch between routing wire lengths, different step-response characteristics between different supply levels, process variations and thermal effects. In real systems, delay-line tuning becomes a sound solution to align the amplitude and phase path signals. However, the alignment cannot be made perfect and is limited to the tuning accuracy, as well as the effectiveness of the calibration algorithm. Therefore, in the AMO nonlinearity compensation simulations, we also experiment with the intentionally added delay between the two paths to estimate the compensator quality.

3.2.2 Iteration-based Off-line Predistortion for LINC/AMO Systems

Iteration Algorithm

Before answering the question of the compensator model's structure, accuracy and complexity, it is illustrative to explore the improvements that can be gained by tailoring the transmitted sequence to each input sample sequence. Under this more relaxed situation, the compensated sequence we are able to produce should outperform any compensator model and hence serves as an upper bound of all possible compensator models. If we have the success of compensating any given sequence in this off-line fashion, then we are assured that a dynamical compensator model does exist.

In order to test whether we are able to obtain the off-line compensator, we abstract the problem as the following. Define $N(x)$ as an aggregate nonlinear function representing the transformation from transmitted samples to received samples. As shown in Figure 3-8, it includes the SCS, LINC/AMO PA system, and the demodulator. Note that Figure 3-8 is only a concept illustration of the function $N(x)$, but not indicating the placement of the actual compensator. Then the off-line compensation has a goal of

$$N(V_c) = V_i, \tag{3.3}$$

where V_i is the desired received samples and V_c is the predistorted sequence, i.e. a solution to the offline compensation question. Define

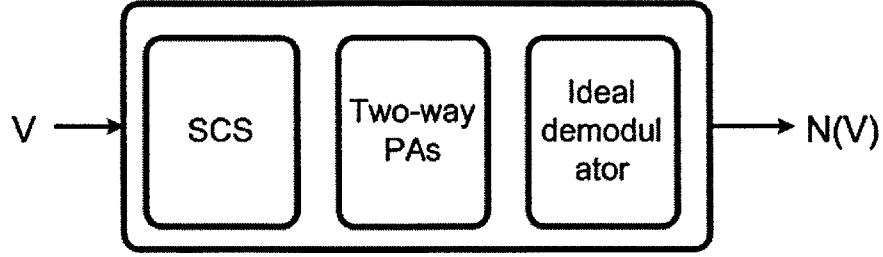


Figure 3-8: PA system under compensation.

$$\Delta(x) = N(x) - x, \quad (3.4)$$

then we can rewrite (3.3) as

$$V_c + \Delta(V_c) = V_i. \quad (3.5)$$

Since nonlinear function $\Delta(x)$'s form is unknown, many nonlinear system solving techniques are infeasible here. The most direct information we are able to obtain on $\Delta(x)$ is its functional value acquired through a simulation with input x . Therefore, we can use the following iterations to solve for the V_c satisfying (3.5).

$$\begin{aligned} V_c^{k+1} &= V_i - \Delta(V_c^k), k = 0, 1, 2, \dots \\ V_c^0 &= V_i. \end{aligned} \quad (3.6)$$

For the iteration to converge, function $\Delta(x)$ has to satisfy the following criterion

$$\|\Delta(x_1) - \Delta(x_2)\| \leq \theta \|x_1 - x_2\|, \quad (3.7)$$

$$\theta < 1. \quad (3.8)$$

In other words, in order for the iterations to converge, function $\Delta(x)$ has to have a Lipschitz constant less than 1. Although there seems little that can be done to change the function $\Delta(x)$ once the design is fixed, we can design the compensator

input sample sequence to avoid the regions of $\Delta(x)$ that would fail the convergence criterion. Such regions do exist because of the discontinuous SCS functions in both LINC and AMO architectures. Take the LINC architecture for example. Assume $v \in \mathbb{C}$ is the input sample of SCS, $v_1, v_2 \in \mathbb{C}$ are the two decomposed samples, and a is the amplitude of the decomposed samples, we have

$$\begin{cases} v = v_1 + v_2, \\ |v_1| = |v_2| = a. \end{cases} \quad (3.9)$$

Solve for v_1, v_2 , and we have

$$v_{1,2} = F(v) = \frac{v}{2} \pm j \cdot \frac{v}{|v|} \sqrt{a^2 - \frac{|v|^2}{4}}. \quad (3.10)$$

Calculate the Frobenius norm of Jacobian of function $F(v)$, we have

$$\|\text{Jacobian}_{v_1}\|_F = \frac{2a^2}{|v| \sqrt{4a^2 - |v|^2}}. \quad (3.11)$$

Figure 3-9 plots the function in (3.11) with $a = 1$. The Jacobian approaches infinity when the input sample amplitude closes to zero. Therefore, avoiding the region close to zero should help speed up the iteration (3.6) convergence. For AMO architecture, we arrive at a similar conclusion. From equations

$$\begin{cases} v = v_1 + v_2, \\ |v_1| = a_1, \\ |v_2| = a_2, \end{cases}$$

we have

$$v_{1,2} = G_{1,2}(v) = \frac{v}{2} \gamma \pm j \cdot \frac{v}{|v|} \sqrt{a_1^2 - \frac{|v|^2 \gamma^2}{4}}, \quad (3.12)$$

where

$$\gamma = 1 + \frac{a_1^2 - a_2^2}{|v|^2}, \quad (3.13)$$

a_1, a_2 are the two amplitude levels of the decomposed signals. The Frobenius norm

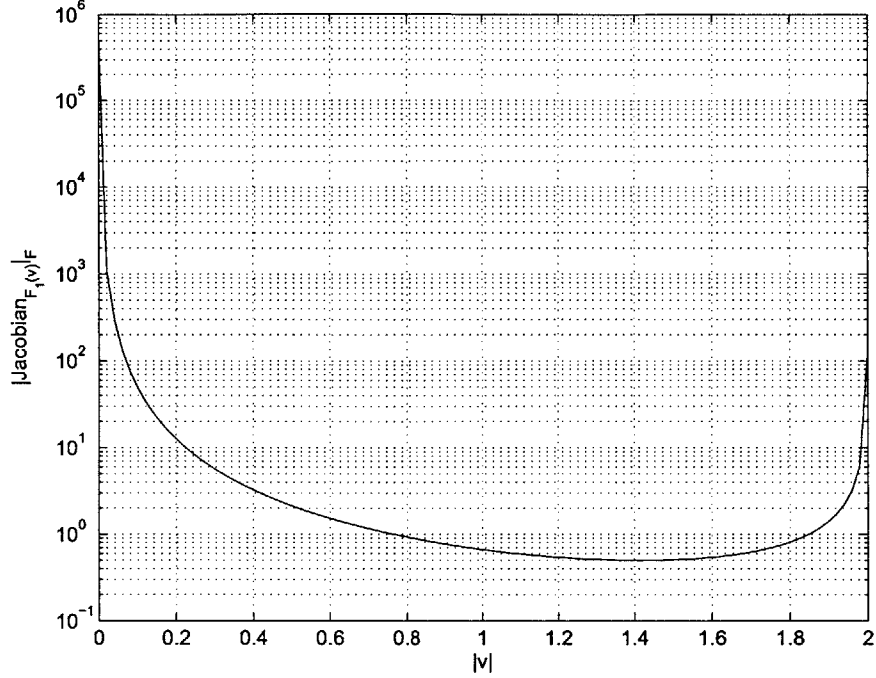


Figure 3-9: Frobenius norm of the Jacobian of the function $v \rightarrow v_1$. $a = 1$.

of the Jacobian of the function $G_1(v)$ (3.14) is shown in Figure 3-10.

$$\|\text{Jacobian}(G_1(v))\|_F = \frac{2a_1a_2}{\sqrt{[(a_1 + a_2)^2 - |v|^2][|v|^2 - (a_1 - a_2)^2]}}. \quad (3.14)$$

Compared to LINC architecture, the Jacobian for the AMO approaches infinity in several more regions where the amplitudes of the decomposed signals switch to different levels. As shown in Figure 3-10, besides the region $|v| \rightarrow 0$, there are levels of $|v| \rightarrow 2.2, 2.5, 2.8, 3.2, 3.6, 4, 4.4$ which also should be avoided. To achieve a sample sequence avoiding all those regions and levels, a special filter is needed to replace the normal shaping filter. Appendix A shows such a sample design for the LINC architecture, where most of the samples are excluded in the defined region which is close to zero while still satisfying the spectral and ISI constraints. A more thorough study of such filters for both LINC and AMO are out of the scope of the thesis and fits into our future work.

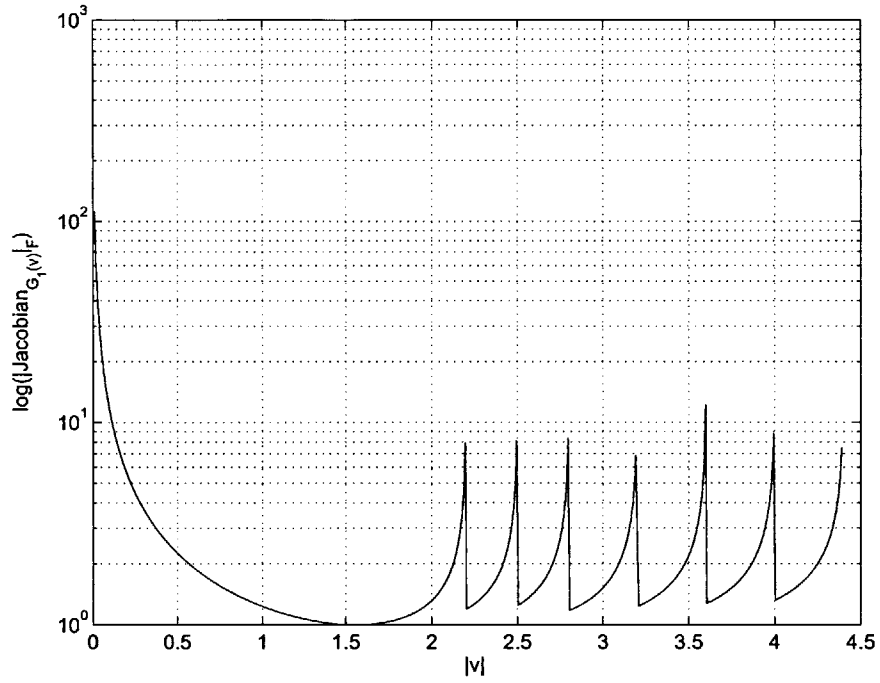


Figure 3-10: Frobenius norm of the Jacobian of the function $G_1(v)$. $a_1, a_2 \in [1.1, 1.4, 1.8, 2.2]$. The corresponding threshold levels for different regions are $[2.2, 2.5, 2.8, 3.2, 3.6, 4, 4.4]$.

Demodulation Method

Since we target to lower the ACPR under -40dB, the iterations have to be carried out as precisely as possible. Besides setting the simulator accuracy level to be conservative, it is also important to use an accurate ideal demodulator shown in Figure 3-8. This 'ideal demodulator' takes the frequency response from the band of interest around the carrier frequency and solves for the exact frequency response of the input. In this way, we are able to isolate the baseband-equivalent system without introducing extra effects from non-ideal down-converting and low-pass filtering. The iteration procedure can be then carried out with greater accuracy. We derive the mathematics for the ideal demodulator as follows. Define the input sample sequence as $v[n]$. After modulation, the continuous-time signal $y(t)$ is

$$\begin{aligned} y(t) &= 2\text{Re}[v(t) \cdot e^{j\omega_c t}], \\ v(t) &= \text{Zoh}(v[n]), \end{aligned} \quad (3.15)$$

where $\text{Zoh}(t)$ is the zero-order sample-and-hold function. ω_c is the carrier frequency in radius, and the Fourier Transform of $v(t)$ can be written as

$$V(j\omega) = \frac{1 - e^{-j\omega T}}{1 - j\omega} V(e^{j\omega T}), \quad (3.16)$$

where T is the sample duration, and we define $V(e^{j\Omega})$ as the discrete time Fourier Transform of $v[n]$, and the Fourier Transform of $y(t)$ is

$$Y(j\omega) = \frac{1 - e^{-j\omega T}}{j(\omega - \omega_c)} V(e^{j\omega T}) + \frac{1 - e^{-j\omega T}}{j(\omega + \omega_c)} \overline{V(e^{-j\omega T})}. \quad (3.17)$$

Replacing variable ω with $\hat{\omega} + \omega_c$, and assuming that there is an integer number of carrier periods in one sample, namely $\omega_c T = 2\pi k$, $k \in \mathbb{Z}$, we have

$$\begin{aligned} Y(j(\hat{\omega} + \omega_c)) &= \frac{1 - e^{-j\hat{\omega}T}}{j\hat{\omega}} V(e^{j\hat{\omega}T}) + \frac{1 - e^{-j\hat{\omega}T}}{j(\hat{\omega} + 2\omega_c)} \overline{V(e^{-j\hat{\omega}T})} \\ &= \frac{1 - e^{-j\hat{\omega}T}}{j\hat{\omega}} (V(e^{j\hat{\omega}T}) + \frac{\hat{\omega}}{\hat{\omega} + 2\omega_c} \overline{V(e^{-j\hat{\omega}T})}). \end{aligned} \quad (3.18)$$

From (3.18), we arrive at the following set of equations

$$\begin{cases} V(e^{j\hat{\omega}T}) + \frac{\hat{\omega}}{\hat{\omega} + 2\omega_c} \overline{V(e^{-j\hat{\omega}T})} = \frac{j\hat{\omega}}{1 - e^{-j\hat{\omega}T}} Y(j(\hat{\omega} + \omega_c)) \\ \frac{-\hat{\omega}}{-\hat{\omega} + 2\omega_c} V(e^{j\hat{\omega}T}) + \overline{V(e^{-j\hat{\omega}T})} = \frac{j\hat{\omega}}{1 - e^{-j\hat{\omega}T}} \overline{Y(j(-\hat{\omega} + \omega_c))}. \end{cases} \quad (3.19)$$

Since $Y(j(\hat{\omega} + \omega_c))$ and $Y(j(-\hat{\omega} + \omega_c))$ are known from the Fourier Transform of the simulation output, we can calculate (demodulate) the value $V(e^{j\hat{\omega}T})$ as the solution of these linear equations with two variables,

$$V(e^{j\hat{\omega}T}) = \frac{A \cdot B \cdot \overline{Y(j(\omega_c - \hat{\omega}))} - B \cdot Y(j(\hat{\omega} + \omega_c))}{A \cdot C - 1}, \quad (3.20)$$

where

$$\begin{aligned} A &= \frac{\hat{\omega}}{\hat{\omega} + 2\omega_c}, \\ B &= \frac{j\hat{\omega}}{1 - e^{-j\hat{\omega}T}}, \\ C &= \frac{-\hat{\omega}}{-\hat{\omega} + 2\omega_c}. \end{aligned}$$

Since in simulation we work with finely oversampled discrete signals Y , (3.20) should be implemented with its discrete Fourier Transform. The same derivation principle follows, with all signals expressed in their discrete Fourier Transform. Starting with (3.16), we have $v(t)$ corresponding to the discrete signal's Fourier Transform expressed as

$$V(e^{j\Omega}) = \frac{1 - e^{-j\Omega M}}{1 - e^{-j\Omega}}, \quad (3.21)$$

where M is the number of time discretization points in one sample duration. Then we have the discrete Fourier Transform for the discretized $y(t)$ as

$$Y(e^{j\Omega}) = \frac{1 - e^{-j\Omega M}}{1 - e^{-j(\Omega - \Omega_c)}} V(e^{j\Omega M}) + \frac{1 - e^{-j\Omega M}}{1 - e^{-j(\Omega + \Omega_c)}} \overline{V(e^{-j\Omega M})}, \quad (3.22)$$

where $\Omega_c = \omega_c \frac{T}{M}$. Similarly, replacing variable Ω with $\hat{\Omega} + \Omega_c$, we obtain the following linear equations with two unknowns

$$\begin{cases} V(e^{j\hat{\Omega}}) + \frac{1 - e^{-j\hat{\Omega}}}{1 - e^{-j(\hat{\Omega}+2\Omega_c)}} \overline{V(e^{-j\hat{\Omega}})} = \frac{1 - e^{-j\hat{\Omega}}}{1 - e^{-j\hat{\Omega}M}} Y(e^{j(\hat{\Omega}+\Omega_c)}) \\ \frac{1 - e^{-j\hat{\Omega}}}{1 - e^{-j(\hat{\Omega}-2\Omega_c)}} V(e^{j\hat{\Omega}}) + \overline{V(e^{-j\hat{\Omega}})} = \frac{1 - e^{-j\hat{\Omega}}}{1 - e^{-j\hat{\Omega}M}} Y(e^{j(-\hat{\Omega}+\Omega_c)}), \end{cases} \quad (3.23)$$

and lastly we solve for the unknown $V(e^{j\hat{\Omega}})$ and arrive at the following expression

$$V(e^{j\hat{\Omega}}) = \frac{\frac{1-e^{-j\hat{\Omega}M}}{1-e^{-j\hat{\Omega}}} \cdot Y_0 - \frac{1-e^{-j\hat{\Omega}M}}{1-e^{-j(\hat{\Omega}+2\Omega_c)}} \cdot Y_1}{\left(\frac{1-e^{-j\hat{\Omega}M}}{1-e^{-j\hat{\Omega}}}\right)^2 - \frac{(1-e^{-j\hat{\Omega}M})^2}{(1-e^{-j(\hat{\Omega}+2\Omega_c)})(1-e^{-j(\hat{\Omega}-2\Omega_c)})}}, \quad (3.24)$$

where

$$Y_0 = Y(e^{j(\Omega_c+\hat{\Omega})}), \quad Y_1 = \overline{Y(e^{j(\Omega_c-\hat{\Omega})})}. \quad (3.25)$$

Finally, the received samples are the inverse Fourier Transform of $V(e^{j\hat{\Omega}})$ and the numerics on samples are done through IFFT,

$$v[n] = \text{Inverse Fourier Transform}(V(e^{j\hat{\Omega}})). \quad (3.26)$$

To summarize, in the demodulation process, we first interpolate and re-sample the output signal, then use equation (3.24) and (3.26) to obtain the received samples. The traditional way of demodulation is to down-convert the simulation output from carrier frequency to baseband and filter out the undesired frequency band. The difference between the two approaches can be traced back to (3.18). The traditional approach assumes $Y(j(\hat{\omega} + \omega_c)) = V(e^{j\hat{\omega}T})$ and ignores the term $\frac{\hat{\omega}}{\hat{\omega}+2\omega_c} \overline{V(e^{-j\hat{\omega}T})}$ as well as the effect from sample-and-hold. For general demodulation purposes, the error it introduces may be tolerable. However, in our situation, the error may hinder the iteration convergence. Therefore, we choose the more accurate demodulation approach with the above equations.

With successfully obtained received samples, we are able to obtain the value of $\Delta(V_c^k)$ at the k^{th} iteration by taking the difference between the transmitted samples and received samples as in (3.5). Then the k^{th} iteration loop (3.6) finishes and we are ready for the next iteration.

Off-line Compensation Results

To test the effectiveness of the iteration scheme, we perform the iterations (3.6) on both LINC and AMO systems with the unit PA design from [54]. The simulation setup is described in Section 3.2.1 and shown in Figure (3-4). Both the LINC and AMO systems operate with carrier frequencies at 45GHz and input sample bandwidth at 2.5GHz. The power supply level of the LINC system is at 2.2V and the AMO system uses four power supplies switching among 1.1V, 1.4V, 1.8V and 2.2V. All input symbols are drawn randomly from the 64-QAM constellation and shaped to input samples with an oversampling rate of 2. We use zero-avoidance and level-avoidance filters to achieve the input sequences avoiding those regions.

Table 3.1 shows the ACPR and EVM performances of the LINC system with a 1024-sample input sequence, which is generated randomly and shaped in spectrum by a real-time zero-avoidance filter described in Appendix A. In the table, $\theta_{1,2}$ and θ represent the Lipschitz constants of a single-way PA and the overall LINC system. As indicated in (3.8), these constants determine the convergence rate of the iteration. The EVM and ACPR columns in the table show the converging performances through 3 iterations. The steady yet slowly decreasing improvements in the two metrics are confirmed by the last column θ . We see that although the Lipschitz constant of each PA stays away from 1 through iterations, the Lipschitz constant of the overall LINC system approaches 1 as iteration goes on hence the improvements diminish along the way. The main reason for the diminishing gain is that the nonlinear effect of the overall LINC system not only depends on the nonlinearity from each of the PAs, but also the way the input sample is decomposed. After the two PA channels, the decomposed signals can no longer perfectly sum up to the original input, and the effective nonlinear effect differs from that of each PA. Therefore, θ shows different

trend compared to θ_1 and θ_2 . Figures 3-11 and 3-12 show the EVM performances

Table 3.1: ACPR and EVM performances of LINC system in off-line iterations, with input sequence generated from a real-time zero-avoidance filter.

Iteration	EVM (%)	ACPR (dB)	θ_1	θ_2	θ
0	4.5	-30.6	NA	NA	NA
1	1.7	-37.6	0.145	0.174	0.289
2	1.1	-42.4	0.192	0.244	0.435
3	1.0	-44.0	0.191	0.228	0.701

before and after the off-line compensation, both compared with an ideal 64QAM constellation. The two EVM figures correspond to an improved EVM performance from 4.5% to 1.0%. Figure 3-13 shows the ACPR performance improvement before and after compensation, from -30.6dB to -44.0dB .

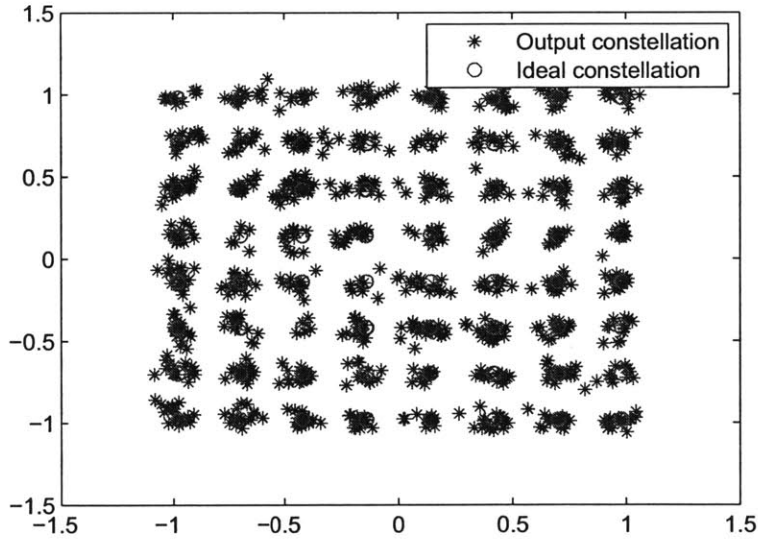


Figure 3-11: EVM of the uncompensated LINC system.

To verify the need for the zero-avoidance filter, we perform the off-line iterations with the input sequence without zero-avoidance property and compare with the previous zero-avoidance input sequence. We carried out the comparisons on two design examples. One of the examples is the design we used to produce the results so far.

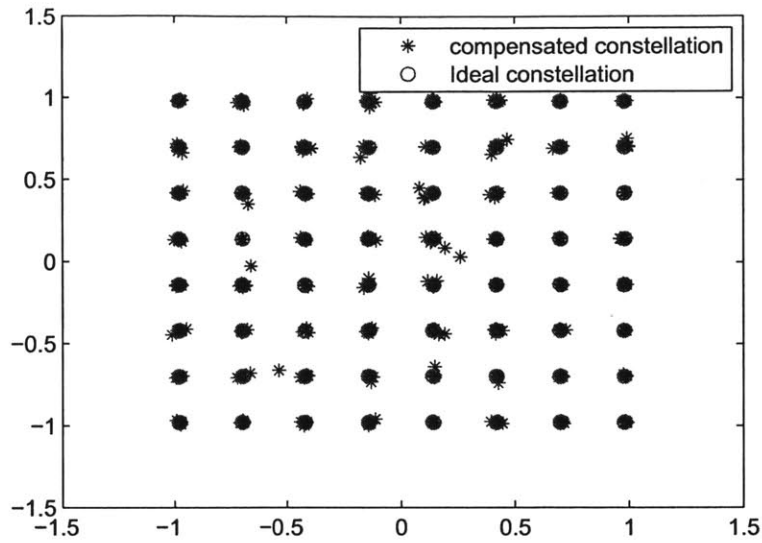


Figure 3-12: EVM of the compensated LINC system, with real-time zero-avoidance input sequence.

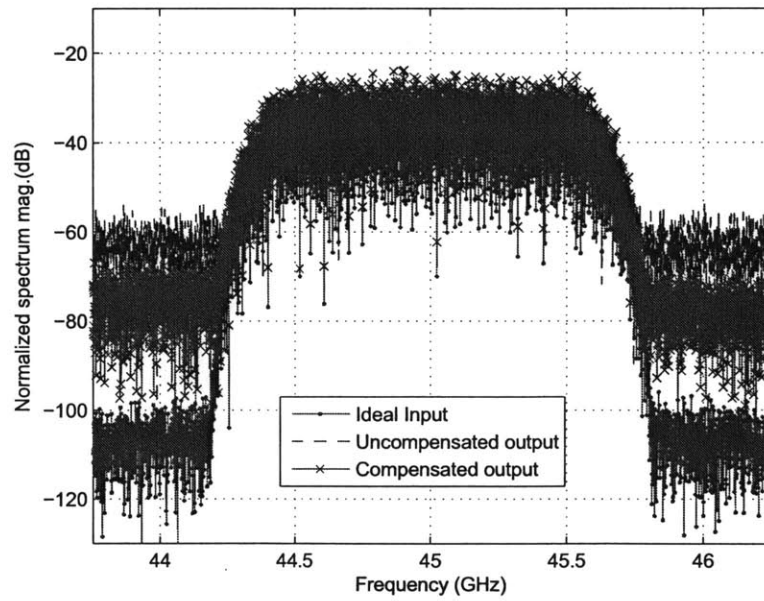


Figure 3-13: Input and output ACPR of the LINC system, with real-time zero-avoidance input sequence.

The other example is a PA design with the same architecture but all transistors' bodies are tied to ground or supply properly. The body-tied PA works at 22.5GHz and has a sample bandwidth of 1.25GSamples/s. The comparison results are shown in the Table 3.2. We observe that for both designs, zero-avoidance sequences help iterations converge better. For body-tied design, the iteration cannot converge at all without the zero-avoidance property in input sequence while it converges nicely with such property. From these comparisons, we are assured the effectiveness of the zero-avoidance in achieving a better converged compensation sequence. Currently, we have techniques to achieve partial zero-avoidance property such as the real-time algorithm shown in Appendix A. We can also obtain the zero-avoidance sequence using offline optimization techniques with better zero-avoidance. Table 3.3 shows the offline iteration result comparison between using a real-time and an offline filter. For the first data set of the body-tied PA design, the offline filter outperforms the real-time filter with around 4 dB better performance in ACPR and slightly better EVM. For the second data set, the two filters lead to roughly the same performances. Therefore, before we are able to realize the offline algorithm with a real-time implementation strategy, the real-time design in Appendix A serves our purpose fairly well.

Table 3.2: ACPR and EVM performance comparisons between using input sequence with and without zero-avoidance property for LINC systems. The zero-avoidance filter has a real-time implementation as shown in Appendix A.

	Floating-body PA		Body-tied PA	
	Zero-avoidance	No zero-avoidance	Zero-avoidance	No zero-avoidance
ACPR(dB)	-30.6 → -44	-30.1 → -39.6	-41.3 → -56.4	-44 → N/A
EVM (%)	4.5 → 1.0	4.2 → 1.7	2.6 → 0.15	0.9 → N/A

For the AMO system, we investigated several scenarios with different parasitics values and path delays. These experiments show different levels of off-line compensation ability, and hence serve as a guideline for circuit and board designers who need compensators for their systems. For instance, the stringent path timing-matching specification for the delay-line design may be relaxed a little bit thanks to the confirmation that a digital compensator is capable of compensating the path mismatch

Table 3.3: Comparison of offline iteration results with real-time and offline zero-avoidance filters.

	Body-tied PA data set 1		Body-tied PA data set 2	
	Real-time	Offline	Real-time	Offline
ACPR(dB)	-37.3 \rightarrow -47.7	-37.7 \rightarrow -51.2	-41.3 \rightarrow -56.4	-38 \rightarrow -56.5
EVM (%)	2.0 \rightarrow 0.43	1.7 \rightarrow 0.40	1.3 \rightarrow 0.15	1.65 \rightarrow 0.17

within a certain range. Designers would also know the range of the switch ringing that could be handled by a compensator and design the circuit boards accordingly. Table 3.4 shows the iteration results for different bump inductance values, as shown in the schematic of Figure 3-6. The input sequences are generated from an offline level-avoidance filter. In both 5pH and 20pH cases, the iterations converge nicely and result in improved EVMs to around 0.7% and ACPRs around -44dB. In the 60pH case, the ringing effect from the bump inductance is so large that the iteration stops converging even in the second iteration, and therefore results in a smaller performance improvement: 2.6% EVM and -34dB ACPR. Figures 3-14 and 3-15 show the EVM comparison before and after the compensation for the 20pH inductance case. Figure 3-16 shows the ACPR improvement accordingly. These results imply both the effectiveness and limitations of digital compensation for different levels of parasitics, and provide a good guideline for PA designers about the capability and conditions necessary for effective digital compensation.

Table 3.4: ACPR and EVM performances of AMO systems with different bump inductances. The input sequences are from offline level-avoidance filtering.

Iter	$L_{\text{bump}}=5\text{pH}$		$L_{\text{bump}}=20\text{pH}$		$L_{\text{bump}}=60\text{pH}$	
	EVM (%)	ACPR (dB)	EVM (%)	ACPR (dB)	EVM (%)	ACPR (dB)
0	4.5	-28.4	5.1	-30.0	6.6	-27.3
1	1.3	-40.6	1.6	-39.5	3.0	-33.1
2	0.8	-44.2	1.0	-42.7	2.6	-33.9
3	0.7	-44.0	0.8	-44.2	NA	NA

Aside from testing with different bump inductances, we also experiment with the situation when a path delay exists between the phase and amplitude path. Table

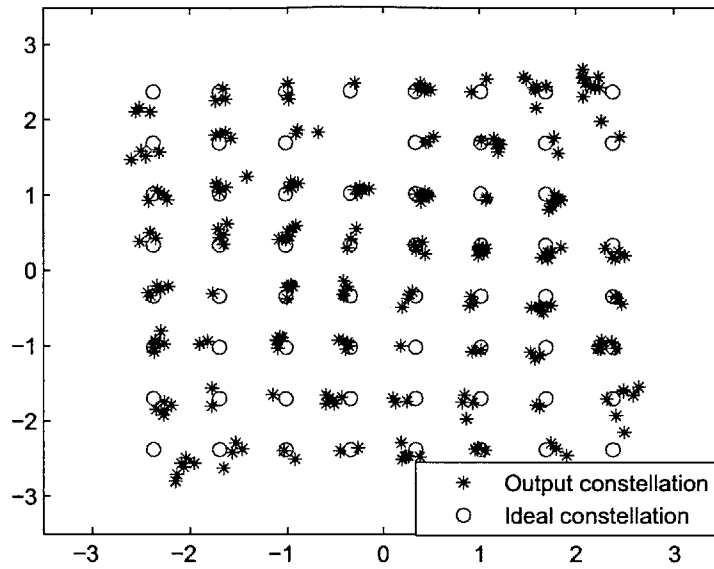


Figure 3-14: EVM of the uncompensated AMO system with $L_{\text{bump}}=20\text{pH}$.

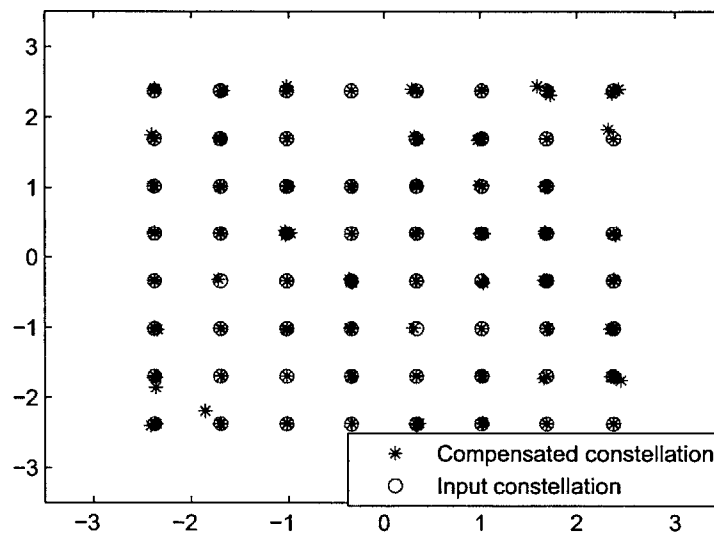


Figure 3-15: EVM of the compensated AMO system with $L_{\text{bump}}=20\text{pH}$. Input sequence is generated from offline level-avoidance filter.

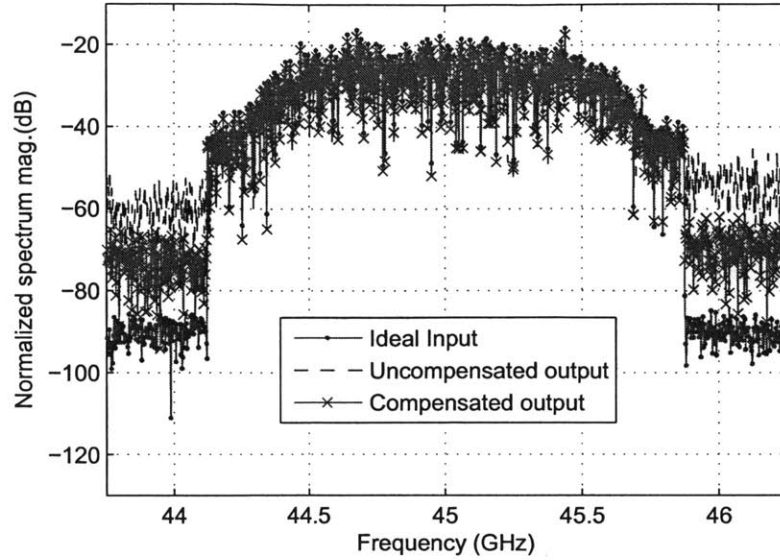


Figure 3-16: Input and output ACPR of the AMO system with $L_{\text{bump}}=20\text{pH}$. Input sequence is generated from offline level-avoidance filter.

3.5 shows the iteration results on EVM and ACPR improvement with 15ps path mismatch. As the results suggest, the compensator is capable of fixing any delay-line tuning residual up to 15ps.

Table 3.5: ACPR and EVM performances of AMO system with 5pH bump inductance and 15ps path mismatch between phase and amplitude paths. The input sequence is generated from offline level-avoidance filter.

Iteration	EVM (%)	ACPR (dB)
0	5.3	-29.7
1	5.3	-29.3
2	1.6	-38.8
3	1.0	-41.6

3.2.3 Analysis of Nonlinearities Throughout the System

With the demonstration of a successful off-line compensation through iterations, we now turn the effort in the following sections to building a dynamical model of the

compensator. First of all, it is important and helpful to understand the structure of the equivalent baseband nonlinear system, as shown in Figure 3-17. The system has the transmitted and received sample sequences $v[n]$ and $u[n]$ as input and output. Its nonlinear system represents the overall nonlinearity from the whole RF signal chain (e.g. PA, modulator etc.) reflected to baseband.

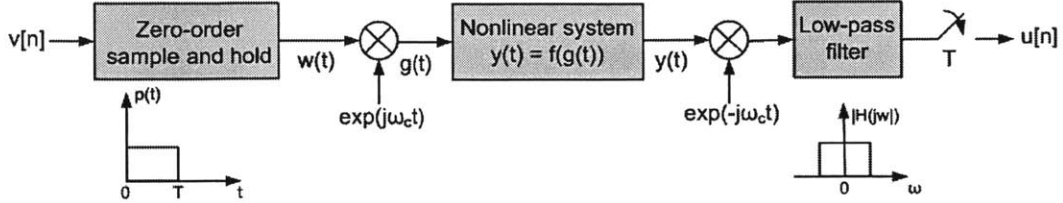


Figure 3-17: Nonlinear system of the overall transceiver signal chain.

In Figure 3-17, discrete sample sequence $v[n]$ first turns into a continuous-time waveform $w(t)$ through a zero-order sample-and-hold system. Its transfer function's time-domain waveform is depicted as $p(t)$ and T denotes the sample duration; $w(t)$ is then modulated to carrier frequency ω_c as $g(t)$ and transmitted. More accurately, the real part of $g(t)$ is transmitted through PA. Any nonlinearities along the way of transmission are abstracted in a Volterra series $f(t)$, which is capable of modeling any time-invariant nonlinear system with memory. With $g(t)$ and $y(t)$ being input and output respectively, we have

$$y(t) = k_0 + \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} k_n(\tau_1, \tau_2, \dots, \tau_n) g(t - \tau_1) g(t - \tau_2) \dots g(t - \tau_n) d\tau_1 d\tau_2 \dots d\tau_n, \quad (3.27)$$

where k_n is the n -th order kernel. The signal $y(t)$ is then demodulated and filtered in the baseband and the down-sampler samples $z(t)$ with a frequency of $1/T$ to discrete samples.

According to this signal chain, we can find out the characteristics of the equivalent nonlinear baseband system, and use them as a guideline for the choice of the compensator model structure. In the following, we will show the expressions of the signals along the signal chain from input sequence $v[n]$ till the received sequence $u[n]$.

The input discrete sample sequence $v[n]$ first transforms to the continuous waveform $w(t)$ as

$$w(t) = \sum_{k=0}^{\infty} v_k p(t - kT), \quad (3.28)$$

where v_k is the k -th sample in the time series; $p(t)$ is the rectangular waveform depicted in Figure 3-17 and T is the sample duration. After modulation, we have $g(t)$ as

$$g(t) = w(t) \times e^{j\omega_c t} = \sum_{k=0}^{\infty} v_k p(t - kT) e^{j\omega_c t}, \quad (3.29)$$

where ω_c is the carrier frequency. Then $g(t)$ goes through the nonlinear system represented by the Volterra series. To make the derivation more tractable, we start from a simpler case where we consider up to the second-order nonlinearity in the Volterra series (3.27), which can be written as

$$y(t) = \underbrace{k_0}_{y_0(t)} + \underbrace{\int_{-\infty}^{\infty} k_1(\tau_1) g(t - \tau_1) d\tau_1}_{y_1(t)} + \underbrace{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) g(t - \tau_1) g(t - \tau_2) d\tau_1 d\tau_2}_{y_2(t)}. \quad (3.30)$$

The three terms in (3.30) represent three systems in parallel, hence we can investigate the three systems separately. Since $y_0(t)$ and $y_1(t)$ represent two LTI systems while $y_2(t)$ represents a nonlinear system with memory, we focus on the output from the nonlinear system $y_2(t)$ in (3.30). We rewrite $y_2(t)$ and substitute the expression of $g(t)$:

$$\begin{aligned} y_2(t) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) g(t - \tau_1) g(t - \tau_2) d\tau_1 d\tau_2 \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) \left(\sum_{l=0}^{\infty} v_l p(t - \tau_1 - lT) e^{j\omega_c(t - \tau_1)} \right) \left(\sum_{m=0}^{\infty} v_m p(t - \tau_2 - mT) e^{j\omega_c(t - \tau_2)} \right) d\tau_1 d\tau_2 \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) \sum_{l=0}^{\infty} \sum_{m=0}^{\infty} v_l v_m p(t - \tau_1 - lT) p(t - \tau_2 - mT) e^{j2\omega_c t} e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2. \end{aligned} \quad (3.31)$$

Among all terms in the double summation, only the terms with overlapping $p(t - \tau_1 - lT)$ and $p(t - \tau_2 - mT)$ yield nonzero values. Figure 3-18 shows an example

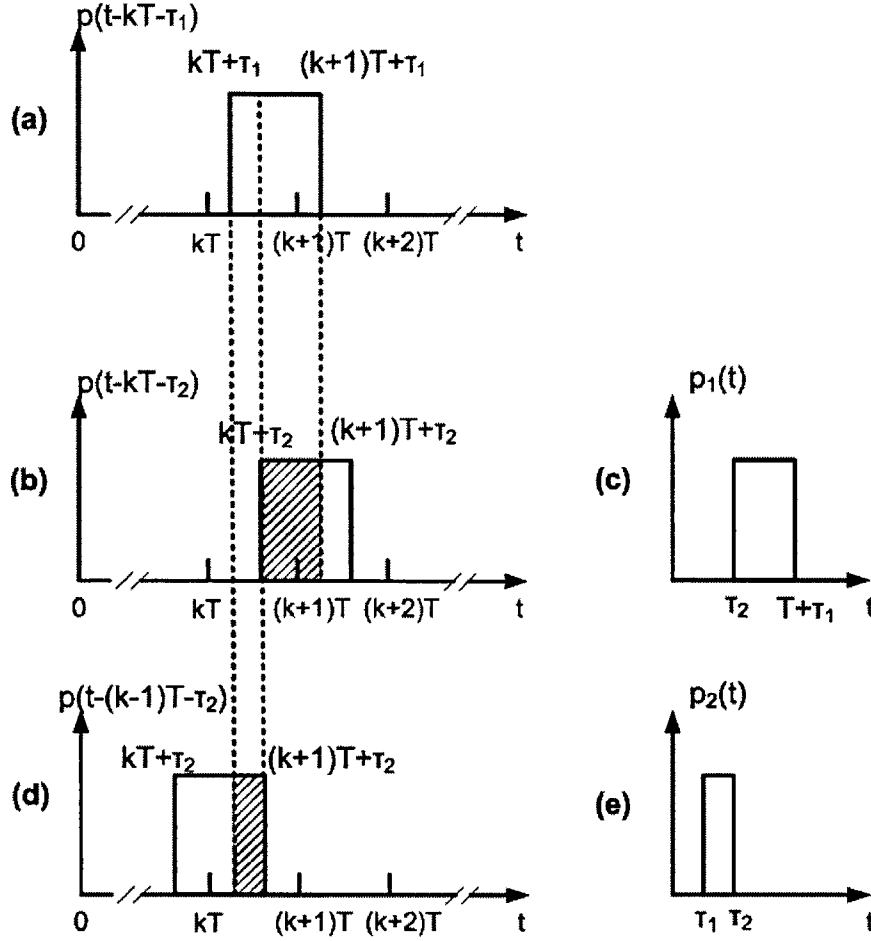


Figure 3-18: Illustration of the derivation for nonzero terms in equation (3.31).

of nonzero terms with the assumption that the nonlinear system has a memory less than one sample duration, namely $0 < \tau_1 \leq \tau_2 \leq T$. Shown in figure 3-18(a), when index l in the first summation takes a value of k , index m must take k or $k - 1$ as in Figure 3-18(b), (d) respectively to yield two nonzero terms. We define $p_1(t)$ and

$p_2(t)$ as shown in Figure 3-18(c) and 3-18(e) and reach the following expressions,

$$\begin{aligned}
y_2(t) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) \sum_{k=0}^{\infty} [(p_1(t - kT)v_k v_{k+1} + p_2(t - kT)v_k^2) \cdot e^{j2\omega_c t} e^{-j\omega_c(\tau_1 + \tau_2)}] d\tau_1 d\tau_2. \\
&= e^{j2\omega_c t} \cdot \sum_{k=0}^{\infty} [v_k v_{k+1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) p_1(t - kT) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2 + \\
&\quad v_k^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) p_2(t - kT) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2] \tag{3.32}
\end{aligned}$$

After demodulation,

$$\begin{aligned}
z_2(t) &= y_2(t) e^{-j\omega_c t} \\
&= e^{j\omega_c t} \cdot \sum_{k=0}^{\infty} [v_k v_{k+1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) p_1(t - kT) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2 + \\
&\quad v_k^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k_2(\tau_1, \tau_2) p_2(t - kT) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2], \tag{3.33}
\end{aligned}$$

where $z_2(t)$ corresponds to the component in $z(t)$ due to the second-order nonlinear system in $f(t)$. In frequency domain, we have

$$\begin{aligned}
Z_2(j\omega) &= \delta(\omega - \omega_c) * \sum_{k=0}^{\infty} (v_k v_{k+1} \cdot H_{1k}(j\omega) + v_k^2 \cdot H_{2k}(j\omega)) \\
&= \sum_{k=0}^{\infty} (v_k v_{k+1} \cdot H_{1k}(j(\omega - \omega_c)) + v_k^2 \cdot H_{2k}(j(\omega - \omega_c))), \tag{3.34}
\end{aligned}$$

where

$$\begin{aligned}
H_{1k}(j\omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_1(t - kT) k_2(\tau_1, \tau_2) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2, \\
H_{2k}(j\omega) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_2(t - kT) k_2(\tau_1, \tau_2) e^{-j\omega_c(\tau_1 + \tau_2)} d\tau_1 d\tau_2. \tag{3.35}
\end{aligned}$$

Low-pass filtering and downsampling follow, which lead to the spectrum expression

for the nonlinear part of $u[n]$ as

$$U_2(e^{j\Omega}) = \sum_{k=0}^{\infty} \left(\underbrace{v_k v_{k+1}}_{\text{NL}} \underbrace{H_{1k}(e^{j(\frac{\Omega}{T} - \frac{2\pi r}{T})})}_{\text{LTI}} + \underbrace{v_k^2}_{\text{NL}} \underbrace{H_{2k}(e^{j(\frac{\Omega}{T} - \frac{2\pi r}{T})})}_{\text{LTI}} \right), \quad (3.36)$$

where $r = \frac{\omega_c}{\omega_s}$, and $\omega_s = \frac{2\pi}{T}$.

Equation (3.36) shows the nonlinear part of the output of the baseband equivalent system in an example of second-order nonlinearity and memory within one sample duration. We can see that the structure of the output consists of two parts: a nonlinear transformation involving the current and previous samples, represented by v_k^2 , $v_k v_{k+1}$ and noted as NL in (3.36), and a linear time-invariant(LTI) system noted as LTI in (3.36). A closer examination of the LTI system H_{1k} reveals that it is the output of the signal $p_1(t - kT)$ through another LTI system. Figure 3-19 shows the real and imaginary parts of its DFT $P_1(e^{j(\frac{\Omega}{T} - \frac{2\pi r}{T})})$ shown in (3.37),

$$P_1(e^{j(\frac{\Omega}{T} - \frac{2\pi r}{T})}) = \frac{e^{-j(\frac{\Omega}{T} - \frac{2\pi r}{T})\tau_1} - e^{-j(\frac{\Omega}{T} - \frac{2\pi r}{T})\tau_2}}{j(\frac{\Omega}{T} - \frac{2\pi r}{T})}. \quad (3.37)$$

An important observation is that the spectrum has a discontinuity at $\pm\pi$, which translates to a long memory in time-domain. In the situation where the system uses a high oversampling ratio from symbol to sample space, this discontinuity has less effect on the band containing symbol information. In another situation where a lower oversampling ratio is used, this discontinuity leads to a more significant effect on the band with symbol information. In high-speed wideband communication systems, the oversampling ratio is usually limited by the speed that the digital baseband and DAC are able to sustain, therefore the latter scenario described is usually encountered and the compensator model should be able to take care of this factor.

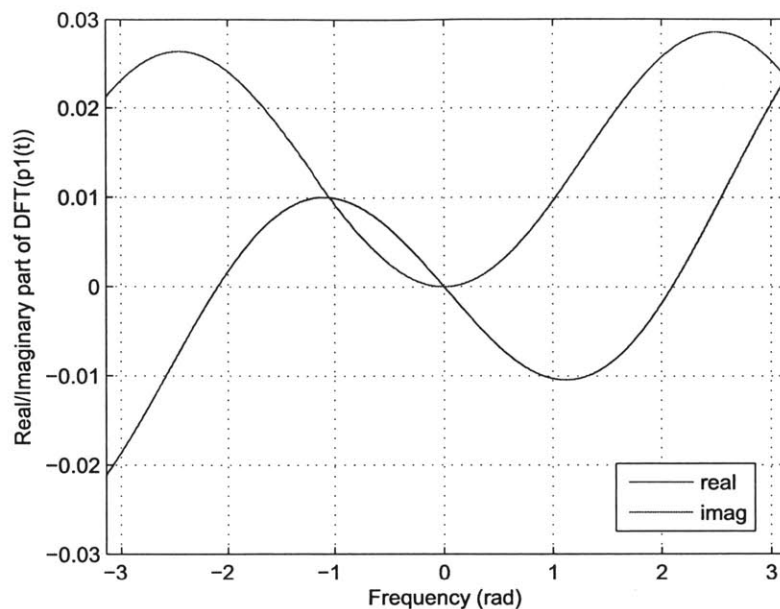


Figure 3-19: An example of DFT $P_1(e^{j(\frac{\Omega}{T}-\frac{2\pi r}{T})})$ with $\tau_1/T = 0.2$, $\tau_2/T = 0.3$, $r = 10$.

3.2.4 Real-time Predistortion Model for LINC/AMO Systems

The off-line compensation through iterations described in Section 3.2.2 not only confirms the possibility of achieving an effective compensator, but also provides input-output pairs of the compensator. However, it does not serve the purpose of compensating in real-time. The compensator coming out of the iterations only compensates the pre-determined input sample sequence and the iteration scheme becomes infeasible for real-time situations. Therefore, having obtained the knowledge of the nonlinear system under compensation, this section is devoted to the effort of modeling the compensator as a dynamical system so that an integrated mixed-signal transmitter system is feasible.

The placement of the compensator is the first question that needs to be answered. From Figure 3-3, we see that there are two possible options. One option is to compensate after the 'Shaped samples' block and before the SCS, while the other option is to

put it after the SCS. The problem with the former comes from the SCS. Because of its nonlinear and discontinuous nature, it is hard to take the SCS into account while modeling the compensator. For example, for the AMO system, a compensated sample V_c may fall into a different decomposition region compared to its uncompensated counterpart V . As a result, since the compensator is built with the knowledge of V , the system we are compensating would be different from the one we build the model on. Therefore, we choose the latter option, and more specifically, the compensator is placed on the two phase paths to correct the phase signals to the PAs, as shown in Figure 3-20. With this placement, the compensator is free of any trouble caused by the SCS and only needs to compensate the nonlinear system from the phase signal to the received sample.

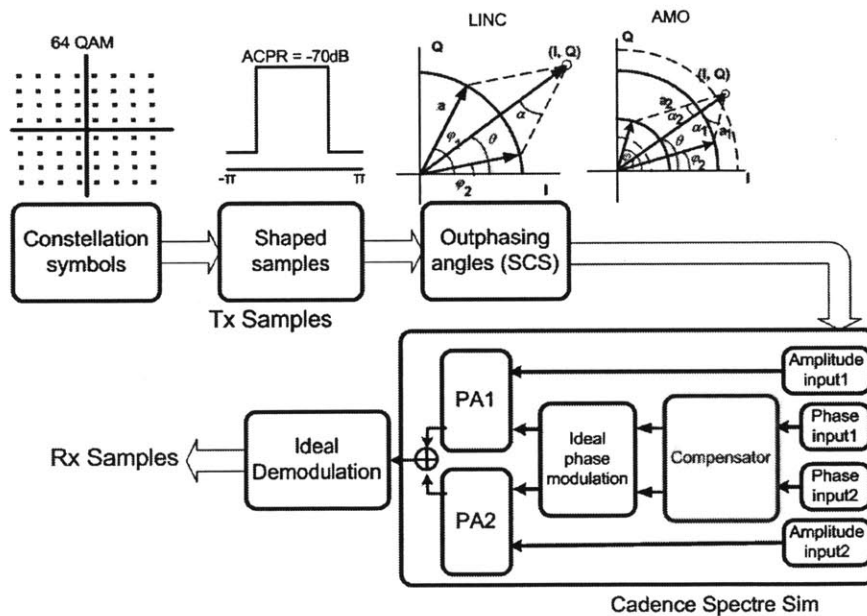


Figure 3-20: Placement of the compensator in the LINC/AMO systems.

Following the analysis in the Section 3.2.3, we see that the equivalent nonlinear baseband system can be represented as a concatenation of a nonlinear transformation with short memory, followed by an LTI system with discontinuities at $\pm\pi$. Furthermore, the dynamical system (3.38) representing the input to the difference between

the output and input follow the same structure.

$$V \rightarrow \Delta(V) = y(V) - V. \quad (3.38)$$

Since $\Delta(V)$ is relatively small compared to input V , the inversion of the input to output function, namely $(V + \Delta(V))^{-1}$ can be approximated by $V - \Delta(V)$. The approximation also corresponds to the first iteration of the off-line compensation in Section 3.2.2, which provides the most of the gain throughout all iterations. Therefore, we can approximate the inversion with the same structure as the forward nonlinear system too.

For the LINC system, we use a model structure as shown in Figure 3-21. The

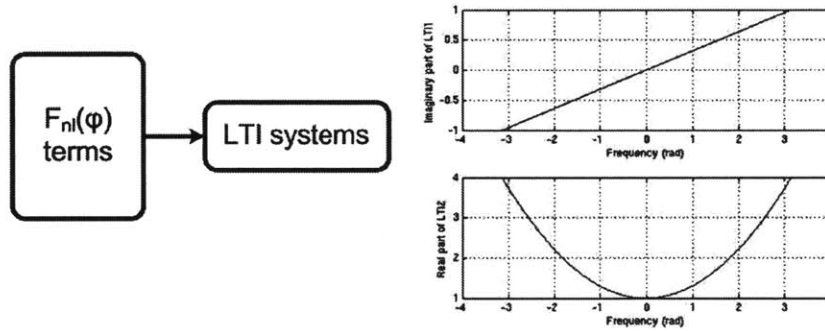


Figure 3-21: Compensator structure.

first part represents the nonlinear function basis terms in a trigonometric polynomial with one memory, as in (3.39).

$$F_{nl}(\phi) = \sum_{i=1}^N a_i \cos(k_{1i}\phi + k_{2i}\phi_d) + b_i \sin(k_{1i}\phi + k_{2i}\phi_d), \quad (3.39)$$

where ϕ is the input phase, ϕ_d is the one-sample delayed version of ϕ , a_i , b_i are the design variables, k_{1i} , k_{2i} are a set of known coefficients determined by the degree of the trigonometric polynomial, N is the number of the terms in the polynomial.

As an example of k_{1i} and k_{2i} values, if the maximal degree on both ϕ and ϕ_d is 2,

then the set of (k_{1i}, k_{2i}) is

$$\{(0, 0), (1, 0), (2, 0), (-2, 1), (-1, 1), (0, 1), (1, 1), (2, 1), (-2, 2), (-1, 2), (0, 2), (1, 2), (2, 2)\}. \quad (3.40)$$

The second part of the compensator structure consists of the LTI functions with discontinuities at $\pm\pi$. Figure 3-21 shows the two LTI functions we use in the fitting. The upper plot shows an LTI system with linear response in frequency, hence its imaginary part is plotted in the figure and the real part is zero. The lower plot shows an LTI system with a second-order response in frequency, hence its real part is as plotted and the imaginary part is zero. With this model structure and a polynomial degree of less than 15 on both current and previous phases, we are able to achieve an EVM performance from 4.5% to 2.5% , and ACPR performance from -30dB to -39dB. The corresponding EVM and ACPR improvements are shown in Figure 3-22 and Figure 3-23. For the AMO system, the nonlinear transformation part should

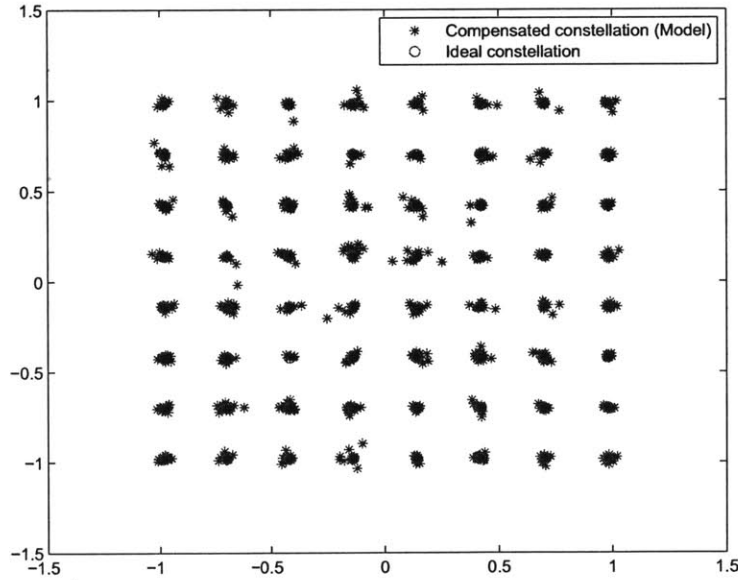


Figure 3-22: EVM of the LINC system with real-time compensator. Input sequence is generated from a real-time zero-avoidance filter.

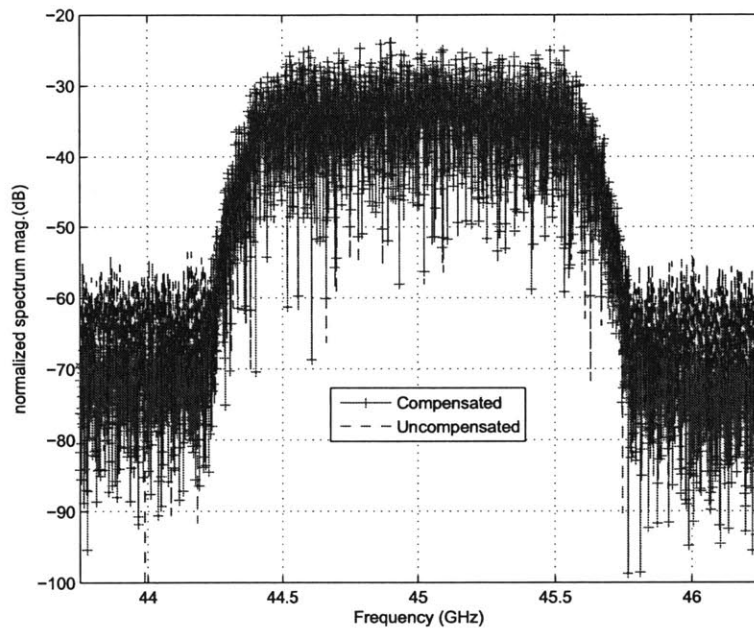


Figure 3-23: ACPR of the LINC system with real-time compensator. Input sequence is generated from a real-time zero-avoidance filter.

also take the amplitudes from the two paths as inputs, as in

$$F_{nl}(\phi, a) = g(a) \cdot \left(\sum_{i=1}^N a_i \cos(k_{1i}\phi + k_{2i}\phi_d) + b_i \sin(k_{1i}\phi + k_{2i}\phi_d) \right), \quad (3.41)$$

where $g(a)$ represents a polynomial function on the amplitude input. In our experiment, we use a linear function on the current and previous amplitude values for each path. After fitting with the design variables a_i and b_i , we obtain the following improved system. With 5pH bump inductance and power switching levels confined to 1.1V and 2.2V, the EVM decreases from 6.7% to 2.7% and ACPR from -27.4dB to -36.2dB. The comparison of the EVM with and without the compensator is shown in Figure 3-24 and 3-25, and the ACPR result is shown in Figure 3-26.

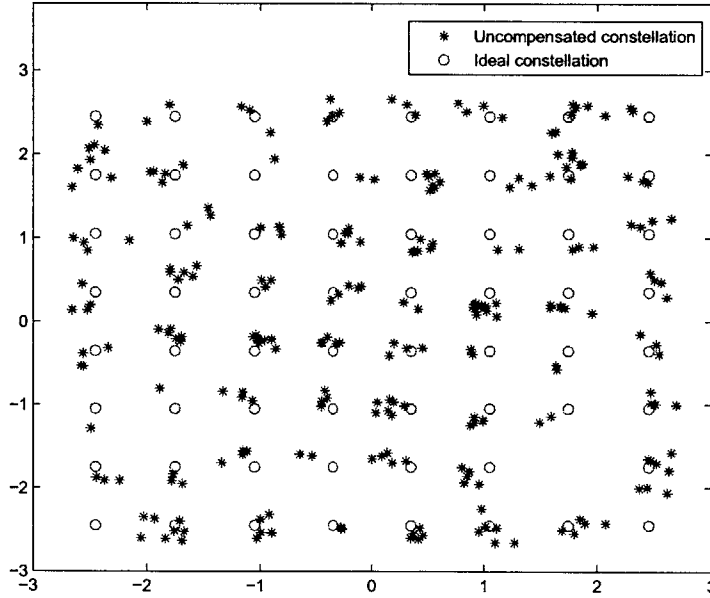


Figure 3-24: EVM of the AMO system without a real-time compensator.

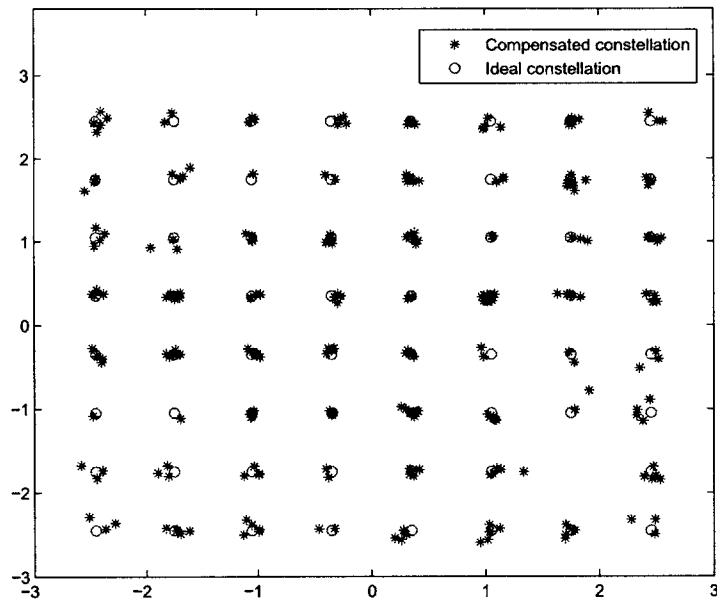


Figure 3-25: EVM of the AMO system with a real-time compensator. The input sequence is generated from an offline level-avoidance filter.

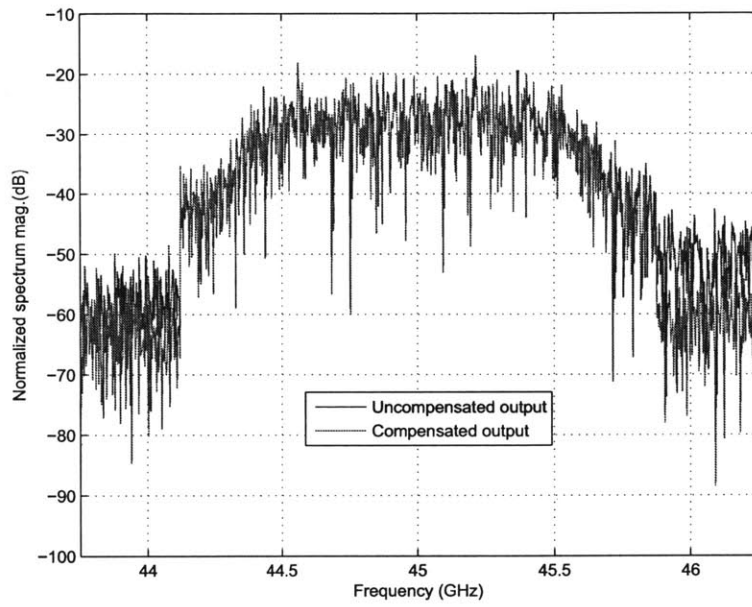


Figure 3-26: ACPR of the AMO system with a real-time compensator. The input sequence is generated from an offline level-avoidance filter.

3.2.5 Implementations

To fully test the model for a real system, we implement the digital baseband containing both the SCS functionality and the compensator, and integrate them with the analog frontend including digital-analog interface, modulator, PA and power supply switches as an overall integrated system solution for a wideband mm-wave transmitter.

The hardware implementation of the compensator is designed to provide the functionality of the model we tested with the simulation data, as discussed in Section 3.2.4, but not limited to it. To fully make use of the silicon area to prepare for the circumstances that are different from what simulator predicts, we also tried to make the hardware as flexible as possible.

Figure 3-28 shows the block diagram of the compensator. It takes the amplitude signals a_1 , a_2 , outphasing angles ϕ_1 , ϕ_2 as the inputs and produces the correction signals $\Delta\phi_1$, and $\Delta\phi_2$ which are added to ϕ_1 , ϕ_2 respectively and passed to the rest of the digital baseband system. The compensator consists of two major parts, corresponding to the nonlinear transformation and LTI system structures in Section 3.2.4. To be more flexible, the nonlinear transformation and LTI system structures have two replicas in the compensator in case the nonlinear system has two modes of behavior.

Figure 3-28 shows the hardware block diagram of the compensator, with its two major pieces indicated in two boxes. The nonlinear transformation part takes signals ϕ_1 , ϕ_2 , a_1 , a_2 , and their one-sample delayed versions as inputs, uses 2 complex-valued nonlinear functions as in two modes and produces 2 complex outputs, or effectively 4 real outputs for each PA. The nonlinear functions are implemented with piecewise linear approximation in two dimensions, an extension from the one dimensional piecewise linear algorithm used in the SCS implementation.

The second part of the compensator structure is an LTI system with discontinuities at $\pm\pi$. As shown in Figure 3-28, it takes the 4 real outputs from each PA and produces the correction signals ΔI and ΔQ . Suppose we need to realize an LTI

system whose real or imaginary part of the transfer function is as shown in Figure 3-27(d). One approach is to directly use an FIR filter to approximate the impulse response. However, the FIR approximation will have a large number of taps because of the discontinuities. Furthermore, since each complex-valued FIR has a complex input, and there are two modes and two PA paths, we have $2 \times 2 \times 2 \times 2 = 16$ effective real-input real-coefficient FIR filtering computations in total. It becomes expensive in power and area to implement 32 such long-tap FIRs. An alternative approach is illustrated in Figure 3-27(a)-(d). The idea is to first upsample the system by 2 in time-domain by use of two-way interleaving in hardware. This leads to the desired transfer function $F(e^{j\Omega})$ scaled in frequency to $[-\pi/2, \pi/2]$, and we are free to design the transfer function in the frequency range $[-\pi, \pi/2]$ and $[\pi/2, \pi]$ such that the new transfer function $F_1(e^{j\Omega})$ is continuous at $\pm\pi$, as shown in Figure 3-27(a). Then the output from the newly designed LTI system F_1 passes through the linear-phase brickwall filter shown in Figure 3-27(b) and produces the frequency response shown in (c). Finally, downsampling by 2 brings us the desired frequency response. The advantage of this approach is that the impulse response of $F_1(e^{j\Omega})$ can be realized with a short-tap FIR, and though the linear-phase brickwall filter matches to a long-tap FIR, it is shared by several short FIRs from different modes from two PA paths. Therefore, in the 'FIR filtering' block of Figure 3-28, for each PA, we have 4 complex inputs interleaved in two-way, and pass through 2 complex-valued short-tap FIR, which produces $4 \times 2 \times 2 = 16$ ways of results in hardware. Then the real and imaginary parts from the total 32-way results are combined together into 4 ways which are still in the two-way interleaved fashion, and pass through the 4 long-tap brickwall filters. As a result, we save from 16-long tap FIRs to 4-long tap FIRs using this approach.

There is one last block ' $\Delta I, \Delta Q$ to $\Delta\phi_1, \Delta\phi_2$ ' which translates the correction signals from Cartesian coordinates to polar coordinates. The formulation is obtained

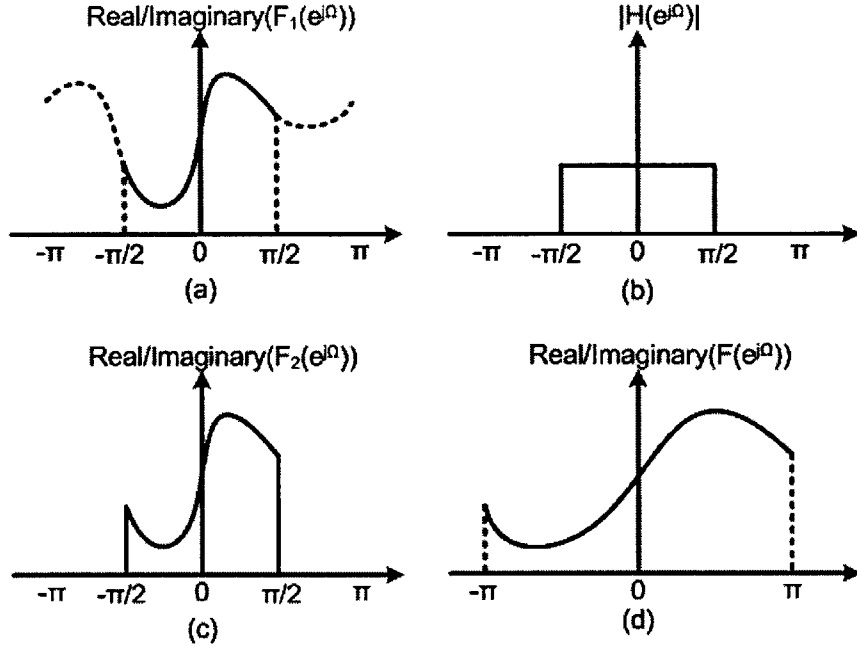


Figure 3-27: Construction of the LTI system with discontinuities at $\pm\pi$.

by taking the full derivatives on the functions $I(\phi_1, \phi_2)$ and $Q(\phi_1, \phi_2)$, where

$$\begin{aligned}
 I(\phi_1, \phi_2) &= a_1 \cos(\phi_1) + a_2 \cos(\phi_2), \\
 Q(\phi_1, \phi_2) &= a_1 \sin(\phi_1) + a_2 \sin(\phi_2).
 \end{aligned}
 \tag{3.42}$$

Therefore we have

$$\begin{aligned}
 \Delta\phi_1 &= \frac{\Delta I \cos \phi_2 + \Delta Q \sin \phi_2}{a_1 \sin(\phi_1 - \phi_2)} \\
 \Delta\phi_2 &= \frac{\Delta I \cos \phi_1 + \Delta Q \sin \phi_1}{a_2 \sin(\phi_2 - \phi_1)}.
 \end{aligned}
 \tag{3.43}$$

With the above compensator structure, we implement the whole digital baseband system whose block diagram as shown in Figure 3-29, which includes the SCS functionality as well as nonlinear compensation capability. The system accepts the input symbol from either FPGA board or on-chip PRBS, and the shaping filter generates the shaped samples to AMO SCS. The SCS uses a single-way to process the computa-

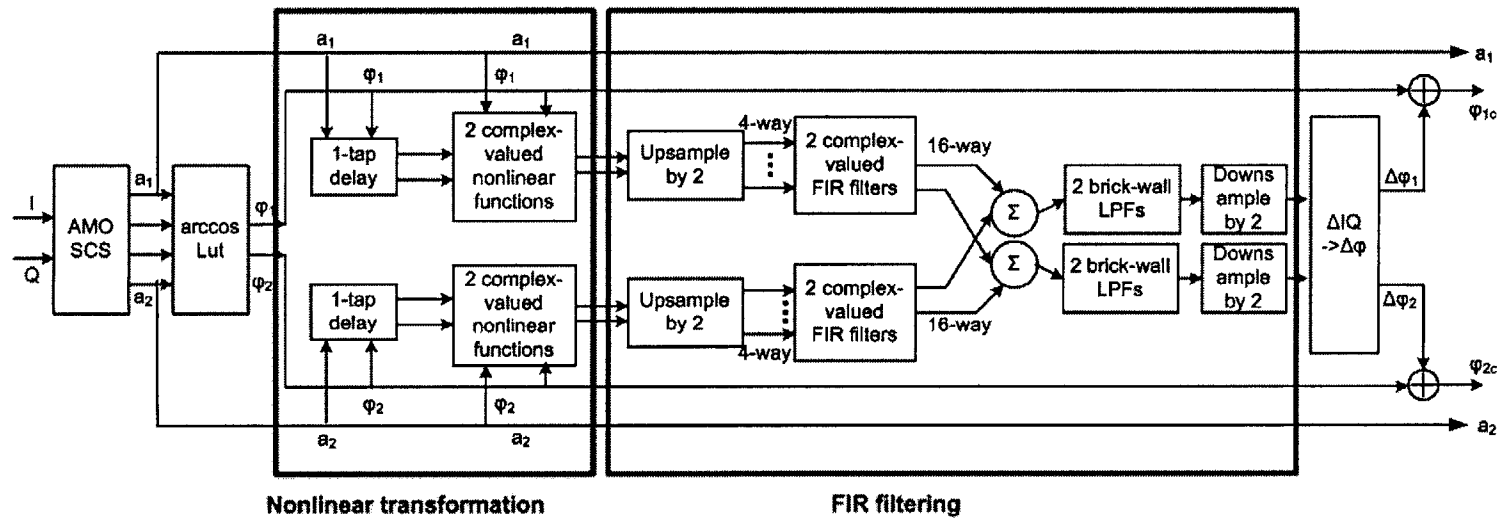


Figure 3-28: The block diagram of the compensator hardware implementation.

tions and provide the amplitude and phase signals along the way to the compensator. The nonlinear compensator outputs the corrected phases and a compensator enabling signal determines whether to output the compensated phases or not. The ' $\frac{1}{1+\tan\phi}$ ' block finally computes the output using the phase inputs. The overall system is an integration of the digital baseband, digital-analog interface, the phase modulator and the 16-way PA. Figure 3-30 shows the placements of major design blocks, and there are compensator sub-blocks, phase modulator, and 16-way PA block from left to right. In the digital baseband design, in order to leave enough room for design space exploration of the compensator, all the parameters in the compensator are programmable and they are implemented with registers, hence the compensator consumes the majority of the area and leaves AMO SCS a relatively small block in the middle. The chip is fabricated with 45nm SOI process, and consumes an area of 3mm×6mm with a total of 4964489 standard cell gates. Figure 3-31 shows the area breakdown of the digital baseband blocks. The majority of the area consumption is from the compensator nonlinear transformation and filtering blocks. The nonlinear transformation part has 4 complex-valued nonlinear functions realized by the 2-dimension piece-wise linear approximation and the filtering part contain 4 100-tap FIR filters, therefore consume a large storage area. With reported post-place-and-route power estimation values, Figure 3-32 shows the power breakdown of the overall system, where the compensator's two major blocks dominate the power consumption. Significant amount of leakage power exists in the nonlinear transformation block of the compensator. This is due to the large LUT size, as well as the register-based LUT design. For the filtering block, dynamic power dominates because of the high activity of the FIR computation. In lab testing, the chip runs up to 500MSamples/s with the power supply at 1.2V and power consumption of 2W. Given that 30% of power at 1GSamples/s throughput is leakage power (mostly due to a leaky register scan chain the revised design with low-leakage scan registers would consume roughly 1W at 500MSamples/s. This represents the cost of 2nJ/sample. For our target AMO PA with 5W of output power and 20% peak-efficiency, this would represent a system with a total power-added efficiency of 19.2% with a penalty from baseband of less

than 0.8%.

3.3 Limitations of the Models

We see from Section 3.2.4 that there is still some gap in the compensation capability between the dynamic model and the iterative solution obtained from simulation iterations. Aside from the model choice itself, one fundamental limiting factor is how stable the underlying nonlinear system is with the setup we have. To answer this question, we devise some experiments to test one of the necessary conditions for a nonlinear system to be stable. For instance, one necessary condition is that a small enough perturbation to the system's initial state does not change the converged state solution.

With this criterion, we simulate a single PA system with phase-modulated input sample sequences, whose phase commands have the structure shown in Figure 3-33. The phase signal starts with 20 zero values for warm-up, followed by a sequence of 10 batches of phases. In each batch, there are 4 sets of patterned phases, where the first 11 phases are the same for all 4 sets and the following 39 phases are randomly chosen differently for the 4 sets. With this setup, we are able to compare the output response differences with the same current input phase and a certain number of previous inputs. Ideally, for a stable system, as long as all the previous states are identical, the outputs should yield the same results with the same current inputs. Since in each 11+39 samples, the first 11 samples have the same phase signals for 4 times, we can compare the k^{th} sample output differences with the previous $k - 1$ samples of the same phases. An example of considering 4 previous and 1 current phases as the state of the system is shown in Figure 3-34 for one of the ten such batches. Keeping the 4 previous phases to be the same, we investigate the differences of the output waveforms of samples 55 versus 5, 105 versus 5, and 155 versus 5. Figure 3-35 shows the waveform differences for all 10 batches, and considering that the output amplitude is around 3 volts, we see up to 4.7% output difference in amplitude. To quantify the waveform differences, we calculate the relative l_2 norm of the discrepancies and Figure 3-36 shows the error

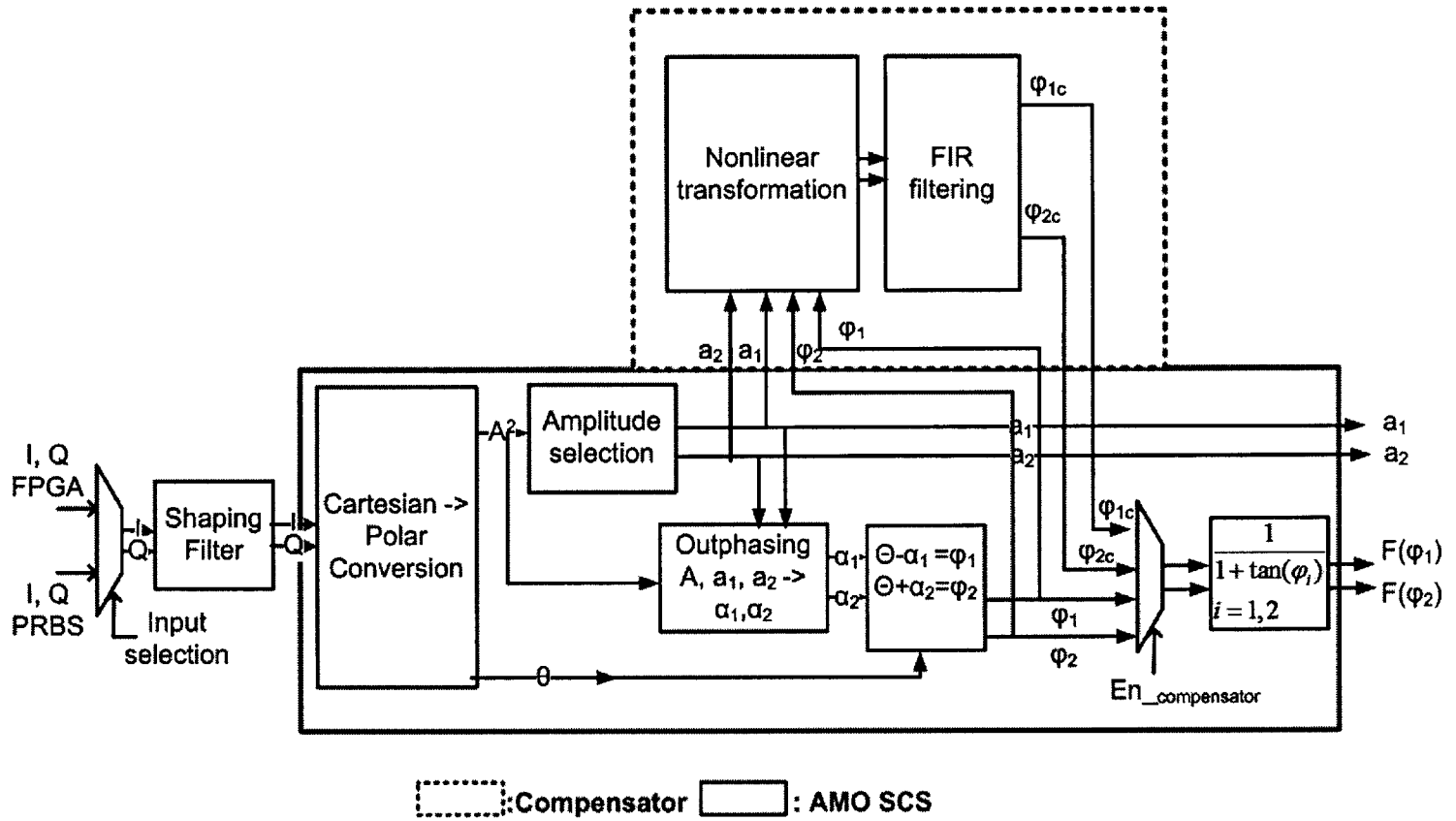


Figure 3-29: The block diagram of the digital baseband with AMO SCS and nonlinear compensator.

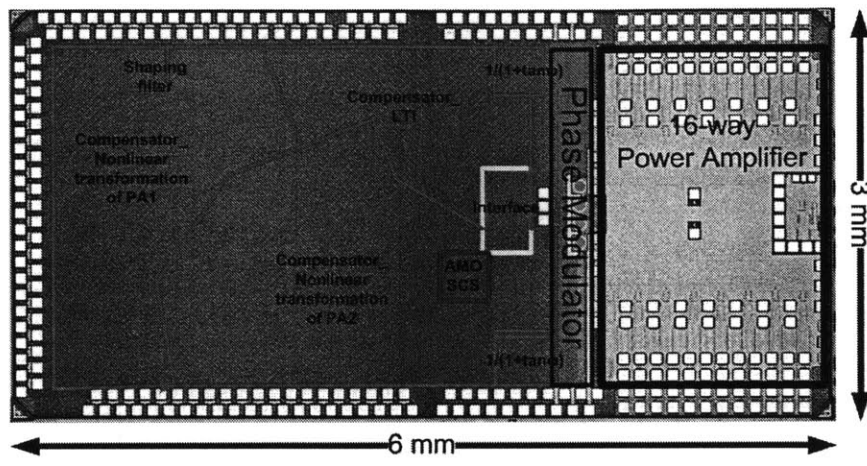


Figure 3-30: Overview of the integrated transmitter system with digital baseband nonlinear compensation.

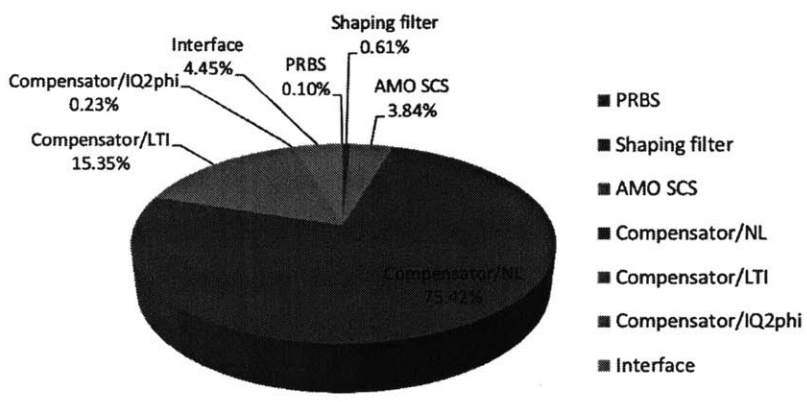


Figure 3-31: The area breakdown of the digital baseband with nonlinear compensator.

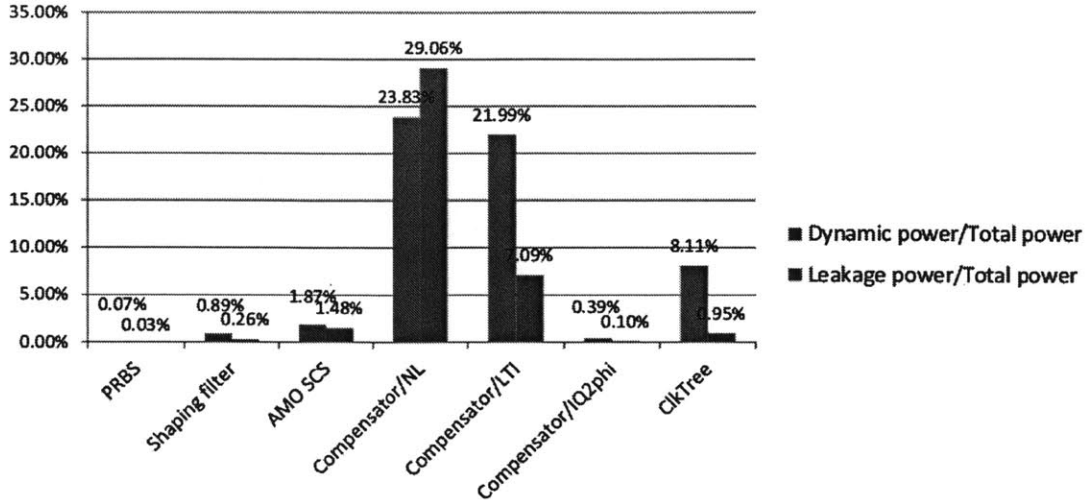


Figure 3-32: The power breakdown of the digital baseband with nonlinear compensator. This is an estimation from post-layout analysis at 1GHz clock frequency.

values. The x-axis represents the data sets. Since we have 10 input phase batches and in each batch we have four sets of repeating input phases lasting for 11 samples, there are a total of $10 \times (4 - 1) = 30$ comparisons and difference errors to calculate. The y-axis represents the relative l_2 norm error calculated as

$$\text{error}_{l_2} = \frac{\text{norm}_{l_2}(\text{target sample waveform} - \text{reference sample waveform})}{\text{norm}_{l_2}(\text{reference sample waveform})}. \quad (3.44)$$

The interesting point we find from the 5 different curves in Figure 3-36 is that as

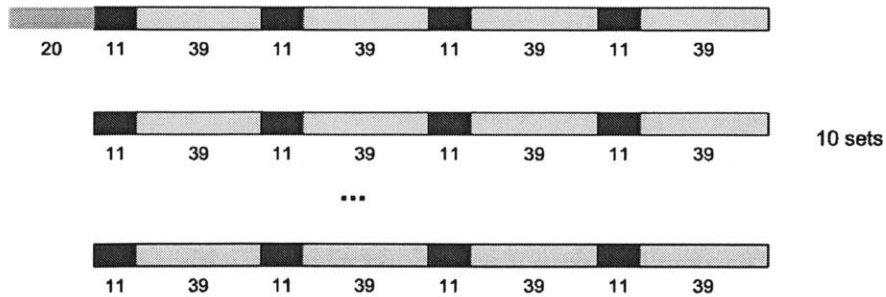


Figure 3-33: Input sequence structure to test the nonlinear system's stability.

more previous samples are included into the state, little or no improvement in l_2 norm

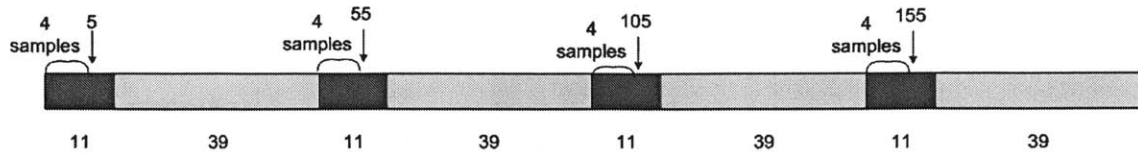


Figure 3-34: An example of constructing output comparisons with fixed previous 4 and the current samples.

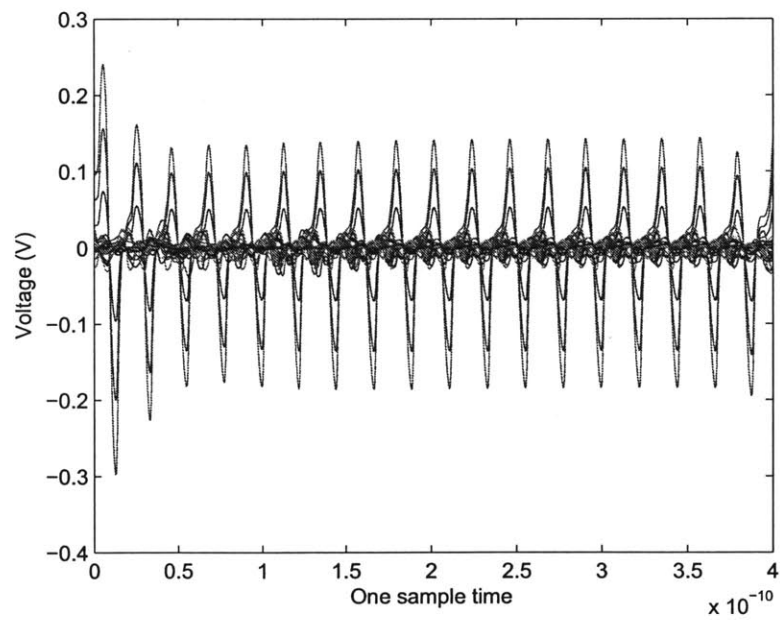


Figure 3-35: An example of output waveform differences with the same previous state and current input phases. The state includes four previous samples.

error can be found, and the error can be as large as nearly 5%. This observation leads to the conclusion that a certain degree of instability exists in the system, and is a fundamental limit of the model quality we are able to achieve.

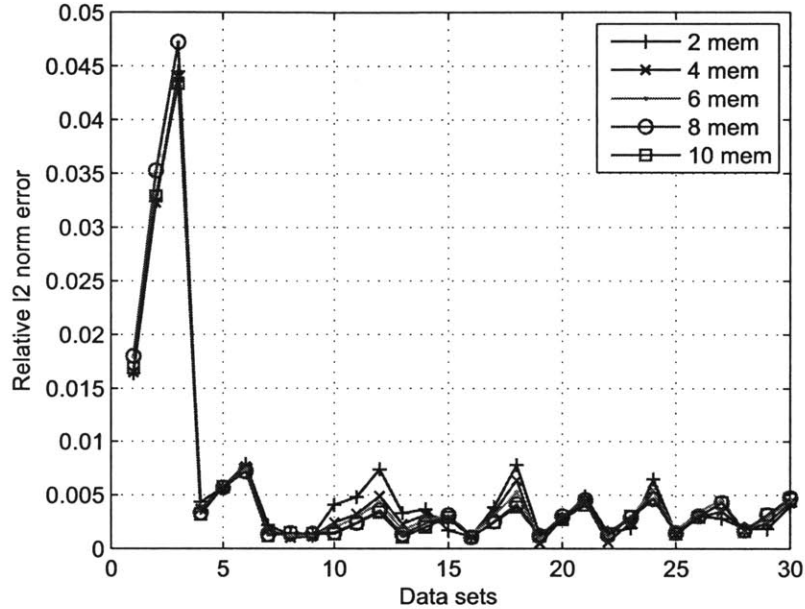


Figure 3-36: The l_2 norm error of all data sets with the same previous state and current input phase. Different curves represent the states containing different number of past samples.

One presumption for the instability factor of the system is the usage of the SOI process in this design. SOI technology has the advantage of reduced junction capacitance, due to the use of insulator rather than silicon as the substrate material in bulk CMOS process. The same factor also leads to the transistor body voltage to be unbiased and fluctuate based on the the switching history of the transistor [55]. This hysteretic behavior in transistor body voltage explains our above observation well because the memory we investigate e.g. 1-10 sample-time, becomes trivial compared to the body voltage memory history. To justify the conclusion, we use a reference PA design with the same architecture but all transistors body-tied and examine the same output. The reference design runs at 22.5GHz carrier frequency and the sample duration is 800ps. The same set of input sequence is used and we analyze the output

waveforms in the same manner as described above and obtain the results in Figure 3-37. Aside from the observation that the absolute l_2 norm relative error values are significantly smaller (over 5x), we also see that as the state includes more sample memory, the differences between the output at different times diminishes drastically and with 10-sample memory in the state, the error is less than 0.03%. This is the behavior that we would expect from a stable system, and the contrast in the error trend partially proves that the floating-body design does present a limit in the system modeling with stable models.

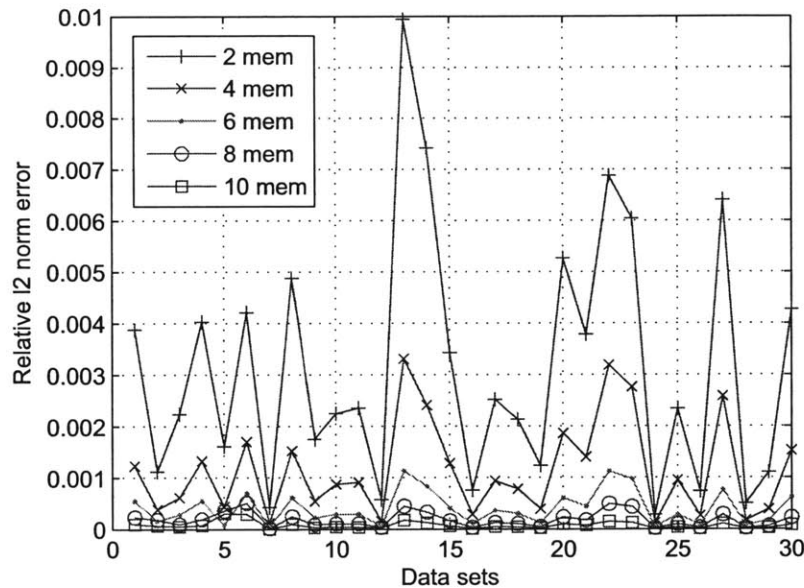


Figure 3-37: The l_2 norm error of all data sets with the same previous state and current input phase for the body-tied PA design. Different curves represent the states containing different number of past samples.

Aside from the perspective of phase memory, we also observe the difference between the body-tied and floating-body designs in their off-line iterations. As we have seen in Table 3.2, with body-tied design, the off-line iterations produce better converged compensation sequence with 10dB improvement in ACPR. This implies

3.4 Summary

In this chapter, we continue to use the outphasing amplifiers (LINC/AMO) as an example to show the strength of the digital compensation, even in such a complex analog system with nonlinearities happening along the whole signal chain. The digital compensation is able to generate the right set of inputs, with which the system yields the output that satisfies the linearity metrics. The introduction of the digital assistance to the analog system also helps relax the analog system blocks design specifications, such as the switch design, delay matching etc in the example we show in this chapter.

The key to the success of the digital compensator design is a detailed analysis of the system structure, leading to the right choice of model for the compensator. We first show the existence of a successful compensator through an iterative simulation strategy, and then demonstrate the effectiveness of the proposed dynamic model. To achieve a full integrated solution of a wideband mm-wave transmitter, we implemented the digital baseband with both SCS and compensation functionalities, and integrated it with the analog frontend. There is also room for further reduction in area and power, taking into account the large register-based LUTs. Finally, we further investigate the limitations of the compensator modeling and present a plausible explanation by comparing the different system behaviors of the floating-body and body-tied PA designs.

Chapter 4

A Hierarchical System Design Methodology

The system identification technique can also be extended to a general hierarchical system design methodology. In today's complex communication systems, one challenge for a system designer is to enable the link between top level design specifications and each subsystems' characteristics, such that both the top-level and subsystems are aware of the characteristics of each other. The system's optimal solution can only be found with a high-degree of interactions among different hierarchical levels. The involvement of such interactions in the system design leads to a high dimensional design space, and places a great challenge in hierarchical system design. Here, we show that this difficulty can be overcome with a combination of the equation-based design models that reduces the design space and the system identifications over a reduced space to help improve the accuracy of the subsystem's abstraction.

4.1 Proposed Hierarchical Design Methodology

In this design flow, all subsystems are characterized by a set of optimal design points parameterized by performance metrics of interest, which is essentially a Pareto set. Since a Pareto set is expected to be a smooth surface, we are able to fit the set with a parameterized continuous function, which not only abstracts the Pareto set, but also

interpolates the designs in between the points in the set. In this way, we are able to express system-level objectives and constraints as functions of subsystems' performances. This optimization problem can then be solved to an optimal or near-optimal design point, depending on the optimization problem's formulations. Figure 4-1 shows the above concept on a radio system example. Each circuit block is abstracted by a Pareto surface and a parameterized compact model from system identification, which represents the trade-off among different performance metrics. These Pareto surfaces and compact models are then used at a higher levels to obtain new design trade-offs.

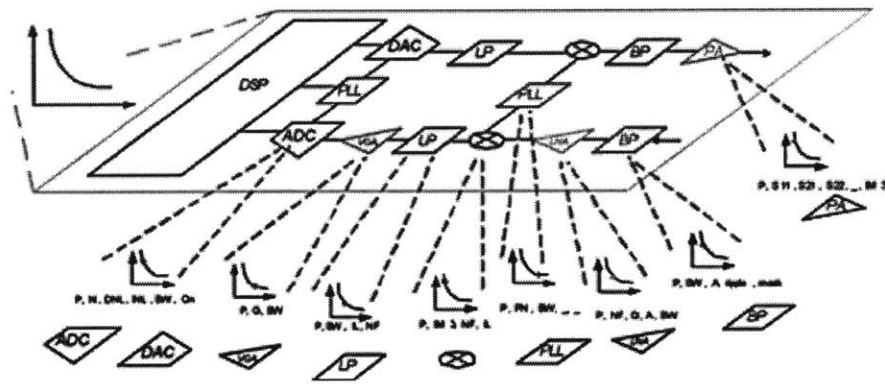


Figure 4-1: Hierarchical system design.

There are several critical points in the hierarchical design flow, which are shown in the following sections.

4.2 Design Space Reduction

In the hierarchical design flow, Pareto surfaces help reduce the design's dimensionality and space by abstracting each subsystem with its trade-off characteristics among key performance metrics. Those performance metrics connect subsystems with higher level design constraints. Therefore, once the right performance metrics of each subsystem are determined from higher-level design, the optimal subsystem is also available because it is on the Pareto surface. The generation of Pareto sets can be done in the

framework of equation-based circuit optimization flow.

4.2.1 Equation-based Circuit Optimization

With a given topology and specification, designers work on formulating circuit performances and biasing conditions as equations. With these equations and fitted transistor models, an optimization problem can be constructed based on design specifications and objectives. Finally, the optimization problem is solved by an optimizer which gives an optimal or near optimal design with all sizing and biasing information [56]. With this circuit optimization framework, it is easy to generate a desired Pareto set, simply by sweeping the correspondent performance specifications. For a two-stage operational-amplifier (Op-amp) with schematic shown in Figure 4-2, Figure 4-3 shows the result, where gain and load capacitance specifications are swept to obtain this Pareto set.

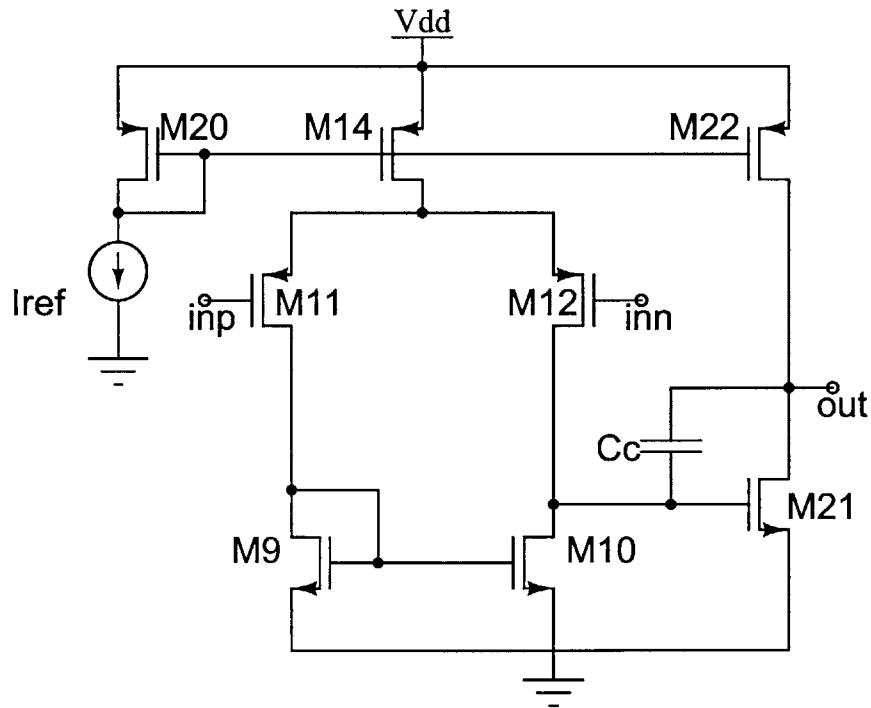


Figure 4-2: The two-stage Op-amp schematic.

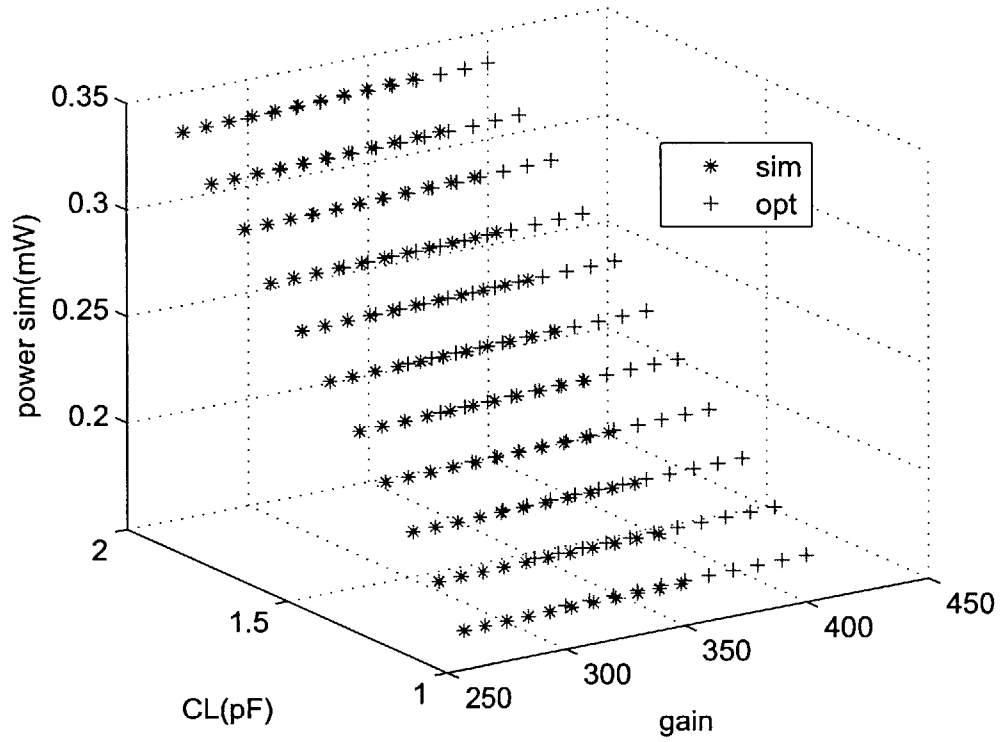


Figure 4-3: Design trade-off sets between gain and load capacitance, corresponding to grid 1.

4.2.2 Equation-based Robust Circuit Optimization

As transistor technologies continue to scale, analog designers must increasingly take into account the effects of process variation and mismatch. Therefore, in this hierarchical system design flow, it is important to incorporate a methodology of robust design within the equation-based circuit optimization framework. In [57], we show an iterative robust circuit optimization algorithm, which deals with random process variations, and is able to generate a design according to the yield specification.

Proposed Robust Circuit Optimization Algorithm

The intuition behind this algorithm is illustrated in Figure 4-4. In Figure 4-4, (a)-(e) depict the impact of variations on the feasible design space, the nominal design point, and how the algorithm pushes the nominal design point to be more robust in two iterations. Here, we assume a circuit design problem with underlying design variables as circuit sizing parameters (x-y plane, not shown here) and the objective to minimize the power consumption, represented by different level contours. The feasible design space is constrained by a set of biasing and performance constraints simplified to a polygon in the figures. In (a), a nominal design point is marked with a dot and one of the constraints (e.g. gain) is active. Under variations, the gain constraint shifts with respect to the x-y plane (the dashed line) and nominal design point now violates the design specification with some probability. Although the objective can also change under variations, it is not shown here for simplicity. In (c), the algorithm generates a variation-aware circuit scenario (adds constraints) to exclude the design space that would fail the gain specification under variations and push the design to a more robust design point, the triangle marker. In (d) and (e), the algorithm iterates once again, expanding the variation set.

A Fully Differential Folded-cascode Op-amp with Common-mode Feedback

The details of the algorithm are presented in Appendix B. Here, we show the results of applying the algorithm on a folded-cascode op-amp. Figure 4-5 shows the

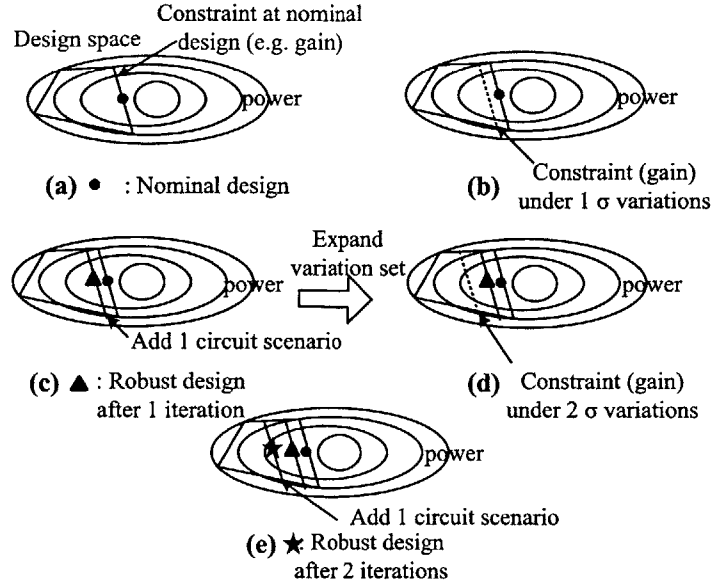


Figure 4-4: An intuitive illustration of the two iterations of the algorithm.

schematic of a fully differential folded-cascode op-amp with common-mode feedback (CMFB). The specifications considered in this example are listed in Table 4.1, with the objective to minimize a weighted sum of power and area. The variation sources under consideration are the threshold and current factor variations from a total of 32 transistors.

Table 4.2 shows the robust design evolution during iterations. The initial design has a transconductance value just on the lower bound of the specification and this critical constraint is found in the first maximization. To prevent this from happening, a circuit scenario is instantiated and along with the nominal circuit, it pushes the new robust design to a high transconductance value close to the upper bound. Therefore, in the second iteration, maximization finds another critical constraint, the transconductance upper bound constraint, and instantiates another circuit scenario to guard the upper bound. The next robust design that comes out of a three-scenario circuit optimization gives a lower G_m value and a higher yield. As iterations continue, no more constraint violations are found and the variation set keeps expanding, until the yield is satisfactory. As expected, the transconductance G_m from the optimizer

finally converges to the middle point of the specification range, i.e. 5.5 mS. Figure 4-6 shows the Monte-Carlo performed in the optimization domain of the initial and final robust design's Gm and the area and power consumption are shown in Table 4.2 . Here, we can see the tradeoff between the circuit performance and area and power costs.

Table 4.1: The folded-cascode op-amp example: specifications for nominal design

DC gain	50dB
phase margin	60°
Gm	[0.5 mS, 0.6 mS]
CMFB DC gain	60 dB
CMFB unity-gain bandwidth ($\omega_{u_{CMFB}}$)	5.9 MHz

Table 4.2: Folded-cascode op-amp: iterations of the robust designs from optimization: $gm \in [0.5 \text{ mS}, 0.6 \text{ mS}]$. k denotes the range of the variability. Please refer to Section B.

k	violated con- straints	Gm (mS)	yield (%)	area (μm^2)	power (mW)
0	$gm \geq 0.5$	0.5	49	539	0.27
0.2	$gm \leq 0.6\text{m}$	0.59	69	544	0.30
0.4	None	0.58	84	627	0.31
0.6	None	0.57	92	659	0.32
0.8	None	0.55	94	668	0.33

4.3 Subsystem Abstraction

For hierarchical systems, the complexity of the system usually prohibits the system simulations performed many times during design iterations. Therefore, we need a parameterized dynamic model for each subsystem, so that cascaded models could replace time-consuming spice-level simulations. The parameter dependence includes the performance variables in the Pareto surface, as well as input and output loads

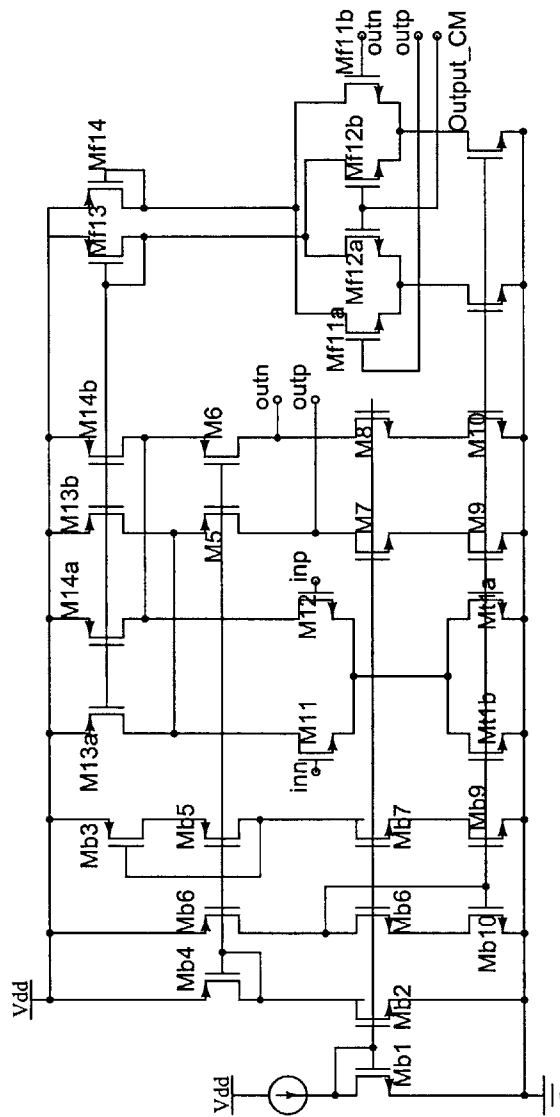


Figure 4-5: The folded-cascode example with $G_m \in [0.5 \text{ mS}, 0.6 \text{ mS}]$ from optimization: Monte-Carlo check with the initial and final robust design.

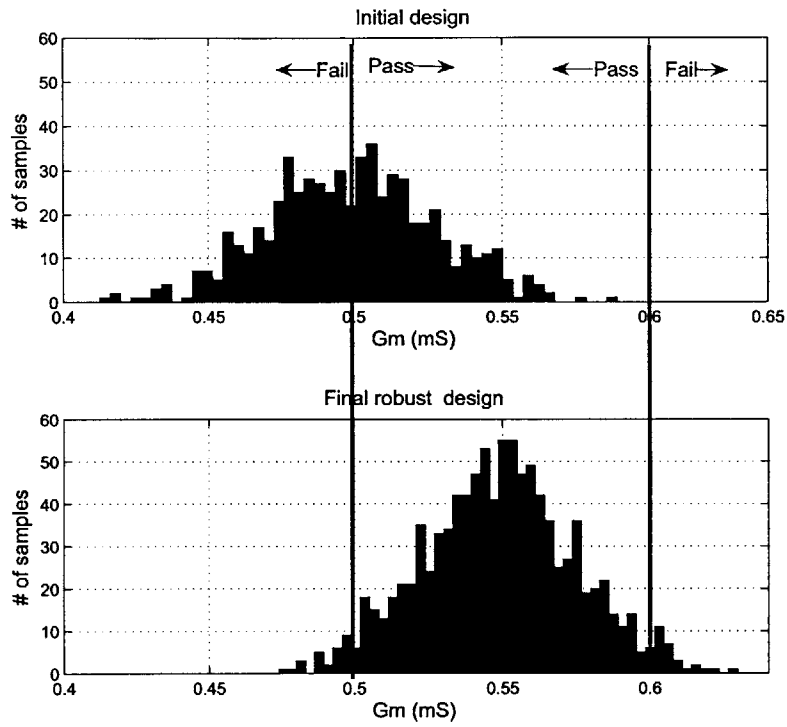


Figure 4-6: Monte-Carlo check with the initial and final robust designs: G_m of the folded-cascode op-amp.

to ensure the correct interface between two cascaded subsystems. With the system identification technique proposed in [58], we will show next how to apply the technique to obtain a nonlinear model of circuit design instances from the set that can be parameterized in circuit macro property parameters (gain, bandwidth, etc).

According to [58], we can build up a parameterized circuit system model:

$$\begin{aligned} F(Y[t], U[t], z) &= Q(y[t-1], \dots, y[t-m], u[t], \dots, u[t-k], z) \\ y[t] - p(y[t-1], \dots, y[t-m], u[t], \dots, u[t-k], z) &= 0, \end{aligned} \quad (4.1)$$

where $z \in \mathcal{R}^{N_z}$ are parameters. Q and p are polynomials on $Y[t], U[t]$ and z . The degrees of the polynomials on $Y[t]$ and $U[t]$ are ρ_{f_p} for p and ρ_{f_q} for Q ; and the degrees of polynomials on z are ρ_{z_p} for p and ρ_{z_q} for Q .

Before identification, we first need to obtain a set of design instances with different gain and load capacitance. We use the two-stage op-amp shown in Figure 4-2 as an example and this can be achieved by using the equation-based circuit optimization framework mentioned in the previous section. By simply sweeping the gain and load capacitance specifications, we are able to achieve designs optimized for different combinations of parameters.

The input signal we use for both training and testing are combinations of sinusoidal signals, with different frequencies, phases, and amplitudes, as shown in (4.2).

$$\text{inp} - \text{inn} = \text{vin}(t) = \sum_i A_i \sin(2\pi f_i t + \phi_i), \quad (4.2)$$

where A_i , f_i and ϕ_i are all randomly generated from uniform distributions. For our training purpose, we use 5 such sinusoids. We sweep the parameters gain and load capacitance from 300 to 400 and 1pF to 2pF, with intervals of 10 and 0.1pF, respectively. The parameter grid is shown in Figure 4-7, where each black circle and red dot represents a set of specifications on gain and load parameters. We have a total of 11×11 points on the parameter grid, which means that we need to generate 121 input-output training sets to use in the identification program. This makes the

optimization problem intractable. However, since the trade-off surface is smooth, we can sub-sample the parameter grid and use only 9 points to represent the whole parameter sets. The 9 points are illustrated as the red dots in Figure 4-7.

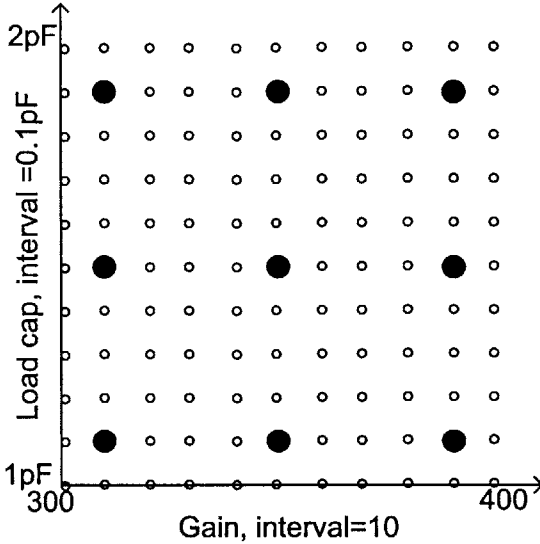


Figure 4-7: Parameter grid. Red dots are the designs used to generate parameterized model.

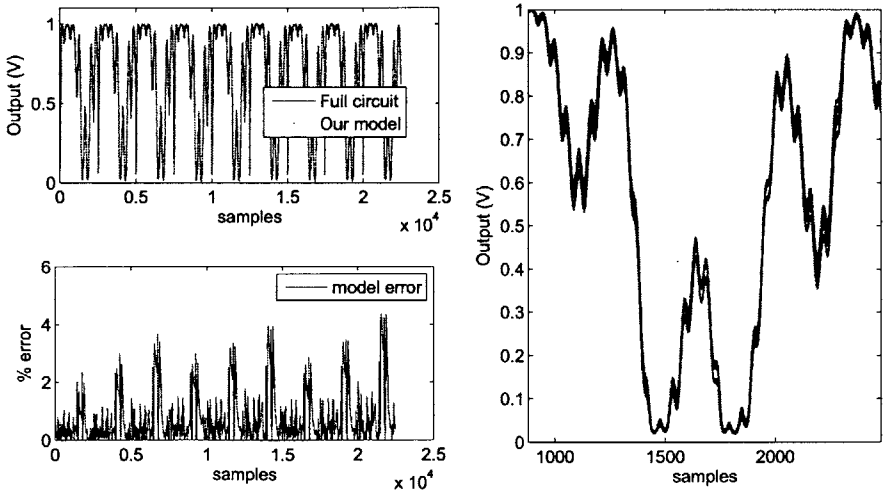


Figure 4-8: Parameterized system identification results.

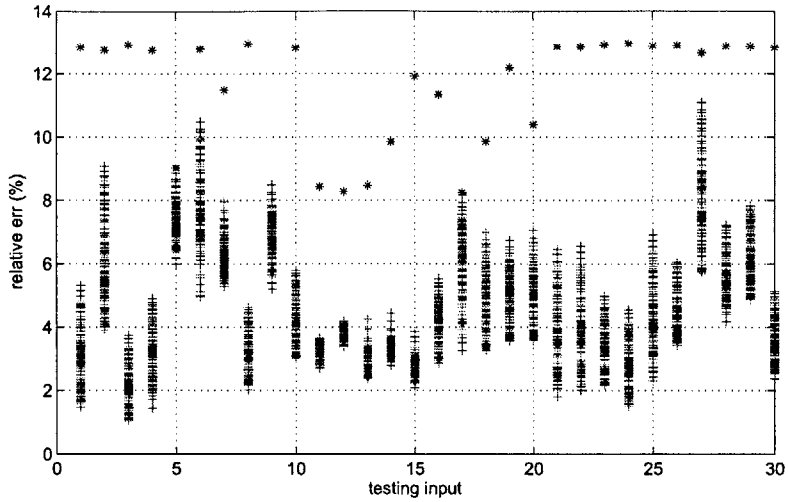


Figure 4-9: Testing results. The x-axis represents 30 testing input patterns and for each x value, there are 121 y values, denoted as colored '+', to represent the maximal relative error of the model output compared with the spice output in time domain. The '*' represent the maximal difference of the outputs from spice among the 121 designs for a certain input pattern (some x value). The 30 input patterns have random frequencies in the ranges [DC 146MHz] for the patterns 1-10, [DC 1MHz] for the patterns 11-15, [1MHz 10MHz] for the patterns 16-20, [10MHz 50MHz] for the patterns 21-25, [50MHz 146MHz] for the patterns 26-30.

Figure 4-8 shows the results by using only 9 input-output training pairs in the identification. The right top sub-figure is part of the samples where the 9 outputs are overlapped. This model has 2 input and 2 output memories, and the polynomial order of the denominator and numerator is 3 and 4, respectively. The total number of terms of the rational polynomial functions is 157 and this model shows a good accuracy ($< 6\%$) for the training data sets. For testing, we want to make sure that:

- (1) the model can produce accurate results for inputs other than the training inputs;
- (2) the model is able to accurately interpolate the designs on the overall parameter grid (121 design points). So, for each design on the grid we generate 30 input signals which have random frequencies from DC to unity-gain bandwidth. The testing results are shown in Figure 4-9. We see that with only 9 training sets we are able to build a model with accuracy of around 10% with random inputs which can interpolate the

design space consisting of 121 designs.

We use a coarser grid to test the applicability of the sub-sampling approach further. Figure 4-10 shows the new grid, where we double the range for gain and load capacitance but keep the number of samples in each dimension the same as before. We again use the 9 red dots on the grid to train the model. The accuracy of the identified model is still very good ($< 6\%$) for the training sets, and the model has the same complexity as before. We also use the same testing scheme as before to test all designs on the grid and with random testing inputs. The testing results are shown in Figure 4-11, with $< 15\%$ relative output error.

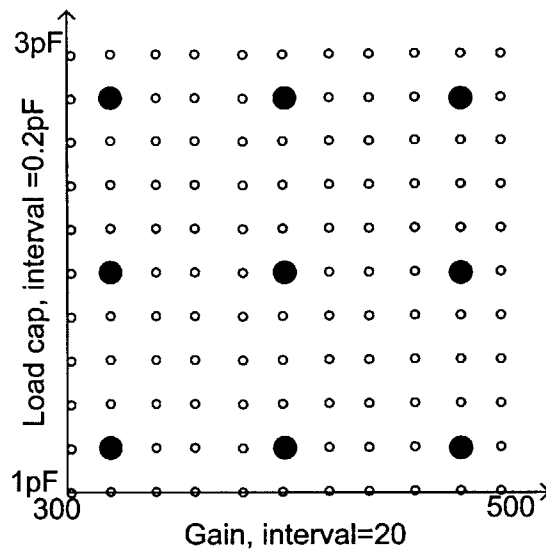


Figure 4-10: A coarser parameter grid 2.

4.4 Summary

This chapter, as an extension of the system modeling techniques discussed in the previous chapter, briefly introduces the hierarchical system optimization methodology. Two key points in the design flow are pointed out and some design examples and results are shown. As a future task, we can work out the whole flow on a complete hierarchical system, employing the techniques discussed in this chapter.

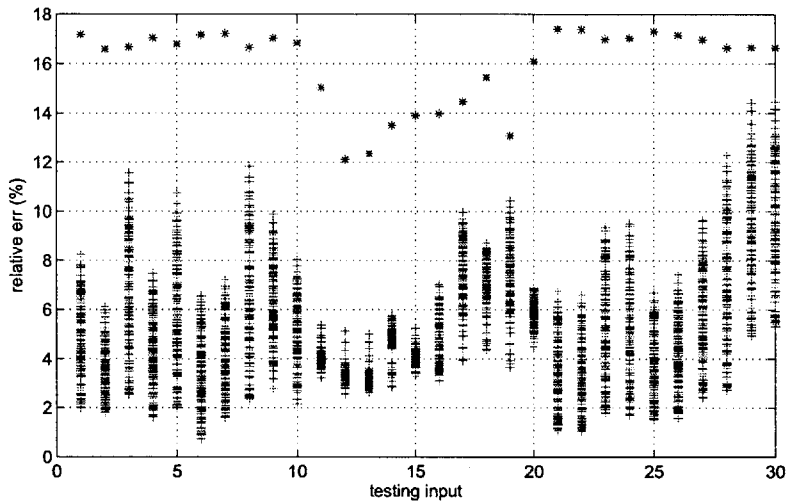


Figure 4-11: Testing results of the doubled-range parameter model. The x-axis represents 30 testing input patterns and for each x value, there are 121 y values, denoted as colored '+', to represent the maximal relative error of the model output compared with the spice output in time domain. The '*' represent the maximal difference of the outputs from spice among the 121 designs for a certain input pattern (some x value). The 30 input patterns have random frequencies in the ranges [DC 146MHz] for the patterns 1-10, [DC 1MHz] for the patterns 11-15, [1MHz 10MHz] for the patterns 16-20, [10MHz 50MHz] for the patterns 21-25, [50MHz 146MHz] for the patterns 26-30.

Chapter 5

Conclusions and Future Research Directions

In this thesis, we demonstrate the strength of the digital assistance through the digital baseband design for outphasing PAs. Two aspects of the digital assistance are explored: efficient nonlinear signal processing and real-time nonlinear compensator design. Both systems involve the design effort from both the algorithmic and hardware sides, and more importantly, reflect the concept of the co-optimization of the two aspects simultaneously to achieve the best balance between efficient algorithm and hardware. In the end of the thesis, we also shed light on a hierarchical system optimization by use of the system identification and equation-based robust optimization techniques.

The nonlinear signal processing unit, e.g. the SCS unit, is a necessary part of the modulation scheme used by the outphasing architecture. The core technology needed to realize its functionality is the elementary function computations. We propose a novel fixed-point piece-wise linear approximation for the nonlinear functions in the SCS. The approximation makes use of three basic energy-efficient computations: LUT, adder and multiplier. Also, the way the formulation is constructed minimizes the size of the operands of all the operations, therefore yields a design with high-throughput while being energy-efficient. We implement the SCS system in 45nm SOI technology. The tested system runs up to 3.4GSamples/s at 1.1V and at the

minimal-energy point, it has an energy-efficiency of 58pJ per sample with 800MSamples/s throughput and supply level of 0.7V. Compared to other projected previous techniques, this implementation shows a 2x improvement in energy-efficiency and 25x in area-efficiency.

To further relieve the tradeoff between linearity and power efficiency for the out-phasing PA, we design and implement the digital nonlinear compensator in the baseband system. We show an off-line iteration scheme to compensate for each particular input sequence, hence gives us the confirmation of the existence of a nonlinear compensator. To realize a real-time compensator, we analyze the structure of the nonlinear equivalent baseband system and propose the model for the PA inverse system. With the dynamical compensator model, we are able to achieve 10dB improvement in ACPR and up to 2.5% of EVM. As a complete solution for a mm-wave wideband transmitter system, we fabricate the baseband system with both SCS and nonlinear compensator, together with the phase modulator and 16-way PA in 45nm SOI technology. The tested system runs up to 500MSamples/s and the power-added efficiency penalty from this baseband is less than 1% with the PA system of 20% peak-efficiency and 5W output power.

For the further development on the algorithmic side, there are still several directions for further research. As we have seen, the zero-avoidance and level-avoidance are crucial to reach a satisfied compensated sequence through the off-line iterative scheme, filters with zero and level-avoidance properties are highly desirable. We suggest one type of such filters with heuristic in Appendix A, however there is no theoretical guarantee to remove all the samples out of the unwanted regions. Further theoretical development for this type of filters as well as their practical implementations are highly valuable.

We briefly discuss the limitations of the modeling and show that floating-body effect is one of the limiting factors in the modeling. It exhibits itself as an ultra-long memory effect. It will be beneficial to carry out more long simulations and analyze such effect more carefully. These simulations may provide more insight to the model structure that can capture such long memory effect.

On the implementation side, we should be able to further improve the energy-efficiency and area-efficiency by use of different types of storage, like SRAM for both SCS and compensator LUT implementations. The use of SRAM would help decrease both the leakage power and area. Currently, the nonlinear transformation part dominates the power consumption of the digital baseband, which are realized by the two-dimensional piece-wise approximation. Further improvement in energy-efficiency would require us to rethink other approaches to realize the nonlinear function computations in this block.

Last but not least, the compensator model should be trained with the testing data. A successfully working RF frontend will provide a lot more opportunities and challenges for the compensator design. We may need new ideas to acquire, analyze data, and build the compensator model.

Appendix A

Zero-avoidance Filter Design

Example Using Heuristics

The zero-avoidance filter is designed to replace the shaping filter in the communication system and serves an extra purpose of generating the output sequence with absolute value away from certain positive threshold, aside from the original purpose of the shaping filter: shape the spectrum with non-inter-symbol-interference (non-ISI) output sequence. Here, we present a design with heuristics where we achieve an output with significantly less points in the 'forbidden zone' as well as satisfying the spectrum specification and non-ISI property.

The basic idea of this design is to start with a normally shaped sample sequence, and select the samples that fall into the 'forbidden region'. The selected samples then pass through another system in such a way that when added back, the new sample sequence yields less ('no' in the best situation) points in the zone. Figure A-1 shows such a 'forbidden zone' depicted with shadow in the two-dimensional IQ plot. The goal is to move the violation points, denoted in red in the zoomed part, outside of the zone. Figures A-2(a)-(e) illustrate the design algorithm. The original shaping filter is first used to generate the shaped samples from a symbol sequence, as shown in A-2(a) and (b), where the lines with dots denote the symbols. Then according to the threshold that defines the 'forbidden zone', we can select samples in the zone. Then the selected samples form a sequence where all other samples are zero, as shown

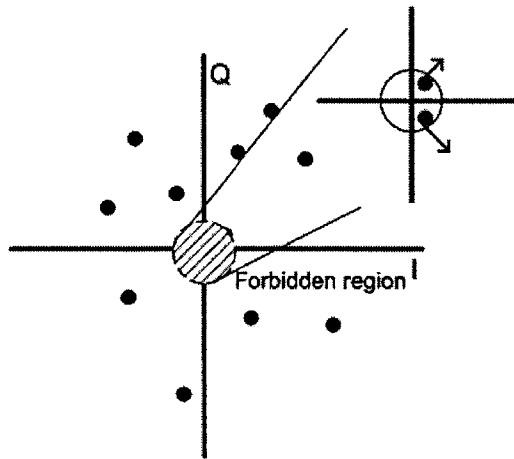


Figure A-1: Illustration of the zero-avoidance zone.

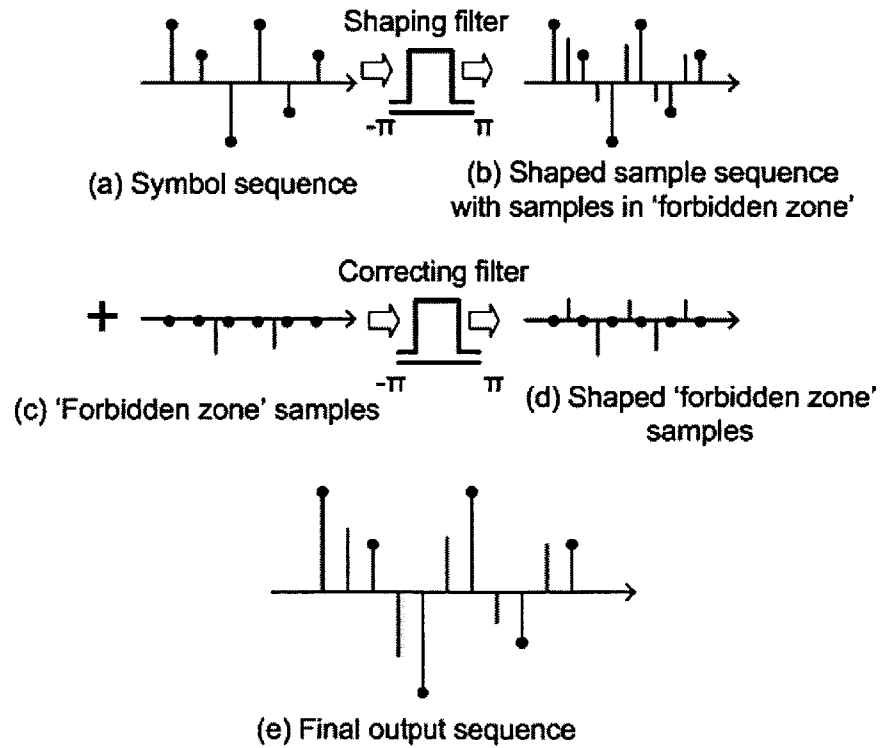


Figure A-2: Illustration of the zero-avoidance filter design.

in Figure A-2(c). The newly selected sample sequence passes through a new filter, the 'correcting filter', which generates a shaped 'forbidden zone' sample sequence. The correcting filter is designed with the same spectral and non-ISI constraints as the original shaping filter, therefore the generated 'forbidden zone' sample sequence has no ISI and its spectrum meets the same spectral mask as the original shaped sample sequence. Finally, the shaped 'forbidden zone' sample sequence is added back to the sample sequence out of the original shaping filter and generates the final output sequence, as shown in Figure A-2(e). This sequence still meets the non-ISI and spectral mask specification, and with less or none points in the 'forbidden zone'.

A design result is shown in Figure A-3, comparing among the original sampled sequence, a sequence after zero-avoidance filtering and the constellation. We can see that the zero-avoided sequence has obvious less points in the zone, whose radius is defined to be 12.7% of the constellation point with the largest amplitude. Quantitatively, the number of points in the zone is reduced from 89 out of 4096 samples to 43.

As we have seen from the algorithm and result, this heuristic method is effective, however with no guarantee in certainty to move the samples out of the zone. More developments have to be carried out in this topic to achieve more effective zero-avoidance.

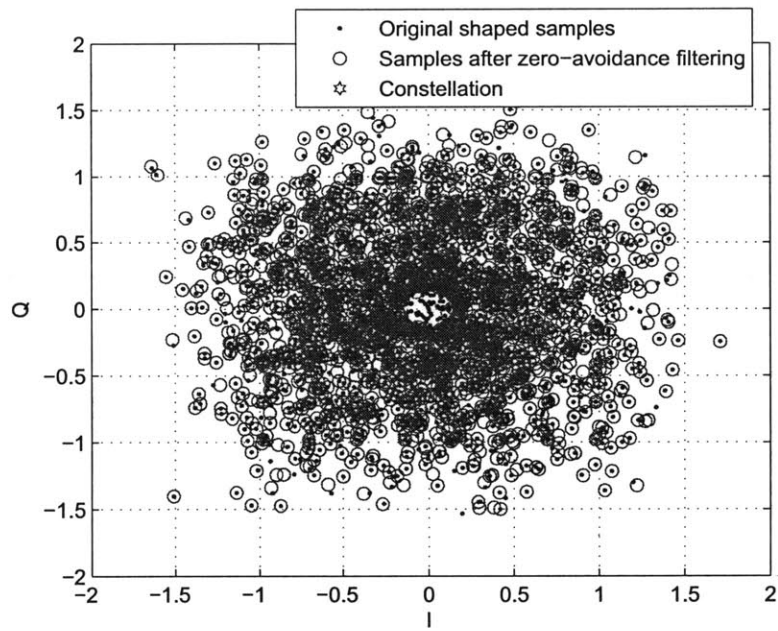


Figure A-3: Result of a zero-avoidance filter design.

Appendix B

Robust Iterative Optimization Algorithm for Analog Circuits

B.1 Algorithm

In this algorithm we focus on with-in die mismatch induced by random dopant fluctuations and line-edge roughness. Pelgrom's model [59] and derivative work, to the first order show that the impact of process variations on deviations of threshold voltage V_T and the relative current factor β are inversely dependent on the area of the transistor, as

$$\sigma_{V_T} = \frac{A_{V_T}}{\sqrt{W \cdot L}}, \text{ and } \frac{\sigma_\beta}{\beta} = \frac{A_\beta}{\sqrt{W \cdot L}}, \quad (\text{B.1})$$

where W and L are the width and length of a transistor; A_{V_T} and A_β are mismatch parameters.

B.1.1 Proposed robust circuit optimization framework

Figure B-1 shows the flow of the algorithm. The following subsections elaborate and map the algorithm steps on a two-stage op-amp circuit example in Figure B-2, to illustrate the specific circuit-related refinements.

B.1.2 Sources of Variability

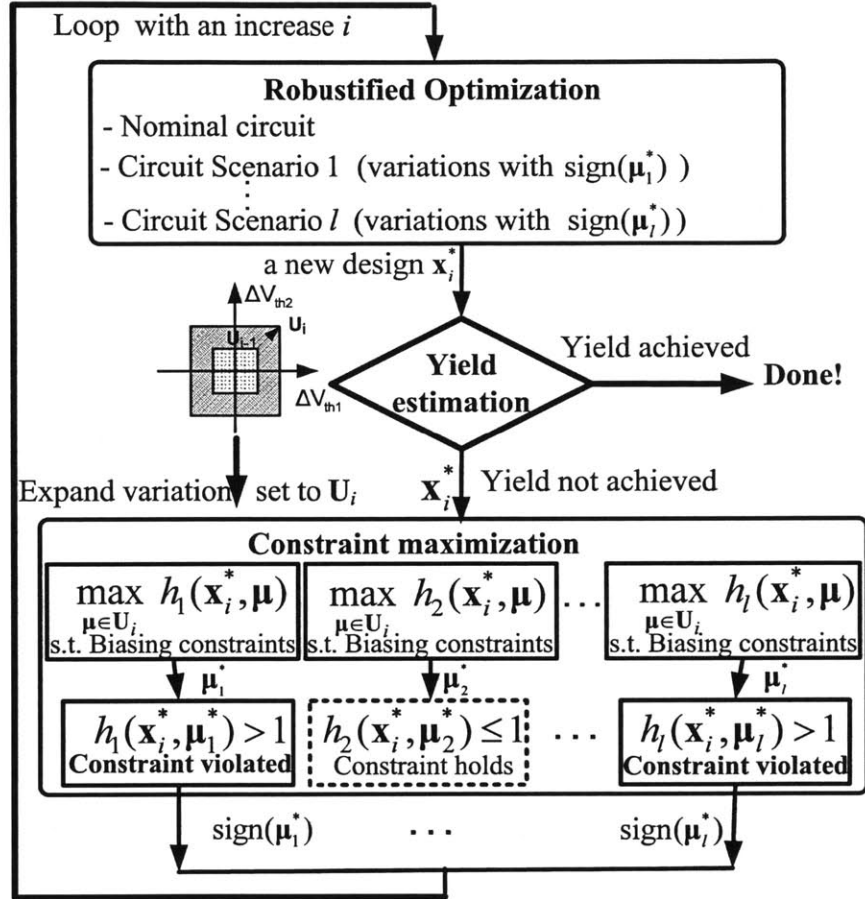


Figure B-1: Flow of the iterative robust algorithm.

Initial design

The initial design is a nominal circuit optimization problem without consideration of variations. It corresponds to the Robustified circuit optimization block without any

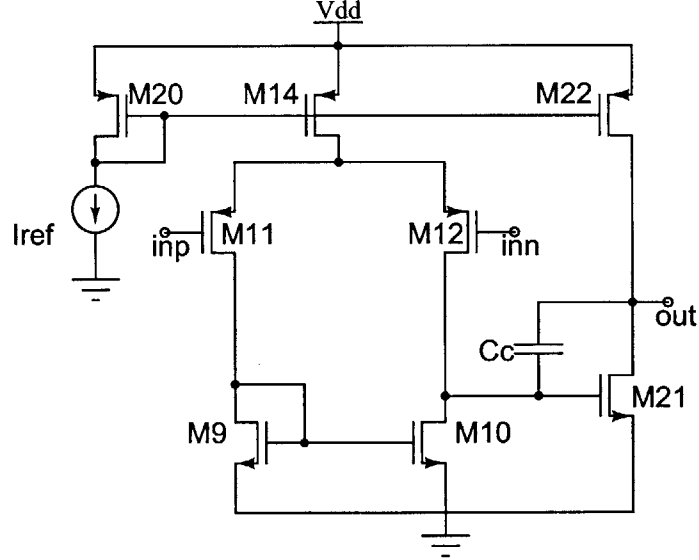


Figure B-2: The two-stage Op-amp schematic.

added circuit scenarios in Figure B-1. The formulation is as the following:

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} && f_0(\mathbf{x}, \boldsymbol{\mu}^*) \\
 & \text{subject to} && f_i(\mathbf{x}, \boldsymbol{\mu}^*) \leq 1, i = 1, 2, \dots, n, \\
 & && g_j(\mathbf{x}, \boldsymbol{\mu}^*) = 1, j = 1, 2, \dots, m, \\
 & && h_k(\mathbf{x}, \boldsymbol{\mu}^*) \leq 1, k = 1, 2, \dots, l, \\
 & && \boldsymbol{\mu}^* = 0,
 \end{aligned} \tag{B.2}$$

where $f_i(\mathbf{x}, \boldsymbol{\mu}^*)$ and $g_j(\mathbf{x}, \boldsymbol{\mu}^*)$ are biasing constraints and $h_k(\mathbf{x}, \boldsymbol{\mu}^*)$ are performance constraints. $\boldsymbol{\mu}^* = 0$ means that no variation is considered here. In the two-stage op-amp example, we minimize a weighted sum of power and area, subject to the performance constraints illustrated in Table B.1, and biasing constraints (KVL, KCL, transistor regions).

After the initial design, we make the optimization variation-aware in the following iterations by adding the critical variation direction information. To obtain the critical variation directions, the next subsection Constraint maximization performs a smart

search and adds the information in the form of new circuit scenarios to the Robustified circuit optimization block.

Table B.1: Specifications for nominal design

open loop gain	300
unity-gain bandwidth (ω_u)	160MHz
phase margin	60°
CMRR	100
slew rate	100MV/s
input-referred spot noise @ 1MHz (vnoise)	64nV/ $\sqrt{\text{Hz}}$

Constraint maximization

Constraint maximization identifies the set of critical performance constraints that would violate the corresponding specifications under process variations. It involves solving the optimization problem defined in (B.3) for each **performance** constraint. This makes the algorithm scale well, since the number of performance constraints does not necessarily grow as the size of the circuit grows.

The problem can be formulated as

$$\begin{aligned}
 & \underset{\boldsymbol{\mu}}{\text{maximize}} && h_k(\mathbf{x}^*, \boldsymbol{\mu}), k = 1, 2, \dots, l, \\
 & \text{subject to} && \boldsymbol{\mu} \in \mathbb{U}, \\
 & && \text{Biasing constraints,}
 \end{aligned} \tag{B.3}$$

where $h_i(\mathbf{x}^*, \boldsymbol{\mu})$ is the performance constraint evaluated at design point \mathbf{x}^* from previous circuit optimization step, with variation variable vector $\boldsymbol{\mu}$. The constraints are bounding and biasing constraints. No probability distribution is assumed. The range for the i^{th} variation variable can be written to be proportional to its standard deviation as $\mu_i = [-k\sigma_i, k\sigma_i]$, where σ_i is the standard deviation of the i^{th} variation and k is a constant. In each iteration, the variation set can be expanded by increasing k . In the two-stage op-amp example, for the bandwidth constraint maximization, the

objective in (B.3) becomes $h_k(\mathbf{x}^*, \boldsymbol{\mu}) = \text{spec.}\omega_u/\omega_u(\mathbf{x}^*, \boldsymbol{\mu})$.

The purpose of expanding the variation set in each iteration is to take more variations into consideration and search for more critical variation directions that would fail the design. Therefore, as iterations go on, the robustified circuit optimization is aware of more critical directions and pushes the design to become more and more robust. By expanding the variation set iteratively, we achieve a design just meeting the yield target, avoiding over-design.

The key circuit-specific refinement in this step is to realize that during the maximization, circuit biasing constraints have to be satisfied even in the presence of random variations for the circuit to be realizable. Therefore, all circuit descriptions are defined with the macro transistor model in Figure B-3. Figure B-3(a) reflects the effect of threshold and current factor variations through the voltage source ΔV_{Th} and the current source ΔI_d , defined in (B.1). We can further simplify the model to Figure B-3(b), which has an aggregate standard deviation accounting for both threshold and current factor variations, as shown in (B.4),

$$\sigma_{V_r}^2 = \frac{A_\beta^2}{W \cdot L} \left(\frac{I_d}{gm} \right)^2 + \frac{A_{V_{Th}}^2}{W \cdot L}. \quad (\text{B.4})$$

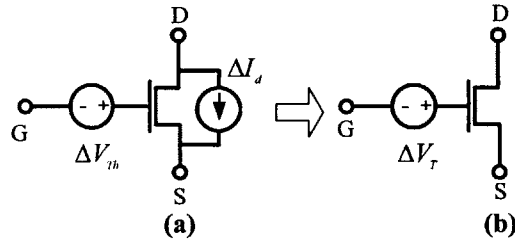


Figure B-3: Transistor Macro model.

Although (B.3) seems to be a general optimization problem, a close examination on the dependence of circuit performances on process variations reveals that circuit performances can be assumed to be locally monotonic on process variations (but not circuit design parameters). This assumption holds reasonably well according to the

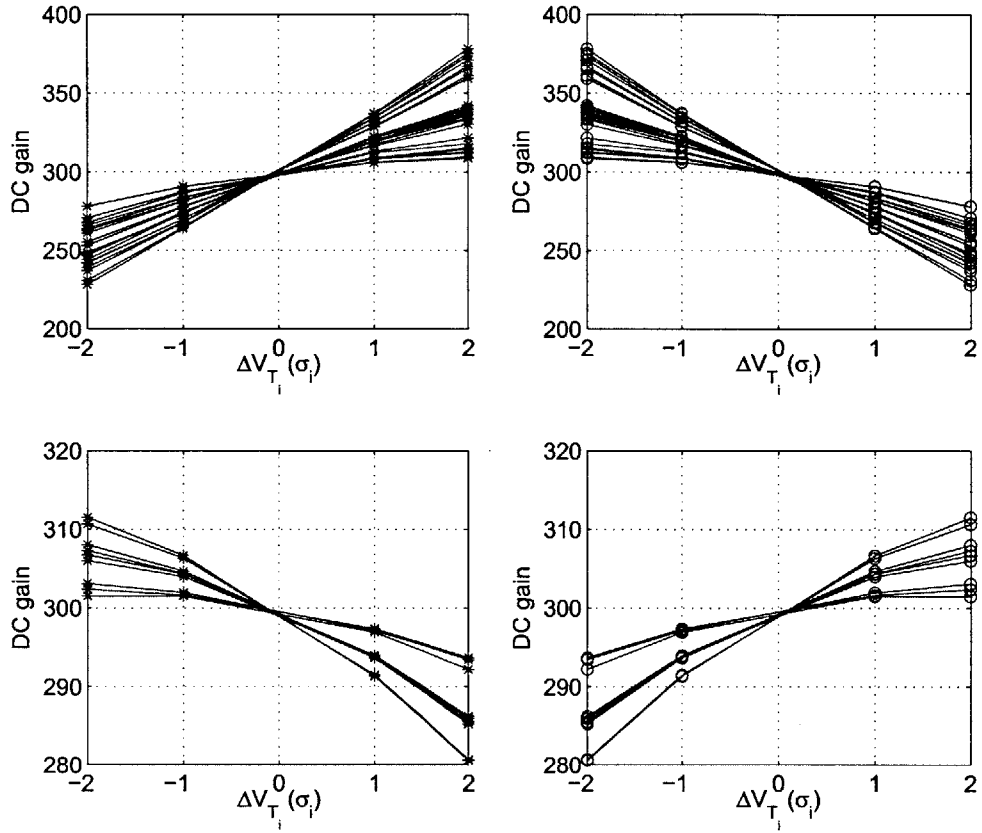


Figure B-4: Monotonicity of circuit performances on variation variables in a two-stage op-amp example.

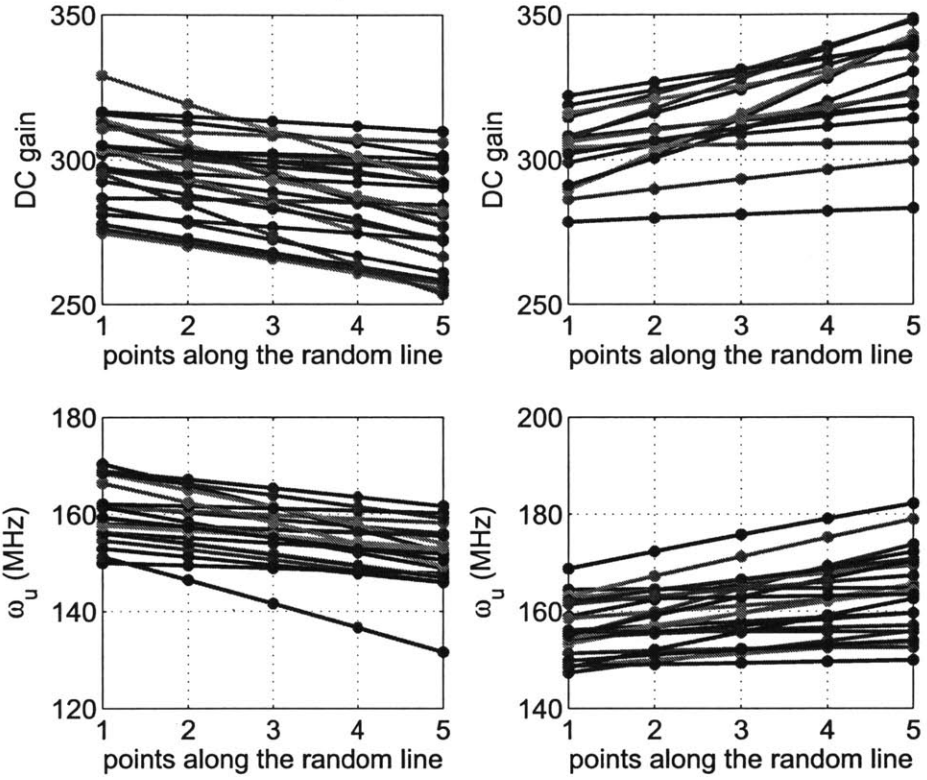


Figure B-5: Monotonicity of circuit performances along random lines in variation variable space in a two-stage op-amp example.

work in RSM, since linear regression model is a traditional way to fit the circuit performance on variation variables [60] [61] and linear model is a monotonic function on fitting variables, the process variations. Furthermore, the variation set is always set to be small initially, i.e. $[-0.2\sigma_{V_T}, 0.2\sigma_{V_T}]$, and often does not need to be expanded to as large as $[-3\sigma_{V_T}, 3\sigma_{V_T}]$. In the two examples we show in the next section, k only increases to around 1 with yield already approaching 100%. Under this small variation range, the accuracy of the linear model is satisfactory, hence our assumption on monotonicity is also reasonable.

To validate the assumption on this op-amp example, we let $k = [1, 2]$, with each transistor's ΔV_T ranging from $-k\sigma_{V_T}$ to $k\sigma_{V_T}$. For each k , we select random variation corners out of the total 2^8 corners and record the DC gain and unity-gain bandwidth of the design under these variations. The performances show nicely monotonic functions as we increase the k , as shown in Figure B-4, where the x-axis denotes equally spaced points along the line. A more general test is to randomly select a line in the variation variable space, i.e., $\mathbf{s}, \mathbf{v} \in [-2\sigma_{V_T}, 2\sigma_{V_T}]$ and let $\mathbf{s} + t \cdot \mathbf{v}$ be the variation vector with t as a varying scalar. Figure B-5 shows monotonic performance variations along different random lines. Because of the monotonicity of performances on variation variable, the objective function in (B.3) becomes monotonic together with bounding constraints, and the optimal solution of the maximization problem can be found easily with general gradient-based optimization solvers.

Although there is no theoretical proof showing the monotonicity property on variations, it is confirmed in the two examples presented here. In the worst situation where the monotonicity does not hold, the algorithm should still work, with the price of adding sub-worst circuit scenarios during iterations. Those sub-worst circuit scenarios can still help push the design to be more robust.

We also notice that since the optimization solver is inherently a GP-based solver, it requires all design variables to be positive. However, ΔV_T range obviously includes negative values. A brute-force way to overcome this problem is to prefix the sign of ΔV_T and maximize over all sign combinations, which makes the number of maximization grow exponentially with the number of variation variables. However, with the

objective function being monotonic over variation range, it is easy to map the solution from the maximization done under the positive variations to a solution under the whole variation range. This is illustrated in Figure B-6, in which case two transistor variations are considered. Suppose we maximize $\text{spec.}\omega_\mu/\omega_\mu$ over the shadow area. Then, because of monotonicity, the optimal solution should be on one of the four vertices of the shadow area. If we obtain the optimal solution at O_1 and know that the objective function increases as μ_{VT_2} decreases, then if not restricted to positive range, the solution O_1 should shift to O'_1 . This simple mapping can help obtain the optimal solution to the maximization problem efficiently.

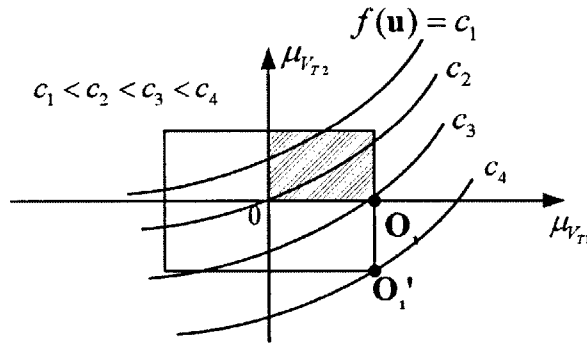


Figure B-6: Constraint maximization with two variation variables.

Robustified circuit optimization

With the critical variation directions obtained in constraint maximization, the next step is the Robustified circuit optimization block in Figure B-1, to achieve a new robust design. If the design has not been robust over the whole variation range, we should be able to identify a nonempty set of constraints whose right-hand sides are violated, i.e. $h_k(\mathbf{x}^*, \boldsymbol{\mu}^*) > 1$. Since each of the worst-case variation vectors $\boldsymbol{\mu}_i^*$ can cause a different biasing condition, a circuit scenario has to be instantiated for each of them. The new circuit scenario is setup such that it has the same topology and shares the optimization variables \mathbf{x} with the nominal circuit, except that it uses the macro model in Figure B-3(b) for transistors, with variability vector having the sign of $\boldsymbol{\mu}_i^*$ and magnitudes parameterized by optimization variable \mathbf{x} .

The key point in the context of the circuit optimization, e.g. two-stage op-amp, is that the maximization solution $\mu^* = \Delta V_T^*$ is used to determine only the polarity of ΔV_T , and the value of ΔV_T in the macro model is a function of design variables as $k\sigma_{V_T}$. This enables the optimizer to recognize that resizing the circuit will help decrease the variation and improve the worst performance. Here, k is the constant used to determine the range of the variation set in maximization, and σ_{V_T} is defined in (B.4). Therefore, the new scenario reflects the real situation when the nominal circuit is under variation with degraded performance. By doing multi-scenario optimization, we ensure that the degraded performance meets the specification, thus giving the nominal performance more margin to fail.

From the perspective of computation cost, adding clone circuits does create a larger optimization problem to solve. However, notice that the number of clone circuits is equal or less than the number of performance constraints, which in usual case is on the order of ten or less. Besides, the interior-point method used in GP-based solvers scales gracefully with increase in the number of constraints [62], since the number of optimization variables remains the same as in the nominal optimization problem.

Yield estimation

Yield estimation closes the loop in Fig B-1. Although many fast yield estimation methods exist, for instance importance sampling [63], pseudo-noise analysis [64], here we use the direct Monte-Carlo sampling method performed in optimization domain. For each Monte-Carlo variability sampling point the optimizer solves a feasibility problem with all design variables fixed, checking the yield with proper biasing and performance constraints. In the result section we will see that the yield estimated in optimization domain follows the yield obtained from Hspice simulation, making these other simulation-based yield estimation methods possible to use in this context as well.

B.2 Experimental results

A two-stage op-amp example is developed here to show that the algorithm is able to add more robustness to the circuit and increase the design yield. The optimizer uses signomial transistor models derived from a 90 nm predictive technology model.

B.2.1 A two-stage op-amp example

This is the example discussed in Section B.1.1, with the schematic shown in Figure B-2. The specifications of the op-amp are listed in Table B.1 with 2 pF load capacitance and design objective to minimize a weighted sum of gain and area. The variation sources under consideration are threshold voltage and current factor of total 8 transistors.

One-corner initial design

The initial nominal design is optimized under **tt** corner with 1 V supply voltage, 10 μ A reference current and temperature at 298 K. The nominal design just meets the gain and bandwidth (ω_u) constraints. In the first iteration, only these two constraints are found to violate the specifications after maximizations. Based on optimal solutions from maximization, we instantiate two circuit scenarios where the polarity of the ΔV_T voltage sources are determined from the sign of maximization solutions. Then a 3-scenario circuit optimization is solved and we reach a robust design with gain and ω_u improved to 311 and 163 MHz (measured without variability). As iterations go on, the design margins increase at the same time, shown in the two columns under optimization in Table B.2. Table B.2 also shows the simulation results for verification. Because of the transistor model's signomial fitting error, there is around 20% mismatch between Hspice simulations and optimization results. However, the simulation results do show the same trend of the improved performance and yields, with yields shown in the top two figures in Figure B-7 (the Hspice yields are calculated with respect to the *simulated* nominal design performance without variations). The cost of robustified design are increased power and area shown in the bottom two

figures in Figure B-7.

Table B.2: Robust two-stage op-amp designs in iterations from optimization and Hspice simulation

k	optimization		simulation	
	DC gain	ω_u (MHz)	DC gain	ω_u (MHz)
<i>N/A</i>	300	160	263	125
0.2	311	163	268	130
0.4	320	166	273	133
0.6	328	168	275	135
0.8	337	170	281	137
1	344	172	282	139
1.2	351	173	285	141
1.4	360	174	288	142

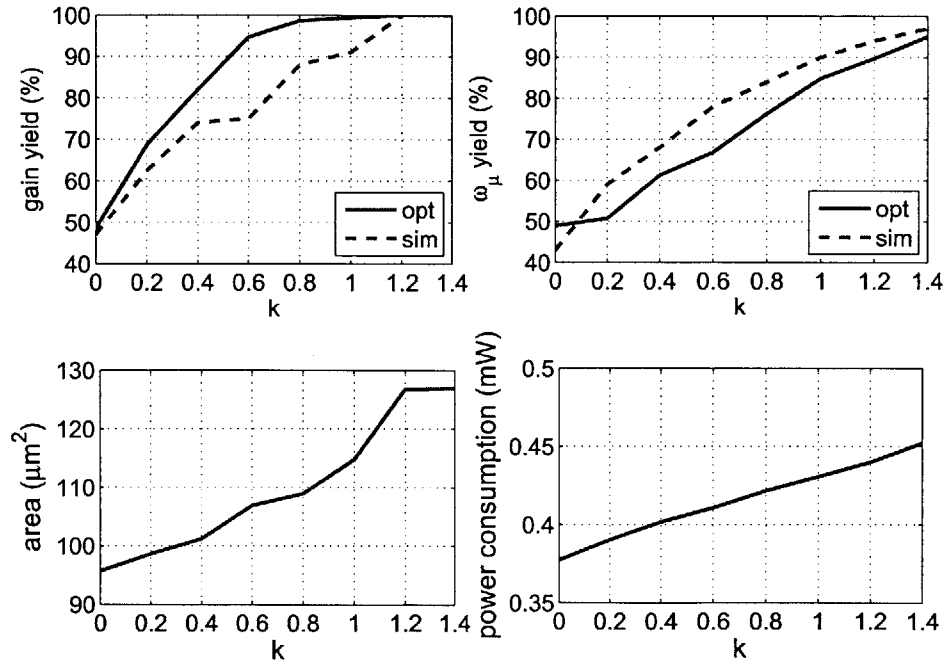


Figure B-7: Two-stage op-amp with 1-corner initial optimization design: yield improvement of gain and ω_u , power and area consumptions.

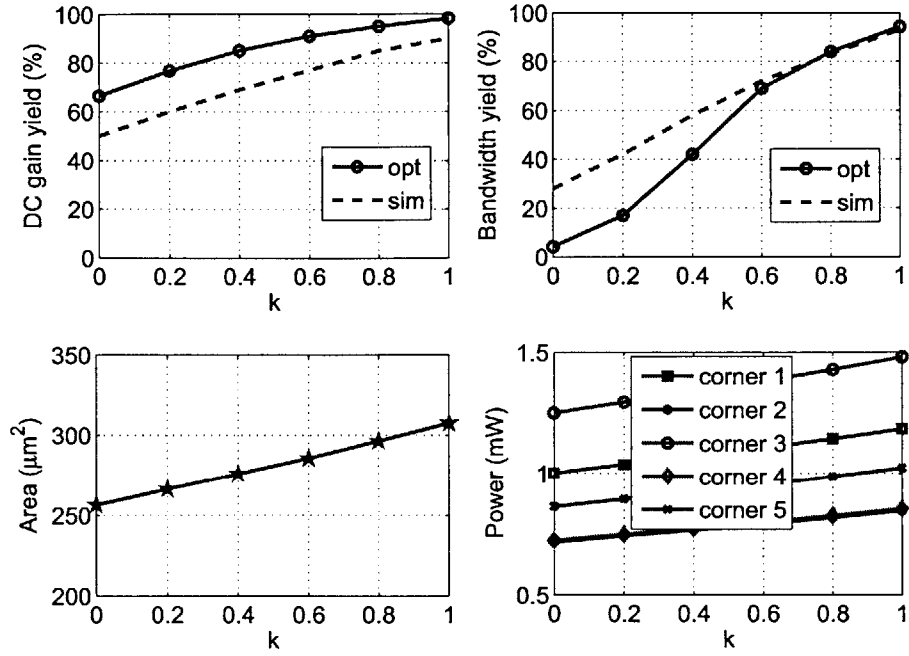


Figure B-8: Two-stage op-amp five-corner optimization design: yield improvement on gain, ω_u and power, area consumptions.

Table B.3: Five-corner of the two-stage op-amp initial design.

number	corner	temp (K)	vdd (V)	Iref (μA)
1	tt	298	1	10
2	ss	398	0.9	8
3	ff	233	1.1	12
4	fs	398	0.9	8
5	sf	233	1.1	8

Five-corner initial design

Here, we consider the situation where designers start from a fairly robust design. The initial design is a multi-scenario optimized design for the 5-corner listed in Table B.3. After applying random variations to the optimized design, gain and bandwidth at the second corner are found to violate the specifications and the corresponding yields for that corner drop to around 60% and 20% respectively, as shown in the top two figures in Figure B-8. This implies that in a design optimized over multiple PVT corners, even though some PVT corners show performances well above specifications even under random variations, the yield of some corners can be very low under random variations.

After adding two circuit-scenarios representing gain and bandwidth with variation vectors found in maximization at the second corner (*ss*), and optimizing the multi-scenario circuit, the yields for the second corner gradually increase to around 100%, shown in Figure B-8. The final robust design achieved in optimization is shown in Figure B-9, compared with the initial design. The cost to achieve the improvement is shown in the bottom two figures in Figure B-8. The simulations in Hspice of the second corner again show the same trend in yields in Figure B-8.

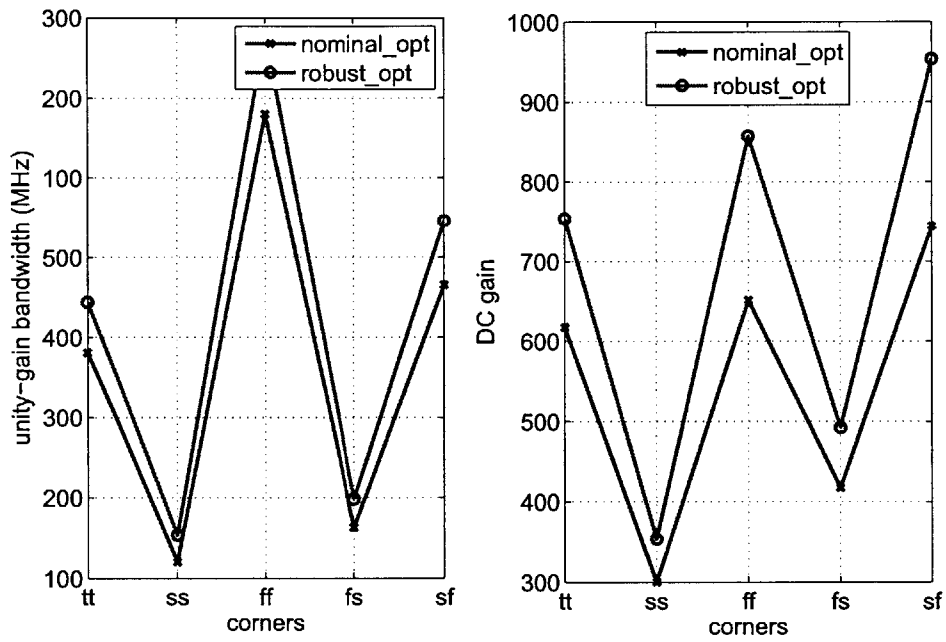


Figure B-9: Two-stage op-amp five-corner optimization design: DC gain and ω_u comparison of initial and final robust designs.

Bibliography

- [1] J. Warnock, "Circuit design challenges at the 14nm technology node," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, june 2011, pp. 464 –467.
- [2] J. Sevic, K. Burger, and M. Steer, "A novel envelope-termination load-pull method for acpr optimization of rf/microwave power amplifiers," in *Microwave Symposium Digest, 1998 IEEE MTT-S International*, vol. 2, jun 1998, pp. 723 –726 vol.2.
- [3] W. Bosch and G. Gatti, "Measurement and simulation of memory effects in pre-distortion linearizers," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 37, no. 12, pp. 1885 –1890, dec 1989.
- [4] J. Lajoinie, E. Ngoya, D. Barataud, J. Nebus, J. Sombrin, and B. Rivierre, "Efficient simulation of npr for the optimum design of satellite transponders sspas," in *Microwave Symposium Digest, 1998 IEEE MTT-S International*, vol. 2, jun 1998, pp. 741 –744 vol.2.
- [5] A. Lim, V. Sreeram, and G. qing Wang, "Digital compensation in iq modulators using adaptive fir filters," *Vehicular Technology, IEEE Transactions on*, vol. 53, no. 6, pp. 1809 – 1817, nov. 2004.
- [6] M. Perrott, I. Tewksbury, T.L., and C. Sodini, "A 27-mw cmos fractional-n synthesizer using digital compensation for 2.5-mb/s gfsk modulation," *Solid-State Circuits, IEEE Journal of*, vol. 32, no. 12, pp. 2048 –2060, dec 1997.
- [7] S. Mazlouman, S. Sheikhaei, and S. Mirabbasi, "Digital compensation techniques for frequency-translating hybrid analog-to-digital converters," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 3, pp. 758 –767, march 2011.
- [8] P. Nikaeen and B. Murmann, "Digital compensation of dynamic acquisition errors at the front-end of high-performance a/d converters," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 3, pp. 499 –508, june 2009.
- [9] J. Laskar, S. Pinel, S. Sarkar, B. Perumana, and P. Sen, "The next wireless wave is a millimeter wave," *Microwave Journal*, vol. 50, no. 8, pp. 22 –36, august 2007.

- [10] S. Pinel, P. Sen, S. Sarkar, B. Perumana, D. Dawn, D. Yeh, F. Barale, M. Leung, E. Juntunen, P. Vadivelu, K. Chuang, P. Melet, G. Iyer, and J. Laskar, "60ghz single-chip cmos digital radios and phased array solutions for gaming and connectivity," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 8, pp. 1347–1357, october 2009.
- [11] C. Marcu, D. Chowdhury, C. Thakkar, L.-K. Kong, M. Tabesh, J.-D. Park, Y. Wang, B. Afshar, A. Gupta, A. Arbabian, S. Gambini, R. Zamani, A. Niknejad, and E. Alon, "A 90nm cmos low-power 60ghz transceiver with integrated baseband circuitry," in *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, feb. 2009, pp. 314–315.
- [12] M. Tabesh, J. Chen, C. Marcu, L. Kong, S. Kang, E. Alon, and A. Niknejad, "A 65nm cmos 4-element sub-34mw/element 60ghz phased-array transceiver," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, february 2011, pp. 166–168.
- [13] S. Nicolson, K. Yau, S. Pruvost, V. Danelon, P. Chevalier, P. Garcia, A. Chantre, B. Sautreuil, and S. Voinigescu, "A low-voltage sigc bicmos 77-ghz automotive radar chipset," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 56, no. 5, pp. 1092–1104, may 2008.
- [14] R. Ben Yishay, R. Carmon, O. Katz, and D. Elad, "A high gain wideband 77ghz sigc power amplifier," in *Radio Frequency Integrated Circuits Symposium (RFIC), 2010 IEEE*, may 2010, pp. 529–532.
- [15] A. Arbabian, B. Afshar, J.-C. Chien, S. Kang, S. Callender, E. Adabi, S. Toso, R. Pilard, D. Gloria, and A. Niknejad, "A 90ghz-carrier 30ghz-bandwidth hybrid switching transmitter with integrated antenna," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 420–421.
- [16] B. Shi, S. Member, and L. Sundström, "A 200-MHz IF BiCMOS Signal Component Separator for Linear LINC Transmitters," *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 7, pp. 987–993, 2000.
- [17] L. Panseri, L. Romano, S. Levantino, C. Samori, and A. Lacaita, "Low-power signal component separator for a 64-qam 802.11 linc transmitter," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 5, pp. 1274–1286, may 2008.
- [18] W. Gerhard and R. Knoechel, "Linc digital component separator for single and multicarrier w-cdma signals," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 53, no. 1, pp. 274–282, january 2005.
- [19] T.-W. Chen, P.-Y. Tsai, D. De Moitie, J.-Y. Yu, and C.-Y. Lee, "A low power all-digital signal component separator for uneven multi-level linc systems," in *ESSCIRC (ESSCIRC), 2011 Proceedings of the*, september 2011, pp. 403–406.

- [20] J. E. Volder, "The cordic trigonometric computing technique," *Electronic Computers, IRE Transactions on*, vol. EC-8, no. 3, pp. 330–334, september 1959.
- [21] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. Krieger Publishing Company, 2006, vol. 1.
- [22] T. Nojima and T. Konno, "Cuber predistortion linearizer for relay equipment in 800 mhz band land mobile telephone system," *Vehicular Technology, IEEE Transactions on*, vol. 34, no. 4, pp. 169–177, nov. 1985.
- [23] J. Cha, J. Yi, J. Kim, and B. Kim, "Optimum design of a predistortion rf power amplifier for multicarrier wcdma applications," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 52, no. 2, pp. 655–663, feb. 2004.
- [24] R. Wilkinson and P. Kenington, "Specification of error amplifiers for use in feedforward transmitters," *Circuits, Devices and Systems, IEE Proceedings G*, vol. 139, no. 4, pp. 477–480, aug 1992.
- [25] J. Cavers, "Adaptation behavior of a feedforward amplifier linearizer," *Vehicular Technology, IEEE Transactions on*, vol. 44, no. 1, pp. 31–40, feb 1995.
- [26] B. Kim, Y. Y. Woo, Y. Yang, J. Yi, J. Nam, and J. Cha, "A new adaptive feedforward amplifier using imperfect signal cancellation," in *Microwave and Millimeter Wave Technology, 2002. Proceedings. ICMMT 2002. 2002 3rd International Conference on*, aug. 2002, pp. 928–931.
- [27] T. Arthanayake and H. Wood, "Linear amplification using envelope feedback," *Electronics Letters*, vol. 7, no. 7, pp. 145–146, 8 1971.
- [28] Y. Akamine, S. Tanaka, M. Kawabe, T. Okazaki, Y. Shima, Y. Masahiko, M. Yamamoto, R. Takano, and Y. Kimura, "A polar loop transmitter with digital interface including a loop-bandwidth calibration system," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, feb. 2007, pp. 348–608.
- [29] J. Dawson and T. Lee, "Cartesian feedback for rf power amplifier linearization," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1, 30 2004-july 2 2004, pp. 361–366 vol.1.
- [30] M. Honarvar, M. Moghaddasi, and A. Eskandari, "Power amplifier linearization using feedforward technique for wide band communication system," in *Radio-Frequency Integration Technology, 2009. RFIT 2009. IEEE International Symposium on*, 9 2009-dec. 11 2009, pp. 72–75.
- [31] D. Cox, "Linear amplification with nonlinear components," *Communications, IEEE Transactions on*, vol. 22, no. 12, pp. 1942–1945, december 1974.

- [32] Y.-C. Chen, K.-Y. Jheng, A.-Y. Wu, H.-W. Tsao, and B. Tzeng, "Multilevel linc system design for wireless transmitters," in *VLSI Design, Automation and Test, 2007. VLSI-DAT 2007. International Symposium on*, april 2007, pp. 1–4.
- [33] S. Chung, P. Godoy, T. Barton, E. Huang, D. Perreault, and J. Dawson, "Asymmetric multilevel outphasing architecture for multi-standard transmitters," in *Radio Frequency Integrated Circuits Symposium, 2009. RFIC 2009. IEEE*, june 2009, pp. 237–240.
- [34] P. Godoy, S. Chung, T. Barton, D. Perreault, and J. Dawson, "A 2.5-ghz asymmetric multilevel outphasing power amplifier in 65-nm cmos," in *Power Amplifiers for Wireless and Radio Applications (PAWR), 2011 IEEE Topical Conference on*, jan. 2011, pp. 57–60.
- [35] J. Hur, O. Lee, K. Kim, K. Lim, and J. Laskar, "Highly efficient uneven multi-level linc transmitter," *Electronics Letters*, vol. 45, no. 16, pp. 837–838, 30 2009.
- [36] P. Eloranta, P. Seppinen, S. Kallioinen, T. Saarela, and A. Parssinen, "A multi-mode transmitter in 0.13 um cmos using direct-digital rf modulator," *Solid-State Circuits, IEEE Journal of*, vol. 42, no. 12, pp. 2774–2784, december 2007.
- [37] T.-W. Chen, P.-Y. Tsai, J.-Y. Yu, and C.-Y. Lee, "A sub-mw all-digital signal component separator with branch mismatch compensation for ofdm linc transmitters," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 11, pp. 2514–2523, november 2011.
- [38] C. Conradi, J. McRory, and R. Johnston, "Low-memory digital signal component separator for linc transmitters," *Electronics Letters*, vol. 37, no. 7, pp. 460–461, march 2001.
- [39] K. Eriksson and D. Esten, *Applied Mathematics: Body and Soul: Volume 2: Integrals and Geometry in \mathbb{R}^n* . Springer, 2010, vol. 2.
- [40] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of cordic: Algorithms, architectures, and applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 9, pp. 1893–1907, september 2009.
- [41] A. Strollo, D. De Caro, and N. Petra, "Elementary functions hardware implementation using constrained piecewise-polynomial approximations," *Computers, IEEE Transactions on*, vol. 60, no. 3, pp. 418–432, march 2011.
- [42] S. Nasayama, T. Sasao, and J. Butler, "Programmable numerical function generators based on quadratic approximation: architecture and synthesis method," in *Design Automation, 2006. Asia and South Pacific Conference on*, jan. 2006, p. 6 pp.

- [43] J. Cao, B. Wei, and J. Cheng, "High-performance architectures for elementary function generation," in *Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium on*, 2001, pp. 136–144.
- [44] T. Sasao, S. Nagayama, and J. Butler, "Numerical function generators using lut cascades," *Computers, IEEE Transactions on*, vol. 56, no. 6, pp. 826–838, june 2007.
- [45] B. Shi and L. Sundstrom, "An if cmos signal component separator chip for line transmitters," in *Custom Integrated Circuits, 2001, IEEE Conference on.*, 2001, pp. 49–52.
- [46] A. Panigada and I. Galton, "A 130 mw 100 ms/s pipelined adc with 69 db snr enabled by digital harmonic distortion correction," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 12, pp. 3314–3328, dec. 2009.
- [47] J. Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *Vehicular Technology, IEEE Transactions on*, vol. 39, no. 4, pp. 374–382, nov 1990.
- [48] A. D'Andrea, V. Lottici, and R. Reggiannini, "Nonlinear predistortion of ofdm signals over frequency-selective fading channels," *Communications, IEEE Transactions on*, vol. 49, no. 5, pp. 837–843, may 2001.
- [49] D. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of rf power amplifiers," *Signal Processing, IEEE Transactions on*, vol. 54, no. 10, pp. 3852–3860, oct. 2006.
- [50] A. Zhu, P. Draxler, C. Hsia, T. Brazil, D. Kimball, and P. Asbeck, "Digital predistortion for envelope-tracking power amplifiers using decomposed piecewise volterra series," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 56, no. 10, pp. 2237–2247, oct. 2008.
- [51] C. Yu, L. Guan, E. Zhu, and A. Zhu, "Band-limited volterra series-based digital predistortion for wideband rf power amplifiers," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 60, no. 12, pp. 4198–4208, dec. 2012.
- [52] S. Hong, Y. Y. Woo, J. Kim, J. Cha, I. Kim, J. Moon, J. Yi, and B. Kim, "Weighted polynomial digital predistortion for low memory effect doherty power amplifier," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 55, no. 5, pp. 925–931, may 2007.
- [53] L. Aguirre, M. C. S. Coelho, and M. Correa, "On the interpretation and practice of dynamical differences between hammerstein and wiener models," *Control Theory and Applications, IEE Proceedings -*, vol. 152, no. 4, pp. 349–356, July.
- [54] W. Tai, "Efficient watt-level power amplifiers in deeply scaled cmos," Ph.D. dissertation, Carnegie Mellon University, 2011.

- [55] A. Wei, M. Sherony, and D. Antoniadis, "Effect of floating-body charge on soimofet design," *Electron Devices, IEEE Transactions on*, vol. 45, no. 2, pp. 430–438, feb 1998.
- [56] M. del Mar Hershenson, "Design of pipeline analog-to-digital converters via geometric programming," in *Computer Aided Design, 2002. ICCAD 2002. IEEE/ACM International Conference on*, nov. 2002, pp. 317 – 324.
- [57] Y. Li and V. Stojanovic, "Yield-driven iterative robust circuit optimization algorithm," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, july 2009, pp. 599 –604.
- [58] B. N. Bond, Z. Mahmood, Y. Li, R. Sredojevi, A. Megretski, V. Stojanovi, Y. Avniel, and L. Daniel, "Compact Modeling of Nonlinear Analog Circuits Using System Identification Via Semidefinite Programming and Incremental Stability Certification," *Computer-Aided Design*, vol. 29, no. 8, pp. 1149–1162, 2010.
- [59] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of mos transistors," *IEEE JSSC*, vol. 24, no. 5, pp. 1433–1440, 1989.
- [60] S. Nassif, "Modeling and analysis of manufacturing variations," *Custom Integrated Circuits, 2001, IEEE Conference on.*, pp. 223–228, 2001.
- [61] X. Li, J. Le, P. Gopalakrishnan, and L. T. Pileggi, "Asymptotic probability extraction for nonnormal performance distributions," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 16–37, Jan. 2007.
- [62] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univeristy Press, 2004.
- [63] R. Kanj, R. Joshi, and S. Nassif, "Mixture importance sampling and its application to the analysis of sram designs in the presence of rare failure events," in *DAC '06: Proceedings of the 43rd annual conference on Design automation*. New York, NY, USA: ACM, 2006, pp. 69–72.
- [64] J. Kim, K. D. Jones, and M. A. Horowitz, "Fast, non-monte-carlo estimation of transient performance variation due to device mismatch," in *DAC '07: Proceedings of the 44th annual conference on Design automation*. New York, NY, USA: ACM, 2007, pp. 440–443.