Digitally-Assisted Design, Simulation and Testing Techniques for Optimization of Analog/RF Integrated Circuits

A Dissertation Presented

by

HARI CHAUHAN

To

The Department of ELECTRICAL AND COMPUTER ENGINEERING

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the field of

ELECTRICAL ENGINEERING

Northeastern University
Boston, Massachusetts
April 2016
Abstract

High-performance low-cost radio frequency (RF) transceivers are essential in today’s wireless systems. Contemporary manufacturing process technologies and device scaling have helped the integration of analog and RF circuits with high performance, but often with high sensitivity to increasing process variations when chips are fabricated in high volumes. This trend has motivated designers to incorporate more digital circuits for performance optimizations of analog/RF circuits. A repercussion of adding on-chip circuits is the rise of design and verification complexity of RF devices, which is paralleled by demands for shorter design cycles and better reliability. These challenges create the need for improved simulation and testing methodologies.

This dissertation focuses on the design and integration of digital circuits with analog/RF circuits for performance optimizations. Spectral analysis for the evaluation of analog/RF circuits is a standard procedure for which the fast Fourier transform (FFT) algorithm is widely used. However, the majority of existing FFT implementations on chips consume excessive area and power for built-in testing applications. In this research, an FFT-based performance monitoring technique with multi-tone test signals has been created for efficient on-chip spectral analysis of analog/RF circuits. This method enables to estimate third-order intermodulation components of up to 50 dB below the fundamental tones with an accuracy of ±1.5 dB based on the output spectrum of analog circuits. The capability of this technique to accurately determine the power of two test tones as well as their distortion components and intermodulation products was demonstrated by designing an on-chip linearity calibration scheme for a tunable low-noise amplifier.

A key aspect of practical circuit and system design is to ensure high performance with high reliability and low cost. Therefore, it is advantageous to utilize test and simulation methods for simultaneous optimization of a design under test and a performance enhancement technique (e.g., self-calibration circuits and linearization methods) prior to fabrication of chips or systems. In this research, a simulation approach to...
concurrently design and optimize an entire system at a desired abstraction level was developed for integrated amplifiers. The design and optimization of a 10 W inverted Doherty power amplifier (PA) with a digital predistortion (DPD) linearization technique was completed to exemplify the effectiveness of the simulation platform. In addition, an integrated hardware-software co-design approach that allows DPD tuning to meet specification requirements with minimum resources was developed together with industrial collaborators. Tuning of the DPD technique was performed with several off-the-shelf power amplifiers for performance optimization of the integrated PA-DPD system. Furthermore, a tuning method was developed that incorporates a digital-to-analog converter in the integrated PA-DPD system. With this approach, the error signal generated by the digital predistortion technique can be utilized to automatically tune a bias voltage in the power amplifier for optimal performance. Simulation results from a closed-loop system consisting of an inverted Doherty power amplifier with digital predistortion were evaluated to validate the tuning mechanism.
Acknowledgements

First and foremost, I would like to express my deepest appreciation and thanks to my dissertation advisor, Prof. Marvin Onabajo, for his excellent guidance and tremendous support throughout these past years without which this dissertation would not have been possible. I would also like to thank my committee members, Prof. Yong-Bin Kim and Prof. Matteo Rinaldi, for their guidance through the final stages of my Ph.D. degree completion; especially Prof. Kim who made chip fabrication possible with his support and mentorship. I also thank my colleagues Chun-hsiang, In-Seok Jung, and Yongsuk Choi for working together on an RF built-in testing project. In addition, a thank you to all the members of Analog and Mixed-Signal Integrated Circuits (AMSIC) and High Performance VLSI (HPVLSI) research labs for the help during the research projects.

I would like to thank Vlad Kvartenko, Martin McCormick, Theophane Weber (previously at Analog Devices Lyric Labs and now with Google DeepMind), Robin Coxe (previously at Analog Devices Lyric Labs and now with Ettus Research), and Brian Donnelly for their thorough guidance and support during the power amplifier linearization project. Without their guidance, hard work and the sponsorship of the project from Analog Devices, I could not have developed the simulation and verification platform for co-optimization of power amplifiers with digital predistortion.

Finally, to my family (Sh. Dhir Singh, Smt. Prabha Devi, Paridhi, Harsh, Dunesh, Dev, Sonali, Soma, Divya, Yash, Parih, Nattu and Dhairesh) and to my wife Arti, thank you for encouragement and support, without you I could not have finished my Ph.D. study.
# Table of Contents

1. Introduction .................................................................................................................. 13

1.1 Overview of digitally-assisted analog/RF design ....................................................... 13

1.2 Spectral analysis ........................................................................................................... 15

1.3 Simulation and verification techniques ......................................................................... 17

1.4 Contributions of this research ..................................................................................... 18

1.4.1 On-chip spectral analysis technique ........................................................................ 18

1.4.2 Digitally-assisted analog/RF design ....................................................................... 19

1.4.3 Simulation and verification techniques for analog/RF designs .............................. 20

1.5 Dissertation structure ................................................................................................. 20

2. On-Chip Spectral Analysis Technique .......................................................................... 21

2.1 Single-tone testing ....................................................................................................... 23

2.2 Two-tone testing ........................................................................................................... 23

2.3 FFT for built-in testing applications ............................................................................. 24

2.4 Conventional coherent sampling ................................................................................ 25

2.5 Proposed spectral analysis technique .......................................................................... 26

2.5.1 Single-tone testing ................................................................................................... 26

2.5.2 Multi-tone testing .................................................................................................... 27

2.6 Matlab simulation results ........................................................................................... 28

2.6.1 Single-tone with high-order harmonics .................................................................. 28

2.6.2 Two-tone signal ....................................................................................................... 31

2.6.3 ADC resolution requirement .................................................................................... 33

2.7 Selection of NFFT and the ADC resolution .................................................................. 38

2.8 Implementation example ............................................................................................. 39

3. Digitally-Assisted Analog/RF Design .......................................................................... 43

3.1 On-chip amplifier linearity calibration with the fast Fourier transform ................. 44

3.1.1 LNA with tuning functionality ................................................................................ 46
3.1.2 Calibration scheme ................................................................. 48
3.1.3 Simulation results ................................................................. 52

3.2 On-chip amplifier calibration using envelope response analysis with a two-tone signal ......................................................... 54

3.2.1 Proposed calibration approach ............................................... 57
3.2.2 Digital calibration control ...................................................... 59
3.2.3 Envelope detector ................................................................. 59
3.2.4 Low-frequency circuits under test ......................................... 60
3.2.5 High-frequency circuits under test ....................................... 61
3.2.6 Simulation results ................................................................. 66

4. Simulation and Verification Techniques ........................................ 70

4.1 An optimization platform for digital predistortion of power amplifiers .... 70

4.1.1 Digital predistortion (DPD) technique .................................. 71
4.1.2 Performance and computational complexity trade-offs .............. 73
4.1.3 Hardware-software test platform ........................................ 75
4.1.4 DPD optimization steps ....................................................... 77
4.1.5 Experimental results .......................................................... 78
4.1.6 FPGA implementation results ............................................. 82

4.2 A simulation method for design and optimization of RF power amplifiers with digital predistortion .................................................. 84

4.2.1 Inverted Doherty power amplifier design ............................... 86
4.2.2 Simulation method ............................................................. 88
4.2.3 Simulation results .............................................................. 92
4.2.4 Case study ........................................................................ 93

4.3 A tuning technique for temperature and process variation compensation of power amplifiers with digital predistortion ........................................... 95

4.3.1 Introduction ........................................................................ 95
4.3.2 Digital predistortion overview ............................................ 97
4.3.3 Tuning approach........................................................................................................ 99
4.3.4 Inverted Doherty power amplifier........................................................................ 100
4.3.5 Digital-to-analog converter (DAC)........................................................................ 103
4.3.6 Tuning algorithm.................................................................................................... 105
4.3.7 Performance tuning simulation method ............................................................... 107
4.3.8 Simulation results.................................................................................................. 111

5. Conclusion and Future Work..................................................................................... 113

6. References................................................................................................................ 115
List of Figures

Figure 1. Simplified diagram of an RF transceiver. .......................................................... 13
Figure 2. RF system with tuning features for performance enhancement. ...................... 14
Figure 3. RF power amplifier linearization with a digital predistortion technique. ....... 14
Figure 4. Two-tone test spectra. ....................................................................................... 15
Figure 5. On-chip built-in calibration (BIC) with spectral testing. ................................. 22
Figure 6. Single-tone test spectra..................................................................................... 23
Figure 7. Two-tone test spectra. ....................................................................................... 24
Figure 8. FFT with conventional coherent sampling ($NFFT = 16$): spectrum of the input signal $x(t)$ with a fundamental frequency component of $12.5 \text{ MHz}$ and four harmonics. 29
Figure 9. FFT with the proposed method ($NFFT = 16$): spectrum of the input signal $x(t)$ with a fundamental frequency component of $10 \text{ MHz}$ and four harmonics. ......... 30
Figure 10. Spectrum of the input signal $y(t)$ with $f_1 = 12.5 \text{ MHz}$ (coherent) and $f_2 = 13.5 \text{ MHz}$ (not coherent). ......................................................................................... 32
Figure 11. Spectrum of input signal $z(t)$ with two test tones $f_1 = 3 \text{ MHz}$, $f_2 = 5 \text{ MHz}$, and their third-order inter-modulation (IM3) products at $1 \text{ MHz}$ and $7 \text{ MHz}$, calculated using a 16-point FFT. ......................................................................................... 33
Figure 12. IM3 vs. FFT length ($NFFT$) and ADC resolution ($n$-bits) from a two-tone test with $f_1 = 3 \text{ MHz}$ and $f_2 = 5 \text{ MHz}$. ......................................................................................... 35
Figure 13. Radix-2 16-point FFT processor (bold lines: buses with multiple bits)........ 40
Figure 14. Layout of the FFT engine (0.073 mm$^2$ in 45 nm CMOS technology)......... 40
Figure 15. Output spectrum from post-layout simulation, showing the two test tones ($f_1 = 3 \text{ MHz}$, $f_2 = 5 \text{ MHz}$) and their IM3 products at $1 \text{ MHz}$ and $7 \text{ MHz}$. ................................. 41
Figure 16. Conceptual block diagram of digitally-assisted RF transmitter. ..................... 43
Figure 17. Block diagram of the calibration scheme............................................................ 45
Figure 18. Low-noise amplifier (LNA) with linearity tuning capability [76]. ................. 46
Figure 19. Spectral components of interest during a standard two-tone test. ................. 48
Figure 20. Digital calibration control flow. ........................................................................ 50
Figure 21. Block diagram of proposed calibration scheme. ............................................. 56
Figure 22. Estimated magnitudes superimposed on the actual magnitudes of 10,000 randomly generated sample values. ............................................................................. 58
Figure 23. Magnitude approximation error (in dB) for the example case with α = 1 and β = 1/4. .......................................................... 59
Figure 24. Envelope detector in standard 0.13 μm CMOS technology [2]. .......... 60
Figure 25. At low operating frequencies the magnitude of the IM3 components can be
directly extracted from the response of the CUT using an ADC and FFT engine. .... 60
Figure 26. Estimation of high-frequency IM3 components from the low-frequency
envelope response of a two-tone signal ........................................ 61
Figure 27. (a) Conceptual DPD diagram, (b) system response, (c) indirect learning DPD
system architecture, (d) direct learning DPD system architecture; where w, x and y
represent the original input signal, predistorted signal and linearized PA output
respectively. .............................................................................. 71
Figure 28. Hardware-software test setup. .................................................. 76
Figure 29. |ACLR| vs. LUT size and LUT fraction bit-width: (a) PA#1, (b) PA#2....... 79
Figure 30. Measured ACLR vs. LUT complexity. ........................................ 80
Figure 31. Measured ACLR vs. pruning factor: (a) PA#1, (b) PA#2................... 81
Figure 32. Measured ACLR using the optimized FPGA-based DPD system with LUT
complexity of ‘0.25·P’, with ‘N/2’ number of coefficients and with a 1c x 20 MHz LTE
test signal for the GaAs HBT PA. ...................................................... 83
Figure 33. Measured ACLR using the optimized FPGA-based DPD system with LUT
complexity of ‘0.25·P’, with ‘N/2’ number of coefficients and with a 1c x 20 MHz LTE
test signal for the GaN SiC Doherty PA. ............................................ 84
Figure 34. Co-simulation setup for optimization of a PA with DPD. ................. 86
Figure 35 Inverted Doherty power amplifier (IDPA). .................................... 87
Figure 36. Simulated characteristics of the Inverted Doherty PA (IDPA). .......... 88
Figure 37. Simulation setup in ADS.......................................................... 89
Figure 38. Up-conversion and down-conversion (implemented with Matlab code)..... 92
Figure 39. Example waveforms after time alignment of the scaled baseband input signal
with the down-converted and scaled baseband output signal.......................... 92
Figure 40. RF signals at the input and the output of the PA captured in Matlab after a
transient simulation with ADS......................................................... 93
Figure 41. Spectrum of the PA output with and without application of the DPD. .... 94
Figure 42. Simulated power spectrum at the output of the PA................................. 95
Figure 43. Characteristics of a power amplifier: (a) under normal operating conditions; (b) under PVT variation showing drift of the 1-dB compression point into the deep saturation region for the same input power $P_{IN}$................................................................. 96
Figure 44. Measured ACLR of an off-the-shelf PA with different bias conditions. ...... 97
Figure 45. Conventional PA-DPD system architecture............................................ 98
Figure 46. Conceptual representation of digital predistortion techniques [1].......... 98
Figure 47. Block diagram of the proposed tuning mechanism to compensate for the effects of PVT variations. ......................................................................................... 99
Figure 48. Inverted Doherty power amplifier (IDPA).............................................. 101
Figure 49. Simulated characteristics of the standalone Inverted Doherty PA (IDPA) in the absence of the tuning algorithm and DAC......................................................... 102
Figure 50. 7-bit R-2R digital-to-analog converter (DAC)........................................ 105
Figure 51. Flow chart of the tuning algorithm.......................................................... 106
Figure 52. PA performance calibration simulation setup in ADS............................ 108
Figure 53. Simulated power spectrum at the output of the IDPA with and without the proposed tuning technique: (a) IDPA operating in the near-saturation region with $Q = 15$, and $T = 25^\circ C$, (b) IDPA operating in the saturation region with $Q = 15$, and $T = 50^\circ C$, and (c) IDPA operating in the deep saturation region with $Q = 13$, and $T = 25^\circ C$ ...................................................................................................................................... 112
List of Tables

Table 1. Theoretical and simulated IM3 (in dBc) from a two-tone test with $f_1 = 3$ MHz, $f_2 = 5$ MHz, $f_{IM3} = 1$ MHz .......................................................................................................................... 36
Table 2. Theoretical and simulated HD3 (in dBc) from a single-tone test with $f_0 = 3$ MHz .................................................................................................................................................. 36
Table 3. Simulated IM3 (in dBc) with DNL error (from a two-tone test with $f_1 = 3$ MHz, $f_2 = 5$ MHz, $f_{IM3} = 1$ MHz, $NFFT = 2^4$ ) ............................................................................................................ 36
Table 4. Simulated IM3 (in dBc) with wideband noise (from a two-tone test with $f_1 = 3$ MHz, $f_2 = 5$ MHz, $f_{IM3} = 1$ MHz, $NFFT = 2^4$ ) ............................................................................................................ 37
Table 5. Simulated IM3 (in dBc) with wideband noise and ±0.5 LSB DNL error (from a two-tone test with $f_1 = 3$ MHz, $f_2 = 5$ MHz, $f_{IM3} = 1$ MHz, $NFFT = 2^4$ ) ........................................... 37
Table 6. Results from a two-tone test: calculated FFT of the input signal (65536 points) vs. output of the 16-Point FFT engine .................................................................................................................. 42
Table 7. Performance summary for the linearized LNA .......................................................................................................................................................................................... 47
Table 8. Simulated specifications of the linearized LNA for different codes that control the $C_{GD2,EXT}$ value ........................................................................................................................................... 47
Table 9. Verification of the calibration algorithm through simulation with a two-tone test ($f_1 = 3$ MHz, $f_2 = 4$ MHz) using a 10-bit ADC and 16-point FFT ..................................................................................... 54
Table 10. Simulated IM3 (in dBc) from a two-tone test with $f_1 = 1$ GHz, $f_2 = 1.001$ GHz using a 16-bit ADC and $2^{16}$-point FFT ........................................................................................................ 63
Table 11. Simulated IM3 (in dBc) from a two-tone test with $f_1 = 1$ GHz, $f_2 = 1.001$ GHz using a 10-bit ADC and FFT lengths varying from $2^4$ to $2^{16}$ points ............................................................................. 65
Table 12. Simulation results for each iterations from a two-tone test with $f_1 = 3$ MHz, $f_2 = 4$ MHz using a 10-bit ADC and a 16-point FFT .................................................................................................. 67
Table 13. Simulation results for each iteration from a two-tone test with $f_1 = 2.4$ GHz and $f_2 = 2.042$ GHz using a 10-bit ADC and 64-point FFT .................................................................................... 68
Table 14. Power amplifier specifications, test signals and measured ACLR ................................................................................................................................................. 78
Table 15. Performance comparison of Matlab-based DPD results and measurements of the FPGA-based DPD implementation ........................................................................................................... 82
Table 16. Potential reconfigurable elements of an RF transmitter ................................................................................................................................................ 94
Table 17 Simulated output 1-dB compression point with varying quality factor (Q) and temperature (T) ........................................................................................................................................................................ 102
Table 18. Simulated ACLR and Normalized Mean Square Error (NMSE) of the IDPA with a 2c x 20 MHz test signal resulting in 36 dBm output power .................................................. 103
Table 19. Comparison of the simulated characteristics of the standalone IDPA and the IDPA with integrated DAC operating with different bias voltages for the peaking amplifier ........................................................................................................................................................................ 104
1. Introduction

High performance low-cost radio frequency (RF) transceivers provide wireless connectivity in an increasing number of devices and systems. Modern manufacturing process technologies and device scaling eased the integration of a large number of analog and RF circuits with high performance, but a common tradeoff is sensitivity to process variations that are trending to increase. As a consequence, many emerging design methods employ on-chip digital circuits for performance optimizations of analog/RF circuits. A repercussion of adding circuits is the steep rise of design and verification complexity, especially for RF blocks within systems. In addition, there is an increasing need to design reliable circuits in shorter time. Hence, efficient simulation and verification methodologies are desired, such as the optimization platform for power amplifiers (PAs) with digital predistortion (DPD) in [1] for example.

1.1 Overview of digitally-assisted analog/RF design

Digitally-assisted analog/RF design [2]-[5] and integrated transceiver calibration [6]-[8] approaches are gaining popularity to ensure efficient and reliable mixed-signal systems in nanoscale complementary metal-oxide-semiconductor (CMOS) technologies. An RF transceiver typically consists of amplifiers [e.g., low-noise amplifier (LNA), power amplifier (PA)], mixers, filters [e.g., bandpass filter (BPF)] and frequency synthesizers [or a local oscillator (LO)] as depicted in Fig. 1. RF systems are often subjected to different operating conditions and should deliver optimal performance under varying scenarios and requirements. For example, when the input signal strength is low, it is
beneficial to adjust the bias voltage of a power amplifier to achieve better efficiency. To meet stringent requirements it is desirable to intelligently tune different elements of a system. Additional analog/RF circuits can be incorporated to extract parameters such that a digital signal processor (DSP) can tune a device under test (DUT) for optimum performance. For example, on-chip amplitude or power detectors can be employed to evaluate gains or large-signal linearity characteristics, and more complex systems can be designed to characterize frequency responses on the chip [9]. Fig. 2 shows a conceptual block diagram that generalizes RF systems with built-in tuning knobs to adjust certain elements such as the input matching network (IMN), output matching network (OMN) or power supply for optimum performance.

An example of a digitally-assisted RF system is the linearization of RF power ampli-
fiers using digital predistortion (DPD) techniques [10]-[12]. As visualized in Fig. 3, DPD algorithms typically use an inverse model of the PA to predistort the input signal based on estimated PA linearity coefficients. With such a scheme, the linearity improvement depends on the cancellation of nonlinear signal components at the PA, which depends on the accuracy by which the estimated linearity coefficients used in the inverse model correlates with the actual linearity coefficients of the PA. However, the performance requirements vary depending on the application, which can result in a significant cost overhead for an integrated DPD [13]-[15]. Variations of devices under test (e.g., power amplifiers) place additional limitations on the performance of the DPD. This has led to the development of design, simulation and verification techniques to optimize the performance of the overall all system. In this dissertation research, an emphasis has been placed on the development of simulation approaches that provide new capabilities to designers for customization and optimization of integrated mixed-signal systems with the goal to circumvent hardware-based testing during each design iteration.

1.2 Spectral analysis

Many performance characteristics can be observed from the output spectrum of a circuit under test (CUT) or a chain of analog blocks [2]-[3], [16]. One way to characterize a nonlinear circuit is by applying a test tone at the input and assessing the harmonic distortion (HD) magnitudes in the spectrum at multiples of the input frequency. This approach is feasible at low frequencies but less practical at radio frequencies because of the difficulties associated with measuring high-frequency harmonics. Furthermore, these harmonics typically fall outside the bandwidth of narrow-band RF amplifiers and other receiver blocks. Alternatively as shown in Fig. 4, the characterization of the third-order

![Figure 4. Two-tone test spectra.](image-url)
linearity can be conducted by applying two closely-spaced input tones \((f_1, f_2)\) and measuring the third-order intermodulation product at \((2f_2 - f_1)\) or \((2f_1 - f_2)\), which can be nearby frequencies within the CUT’s passband \([2]\).

The fast Fourier transform (FFT) is a well-known technique used to determine the spectrum, however the FFT algorithm calculates the spectrum of an input waveform only for certain discrete frequencies known as FFT bins which are separated by the fundamental frequency of the FFT. To obtain an accurate measurement for a frequency component of the input waveform, its frequency has to be an exact integer multiple of the FFT fundamental frequency. Even a slight variation of the input frequency components will result in spectral leakage \([17]-[18]\). An additional problem that may result from spectral leakage is that the spectral leakage from a large signal component may mask the weak signal components at the different frequencies. Therefore, the simple FFT algorithm is most useful for single-tone testing when the input waveform contains a single frequency component and its higher-order harmonics. On the other hand, spectral leakage is almost unavoidable in multi-tone testing cases, where one or more of the input tones are not integer multiple of the FFT fundamental frequency. The FFT fundamental frequency \((\Delta f)\) strictly depends on the sampling frequency \((f_{\text{samp}})\) and the length of the FFT \((N_{\text{FFT}})\) as follows: \(\Delta f = \frac{f_{\text{samp}}}{N_{\text{FFT}}}\). Thus, to achieve high frequency resolution, either the sampling frequency has to be decreased or the length of FFT has to be increased \([19]-[22]\). Since the sampling frequency should meet the Nyquist criterion, the sampling frequency condition can typically not be relaxed when the goal is to perform on-chip analysis of the highest possible signal frequency under the constraints of a given CMOS technology. In contrast, increasing the length of the FFT to achieve better frequency resolution results in an additional area overhead and thereby reduces the feasibility of an FFT implantation for on-chip testing applications. Another useful approach to alleviate the leakage effect is to choose a suitable window function that minimizes the energy spreading \([23]-[26]\). However, the use of a windowing technique introduces various other flaws into the spectrum such as frequency/amplitude imprecision, broadening of main lobe width, and creation of side lobes. Moreover, windowing operations require additional computational resources. Various on-chip FFT implementations have been
developed for communication systems such as those using orthogonal frequency-division multiplexing (OFDM), which requires a combination of high throughput rate and accuracy [23]-[24]. These requirements typically translate into a large number of points (e.g., 64-2048), resulting in area and power requirements of $>1 \text{ mm}^2$ and $>20 \text{ mW}$, respectively. It is preferable to have an accurate spectral analysis technique that requires a smaller number of points to make the FFT realization more suitable for on-chip built-in testing and calibration applications requiring area and power efficiency.

1.3 Simulation and verification techniques

Historically, RF circuits were limited to a set of discrete components, which were relatively easy to simulate and verify with traditional bottom-up and top-down methodologies. In the bottom-up approach, individual blocks are designed and verified at the transistor level, and afterwards the blocks are combined in hierarchical manner to form a complete system. This approach has limitations because it makes verification of a complete system a very complex and tedious task, requiring long simulation times. In addition, system-level verification is not possible until the integration of the complete system, and it requires significantly more redesign efforts if an error is detected at the top level. In contrast to the bottom-up approach, in a top-down approach the top-level system design and simulations are carried out first. Hardware description languages (HDLs) such as Verilog-A or VHDL can be utilized to model individual blocks and their interfaces [27]-[28]. After system-level optimizations, the modeled blocks serve as specification targets during the transistor-level design phase. Limited accuracies of the models are a major drawback of the top-down approach. Moreover, the incorporation of extensive measurement circuits and digital logic for adaptive performance enhancements in the presence of rising process variations further add complexity to integrated RF systems [29], which cause more verification and testing challenges as well as require additional resources ranging from simulation time to test equipment hardware. For example, for the integrated power amplifier with digital predistortion (PA-DPD) system, the analog and mixed-signal simulation in Cadence with a transistor-level PA design and a Verilog model of the DPD algorithm can potentially result in excessive simulation times because the simulation time steps has to be small enough to run accurate transient
simulations with PA input/output signals in the gigahertz range, but over the complete duration to execute the DPD algorithm. Furthermore, assessments with modulated signals defined in a given communication standard should be performed to thoroughly verify an RF system based on performance parameters such as the bit error rate (BER) and error vector magnitude (EVM) [30]. Thus, to shorten simulation times, design platforms that allow combined simulations at different abstraction levels (from the transistor level to the algorithm level [31]) are more appealing. Another promising methodology is hardware-software co-design, where a part of the transceiver chain is implemented with the actual hardware components and another part with a software model. In general, it is advantageous to devise universal adaptive design-and-test platforms that allow evaluation and optimization of system performance for practical test conditions and that give flexibility to incorporate additional test constraints during the design phase [1], [32]-[35].

1.4 Contributions of this research

1.4.1 On-chip spectral analysis technique

An accurate FFT-based analysis approach was created for on-chip spectral characterization of multi-tone signals. The approach has been derived from the conventional coherent sampling method. It was demonstrated that it allows the designer to select the appropriate test signal frequencies, analog-to-digital converter (ADC) resolution, and FFT length to achieve the desired frequency resolution in the output spectrum without spectral leakage. The method avoids the use of a large number of FFT points to minimize the required on-chip FFT resources for area and power efficient built-in testing applications. Post-layout simulation results of an FFT implementation in standard 45 nm CMOS technology have shown the feasibility of the approach. For a 16 MHz 16-point FFT computation, the example FFT engine consumes an estimated power of 6.47 mW with 1.1 V supply and occupies an area of 0.073 mm². In addition, a strategy was developed to determine the suitable ADC resolution and the FFT length to obtain a required accuracy. For example, when combined with a 10-bit ADC, the simulated error for third-order intermodulation (IM3) extraction from the output spectrum of the 16-point FFT is within 1.5 dB for IM3 components ≤ 50 dBc [16].


1.4.2 Digitally-assisted analog/RF design

1.4.2.1 On-chip amplifier linearity calibration with an FFT engine

An on-chip linearity calibration scheme based on third-order intermodulation (IM3) testing with a two-tone test signal was formulated. A mathematical model of the proposed calibration scheme was implemented in Matlab to assess the accuracy of the methodology with simulations. The results indicate that IM3 components from 30 dBc to 60 dBc can be extracted with an error of less than 0.05 dB depending on the amount of available on-chip digital signal processing resources. With introduction of limited ADC resolution (10-bit), the IM3 estimation approach still permits evaluation with up to 1.5 dB error. An application of the developed calibration scheme was emulated by performing a Matlab simulation with a tunable low-noise amplifier (LNA) model [36].

1.4.2.2 A built-in calibration system to optimize third-order intermodulation performance of RF amplifiers

A built-in calibration system for performance tuning of RF amplifiers has been designed in collaboration with other researchers from the High Performance VLSI Design (HPVLSI) and Analog & Mixed-Signal Integrated Circuit (AMSIC) groups at Northeastern University. The approach allows on-chip calibration for linearity optimization by estimating the IM3 during two-tone testing. To minimize the area and power overhead of the system, an envelope detector was employed to reduce the speed of operation for the ADC and digital blocks. Furthermore, an efficient IM3 estimation method has been developed. Closed-loop simulations of this calibration system with a low-noise amplifier having digitally-tunable third-order linearity performance showed the feasibility of the synthesized digital blocks in standard CMOS technology [2].

1.4.2.3 Tuning scheme to compensate for temperature and process variations of power amplifiers with digital predistortion

The linearity of PAs is frequency-dependent and can also vary over time due to temperature and other environmental effects. A novel adaptive method to enhance PA linearity and gain parameters through digital control based on the DPD algorithm’s state after each iteration is introduced in this dissertation. This adaptability can help to reduce DPD convergence times and power consumption during periodic calibrations (e.g., after
carrier frequency changes) by enhancing the intrinsic linearity characteristics of the PA for cases in which the results of the first iteration of the DPD algorithm indicate significant PA deficiencies. A simulation platform in Agilent’s Advanced Design System (ADS) design environment with embedded Matlab was constructed to evaluate digitally-assisted amplifier design techniques [37].

1.4.3 Simulation and verification techniques for analog/RF designs

As part of a collaboration with Analog Devices, Inc. (Lyric Labs), an approach to optimize DPD algorithms for enhanced PA performance was demonstrated with a purpose-built integrated hardware-software test platform. The platform enables a comparison of different DPD implementations in order to meet target specifications with minimal computational resources. This optimization process was demonstrated via measurements of commercially available PAs using the Analog Devices DPD solution [1]. An associated simulation method for design and optimization of RF PAs with the DPD was established. The method was demonstrated through simulations of an inverted Doherty PA with an integrated DPD algorithm that was implemented with embedded Matlab in ADS. The simulation method enables designers to jointly optimize PAs and DPDs during design phases, such that costly hardware setups are circumvented during design iterations and are only required in final characterization test phases [31].

1.5 Dissertation structure

The organization of this thesis is as follows. A spectral analysis technique based on an area-efficient on-chip FFT engine for built-in testing and calibration is presented in Chapter 2. Built-in-calibration schemes for RF amplifiers based on the developed FFT engine are discussed in Chapter 3. A simulation and verification technique for design and optimization of analog/RF designs with emphasis on optimization of power amplifiers having digital predistortion is described in Chapter 4. Conclusion and suggestions for future work are provided in Chapter 5.
2. On-Chip Spectral Analysis Technique

Digitally-assisted analog design and integrated transceiver calibration [38]-[40] approaches are gaining popularity to ensure efficient and reliable mixed-signal systems in nanoscale CMOS technologies. One design aspect is to equip analog blocks with performance tuning features that allow the recovery from process variations and faults. Examples of such tuning mechanisms include input impedance matching, gain and center frequency tuning for low-noise amplifiers [40]-[42], second-order nonlinearity and mismatch correction for mixers [43]-[45], as well as linearity enhancements for baseband filters [46]. Another aspect related to digitally-assisted design is the extraction of performance metrics on the chip to enable one-time or periodic calibrations. Many performance characteristics can be observed based on the output spectrum of a circuit under test (CUT) or a chain of analog blocks, which has led to on-chip spectrum analyzers that emulate conventional off-chip instrumentation [47]-[48]. Alternatively, calibration methods have been proposed that incorporate existing or dedicated analog-to-digital converter (ADC) and digital signal processing resources to directly quantize the output signals of analog circuits for computation of the Fast Fourier Transform (FFT) and automatic tuning with digital-to-analog converters (DACs) [49]-[50]. Extraction of circuit linearity parameters with the latter approach calls for efficient FFT implementations, which is the focus of the research described in this chapter. The presented method leverages that the frequencies for two-tone tests can be selected by the designer of the built-in test (BIT) or built-in calibration (BIC) scheme to circumvent inaccuracies due to spectral leakage while using a small number of FFT points. This capability to accurately measure the power of the tones as well as their distortion and intermodulation products with an efficient on-chip FFT can also find application in loopback testing techniques with spectral estimation such as in [51]-[52]. Fig. 5 visualizes the BIC approach that is the target application for the presented FFT realization. The calibration method could be applied to an individual analog block or a cascade of blocks. In practice, the maximum test signal frequency depends on the highest possible ADC sampling frequency and clock frequency for the FFT computation. Thus, a down-conversion within the chain of blocks under calibration is assumed if it involves RF circuits. In such a case, an effective calibration might require sequential injection of test signals at various locations along
the analog signal chain through the use of switches as in [53]. Alternatively, it has been shown that envelope detectors can be placed at the output of an RF circuit to extract the signal characteristic at lower frequencies. For example it is outlined in [54] how an input signal with two test tones ($f_1, f_2$) allows to determine third-order linearity characteristics of the CUT by monitoring the spectral components at the frequencies up to $3\cdot(f_2 - f_1)$, which can be assured to be low by selecting appropriate test tone frequencies with narrow spacing. This permits the use of lower sampling frequencies. The output signal is quantized by the on-chip ADC in Fig. 5 prior to the FFT computation, which may be implemented with existing system resources or extra components. An adaptive biasing block (that typically includes DACs) enables the digitally-assisted performance tuning for the analog circuit(s) based on the digital extraction of gain, linearity, or frequency response parameters. Most related circuit-level and system-level considerations in the abovementioned references are outside of the scope of this research, in which the main focus is on the realization of suitable area and power efficient FFT implementations to support on-chip BIT and BIC approaches.
2.1 Single-tone testing

One way to characterize a nonlinear circuit is by applying a test tone at the input and measuring the harmonic distortion (HD) magnitudes in the output spectrum at multiples of the input frequency as visualized in Fig. 6. Let us assume an input test signal $x(t)$ defined by equation (1) is applied to a nonlinear system modeled with a third-order nonlinear equation as in Fig. 6. When ignoring the high-frequency components, the nonlinear response $y(t)$ of the system is defined by equation (2) below.

$$x(t) = A \cdot \cos(\omega t)$$  \hspace{1cm} (1)

$$y(t) = \frac{\alpha_2 \cdot A^2}{2} + (\alpha_1 \cdot A + \frac{3\alpha_3 \cdot A^3}{4}) \cos(\omega t) + \frac{\alpha_2 \cdot A^2}{2} \cos(2\omega t) + \frac{\alpha_3 \cdot A^3}{4} \cos(3\omega t)$$  \hspace{1cm} (2)

This approach is feasible at low frequencies but less practical at radio frequencies because of the difficulties associated with measuring high-frequency harmonics. Furthermore, the harmonics typically fall outside of the bandwidth of narrow-band RF amplifiers and other receiver blocks due to filtering.

2.2 Two-tone testing

A practical approach is to characterize a high-frequency circuit for nonlinearity such as intermodulation distortion (IMD) that falls within the frequency band of interest. For this reason, two-tone test methods as depicted in Fig. 7 are widely used, where the input test signal $x(t)$ consisting of two closely-spaced input tones ($f_1$, $f_2$) defined by equation (3) is applied to a nonlinear system model. This permits to measure the third-order intermodulation product at $(2 \cdot f_2 - f_1)$ or $(2 \cdot f_1 - f_2)$ as represented by equation (4), which can be nearby frequencies within the CUT’s passband [55].

$$x(t) = A_1 \cdot \cos(\omega_1 t) + A_2 \cdot \cos(\omega_2 t)$$  \hspace{1cm} (3)
\[ y(t) = \alpha_0 + \cdots + \frac{3\alpha_3 A^2}{4} \cos(2\omega_1 - \omega_2)t + \frac{3\alpha_3 A^2}{4} \cos(2\omega_2 - \omega_1)t \] (4)

2.3 FFT for built-in testing applications

Various on-chip FFT implementations have been developed for communication systems such as those using orthogonal frequency-division multiplexing (OFDM), which requires a combination of high throughput rate and accuracy [56]-[57]. These requirements typically translate into a large number of points (e.g. 64-2048), resulting in area and power requirements of more than 1 \( \text{mm}^2 \) and 20 \( \text{mW} \), respectively. The purpose of the proposed FFT construction is to reduce the overheads associated with on-chip FFT implementations for built-in testing applications because mixed-signal systems-on-a-chip frequently do not provide room for inclusion of such extensive digital signal processing resources.

For a discrete time signal \( x[n] \), the basic FFT algorithm represented by equation (6), calculates the spectrum of input waveform only for certain discrete frequencies known as FFT bins, which are separated by the fundamental frequency (\( \Delta f \)) of the FFT. The \( \Delta f \) strictly depends on the sampling frequency (\( f_{\text{samp}} \)) and the length of FFT (\( NFFT \)) as follows: \( \Delta f = f_{\text{samp}}/NFFT \). Thus to avoid spectral leakage and to accurately measure the frequency components of the input waveform, its frequencies has to be an exact integer multiples of the \( \Delta f \). To achieve high frequency resolution, either the sampling frequency has to be decreased or length of FFT has to be increased. Since the sampling frequency should meet the Nyquist criterion, the sampling frequency condition can typically not be relaxed when the goal is to perform on-chip analysis of the highest possible signal.
frequency under the constraints of a given CUT and CMOS technology. In contrast, increasing the length of the FFT to achieve better frequency resolution results in an additional area overhead and thereby reduces the feasibility of an FFT implantation for on-chip testing applications.

\[
x[n] \quad \forall n \in 0 \rightarrow N
\]

\[
FFT[x[n]] = X[k] = \sum_{n=0}^{N-1} x[n] \exp(-i2\pi kn / N)
\]

\[
\forall k \in 0 \rightarrow N - 1
\]

Therefore, the simple FFT algorithm is most useful for single-tone testing when the input waveform contains a fundamental frequency component and its higher-order harmonics. On the other hand, spectral leakage is difficult to avoid in multi-tone testing cases, where one or more of the input tones might not be an integer multiple of \( \Delta f \). A useful approach to alleviate the leakage effect is to choose a suitable window function that minimizes the energy spreading [59]-[60]. However, the use of a windowing technique introduces various other flaws into the spectrum such as frequency/amplitude imprecision, broadening of main lobe width, and creation of side lobes. Moreover, windowing operations require additional computational resources.

### 2.4 Conventional coherent sampling

Coherent sampling is a useful and efficient technique for evaluating spectral characteristics of analog/mixed signal circuits [61]-[64] because it increases the FFT accuracy and eliminates the need for a window function if certain conditions are met. Coherent sampling assures that the signal power in the spectrum is contained in exactly one of the frequency bins. The condition for coherent sampling is given by:

\[
\frac{f_{in}}{f_{samp}} = \frac{N_{cycle}}{NFFT},
\]

where \( f_{in} \) is the input frequency, \( f_{samp} \) is the sampling frequency, \( N_{cycle} \) is the integer number of cycles of the signal to be sampled, and \( NFFT \) is the length of the FFT.

To ensure coherent sampling one can first determine the number (usually prime) of integer cycles (\( N_{cycle} \)) that fits into the predefined sampling window, and use it to approx-
imate the input frequency to the near optimal frequency that exactly matches to one of the discrete frequency bins in the spectrum for the given FFT length [17]-[18]. Under the condition in (7), there will not be any leakage because the coherent sampling guarantees an exact integer number of input signal cycles.

Coherent sampling methods can be used for performing single-tone testing because the nearby optimal input frequency can be calculated for the fundamental signal frequency component. When several input tones are present, it is typically not guaranteed that the tones are integer multiples of the FFT’s fundamental frequency, and thus spectral leakage will occur in many multi-tone testing scenarios.

2.5 Proposed spectral analysis technique

An approach derived from the conventional coherent sampling technique is presented here for built-in testing applications. It follows the rules of coherent sampling, but instead of defining the near optimal fundamental FFT frequency for a single test tone, the coherent sampling frequency is determined based on a fixed-frequency difference between the tones in multi-tone test cases. Consequently, it circumvents the spectral leakage problems associated with basic coherent sampling during multi-tone testing. Both in the single-tone and multi-tone test cases, the method exploits that the FFT engine can be co-designed with the requirements of a given built-in test scenario, which often leaves room to select test signal frequencies in a way that the required on-chip FFT resources are minimized.

2.5.1 Single-tone testing

For single-tone test cases, the coherent sampling rule is rearranged and repeated below to introduce the alternative approach.

$$f_{\text{sampCoh}} = \frac{f_{\text{in}} \cdot NFFT}{N_{\text{cycle}}}$$  \hspace{1cm} (8)

The new sampling frequency, called coherent sampling frequency ($f_{\text{sampCoh}}$) here, is calculated for the desired input test tone ($f_{\text{in}}$) with a predefined FFT length ($NFFT$) and a chosen integer number of cycles of the input signal ($N_{\text{cycle}}$).
2.5.2 Multi-tone testing

For multi-tone test cases, the FFT fundamental frequency defined as \( \Delta f \) can be used in lieu of \( f_{in} \) in equation (8), which represents the effective resolution for suitable test tone frequencies. In a two-tone intermodulation test scenario for example, this implies that the spacing between the test tones must be equal to \( \Delta f \) or a multiple of \( \Delta f \). This frequency resolution in the spectrum can be set to the desired value for a particular on-chip testing application. With this predetermined value for \( \Delta f \), the corresponding value of \( f_{sampCoh} \) can be obtained according to:

\[
f_{sampCoh} = \frac{\Delta f \cdot NFFT}{N_{cycle}} .
\]

(9)

The choice of test tones should abide to the following rules:

1. Test tone frequencies must be less than half of the value of \( f_{sampCoh} \) in order to meet the Nyquist criterion.
2. Test tones must be separated by integer multiple of \( \Delta f \).

This approach not only eliminates the spectral leakage problem, but it also offers the following benefits when used either for single-tone or multi-tone testing:

1. Accurate spectral characteristics can be determined for a number of higher-order harmonics of the input test tone.
2. Flexibility in choosing \( N_{cycle} \) and \( NFFT \) for the given input test tone \( f_{in} \). Any values for \( N_{cycle} \) and \( NFFT \) can be used for spectral testing provided that the ratio of \( NFFT \) and \( N_{cycle} \) is chosen such that the calculated \( f_{sampCoh} \) satisfies the Nyquist criterion for the given \( f_{in} \) and for the higher-order harmonics of \( f_{in} \) to be assessed. For example: with \( N_{cycle} = 1 \) and \( NFFT = 16 \), \( f_{sampCoh} \) is \( 16f_{in} \) which means that the spectrum of \( f_{in} \) and its harmonics until the 7th-order can be precisely calculated without aliasing or spectral leakage.
3. Accurate spectral characteristics are achievable even with short FFT length, e.g., an FFT with 16 points. Thus, the presented approach offers significant savings in terms of area, power, and the required computational resources.
4. The noise levels with the discussed approach are generally low because spectral leakage is avoided based on the coherent sampling characteristics.

In a production test environment, test signals with coherent input frequencies can be generated with standard automatic test equipment. Even though the proposed on-chip FFT engine could be utilized to reduce off-chip resource requirements and the number of required test outputs of the chip for test cost reduction, it is envisioned to be more valuable during in-field testing and self-calibrations. On-chip test signals can be generated by dedicated circuits for built-in self-test such as the 40 MHz generator in [67], or by sinusoidal oscillators with wide frequency tuning range such as the 1-25 MHz oscillator in [68]. Signal generation methods with a digital foundation are advantageous to ensure coherence with the proposed approach. For example, the 41 MHz signal generator in [69] contains a block which creates a stepwise approximation of a sine wave using a digital master clock \(f_{\text{clk}}\) that is \(M = 16\) times higher than the output frequency. This synthesized sine wave is subsequently processed by an analog filter to generate a purer sinusoidal output with a 67 dB spurious-free dynamic range. Since the signal generator in [69] consumes only 0.1 \(\text{mm}^2\) in 0.35 \(\mu\text{m}\) CMOS technology, it would be a good candidate for applications that require the generation of coherent input signals on the chip at frequencies below 50 MHz. With such a signal generator, notice that the master clock that produces \(f_{\text{in}} = f_{\text{clk}}/M\) in (8) can be directly derived from \(f_{\text{sampCoh}}\) with a simple digital divider. Alternatively, the designer of the test might decide to use \(f_{\text{sampCoh}} = f_{\text{clk}}\) which would be possible with \(N_{\text{FFT}} = M = 16\) and \(N_{\text{cycle}} = 1\) in (8) for instance.

2.6 Matlab simulation results

This section presents simulation results to compare the conventional coherent sampling method and the developed approach. A single-tone test case and a two-tone test case are discussed to highlight the differences.

2.6.1 Single-tone with high-order harmonics

Consider the following input test signal with a fundamental frequency of \(f_{\text{in}}\) and four harmonic components:
\[ x(t) = 10^3 \cdot \sin(2\pi \cdot f_{in} \cdot t) + 10^3 \cdot \sin(2\pi \cdot 2f_{in} \cdot t) \\
+ 10^2 \cdot \sin(2\pi \cdot 3f_{in} \cdot t) + 10 \cdot \sin(2\pi \cdot 4f_{in} \cdot t) \]  

(10)

Let the above signal be sampled at a sampling rate of \( f_{samp} \), and analyzed using a 16-point FFT.

### 2.6.1.1 Conventional coherent sampling approach

Let \( f_{in} = 10 \text{ MHz} \), \( NFFT = 16 \), and \( f_{samp} = 200 \text{ MHz} \): \( \Delta f = f_{samp} / NFFT = 12.5 \text{ MHz} \). Using equation (7):

\[ N_{cycle} = \frac{f_{in} \cdot NFFT}{f_{samp}} = 0.8 \equiv 1 \]  

(11)

Therefore, the nearest optimal input frequency (\( f_{inCoh} \)) is:

\[ f_{inCoh} = \frac{N_{cycle} \cdot f_{samp}}{NFFT} = 12.5 \text{ MHz} \]  

(12)

It should be noted that the difference between \( f_{in} \) and \( f_{inCoh} \) is 2.5 MHz. In traditional testing applications (e.g., test of an ADC block with fixed sampling rate) it is normally feasible to generate a single test tone with a fundamental frequency of \( f_{inCoh} \) instead of \( f_{in} \). Fig. 8 shows the spectrum of the coherent (12.5 MHz fundamental) input signal \( x(t) \) obtained using a 16-point FFT and conventional coherent sampling. It can be observed

![Figure 8. FFT with conventional coherent sampling (NFFT = 16): spectrum of the input signal \( x(t) \) with a fundamental frequency component of 12.5 MHz and four harmonics.](image-url)
that the amplitudes at the signal’s fundamental and harmonic frequencies are accurately represented by the magnitudes of the frequency bins at the corresponding multiples of \( f_{\text{inCoh}} \).

2.6.1.2 Proposed approach

With \( f_{\text{in}} = 10 \, MHz \), \( NFFT = 16 \), \( N_{\text{cycle}} = 1 \) from (11), and using equations (8) and (9):

\[
f_{\text{ SampCoh}} = \frac{f_{\text{in}} \cdot NFFT}{N_{\text{cycle}}} = 160 \, MHz
\]

\[
\Delta f = \frac{f_{\text{ SampCoh}} \cdot N_{\text{cycle}}}{NFFT} = 10 \, MHz
\]

Notice that \( \Delta f \) in this case is exactly 10 \( MHz \), which allows calculating the spectrum precisely at the desired fundamental input frequency and its harmonics. This implies that the ADC sampling frequency and FFT rate must be selected accordingly, which exploits the design freedom in BIT and BIC scenarios with dedicated ADC and FFT engine. The spectrum of \( x(t) \) calculated with the proposed approach is displayed in Fig. 9, showing that the amplitudes of the fundamental and harmonic components are in exact agreement with those calculated using conventional coherent sampling.

Figure 9. FFT with the proposed method (\( NFFT = 16 \)): spectrum of the input signal \( x(t) \) with a fundamental frequency component of 10 \( MHz \) and four harmonics.
2.6.2 Two-tone signal

2.6.2.1 Conventional coherent sampling approach

Two-tone signals are frequently used to characterize the intermodulation components generated by second-order or third-order nonlinearities of a circuit under test. The frequency separation between the two test tones is typically much smaller than the frequency of each tone. Hence, it is not feasible to satisfy coherent sampling by locating the second test tone at a frequency that is a multiple of the first tone’s fundamental frequency. As demonstrated by the simulation in this subsection, the FFT spectrum will exhibit severe inaccuracies due to leakage when the input signal is comprised of one tone that satisfies coherent sampling and another tone that does not. Consider the input signal $y(t)$ such that:

$$y(t) = 10^5 \cdot \sin(2\pi \cdot f_1 \cdot t) + 10^5 \cdot \sin(2\pi \cdot f_2 \cdot t).$$

(15)

Signal $y(t)$ consists of two ideal test tones at frequencies $f_1$ and $f_2$ without intermodulation components. After using $f_1 = 12.5 \, MHz$ [equal to the coherent input frequency ($f_{inCoh}$) from equation (12)] and $f_2 = 13.5 \, MHz$ in the simulation, the spectrum in Fig. 10 was obtained with $NFFT = 16$ and $f_{samp} = 200 \, MHz$. From the signal contents at frequencies other than $f_1$ and $f_2$, it is evident that spectral leakage occurs because $f_2$ does not satisfy the coherent sampling criterion.

It is not practical to increase $NFFT$ significantly due to the corresponding die area requirement for the FFT implementation. Furthermore, in the discussed example case, such an approach would require an approximately ten times larger $f_{samp}$ in order to obtain $f_{inCoh}$ around $1 \, MHz$ based on equation (12), which would allow placing the test tones at multiples of $f_{inCoh}$ with a separation close to $1 \, MHz$. However, operating the on-chip ADC and FFT engine at $2 \, GHz$ would be impractical or consume too much power in many applications.

2.6.2.2 Proposed approach

To avoid spectral leakage in multi-tone BIT and BIC setups, the test tone frequencies can be chosen such that they and their intermodulation products are multiples of $\Delta f$. A secondary consideration is the need to maintain a sampling rate that is low enough (e.g.,
< 200 MHz) for on-chip ADC and FFT engine implementations. The example in this subsection demonstrates the benefit of selecting the sampling frequency based on the required frequency resolution (∆f) between tones. Consider the input signal z(t) such that:

\[ z(t) = 10^5 \cdot \sin(2\pi \cdot f_1 \cdot t) + 10^5 \cdot \sin(2\pi \cdot f_2 \cdot t) + 10 \cdot \sin(2\pi \cdot [2f_1 - f_2] \cdot t) + 10 \cdot \sin(2\pi \cdot [2f_2 - f_1] \cdot t) \]

which consists of test tones at f_1 and f_2 as well as their third-order intermodulation (IM3) products at the output of a circuit under test during standard two-tone testing of a nonlinear analog circuit. If ∆f = 1 MHz, then evaluation of equation (9) with NFFT = 16 and N_{cycle} = 1 yields:

\[ f_{\text{samp Coh}} = \frac{\Delta f \cdot \text{NFFT}}{N_{\text{cycle}}} = 16 \text{ MHz} \]

With this approach, frequency components up to \( f_{\text{samp Coh}} / 2 = 8 \text{ MHz} \) can be accurately measured, while the test tones have to be selected at multiples of ∆f = 1 MHz to avoid leakage. Compared to the approach with increased NFFT, this method also has the advantage that the sampling frequency is not several orders of magnitude higher than the frequencies of interest. Instead, the band of interest spans \( f_{\text{samp Coh}} / 2 \), which is 8 MHz in this example to show an FFT realization that would be useful for the characterization of

Figure 10. Spectrum of the input signal y(t) with \( f_1 = 12.5 \text{ MHz} \) (coherent) and \( f_2 = 13.5 \text{ MHz} \) (not coherent).
baseband circuits in wireless receivers or other low-frequency analog circuits. Fig. 11 displays the corresponding spectrum of $z(t)$ for a case in which $f_1 = 3 \text{ MHz}$ and $f_2 = 5 \text{ MHz}$. The result demonstrates that the two input test tones and their third-order inter-modulation components are captured with high accuracy and without any spectral leakage.

2.6.3 ADC resolution requirement

The amplitudes of harmonic distortion (HD) and intermodulation (IM) distortion components are influenced by the number of ADC bits, $n$. For a single-tone signal digitized by an $n$-bit ADC with a unit step size, the amplitude of the $p^{th}$ harmonic component is defined as [70]:

$$A_p = \delta_{p1} \cdot 2^{n-1} + \sum_{m=1}^{\infty} \frac{2}{m \cdot \pi} J_p (2 \cdot m \cdot \pi \cdot 2^{n-1}) \quad \text{if } p \text{ is odd,}$$

(18)

where $A_p = 0$ if $p$ is even, $\delta_{pq} = 0$ if $p \neq q$, $\delta_{pq} = 1$ if $p = q$, and $J_p$ is the Bessel function. Quantization of a two-tone signal with frequency components $f_1$ and $f_2$ produces IM distortion products with the following amplitudes ($A_{pq}$) and frequencies ($F_{pq}$) [70]:

$$A_{pq} = \delta_{p1} \cdot \delta_{q0} \cdot 2^{n-2} + \delta_{p0} \cdot \delta_{q1} \cdot 2^{n-2} + \sum_{m=1}^{\infty} \frac{2}{m \cdot \pi} J_p (2 \cdot m \cdot \pi \cdot 2^{n-2}) \cdot J_q (2 \cdot m \cdot \pi \cdot 2^{n-2}),$$

(19)
\[ F_{pq} = p \cdot f_1 + q \cdot f_2 \, , \]  

where \( p \) and \( q \) are integers with an odd and positive sum. From equations (18) and (19) it can be shown that increasing the resolution of an ideal ADC by 1-bit reduces the third-order harmonic distortion (HD3) component by 9 dB and IM3 components by 12 dB.

Another factor influencing the accuracy of the spectral analysis is the length \((NFFT)\) of the FFT. The digitized samples at the output of ADC contain the applied input signals with their distortion components, generated quantization noise and distortions, as well as ADC device noises. The finite size of the FFT engine puts an additional limitation on the accuracy of the spectral measurement in the presence of distortion components. To achieve high accuracy with an FFT that is preceded by an ADC, a fine frequency resolution is desired, suggesting the use of a large number of FFT points \([71]\). To avoid excessive area and power consumption due to a large-sized FFT engine, the selections of ADC resolution and FFT length have to be made under the given accuracy requirements and design constraints.

Matlab simulations were performed to show the impact of finite ADC resolution and FFT length on HD and IM components. The ADC was modeled with an n-bit quantizer, and the developed approach described in Section 2.5 was used to calculate the power spectrum of the signal. Fig. 12 shows the IM3 power in \(dBc\) (i.e., IM3 power magnitude in decibels below the fundamental components from a two-tone test) as the FFT length and the ADC resolution are swept from 16 to 2048 points and from 4 to 14 bits, respectively. The IM3 power is 57 \(dBc\) when a 10-bit ADC resolution is combined with a 16-point FFT, as annotated in Fig. 12. If necessary, higher accuracy can be achieved by increasing the FFT length. As shown in Fig. 12, the simulated power of the same IM3 component is 85 \(dBc\) with 10-bit ADC resolution but with a 2048 point FFT.

To give additional insights into the trade-offs of ADC resolution and FFT length, theoretical values (ADC distortion only, \(NFFT = \infty\)) obtained from equations (18) and (19) were compared with simulated results using \(NFFT = 2^4, 2^5, 2^6, 2^7, \) and \(2^{15}\). Tables 1 and 2 summarize the IM3 and HD3 components (in \(dBc\)) for the respective two-tone and single-tone test cases discussed in Section 2.6. Notice that only the test tones were applied, such that the listed IM3 and HD3 levels represent the accuracy limit of the ADC-FFT combination under the given conditions. For a 10-bit ADC, HD3 levels down
to 72 dB below the fundamental can be extracted with $NFFT = 2^4$. In contrast, the selection of $NFFT = 2^{15}$ would allow to identify HD3 down to 89 dB, which is close to the 90 dBc theoretical HD3 generated by the 10-bit ADC.

An important metric for ADC nonlinearity is differential nonlinearity (DNL), which can be as low as 0.09 LSB to ±0.5 LSB ([72]-[73]) for example. To assess the impact of the ADC’s static nonlinearity, simulations were performed with DNL error of ±0.3 LSB and ±0.5 LSB by generating random offsets of each reference voltage level with a Gaussian distribution such that the DNL value is equal to 3 times the standard deviation ($\sigma$). Table 3 summarizes the mean ($\mu$) and the standard deviation ($\sigma$) of the IM3 components from 20 simulations for the two DNL cases with $NFFT = 2^4$. Assuming a 10-bit ADC with ±0.3 LSB DNL, a mean IM3 of 65.32 dBc with 4.86 dB standard deviation is observed. This suggests that the inherent linearity of the ADC and FFT combination would allow to measure IM3 levels over 50 dBc (3∙$\sigma$ lower than the mean) in 99.8% of the cases.
To capture the effect of noise on the IM3 characterization capability, white noise was introduced at the ADC input in Matlab simulations such that the ADC’s effective number of bits (ENOB) reduces by 1-bit. The mean (µ) and the standard deviation (σ) of the measured IM3 component from 20 simulations are listed in Table 4. The results

Table 1. Theoretical and simulated IM3 (in dBc) from a two-tone test with \( f_1 = 3 \text{ MHz}, f_2 = 5 \text{ MHz}, f_{IM3} = 1 \text{ MHz} \)

<table>
<thead>
<tr>
<th>NFFT</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>92.5</td>
</tr>
<tr>
<td>9</td>
<td>96.31</td>
</tr>
<tr>
<td>10</td>
<td>100.69</td>
</tr>
<tr>
<td>11</td>
<td>104.3</td>
</tr>
<tr>
<td>12</td>
<td>108.37</td>
</tr>
<tr>
<td>13</td>
<td>114.4</td>
</tr>
<tr>
<td>14</td>
<td>126.4</td>
</tr>
</tbody>
</table>

Table 2. Theoretical and simulated HD3 (in dBc) from a single-tone test with \( f_0 = 3 \text{ MHz} \)

<table>
<thead>
<tr>
<th>NFFT</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>83.31</td>
</tr>
<tr>
<td>9</td>
<td>86.42</td>
</tr>
<tr>
<td>10</td>
<td>92.91</td>
</tr>
<tr>
<td>11</td>
<td>99.34</td>
</tr>
<tr>
<td>12</td>
<td>106.99</td>
</tr>
<tr>
<td>13</td>
<td>113.49</td>
</tr>
<tr>
<td>14</td>
<td>120.4</td>
</tr>
</tbody>
</table>

Table 3. Simulated IM3 (in dBc) with DNL error (from a two-tone test with \( f_1 = 3 \text{ MHz}, f_2 = 5 \text{ MHz}, f_{IM3} = 1 \text{ MHz}, NFFT = 2^4 \) )

<table>
<thead>
<tr>
<th>DNL (LSB)</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>41.45</td>
</tr>
<tr>
<td>8</td>
<td>44.57</td>
</tr>
<tr>
<td>9</td>
<td>50.80</td>
</tr>
<tr>
<td>10</td>
<td>57.22</td>
</tr>
<tr>
<td>11</td>
<td>64.09</td>
</tr>
<tr>
<td>12</td>
<td>72.11</td>
</tr>
<tr>
<td>13</td>
<td>84.3</td>
</tr>
<tr>
<td>14</td>
<td>84.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>±0.3 (µ)</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>49.75</td>
<td>53.15</td>
</tr>
<tr>
<td>59.52</td>
<td>65.32</td>
</tr>
<tr>
<td>66.30</td>
<td>75.04</td>
</tr>
<tr>
<td>80.71</td>
<td>87.58</td>
</tr>
<tr>
<td>106.99</td>
<td>113.49</td>
</tr>
<tr>
<td>120.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>±0.3 (σ)</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.64</td>
<td>4.92</td>
</tr>
<tr>
<td>5.29</td>
<td>4.86</td>
</tr>
<tr>
<td>7.37</td>
<td>7.73</td>
</tr>
<tr>
<td>10.69</td>
<td>11.05</td>
</tr>
<tr>
<td>12.45</td>
<td>12.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>±0.5 (µ)</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>48.70</td>
<td>52.90</td>
</tr>
<tr>
<td>58.10</td>
<td>70.40</td>
</tr>
<tr>
<td>70.98</td>
<td>75.59</td>
</tr>
<tr>
<td>93.98</td>
<td>98.58</td>
</tr>
<tr>
<td>104.5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>±0.5 (σ)</th>
<th>n-bit ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.01</td>
<td>5.45</td>
</tr>
<tr>
<td>5.17</td>
<td>5.51</td>
</tr>
<tr>
<td>5.85</td>
<td>6.25</td>
</tr>
<tr>
<td>7.95</td>
<td>8.35</td>
</tr>
<tr>
<td>8.65</td>
<td>9.05</td>
</tr>
</tbody>
</table>

To capture the effect of noise on the IM3 characterization capability, white noise was introduced at the ADC input in Matlab simulations such that the ADC’s effective number of bits (ENOB) reduces by 1-bit. The mean (µ) and the standard deviation (σ) of the measured IM3 component from 20 simulations are listed in Table 4. The results
indicate that IM3 levels down to 52 dBc (\(\mu - 3\sigma\)) can be extracted with 99.8% confidence using \(NFFT = 2^4\) and an ADC having an ENOB of 10-bit. As shown in Table 5, in presence of both DNL of \(\pm 0.5\) LSB and wideband noise, IM3 components of 52.5 dBc (\(\mu - 3\sigma\)) or higher can be accurately determined with a 16-point FFT and ENOB = 10-bit. The results suggest that presence of DNL causes distribution of power over more frequency bins, which leads to increased variation of the IM3 measurement capability but with a higher mean. The DNL of the ADC causes distribution of power over frequency bins due to the generation of more intermodulation products, which depend on the random DNL variation across the reference voltages. Generally, non-zero DNL results in less power at the IM3 frequencies but higher power at other frequencies due to the generation of different intermodulation products. However, the power levels in all frequency bins exhibit higher variations with non-zero DNL.

To investigate the effect of ADC bandwidth limitation on the IM3 characterization capability, a third-order lowpass filter (LPF) with a cut-off frequency of 3.37 MHz was placed in front of an ideal behavioral model of a 10-bit ADC to model frequency limitation during Spectre simulations in Cadence with a 16 MHz sampling frequency. The characteristics of the LPF were such that the ADC has an ENOB of approximately 10-bit.
and 6-bit for \( f_{in} \leq 3 \text{ MHz} \) and at \( f_{in} = 7 \text{ MHz} \), respectively. Furthermore, the ENOB with input frequencies of 5 MHz and 6 MHz is approximately the same (ENOB \( \approx 7 \)-bit), which is why they were selected for a two-tone test simulation. After taking a 16-point FFT of the ADC output, the IM3 components of 50.96 \( dBc \) and 55.64 \( dBc \) were obtained at the frequencies of 4 MHz and 7 MHz, respectively. This estimated test accuracy at frequencies with an ENOB \( \approx 7 \)-bit deviates by 2-7 \( dB \) from the IM3 limit for the 7-bit ADC case with DNL in Table 3. Note that the test tones experience different attenuations and phase shifts because the ADC bandwidth limitation is modeled with a third-order LPF, which impacts the IM3 results. Nevertheless, the result indicates that the ADC ENOB dictates the test accuracy as in the previously discussed example with wideband noise.

For on-chip implementation, it is desirable to achieve small area and power with sufficient accuracy. The combination of a 10-bit ADC and 16-point FFT were chosen for the implementation example discussed in the next section because it provides appropriate accuracy for various built-in test scenarios. Efficient ADCs can be designed with 10-80 MS/s for the anticipated applications. For example, a 10-bit ADC with 40 MS/s sampling rate implemented in 65 nm CMOS technology was reported in [74] with an active area of 0.06 mm\(^2\) and power dissipation of 1.21 mW.

### 2.7 Selection of NFFT and the ADC resolution

Based on the analysis presented in Section 2.6, the following procedure can be adopted to aid in selecting the appropriate ADC resolution and the FFT length:

**Step 1:** Identify the required accuracy of the IM3 (or HD3) assessment (in \( dBc \)). For example, a transistor-level simulation of a CUT using test tones with a specified amplitude will result in a certain output IM3 that should be lower than the IM3 limits in the tables.

**Step 2:** Based on Table 1, select the appropriate ADC resolution \( (n) \) and the FFT length \( (NFFT) \). If one of the two is too high for efficient on-chip implementation based on available resources, then the test tone amplitudes in step 1 can be increased to obtain lower output IM3 (in dBc), relaxing the requirements for the on-chip test resources.

**Step 3:** Define the desired frequency resolution \( (\Delta f) \) in the output spectrum depending on the application-specific requirements and available test tone generation circuitry.
**Step 4:** Using equation (9), determine the coherent sampling frequency ($f_{sampCoh}$). Note that $f_{sampCoh}$ should be chosen such that the desired IM3 (or HD3) components fall within $f_{sampCoh}/2$.

**Step 5:** Choose the input test-tone frequencies ($f_1$, $f_2$) as an integer multiple of the frequency resolution ($\Delta f'$).

### 2.8 Implementation example

A radix-2 16-point FFT engine was implemented to determine the spectral characteristics of the signal generated at the output of an ADC. The FFT engine is based on the standard Decimation-in-Time (DIT) algorithm. It is designed as a serialized, streaming I/O FFT block that accepts streaming complex input and generates streaming complex output continuously with every clock cycle after an initial latency of 34 clock cycle, where each output corresponds to a frequency bin. The input and the output data streams are represented in 2’s complement Q10.4 and Q13.4 format, respectively. The output of the 10-bit ADC represents the integer portion of the input data where 4 additional fractional bits are appended to achieve a resolution of -84 dBc. The integer portion of the output data is comprised of 13 bits to capture the overflow that is generated during the FFT computation.

Fig. 13 shows the block diagram of the FFT engine, where $I_{in}$, $I_{out}$ and $Q_{in}$, $Q_{out}$ represent the real and the imaginary parts of the input and the output data streams respectively. The input data is passed to the FFT Logic unit and the processed data is carried to the Butterfly unit for further arithmetic operations or to the DualPort RAM unit for storage in the registers. The functions of FFT Logic unit are to reorder the output bins of the FFT engine, to calculate addresses for the Butterfly unit, and to count the delay for feedback registers. Its outputs are the twiddle indexes, addresses for the DualPort RAM, calculated real and imaginary parts of data, and FFT outputs ($I_{out}$, $Q_{out}$). The twiddle indexes are passed to the Twiddle Table unit, where the sine and cosine values are selected for the calculations inside the Butterfly unit. The outputs of the Butterfly unit and the DualPort RAM are two pairs of real and imaginary numbers that are calculated in parallel. To reduce hardware complexity, the DualPort RAM serves as a set of feedback delay registers, and a minimized Butterfly unit is used.
The FFT engine was implemented in Verilog HDL and synthesized to the gate level netlist with Cadence RTL compiler using the publicly available 45 nm PDK [75]. The generated gate level netlist was ported to the Cadence Encounter tool to complete the physical layout of the FFT engine and to evaluate the overall area and power require-
The layout is displayed in Fig. 14. It occupies a total chip area of around 0.073 mm² with 87% density in 45 nm CMOS technology. The total estimated power dissipation for 16 MHz 16-point FFT computations at 1.1 V supply voltage is 6.47 mW.

From Table 1, it should be noted that an ideal two-tone input signal when quantized using a 10-bit ADC contains an IM3 component at 57.2 dBc in the output spectrum obtained with a 16-point FFT. Thus, to accurately determine the IM3 of an input signal from a circuit under test, this IM3 component should be larger than the intrinsic linearity limitation of the chosen combination of the ADC resolution and the number of FFT points. As an example, a two-tone test signal represented by equation (16) but with the known IM3 components of 50 dBc was applied during the post-layout simulation of the 16-point FFT engine implementation together with a 10-bit ADC (Verilog model). The extracted parasitic resistances and the capacitances were included in the post-layout simulation. The output bit streams from the post-layout simulation were imported to Matlab to plot the resulting output spectrum that is shown in Fig. 15. Table 6 compares the post-layout simulation results of the implemented FFT engine with the FFT of the input signal to the ADC obtained with the Cadence calculator using 65536 points. The column named “Actual Input” lists the differences (in dBc) between the input components specified for the signal sources at the frequencies of interest relative to the full-

![Graph](image)

Figure 15. Output spectrum from post-layout simulation, showing the two test tones \( (f_1 = 3 \text{ MHz}, f_2 = 5 \text{ MHz}) \) and their IM3 products at 1 MHz and 7 MHz.
As reference for comparison to the post-layout 16-point FFT results, the “Calculated Input” column includes the corresponding frequency component values obtained for the same input with an ideal 65536-point FFT. An error of less than ±0.02 dB is observed for the fundamental components at 3 MHz and 5 MHz, whereas the applied 50 dBc IM3 components at 1 MHz and 7 MHz are captured with an error of 0.72 dB and 1.32 dB respectively. The post-layout FFT engine simulation results indicate that the chosen combination of a 10-bit ADC and a 16-point FFT in this example is suitable to accurately determine the IM3 components of ≤ 50 dBc.

Table 6. Results from a two-tone test: calculated FFT of the input signal (65536 points) vs. output of the 16-Point FFT engine

<table>
<thead>
<tr>
<th>Frequency (MHz)</th>
<th>Actual Input (dBFS)</th>
<th>Calculated Input (dBFS)</th>
<th>Post-Layout Output (dBFS)</th>
<th>Actual IM3 (dBc)</th>
<th>Calculated IM3 (dBc)</th>
<th>Post-Layout Output IM3 (dBc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1, f_2 = 3, 5$</td>
<td>-11.37</td>
<td>-11.38</td>
<td>-11.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$f_{IM3} = 1$</td>
<td>-61.37</td>
<td>-61.31</td>
<td>-62.08</td>
<td>50.00</td>
<td>49.93</td>
<td>50.72</td>
</tr>
<tr>
<td>$f_{IM3} = 7$</td>
<td>-61.37</td>
<td>-61.36</td>
<td>-60.04</td>
<td>50.00</td>
<td>49.98</td>
<td>48.68</td>
</tr>
</tbody>
</table>
3. Digitally-Assisted Analog/RF Design

Nowadays, analog/RF circuits have to be optimized by design for reliable high performance with minimal power consumption to reach complete system-on-chip solutions that meet the end user demands in diverse applications. Emerging technologies and device scaling eased the integration of analog/RF circuits but provide insufficient control over the process, temperature and voltage (PVT) variations, resulting in undesirable system characteristics. However, transistor scaling implies faster switching for digital circuits and enables the integration of more digital processing elements in less area with lower power consumption. Consequently, there is a tremendous thrust to incorporate digital techniques to continuously tune analog/RF components and to provide robust mechanisms for achieving efficient solutions under varying operating conditions. For example, a conceptual block diagram of an adaptive RF transmitter is shown in Fig. 16. The response of the system to a predetermined optimum test signal is captured and fed to the digital signal processor (DSP) where the model parameters and the performance metrics of the system components across all tuning knob settings are predicted. A tuning algorithm then uses the predicted performance metric(s) to find the optimal control settings for each individual component.

This chapter introduces on-chip linearity calibration techniques that were developed to tune RF amplifiers. These methodologies are based on the spectral analysis technique described in Chapter 2 to efficiently estimate the spectral components of the amplifiers for tuning algorithms.

Figure 16. Conceptual block diagram of digitally-assisted RF transmitter.
3.1 On-chip amplifier linearity calibration with the fast Fourier transform

To enhance the performance and the reliability of mixed-signal systems, digitally-assisted analog designs [38] and calibration approaches [39]-[40] are significantly gaining importance in current CMOS technologies. Analog designs equipped with digitally tunable capabilities for impedance matching, for gain and frequency tuning of low-noise amplifiers [40]-[42], second-order nonlinearity and mismatch correction for mixers [43]-[45], linearity enhancements for baseband filters [46] are some examples of the current trends that show the relevance of incorporating digital design techniques. Furthermore, digital circuits can be employed on the chip to efficiently extract performance metrics of analog circuits, enabling periodic calibration and tuning of complete systems to meet application-specific requirements. One way of estimating performance metrics is through spectral testing, where performance parameters such as linearity, gain, third-order intermodulation intercept point (IIP3) etc. can be derived from the spectral characteristics of the circuit under test (CUT). Traditionally, spectral components are measured using off-chip instruments, whereas on-chip spectrum analyzers [47]-[48] have been used for built-in calibration (BIC) and built-in self-test (BIST). Alternatively, area-efficient BIC schemes that capitalize on available on-chip resources such as analog-to-digital converters (ADC) and digital-to-analog converters (DAC) to estimate characteristics of CUTs have also been demonstrated [49]-[50].

One way to characterize a nonlinear circuit is through single-tone testing, where one test tone is applied at the input of the CUT and the generated harmonic distortion (HD) components due to nonlinearity of the CUT are measured in the output spectrum. This approach is prominent at low frequencies because the HD components are also at low frequencies and can be efficiently measured using on-chip or off-chip instrumentation. An alternative way to characterize a nonlinear circuit is with a two-tone test. In a standard two-tone test, two closely-spaced input tones at frequencies $\omega_1$ and $\omega_2 = \omega_1 + \Delta \omega$ are applied and their intermodulation (IM) products at frequencies $m \cdot \omega_1 \pm n \cdot \omega_2$ are measured, where $m$ and $n$ are integers. Typically the third-order intermodulation (IM3) products at $(2 \cdot \omega_2 - \omega_1)$ or $(2 \cdot \omega_1 - \omega_2)$ fall within the CUT’s passband and are often used
to characterize the nonlinearity of a CUT [21], [55]. Use of the latter approach to design a low-cost and area-efficient on-chip calibration scheme is the focus of this chapter.

Fig. 17 presents the BIC approach that was developed, in which a two-tone test signal is applied at the input of the CUT. The output of CUT is digitized using an ADC. In order to reduce the spectral leakage in the output spectrum, the signal is sampled at a coherent sampling frequency \( f_{\text{sampCoh.}} \) determined from the conventional coherent sampling approach [19]-[21]. The digitized signal is processed by an on-chip fast Fourier Transform (FFT) engine to calculate the output spectrum of the test signal. The magnitude of the spectral components are used by a digital control block to control the switch bank \( S (S_0, S_1, S_2) \), which in turn tunes the capacitor array \( C (C_0, C_1, C_2) \) to an appropriate value for optimum CUT performance.

Figure 17. Block diagram of the calibration scheme.
3.1.1 LNA with tuning functionality

A subthreshold LNA with a linearity enhancement technique was designed in the Analog & Mixed-Signal Integrated Circuit research group and is reported in [76]. Fig. 18 shows the schematic of the cascode LNA with inductive source degeneration, where the added tunable capacitor $C_{gd2\_ext}$ and inductor $L_g2$ enable linearity improvement. The LNA in Fig. 18 was designed in 0.13 $\mu$m CMOS technology with a 2.4 GHz center frequency. Table 7 summarizes the simulated specification parameters of the LNA for $C_{gd2\_ext} = 140 \, fF$. The subthreshold LNA design achieves high linearity (IIP3: -2.0 dBm, 1-dB compression point: -13.9 dBm) and low power consumption (240 $\mu$W). The value of capacitor $C_{gd2\_ext}$ is tuned in a range from 80 $fF$ to 220 $fF$ with the help of a capacitor array consisting of three parallel capacitors $C_0$, $C_1$, and $C_2$. The selection of the combination of capacitors from the array is performed with a 3-bit control code that controls a switch bank S consisting of three digital switches ($S_0$, $S_1$, and $S_3$). For example, a control word of ‘000’ corresponds to $S_0 = '0'$, $S_1 = '0'$, and $S_3 = '0'$, which means that none of the capacitors from the capacitor bank is selected, and thereby the capacitor $C_{gd2\_ext}$ is tuned to a minimum value of 80 $fF$. On the other hand, a control word of ‘111’ corresponds to $S_0 = '1'$, $S_1 = '1'$, and $S_3 = '1'$, which enables the selection of all the three capacitors and thus tunes the capacitor $C_{gd2\_ext}$ to a maximum value of 220 $fF$. By adjusting the capacitor $C_{gd2\_ext}$ the performance of the LNA in [76] can be tuned for optimum performance to

![Figure 18. Low-noise amplifier (LNA) with linearity tuning capability [76].](image-url)
compensate for manufacturing process variations. Table 8 lists the control codes with corresponding to IIP3 and IM3 simulation results in \textit{dBc} for an input power level of -30 \textit{dBm} with a two-tone test.

The LNA in [76] operates in the radio frequency (RF) range, thus an efficient down-conversion mechanism is required to convert the high-frequency response of the LNA to low frequencies. One way to down-convert the high-frequency response of a CUT is by using a simple envelope detector at the output of the CUT. For example, it has been outlined in [54] that the low-frequency envelope response of the two-tone test can be used to extract the high-frequency nonlinearity characteristic of the CUT at frequencies below 50MHz, such that an area and power efficient ADC can be employed to capture the response. The proposed calibration approach for tunable circuits such as the LNA in
is presented next using a mathematical model in Matlab under the assumption that the frequency response has been translated to low frequencies.

3.1.2 Calibration scheme

Mathematically, the nonlinearity of an analog circuit can be approximated using a third-order nonlinear equation [78]:

\[
y(t) \approx a_0 + a_1 \cdot (x(t)) + a_2 \cdot (x(t))^2 + a_3 \cdot (x(t))^3,
\]

where \( y(t) \) is the transient response of the circuit to the applied input signal \( x(t) \), and \( a_j \) are the coefficients that define the linearity of the CUT. Consider a two-tone test as visualized in Fig. 19, where a two-tone signal defined by equation (22) is applied at the input of a nonlinear CUT modeled by equation (21):

\[
x(t) = A \cdot \sin(\omega_1 \cdot t) + A \cdot \sin(\omega_2 \cdot t),
\]

where \( A \) is the amplitude of the two tones at frequencies \( \omega_1 = \omega_o - \omega_b/2 \) and \( \omega_2 = \omega_o + \omega_b/2 \). By evaluating equation (21) and discarding the DC terms, insignificant harmonics and second-order intermodulation (IM2) components, the components at the fundamental frequencies \( \omega_1 \) and \( \omega_2 \) are defined by:

\[
B_1 = \left( a_1 \cdot A + \frac{9}{4} a_3 \cdot A^3 \right) \cdot \sin(\omega_1 \cdot t),
\]

\[
\left( a_1 \cdot A + \frac{9}{4} a_3 \cdot A^3 \right) \cdot \sin(\omega_2 \cdot t)
\]

The components at frequencies \((3 \cdot \omega_1), (3 \cdot \omega_2), (2 \cdot \omega_1 + \omega_2), \) and \((2 \cdot \omega_2 + \omega_1)\) are at high

![Image](image.png)

Figure 19. Spectral components of interest during a standard two-tone test.
frequencies and can be easily filtered off. The in-band IM3 components at frequencies 
\((2 \cdot \omega_1 - \omega_2)\) and \((2 \cdot \omega_2 - \omega_1)\) are:

\[
B_2 = \left(\frac{3}{4} a_3 \cdot A^3\right) \cdot \sin(2 \cdot \omega_1 - \omega_2) \cdot t ,
\]

\[
\left(\frac{3}{4} a_3 \cdot A^3\right) \cdot \sin(2 \cdot \omega_2 - \omega_1) \cdot t
\]

(24)

The described BIC scheme is based on IM3 testing with a two-tone signal, where the goal is to extract IM3 components to determine the best performance tuning parameters. A key element of the approach is the digital calibration control that is presented next.

3.1.2.1 Digital calibration control

As represented in Fig. 20, the digital calibration control block implements a simple search algorithm that compares the magnitude of the current IM3 component \((CuIM3)\) with the previously known best IM3 component \((BestIM3)\) and saves the binary control code that results in the lowest achievable IM3 components, indicating optimum linearity. It consists of three \(N\)-bit registers Count1, Count2 and BestCode, as well as one \(M\)-bit register BestIM3, where \(N\) and \(M\) represent the width of binary control code and the width of the data word used to represent the magnitude of the spectral components respectively. The \(N\)-bit binary control code can be used to evaluate a number of \(2^N\) settings during the tuning. Register Count2 holds a binary value that corresponds to the number of tuning settings that the user wants to evaluate. Register Count1 is used to keep track of the current control code that is being evaluated, whereas the register BestCode is used to save the control code that results in the best performance of the CUT. Register BestIM3 stores the magnitude of the lowest IM3 components. The value of the BestIM3 register is compared with the magnitude of current IM3 component \((CuIM3)\) to determine the lowest IM3 component. If the condition "\(CuIM3 \leq BestIM3\)" is satisfied, then the value of the register BestIM3 is updated with the corresponding value of CuIM3. The compare-save algorithm is iteratively executed for the number of iterations specified in the register Count2.
For example, consider the look-up table (LUT) in Table 8 with the 3-bit binary control code \((N = 3)\) varying from '000' to '111', corresponding to \(2^3 = 8\) different sets of performance parameters \((a_i)\). The goal is to find the control code that leads to the smallest IM3 component (i.e., the largest value in \(dBc\)) among all the eight (0 to 7, or '000' to '111') possible control codes. During the initialization phase, register Count2 is set to a binary value of '111' that corresponds to the total of number binary-coded control codes, which is \(2^3 - 1 = 7\) in this case. Register Count1 is initialized to '000' and is incremented by '1' after each iteration until it is equal to Count2 to exit the loop. Register BestIM3 is initialized to a large value such that it always satisfies the condition "\(CuIM3 \leq BestIM3\)" when compared with the value CuIM3 during the first iteration of the compare-save algorithm. Usually the magnitudes of the IM3 components are smaller than the magnitudes of the fundamental components. Hence, the magnitude \((A)\) of the fundamental components can be used as an initial value of BestIM3. After the initialization phase, the value of Count1 = '000' is applied to the switch bank S to tune the capacitor \(C_{gd2_{ext}}\) to a value of 80 \(fF\). The output response of the CUT with the current value of \(C_{gd2_{ext}}\) is evaluated to estimate the magnitude of the current IM3 component that is subsequently
saved in the register CuIM3. The value of CuIM3 is then compared to the value stored in register BestIM3 to evaluate the condition "CuIM3 ≤ BestIM3". If the condition is satisfied, the values of BestIM3 and BestCode are updated with the value of CuIM3 and the current control code, respectively. However if the condition "CuIM3 ≤ BestIM3" is not satisfied, then the present value of BestIM3 is preserved whereas the present value of CuIM3 is discarded. After comparing BestIM3 and CuIM3, the value of Count1 is incremented by '1' and is compared with the value of Count2. If the condition Count1 < Count2 holds true, the binary value of register Count1 is applied to the switch bank that sets capacitor C_{gd2_ext} to a new value. The whole loop is repeated until the condition Count1 < Count2 becomes false to terminate the loop. Afterwards, the control code corresponding to the lowest IM3 component is retrieved from the register BestCode and applied to the switch bank to operate the CUT with optimum linearity performance.

3.1.2.2 Sampling frequency

The accuracy of the IM3 estimation is affected by the error introduced due to limited resolution of the ADC as well as due to the presence of spectral leakage in the output spectrum from the limited FFT length. Spectral leakage can be reduced by using an appropriate window function. However, the use of windowing techniques introduces other inaccuracies because of a broadened main lobe, creation of side lobes, and the introduction of amplitude/frequency imprecisions. Another useful and efficient approach to reduce the spectral leakage is conventional coherent sampling, which dictates the following relationship:

\[
\frac{f_{in}}{f_{samp}} = \frac{N_{cycle}}{NFFT},
\]  

where \( f_{in} \) is the input frequency, \( f_{samp} \) is the sampling frequency, \( N_{cycle} \) is the integer number of cycles of the signal to be sampled, and \( NFFT \) is the length of FFT engine. The condition of the conventional coherent sampling technique ensures that the signal energy is constrained to a single frequency bin, thereby eliminating spectral leakage for a single-tone input. However, for a multi-tone input signal it is not guaranteed that all the input tones meet condition (25), which causes spectral leakage.
Equation (25) can be re-arranged to determine the coherent sampling frequency \( f_{\text{sampCoh}} \) that satisfied the coherent sampling condition for input frequency \( f_{\text{in}} \):

\[
f_{\text{sampCoh}} = \frac{f_{\text{in}} \cdot N_{\text{FFT}}}{N_{\text{cycle}}}
\]  

(26)

The condition in equation (26) states that spectral leakage can be avoided provided that the input signal is sampled at a rate of \( f_{\text{sampCoh}} \) for a given FFT length \( N_{\text{FFT}} \) and for given integer number of cycles of the input signal \( N_{\text{cycle}} \). Sampling the input signal at \( f_{\text{sampCoh}} \) ensures that the spectral components in the output spectrum are at integer multiples of frequency \( f_{\text{in}} \). For multi-tone test cases equation (26) can be utilized to achieve the desired frequency resolution \( \Delta f \) in the output spectrum:

\[
f_{\text{sampCoh}} = \frac{\Delta f \cdot N_{\text{FFT}}}{N_{\text{cycle}}},
\]

(27)

where \( \Delta f \) represents the effective resolution for suitable test tone frequencies. For instance, consider a two-tone test signal \( s(t) \) composed of two signals at frequencies \( f_1 = 3 \text{ MHz} \) and \( f_2 = 5 \text{ MHz} \). Based on equation (27), the signal should be sampled at a rate of \( f_{\text{sampCoh}} = \Delta f \cdot N_{\text{FFT}} \) for a frequency resolution \( \Delta f = 1 \text{ MHz} \) with \( N_{\text{cycle}} = 1 \)' and a given FFT length \( N_{\text{FFT}} \).

3.1.3 Simulation results

This section demonstrates the application of the developed calibration scheme to find the best tuning setting for LNA presented in Section 3.1.1. As explained in Section 3.1.1, it is assumed that the spectral information of the LNA output has been down-converted to be captured by the ADC. Consider a test signal \( x(t) \) composed of two tones with equal amplitude of \( A = 0.01 \text{ V} \) (input power level of -30 dBm) and at frequencies \( f_1 = 3 \text{ MHz} \) and \( f_2 = f_1 + \Delta f = 3 \text{ MHz} + 1 \text{ MHz} = 4 \text{ MHz} \):

\[
x(t) = 10^{-2} \cdot \sin(2\pi \cdot f_1 \cdot t) + 10^{-2} \cdot \sin(2\pi \cdot f_2 \cdot t)
\]

(28)

For the input frequencies \( f_1 = 3 \text{ MHz} \) and \( f_2 = 4 \text{ MHz} \), the IM3 components are present at frequencies \( f_{\text{IM3-1}} = 2f_1 - f_2 = 2 \text{ MHz} \) and \( f_{\text{IM3-2}} = 2f_2 - f_1 = 5 \text{ MHz} \). During the Matlab simulation, signal \( x(t) \) was applied to a CUT modeled using equation (21) and signal \( y(t) \) was generated with the IM3 components specified in Table 8:
\[ y(t) = A \cdot \sin(2\pi \cdot f_1 \cdot t) + A \cdot \sin(2\pi \cdot 2f_2 \cdot t) + A_{IM3} \cdot \sin(2\pi \cdot f_{IM3-1} \cdot t) \]
\[ + A_{IM3} \cdot \sin(2\pi \cdot f_{IM3-2} \cdot t) \]  \hspace{1cm} (29)

where \(A_{IM3}\) are the amplitudes of the IM3 components with the various tuning settings of the CUT. The IM3 components at the output of the CUT can be extracted by directly applying the signal \(y(t)\) to the ADC and then to the FFT, or by down-converting the signal before quantizing it with the ADC. Here, with \(NFFT = 16\), \(N_{cycle} = 1\) and using equation (27), the signal \(y(t)\) was directly quantized at a sampling rate of \(f_{sampCoh} = \Delta f \cdot NFFT = 1 \text{ MHz} \cdot 16 = 16 \text{ MHz}\). During the Matlab simulation, the closed loop was iteratively executed for each tuning setting to determine the code that results in the lowest IM3 level (corresponding to the highest value in \(dB_C\); i.e., the difference between the amplitudes of the main tones (A) and the amplitudes of the IM3 tones (\(A_{IM3}\)) in decibels. The results for the iterations are summarized in Table 9. The value of "BestCode" corresponds to the control code that results in the lowest IM3 level, which becomes the final setting for the LNA after the last iteration. It can be observed in Table 9 that the final value of BestCode is ‘011’, which matches to the control code for lowest IM3 level in Table 8.
3.2 On-chip amplifier calibration using envelope response analysis with a two-tone signal

Digitally-assisted calibration methods can incorporate existing or dedicated analog-to-digital converter (ADC) and digital signal processing resources to directly quantize the output signals of analog circuits for computation of the fast Fourier Transform (FFT) and automatic tuning with digital-to-analog converters (DACs) [49]-[50]. Extraction of circuit linearity parameters with the latter approach calls for efficient FFT implementations as discussed in Section 3.1. However, at radio frequencies the feasibility of performing conventional two-tone testing with desired accuracy becomes quite difficult and computationally expensive due to the limitation incurred by the performance of the ADC and the required DSP circuitry. Apart from the performance limitation of the available resources, the cost of test equipment to perform two-tone tests at RF frequencies is exorbitant, thereby limiting the feasibility of classical two-tone tests for built-in self-test (BIST) and built-in calibration (BIC) approaches at RF frequencies. Alternative-

Table 9. Verification of the calibration algorithm through simulation with a two-tone test 
($f_1 = 3$ MHz, $f_2 = 4$ MHz) using a 10-bit ADC and 16-point FFT

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Code</th>
<th>Count1</th>
<th>Count2</th>
<th>BestCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>000</td>
<td>000</td>
<td>111</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>001</td>
<td>001</td>
<td>111</td>
<td>001</td>
</tr>
<tr>
<td>3</td>
<td>010</td>
<td>010</td>
<td>111</td>
<td>010</td>
</tr>
<tr>
<td>4</td>
<td>011</td>
<td>011</td>
<td>111</td>
<td>011</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>100</td>
<td>111</td>
<td>011</td>
</tr>
<tr>
<td>6</td>
<td>101</td>
<td>101</td>
<td>111</td>
<td>011</td>
</tr>
<tr>
<td>7</td>
<td>110</td>
<td>110</td>
<td>111</td>
<td>011</td>
</tr>
<tr>
<td>8</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>011</td>
</tr>
<tr>
<td>Stop</td>
<td>011</td>
<td></td>
<td></td>
<td>011</td>
</tr>
</tbody>
</table>
ly, it has been shown that envelope detectors can be placed at the output of an RF circuit to extract the signal characteristics at lower frequencies. For example, it is outlined in [54] how an input signal with two test tones \((f_1, f_2)\) allows to determine third-order linearity characteristics of the circuit under test (CUT) by monitoring the spectral components at frequencies up to \(3\cdot(f_2 - f_1)\), which can be assured to be low by selecting appropriate test tone frequencies with narrow spacing. This also permits the use of lower sampling frequencies. Use of the latter approach to design a low-cost and area-efficient on-chip calibration scheme is discussed in this section. The LNA described in Section 3.1.1 is used as a CUT to demonstrate the performance of the BIC scheme that was developed. Fig. 21 presents the proposed BIC approach. This calibration method can be efficiently applied to an individual analog block or to a cascade of blocks operating at low or high frequencies with a two-tone test signal at the input of the CUT. In a low-frequency BIC, the output of the CUT can be directly fed to the ADC, after which it is acquired by the FFT engine to calculate the spectrum of the input signal. The output of the FFT engine is usually a complex number. To estimate the absolute magnitude of the spectral components, a magnitude calculator block is used after the FFT. The estimated magnitudes of the spectral components are processed by the digital control block to efficiently tune the CUT for improved performance.
The loop to calibrate circuits operating at low frequencies is represented by the black solid lines in Fig. 21. Incorporating an on-chip ADC and FFT engine to directly process the output signals of analog circuits could be efficient for low-frequency signals, but the performance of the ADC is significantly limited at radio frequencies. For this reason, a down-conversion mechanism is employed at the input of the ADC to efficiently monitor the output of a high-frequency CUT. In the proposed approach, an envelope detector (highlighted with a dashed red line in Fig. 21) is used to down-convert the signal from RF to low frequencies. To use the described approach at RF frequencies, the response of the CUT with a two-tone test signal is fed to the envelope detector to obtain a low-frequency envelope of the two-tone signal. In order to reduce the spectral leakage in the output spectrum, the envelope signal is sampled at coherent sampling frequency \( f_{\text{sampCoh.}} \) determined from the conventional coherent sampling approach [19]-[21], [60]-[62]. The spectral components of the envelope signal are extracted and used by the

Figure 21. Block diagram of proposed calibration scheme.
calibration scheme to tune the CUT.

3.2.1 Proposed calibration approach

The BIC scheme that was developed is based on IM3 testing, where the IM3 components are extracted from the analysis of envelope response of a standard two-tone test signal and are used to determine the tuning setting for optimum performance. The effectiveness of the BIC depends on the accuracy of the magnitude estimation for the IM3 components, which in turn depends on the availability of the digital hardware resources. Thus, two key elements of the proposed BIC scheme shown in Fig. 21 are the magnitude calculator and digital calibration control.

3.2.1.1 Magnitude calculator

The fast Fourier transform (FFT) is a standard mechanism that is widely used to perform spectral analysis. The FFT algorithm calculates the spectrum of the input signal at certain discrete frequencies called as FFT bins separated by the FFT fundamental frequency known as FFT resolution. The FFT engine produces a complex output, \( z(x,y) \), consisting of N-bit real and imaginary parts represented by \( x \) and \( y \) respectively. The traditional way of calculating the magnitude of a complex number requires a square root operation as defined by equation (30).

\[
\text{Magnitude} \{ z \} = |z| = \sqrt{(x)^2 + (y)^2} \tag{30}
\]

The values of the real and the imaginary parts at the output of the FFT engine are only of secondary importance when the power level of the spectral components is the quantity of interest. In order to determine the power level of the spectral components it is required to calculate the magnitude of each spectral component. To achieve the desired measurement accuracy, the numbers are usually represented with fixed-point or floating-point notation which poses a significant area and power overhead for on-chip estimation of the magnitudes. Therefore, the typical way to extract the power spectrum from the FFT output is to transfer the numbers generated at the FFT output to off-chip resources such as a computer, and to exploit mathematical tools such as Matlab to calculate the power spectrum. However, such an approach would be very inefficient and can place limits on on-chip built-in-calibration (BIC) and built-in-test (BIT) approaches where it is very
critical to obtain an estimation of the spectral characteristics for dynamic tuning of the circuit under test. An alternative way to determine the magnitude of a complex number based on the "alpha max plus beta min" algorithm was adopted, which is defined by the following equation [79]:

\[ \text{Magnitude (} z \text{)} = \alpha \cdot \max(|x|, |y|) + \frac{\beta \cdot \min(|x|, |y|)}{4}, \]  

(31)

where \( \max(|x|, |y|) \) and \( \min(|x|, |y|) \) represents the maximum and the minimum absolute values of real and imaginary part respectively. Equation (31) is a linear approximation of a complex number's magnitude that is simple and can be efficiently implemented with on-chip resources. The absolute value operations are easily realized by dropping sign bits. Both the \( \max(|x|, |y|) \) and \( \min(|x|, |y|) \) can be evaluated with one comparison. However, the two coefficients \( \alpha \) and \( \beta \) have to be chosen for the approximation. The values of \( \alpha \) and \( \beta \) can be iteratively determined, depending on the desired accuracy and other performance parameters such as area, power and the available computational resources. Matlab simulations were performed to select the values of \( \alpha \) and \( \beta \) under consideration of accuracy. During the Matlab simulations, the values of \( \alpha \) and \( \beta \) were randomly generated from a Gaussian distribution with different mean and standard deviation to estimate the error introduced by the approximation. Fig. 22 shows the estimated magnitudes for 10,000 samples superimposed on the actual magnitudes with \( \alpha = 1 \) and \( \beta = 1/4 \). The error introduced by the equation (31) is bounded to approximately 1 dB as evident from Fig.

![Simple Magnitude Approximation Algorithm (abs |Mag|)](image)

Figure 22. Estimated magnitudes superimposed on the actual magnitudes of 10,000 randomly generated sample values.
23. The value of $\alpha = 1$ helps to reduce the count of the N-bit fixed-point/floating-point multiplication, which is why $\alpha = 1$ and $\beta = 1/4$ were chosen here, allowing the binary calculation (divide-by-2) to be easily and efficiently implemented by a bit-shift operation. If more accuracy is required for an application, then $\alpha$ and $\beta$ can be selected accordingly at the expense of increased chip area for the magnitude calculation.

### 3.2.2 Digital calibration control

The digital calibration control block described in the Section 3.1.2.1 is employed in this BIC, for which the flow chart is depicted in Fig. 20.

### 3.2.3 Envelope detector

Mathematically, the envelope response of an amplifier can be calculated as an absolute value of the Hilbert transform of the amplifier’s output, which was used during the modelling of the proposed BIC scheme during the closed-loop simulation performed in Matlab and VHDL-AMS environment.

For the validation of the proposed BIC on the circuit level, the envelope detector [2] presented in Fig. 24 was designed by Chun-hsiang Chang from the AMSIC research.

---

Figure 23. Magnitude approximation error (in $dB$) for the example case with $\alpha = 1$ and $\beta = 1/4$. 

Magnitude Error (dB)

<table>
<thead>
<tr>
<th>Number of iterations</th>
<th>Magnitude (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>0</td>
</tr>
<tr>
<td>3000</td>
<td>0</td>
</tr>
<tr>
<td>4000</td>
<td>0</td>
</tr>
<tr>
<td>5000</td>
<td>0</td>
</tr>
<tr>
<td>6000</td>
<td>0</td>
</tr>
<tr>
<td>7000</td>
<td>0</td>
</tr>
<tr>
<td>8000</td>
<td>0</td>
</tr>
<tr>
<td>9000</td>
<td>0</td>
</tr>
<tr>
<td>10000</td>
<td>0</td>
</tr>
</tbody>
</table>

Magnitude (dB)
laboratory at Northeastern University. The envelope detector was designed using a vertical bipolar (NPN) transistor in a standard 0.13 $\mu m$ CMOS technology.

### 3.2.4 Low-frequency circuits under test

At low operating frequencies ($< 100 MHz$) the availability of efficient ADC and FFT resources improves the feasibility and effectiveness of standard two-tone testing. Hence, a two-tone test signal consisting of two closely spaced test tones as shown in Fig. 25 can be applied at the input of the CUT while directly capturing its output (bypassing the envelope detector in Fig. 21) for direct processing with the ADC and FFT engine. The magnitude calculator is then employed at the output of the FFT engine to determine the magnitudes of the spectral components, particularly the magnitude of IM3 components. According to the process outlined in Section 3.1.2.1, the output of the magnitude

![Figure 24. Envelope detector in standard 0.13 $\mu m$ CMOS technology [2].](image-url)

![Figure 25. At low operating frequencies the magnitude of the IM3 components can be directly extracted from the response of the CUT using an ADC and FFT engine.](image-url)
calculator (CuIM3) is then fed to the digital calibration control block where it is compared with the value of the register BestIM3. The proposed BIC scheme will iteratively evaluate the IM3 for each setting of the capacitor bank in the LNA, and fine tune the LNA to select the best available setting.

### 3.2.5 High-frequency circuits under test

At radio frequencies, performing traditional two-tone testing with desired accuracy becomes technically challenging and computationally expensive due to the limitation incurred by the performance of ADC and the required DSP circuitry. Alternatively, it has been shown in [77] that the performance parameter of a CUT can be extracted by adding an envelope detector at the CUT output. As shown in Fig. 21, the output of the CUT in the proposed BIC scheme is acquired by an envelope detector that extracts a low-frequency envelope of the response to the test stimulus. The envelope is then processed by an ADC, FFT engine and digital processing circuits to estimate the magnitude of the spectral components. The extracted IM3 components can be digitally evaluated to control calibration of a CUT operating with a high RF frequency.

To gain insights into the envelope analysis approach, assume that a standard two-tone test signal \( x(t) \) composed of two equal-magnitude tones at closely spaced frequencies is applied at the input of the CUT as visualized in Fig. 26 such that:

\[
\begin{align*}
\omega_1 &= \omega_0 - \omega_b/2 \\
\omega_2 &= \omega_0 + \omega_b/2
\end{align*}
\]

![Figure 26. Estimation of high-frequency IM3 components from the low-frequency envelope response of a two-tone signal.](image)

61
\[ x(t) = A \cdot \sin((\omega_0 - \frac{\omega_b}{2}) \cdot t) + A \cdot \sin((\omega_0 + \frac{\omega_b}{2}) \cdot t), \]  

(32)

where \( A \) is the magnitude of the two tones and \( \omega_b \) is the separation between the tones. The corresponding output signal is:

\[ y(t) \approx a_0 + a_1 \cdot (x(t)) + a_2 \cdot (x(t))^2 + a_3 \cdot (x(t))^3 \]  

(33)

Considering only the first-order and the third-order components of equation (33), the output response \( y(t) \) of the CUT can be written as:

\[ y(t) = B_1 \cdot \sin((\omega_0 - \frac{\omega_b}{2}) \cdot t) + B_1 \cdot \sin((\omega_0 + \frac{\omega_b}{2}) \cdot t) \]

\[ + B_2 \cdot \sin((\omega_0 - \frac{3\omega_b}{2}) \cdot t) + B_2 \cdot \sin((\omega_0 + \frac{3\omega_b}{2}) \cdot t) \]  

(34)

where \( B_1 \) and \( B_2 \) are the magnitudes of the fundamental components and the IM3 components respectively. The envelope response \( r(t) \) of the signal \( y(t) \) can be defined as [77]:

\[ r(t) = \left| 2 \cdot B_1 \cdot \sin\left(\frac{\omega_0 \cdot t}{2}\right) + 2 \cdot B_2 \cdot \sin\left(\frac{3\cdot \omega_0 \cdot t}{2}\right) \right| \]  

(35)

Signal \( r(t) \) is a periodic signal with period \( T_b = 2\pi/\omega_b \) and can be expressed in its Fourier series as:

\[ r(t) = C_0 + \sum D_k \cdot \sin(k \cdot \omega_b \cdot t) + C_k \cdot \cos(k \cdot \omega_b \cdot t), \]  

(36)

where the Fourier series coefficients are defined as:

\[ D_k = 0 \]  

(37)

\[ C_0 = \frac{4}{\pi} (B_1 - \frac{B_2}{3}) \]  

(38)

\[ C_k = \frac{8}{\pi} \left( \frac{B_1 \cdot (-1)^{k+1}}{4 \cdot k^2 - 1} + \frac{3 \cdot B_2 \cdot (-1)^k}{4 \cdot k^2 - 9} \right) \quad k = 1, 2, 3, \ldots \]  

(39)

Coefficients \( C_0 \) and \( C_k \) can be extracted from the spectrum of the envelope response of the two-tone signal. Solving equation (39) for \( k = 1, 2 \) and 3, the coefficients \( C_k \) can be defined in terms of the magnitude of fundamental components (\( B_1 \)) and the magnitude of IM3 components (\( B_2 \)) as:
\[ C_1 = \frac{8}{\pi} \left( \frac{B_1}{3} + \frac{3 \cdot B_2}{5} \right) \]  
(40)

\[ C_2 = \frac{8}{\pi} \left( -\frac{B_1}{15} + \frac{3 \cdot B_2}{7} \right) \]  
(41)

\[ C_3 = \frac{8}{\pi} \left( \frac{B_1}{35} - \frac{3 \cdot B_2}{27} \right) \]  
(42)

Equations in (40)-(42) reveal that the coefficients \( C_1 \), \( C_2 \) and \( C_3 \) in the spectrum of the envelope reflect the power levels of the fundamental components \( B_1 \) and the IM3 components \( B_2 \) at the output of the CUT. From (40)-(42), \( B_1 \) and \( B_2 \) can be written in terms of the coefficients \( C_1 \) and \( C_3 \) as:

\[ B_1 = \frac{1575 \cdot \pi \cdot C_1 + 8505 \cdot \pi \cdot C_3}{6144} \]  
(43)

\[ B_2 = \frac{405 \cdot \pi \cdot C_1 - 4725 \cdot \pi \cdot C_3}{6144} \]  
(44)

The above envelope analysis approach has been demonstrated in theory and with simulations in [77]. In the remainder of this chapter, some design aspects for the practical realization of this approach will described in conjunction with the self-calibration technique. Equation (43) was evaluated by applying a standard two-tone test signal with input frequencies of 1 GHz and 1.001 GHz and a power level of -40 dBm to a CUT modeled by equation (33). Values of parameters \( a_1 \) and \( a_3 \) were varied (keeping \( a_0 = 0 \) and \( a_2 = 0 \) to set the DC component and second-order IM components to zero) such that only the IM3 components ranging from 10 dBc to 80 dBc are present in the CUT’s output response \( y(t) \). The envelope \( r(t) \) of signal \( y(t) \) was extracted using an ideal envelope detector (described in Section 3.2.3), and then applied to a 16-bit ADC. A \( 2^{16} \)-point FFT was used to evaluate the spectral response of \( r(t) \). In this first simulation, a 16-bit ADC and \( 2^{16} \)-point FFT were chosen to keep quantization errors small during the

<table>
<thead>
<tr>
<th>IM3 (dBc)</th>
<th>Input power level: -40 dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied</td>
<td>80</td>
</tr>
<tr>
<td>Extracted</td>
<td>79.9889</td>
</tr>
</tbody>
</table>

Table 10. Simulated IM3 (in dBc) from a two-tone test with \( f_1 = 1 \) GHz, \( f_2 = 1.001 \) GHz using a 16-bit ADC and \( 2^{16} \)-point FFT
estimation of the spectral components, providing a theoretical reference for assessing the impacts of changes in ADC resolution and FFT length on the accuracy of measurement. Coefficients $C_1$ and $C_3$ were measured from the spectrum of the envelope $r(t)$ and equation (44) was used to estimate the magnitude of the IM3 components. The results are summarized in Table 10, showing that IM3 components with an accuracy of around $\pm 0.05 \text{ dB}$ can be estimated from the spectral coefficients $C_1$ and $C_3$ of the low-frequency envelope based on equation (44).

In practice, the use of a 16-bit ADC and $2^{16}$-point FFT combination is not suitable for BIC and BIST applications due to chip area overhead constraints. Furthermore, the accuracy of measurement is affected by the error introduced due to limited resolution of the ADC as well as due to the presence of spectral leakage in the output spectrum introduced by the limited FFT length. Note that the envelope response of a two-tone test signal represented by equation (32) is a low-frequency signal with spectral components at integer multiples of frequency $\omega_0$. Thus, signal $r(t)$ can be sampled at a rate of $f_{\text{sampCoh}}$ based on equation (27) to avoid spectral leakage. For example, consider a two-tone test signal $s(t)$ composed of two signals at frequencies $f_1 = 1 \text{ GHz}$ and $f_2 = 1.001 \text{ GHz}$. The envelope response $r(t)$ of the signal $s(t)$ is a low-frequency signal with a fundamental frequency component ($\Delta f = f_2 - f_1$) of 0.001 GHz. Based on equation (27), with $N_{\text{cycle}} = '1'$ and for a given FFT length ($NFFT$), the envelope signal $r(t)$ can be sampled at a rate of $f_{\text{sampCoh.}} = \Delta f \cdot NFFT$ to avoid spectral leakage, thereby increasing the accuracy in the measurement of coefficients $C_0$, $C_1$, $C_2$ and $C_3$. 
To determine the impact of limited FFT length on the accuracy of the IM3 component estimation, the envelope response of the two-tone signal with input frequencies of 1.000 GHz and 1.001 GHz and an input power level of -30 dBm was sampled at a rate of \( f_{\text{sampCoh.}} = \Delta f \cdot NFFT \), and analyzed with FFT length \((NFFT)\) varying from \(2^4\) points to \(2^{16}\) points. Practical values of IM3 components with -30 dBm input power level typically fall in the range of 30 dBc to 60 dBc based on the results in Table 8. Table 11 summarizes the IM3 level in a range of 30 dBc to 60 dBc estimated from the envelope response \(r(t)\) represented by equation (35) of the two-tone signal. With a large FFT length, for instance \(NFFT = 2^{11}\), the absolute value of IM3 can be estimated with an error of less than 1 dB. However, for BIC purposes the ability to accurately differentiate between the IM3 levels and to find the desired IM3 level (for example minimum IM3) is more important. Furthermore, an increase in FFT length will result in a significant increase in area overhead and thereby reduce the practicality of an FFT implementation for on-chip testing applications. It is therefore recommended to choose the FFT length based on the desired accuracy.

<table>
<thead>
<tr>
<th>(NFFT)</th>
<th>(30)</th>
<th>(35)</th>
<th>(40)</th>
<th>(45)</th>
<th>(50)</th>
<th>(55)</th>
<th>(60)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2^4)</td>
<td>36.72</td>
<td>83.76</td>
<td>41.36</td>
<td>37.85</td>
<td>35.82</td>
<td>35.35</td>
<td>34.54</td>
</tr>
<tr>
<td>(2^5)</td>
<td>31.29</td>
<td>36.94</td>
<td>44.75</td>
<td>55.73</td>
<td>54.65</td>
<td>50.64</td>
<td>50.67</td>
</tr>
<tr>
<td>(2^6)</td>
<td>30.49</td>
<td>35.30</td>
<td>40.77</td>
<td>46.97</td>
<td>54.97</td>
<td>62.05</td>
<td>73.57</td>
</tr>
<tr>
<td>(2^7)</td>
<td>30.13</td>
<td>35.20</td>
<td>40.22</td>
<td>45.07</td>
<td>50.18</td>
<td>56.24</td>
<td>59.20</td>
</tr>
<tr>
<td>(2^8)</td>
<td>29.97</td>
<td>34.93</td>
<td>40.16</td>
<td>44.83</td>
<td>50.01</td>
<td>54.94</td>
<td>59.64</td>
</tr>
<tr>
<td>(2^9)</td>
<td>29.98</td>
<td>35.05</td>
<td>40.07</td>
<td>44.83</td>
<td>50.10</td>
<td>55.04</td>
<td>58.95</td>
</tr>
<tr>
<td>(2^{10})</td>
<td>29.95</td>
<td>35.01</td>
<td>40.06</td>
<td>45.01</td>
<td>50.15</td>
<td>54.80</td>
<td>59.16</td>
</tr>
<tr>
<td>(2^{11})</td>
<td>29.97</td>
<td>34.99</td>
<td>40.03</td>
<td>45.01</td>
<td>50.15</td>
<td>55.00</td>
<td>59.08</td>
</tr>
<tr>
<td>(2^{16})</td>
<td>29.98</td>
<td>34.99</td>
<td>39.99</td>
<td>44.97</td>
<td>49.99</td>
<td>55.02</td>
<td>59.86</td>
</tr>
</tbody>
</table>

Table 11. Simulated IM3 (in dBc) from a two-tone test with \(f_1 = 1\) GHz, \(f_2 = 1.001\) GHz using a 10-bit ADC and FFT lengths varying from \(2^4\) to \(2^{16}\) points.
3.2.6 Simulation results

This section demonstrates the application of the proposed calibration scheme to find the best tuning setting with the LNA linearity parameters presented in Section 3.1.1 that results in minimum IM3 components. For the purpose of simulation, a mathematical model of the proposed BIC scheme presented in Fig. 21 was created in Matlab. The amplifier in Fig. 21 was modeled using equation (33) with the coefficients $\alpha_0$, $\alpha_1$, $\alpha_2$, and $\alpha_3$ [78] that corresponds to the IM3 levels summarized in Table 8. The envelope response of the amplifier was computed as an absolute value of the Hilbert transform of the amplifier’s response to the test signals. The complex output of the FFT engine was converted into absolute magnitude by the magnitude estimation represented by equation (31). Note that for the test cases with RF test tones and an envelope detector, the output of the magnitude calculator represents the coefficients $C_1$, $C_2$, and $C_3$ and the actual IM3 components, i.e., $B_1$ and $B_2$ were estimated by a Matlab subroutine that implements equations (43) and (44). Whereas, for low-frequency test signals without envelope detector, the output of the magnitude calculator represents the actual IM3 components of the CUT. The computed IM3 components are then fed to the control unit implemented in Matlab with calibration control flow as shown in Fig. 20. Matlab simulation results are discussed next for operating frequencies ranging from megahertz (low-frequency amplifier) to gigahertz (RF LNA).

3.2.6.1 Case 1: Low-frequency CUT

Consider an input test signal $x(t)$ composed of two tones with equal amplitude of $A = 0.01 \, V$ (input power level of -30 dBm) and at frequencies $f_1 = 3 \, MHz$ and $f_2 = f_1 + \Delta f = 3 MHz + 1 MHz = 4 MHz$:

$$x(t) = 10^{-2} \cdot \sin(2\pi \cdot f_1 \cdot t) + 10^{-2} \cdot \sin(2\pi \cdot f_2 \cdot t)$$  \hspace{1cm} (45)

For input frequencies $f_1 = 3 MHz$ and $f_2 = 4 MHz$, the IM3 components are at frequencies $f_{IM3-1} = 2f_1 - f_2 = 2 MHz$ and $f_{IM3-2} = 2f_2 - f_1 = 5 MHz$. Signal $x(t)$ was applied to a CUT modeled with equation (33) and signal $y(t)$ was obtained with the IM3 components specified in Table 8:
where $A_{IM3}$ is the amplitude of the IM3 components that varies for different tuning settings of the CUT. For low-frequency test tones, the IM3 components at the output of CUT can be accurately extracted by directly applying the signal $y(t)$ to an efficient ADC prior to processing with the FFT. In this simulation with $NFFT = 16, N_{cycle} = 1$, the signal $y(t)$ was sampled at a rate of $f_{sampCoh} = \Delta f \cdot NFFT = 1 \text{ MHz} \cdot 16 = 16 \text{ MHz}$ based on equation (27). The magnitudes of IM3 components were estimated by the magnitude calculator block, whose output was processed by the digital calibration control block. The closed loop was iteratively executed for each tuning setting to determine the setting that results in the lowest IM3 level. The results of each iteration are summarized in Table 12, revealing the final value of BestCode as ‘011’, which matches the control code with lowest IM3 level.

### 3.2.6.2 Case 2: High-frequency CUT

Consider the signal $x(t)$ defined by equation (45) but with frequencies $f_1 = 2.4 \text{ GHz}$ and $f_2 = f_1 + \Delta f = 2.4 \text{ GHz} + 2 \text{ MHz} = 2.402 \text{ GHz}$. For input frequencies $f_1 = 2.4 \text{ GHz}$ and $f_2 = 2.402 \text{ GHz}$, the IM3 components are at frequencies $f_{IM3-1} = 2f_1 - f_2 = 2.398 \text{ GHz}$ and $f_{IM3-2} = 2f_2 - f_1 = 2.404 \text{ GHz}$. As in the low-frequency case, signal $x(t)$ was applied.
to a CUT modeled with equation (33) and signal $y(t)$ was obtained with the IM3 components specified in Table 8:

$$y(t) = A \cdot \sin(2\pi \cdot f_1 \cdot t) + A \cdot \sin(2\pi \cdot 2f_2 \cdot t) + A_{IM3} \cdot \sin(2\pi \cdot f_{IM3-1} \cdot t) + A_{IM3} \cdot \sin(2\pi \cdot f_{IM3-2} \cdot t)$$ (47)

The high-frequency output signal $y(t)$ was processed with an ideal envelope detector (outlined in Section 3.2.3) to obtain the frequency envelope signal $r(t)$. Here, the difference of the input frequencies was $\Delta f = f_1 - f_2$ is 2 MHz, and the spectral components of the low-frequency envelope $r(t)$ are at integer multiples of $\Delta f$. An FFT of length 64-points was chosen based on the results in Table 11. From equation (27), signal $r(t)$ was sampled at a rate of $f_{\text{samplCoh}} = \Delta f \cdot NFFT = 2 \text{MHz} \cdot 64 = 128 \text{MHz}$ and applied to a 10-bit ADC modeled in Matlab. Coefficients $C_1$ and $C_3$ were obtained from the output spectrum of envelope signal $r(t)$, which were then used to estimate the IM3 level using equation (44) in the digital calibration control block implemented as a subroutine in Matlab. Using the estimated IM3 levels during the simulation, the CUT was automatically tuned to identify the lowest IM3 component level among the different settings (sets of nonlinearity parameters). Table 13 contains the results obtained after each iteration to determine the best control code for the CUT that yields the lowest IM3 level. Despite of

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Code</th>
<th>Count1</th>
<th>Count2</th>
<th>Actual IM3 (dBc)</th>
<th>Estimated IM3 (dBc)</th>
<th>Best IM3 (dBc)</th>
<th>BestCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>000</td>
<td>000</td>
<td>111</td>
<td>-40.6</td>
<td>-41.62</td>
<td>-41.62</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>001</td>
<td>001</td>
<td>111</td>
<td>-43.8</td>
<td>-45.75</td>
<td>-45.75</td>
<td>001</td>
</tr>
<tr>
<td>3</td>
<td>010</td>
<td>010</td>
<td>111</td>
<td>-47.4</td>
<td>-48.97</td>
<td>-48.97</td>
<td>010</td>
</tr>
<tr>
<td>4</td>
<td>011</td>
<td>011</td>
<td>111</td>
<td>-56.0</td>
<td>-69.23</td>
<td>-69.23</td>
<td>011</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>100</td>
<td>111</td>
<td>-47.6</td>
<td>-49.06</td>
<td>-69.23</td>
<td>011</td>
</tr>
<tr>
<td>6</td>
<td>101</td>
<td>101</td>
<td>111</td>
<td>-41.0</td>
<td>-42.39</td>
<td>-69.23</td>
<td>011</td>
</tr>
<tr>
<td>7</td>
<td>110</td>
<td>110</td>
<td>111</td>
<td>-38.2</td>
<td>-38.61</td>
<td>-69.23</td>
<td>011</td>
</tr>
<tr>
<td>8</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>-35.8</td>
<td>-36.04</td>
<td><strong>-69.23</strong></td>
<td><strong>011</strong></td>
</tr>
</tbody>
</table>
the estimation errors, the estimated IM3 values in Table 13 follow the same trend as the actual IM3 values. Hence, the technique allows to identify the tuning code that achieves best IM3 performance from the value of BestCode in Table 13 that captures the correct control code. The capability to extract the differences in power levels makes this approach suitable for BIC and BIST.
4. Simulation and Verification Techniques

4.1 An optimization platform for digital predistortion of power amplifiers

The operation of a power amplifiers (PA) is most linear at low output power levels and least linear at high power levels. On the other hand, PAs are more efficient with high output power levels and less efficient at lower power levels. One of the approaches to improve the linearity of PAs, backing off the output to low power levels, has the undesirable effect of lowering the efficiency [10]. Another approach involves the implementation of a linearization technique [11] such as digital predistortion (DPD) in the RF transmitter [12]. A DPD uses digital signal processing to predistort the baseband signal before up-converting it to RF frequencies. Digital predistortion (DPD) is one of the most favorable linearization techniques employed in today’s RF transceiver systems [10]. In conventional DPD implementations, a predistorter acts on the input of the PA as visualized in Fig. 27(a). The predistorter models the inverse of the gain, phase, and distortion of the operating PA (or the device under test - DUT) such that the overall system (DPD + DUT) response is linear as depicted in Fig. 27(b). Optimum DPD performance requires the predistorter to be an exact inverse of the PA, which is difficult to achieve because the characteristics of the PA depend on parameters such as class, technology, and the operating conditions [10]-[12]. In addition, the linearity requirement (typically measured in terms of ACLR, which is defined as the amount of the transmit power relative to the carrier that leaks into the adjacent channel) of a DUT varies for different transmission standards (e.g., WCDMA, LTE). For example, in a system with wideband signals, memory effects in the DUT are more pronounced [12]. Thus, larger look-up tables (LUTs) are desirable, which can result in an overdesigned system for narrow-band signals. Furthermore, post-fabrication testing of a system (DUT + DPD) could result in several iterations of design and test due to performance limitations of the system. In general, it is advantageous to devise universal adaptive design-and-test platforms that allow evaluation and optimization of system performance for practical test conditions, and that give flexibility to incorporate additional test constraints during the design phase [32]-[35]. A hardware-software co-design and test method that provides
data for both, DPD and PA designers, during system optimizations is introduced in this chapter.

4.1.1 Digital predistortion (DPD) technique

4.1.1.1 Conventional DPD system architecture

The goal of DPD is to create a predistorted sequence \( x \) of the input signal sequence \( w \) such that the output \( y \) of the PA when processing the predistorted sequence \( x \) (after up-conversion to RF) results in a linearized amplification of the original input sequence \( w \). A conventional DPD approach, as shown in Fig. 27(c), incorporates at least two components. The first is an adaptation algorithm, or a ‘postdistorter’, which models the inverse of the PA. Generally, a memory polynomial (MP) is used as a postdistorter to map the baseband output of the PA to the baseband input of the PA, and the coefficients of the MP that minimize the difference between the baseband input and the baseband output of the PA are derived in the process. These coefficients are then fed to the second
component (a ‘predistorter’) that is an exact copy of the postdistorter for predistortion of the original baseband input signal.

The input-output relationship of a causal, time-invariant nonlinear system can be represented by Volterra series with infinite terms. However, due to computational restrictions, a truncated version of Volterra series is usually adopted to define a finite-order nonlinear system such that [80]:

\[ y(n) = \sum_{p=1}^{P} \sum_{i_1=0}^{M} \cdots \sum_{i_p=0}^{M} h_p(i_1, \cdots, i_p) \prod_{j=1}^{p} x(n - i_j), \]

where \( x(n) \) and \( y(n) \) are the input and output of the nonlinear system with a discrete time step \( n \), respectively. Terms ‘P’ and ‘M’ represent the nonlinearity order and memory length, whereas \( h_p(\cdot) \) is known as \( p \)-th order Volterra kernel. Alternatively, the postdistorter function can also be realized using a derivative of the Volterra series like the generalized memory polynomial (GMP), which maps an input ‘\( x \)’ to an output ‘\( y \)’ of the PA inverse model [80]:

\[ y(t) = \sum_{i,j,k} a_{i,j,k} \cdot |x(t-i)|^k \cdot x(t-j), \]

where \( a_{i,j,k} \) are complex coefficients, ‘\( k \)’ is the nonlinearity order (O), and ‘\( i \)’ is the memory depth (or taps T) of the PA model. The terms of equation (49) for which ‘\( i \)’ differs from ‘\( j \)’ are referred to as cross terms (CT). The postdistorter is incorporated in the feedback path, and is often implemented on a microprocessor. The predistorter function evaluated for different values of the input signal amplitude ‘\( x \)’ is implemented in the form of LUTs. The adaptation algorithm determines the values of the LUT coefficients \( (a_{i,j,k}) \) by comparing the output signal of the DUT with the delayed version of the input signal. The predistorter in the main transmit path actually predistorts the input sequence before applying it to the PA. For a fast response time, the predistorter is often implemented on an FPGA. Conventional DPD implementations have several drawbacks. First, the memory polynomial (MP) only provides an approximation of the inverse PA model. As such, the best estimate post-inverse is not necessarily the best estimate of the pre-inverse. Increasing the number of terms in the model equation always improves the quality of the post-inverse, but can lead to decreased performance when used as a pre-inverse. Another drawback is that the coefficients are computed for a particular gain
level, and would not hold for a different gain. Alternatively, in a direct learning DPD (a DPD variant) as shown in Fig. 27(d), the baseband equivalent PA model is estimated from the input and the output of the DUT. The inverse of this PA model is then used as a predistorter [80]-[81].

4.1.1.2 Analog Devices (ADI) DPD

In contrast to conventional DPD algorithms using fixed PA models, the DPD algorithm that was developed by collaborators at Analog Devices, Inc. learns the PA transfer function and directly computes the exact inverse of this transfer function on the fly. The ADI DPD algorithm is built on a Bayesian form of direct learning (as in Fig. 27(d)) using PA models that have known inverses. In the first phase, the algorithm trains the PA model so that the predicted error between the real PA output and the modeled PA output is as small as possible. In a second phase, the algorithm uses the PA model as well as the desired output to compute what the predistorter output should be. This computation provides the best approximate pre-inverse (as opposed to post-inverse in conventional DPD) to the internal PA.

4.1.2 Performance and computational complexity trade-offs

4.1.2.1 PA model and test signal

PA nonlinearity depends on variety of parameters. Therefore, the selection of the predistorter model described by a model equation such as equation (49) should be made based on the analysis of both the signal and the DUT. For example, the authors in [13] evaluate the performance of DPDs with different model equations to characterize PA nonlinearity. Moreover, the accuracy of the predistorter model also depends on T, CT, and O (defined in Section 4.1.1.1). For instance, low-power PAs or narrow-band signals exhibit weak memory effects for which the predistorter model of equation (49) with small memory depth is sufficient; whereas high-power PAs or wideband signals have strong memory effects, requiring higher nonlinearity order and memory depth [10]-[12].

4.1.2.2 Predistorter (look-up table size and computational precision)

LUT-based predistorters are extensively used in DPD systems to implement the inverse of the PA model defined by the memory polynomial or generalized memory polynomial such as equation (49). The model equation is evaluated for different ampli-
tudes of the input signal and the results are stored in a LUT [14]. During the actual linearization, the sampled input signal indexes the LUT to fetch the corresponding pre-computed value of the model equation. Ideally, the LUT should contain the entries for all possible values of the input signal sample, which becomes impractical due to memory limitations. A direct consequence of a larger LUT size is slow convergence as reported in [14]. Also, based on equation (49), computation of the predistorted sequence involves arithmetic operations such as multiplication and addition. These arithmetic operations produce results that can have a higher number of bits than the operands. The computed results are either truncated or rounded off, causing a loss of information. Therefore, the amount of resources (area, power, and time) consumed by a computation engine depends on the internal representation of the data. Thus, proper selection of the data format is critical.

4.1.2.3 Adaptation algorithm complexity, processing time and convergence time

The coefficients of equation (49) can be estimated using any flavor of the least-square algorithm. Commonly used adaptation algorithms are least squares (LS) and recursive least squares (RLS) for example. Each algorithm offers different computational complexity and convergence time [15]. The adaptation algorithm in the ADI DPD makes use of the least-square (LS) as well as the recursive least-square (RLS) algorithms in a proprietary multi-step process, and it achieves RLS-like results with higher computational efficiency. Real-time processing requires coefficients to be updated sample-by-sample, making proper selection and optimization of the adaptation algorithm essential for the minimization of the convergence time.

4.1.2.4 Number of coefficients and pruning algorithm

The coefficients $a_{i,j,k}$ of the model equation (49) computed during the adaptation process are passed to the predistorter where they are multiplied by the sampled input baseband signal to generate the predistorted input sequence. The number of coefficients governs the computation requirements of both the postdistorter and the predistorter. Therefore, optimization of the number of coefficients is essential for an efficient DPD system. A pruning approach that iteratively reduces the number of coefficients by a pruning factor 'K' (defined as the number of removed coefficients) was developed in this research. Two lists, 'CurList' and 'PreList', are used in the algorithm. CurList holds the
current list of coefficients to predistort the input signal, whereas PreList holds the coefficients from the previous iteration. The key steps for pruning the coefficients are as follows:

Step I: The optimum values for T, CT and O of the model equation are determined based on the DPD architecture optimization process described later (in Section 4.1.4).

Step II: The original input baseband signal is up-converted to radio frequencies and applied to the PA. Since at this point the response of the PA is not known, the initial values of the coefficients in both lists are set to '1' such that the input baseband signal is not predistorted.

Step III: The output of the PA is measured to compute and update all coefficients in the CurList through the adaptation process. Then, the CurList is copied to the PreList.

Step IV: Using CurList (with an excessive number of coefficients), the input baseband signal is predistorted and applied to the PA. The ACLR performance metric is measured at the output of the PA. Since a complete list of coefficients is used, the measured value represents the best possible value and can be used as accurate reference.

Step V: Based on the chosen pruning factor (K), the K coefficients with the smallest error reduction impacts are removed from the CurList. The number of pruned coefficients can be decided by the designer.

Step VI: Using the pruned CurList, the input baseband signal is predistorted and applied to the PA for measurement of the ACLR at the output of the PA. If it meets the desired requirements, the coefficients in the CurList are re-computed and updated through the adaptation process, and the CurList is copied to the PreList. Otherwise, the PreList becomes the CurList and the optimization flow is stopped. (Steps V and VI are repeated until the target ACLR specification is met.)

4.1.3 Hardware-software test platform

Fig. 28 depicts the test platform that was developed to optimize DPD algorithms. It consists of an evaluation board with a dual-channel RF integrated transceiver test chip mounted on it. Each RF integrated transceiver test chip channel consists of a transmitter and an observation path receiver. The evaluation board is interfaced to a board with a Virtex 6 FPGA through a high-speed JESD204B link over an FMC connector. The
FPGA board is connected to a PC through a USB interface and is used as a buffer to store the signal transmitted to and received from the transceiver. The ADI DPD adaptation algorithm was implemented in Matlab running on a PC. Note that the test platform employs a Matlab-based DPD algorithm to explore the trade-offs described in Section 4.1.2 for optimization of FPGA-based DPD systems.

The baseband signal generated in Matlab is stored in RAM on the FPGA board. The buffered baseband signal from the FPGA board is then sent to the RF integrated transceiver test chip board where it is up-converted to the appropriate frequency band. The transmitter output is amplified through a series of highly-linear wideband pre-amplifier stages to drive the PA with the desired power level. The RF output of the PA is attenuated before being routed to a spectrum analyzer and to the observation receiver path on the RF integrated transceiver chip evaluation board via a 3-dB splitter. The received RF input signal is down-converted to baseband frequencies, digitized, and transferred to the FPGA. Note that in theory, an N-bit converter allows a signal-to-noise ratio (SNR) of ‘6.02×N + 1.76 dB’ and ‘6.02×N - 4.02 dB’ in the best case (without linearity error) and in the worst case (with ½ LSB linearity error), respectively. For example, a setup with 12-bit converters can allow linearization of the DUT to a noise floor level of approxi-
mately 68 dBc. However, to leave margin and to ensure high performance, a 14-bit DAC and a 14-bit ADC were integrated into the presented system. Furthermore, it is advisable to sample the baseband input signal of bandwidth ‘x’ with a sampling rate that is at least five times higher than the original input signal bandwidth (i.e., ‘\( f_s = 5 \cdot x \)’). This helps to capture the nonlinear distortion components around the fundamental signal. For example, with this approach a 20 MHz wide LTE signal implies a baseband sampling rate of at least 100 MS/s. Thus, to accommodate up to 40 MHz bandwidth signal, a DAC/ADC sampling rate of 245.76 MS/s was selected for the presented system. This baseband received signal is retrieved over the USB link from the FPGA board by the DPD algorithm running on the PC, where it is used for adaptation and predistortion of the signal. The predistorted signal is stored on the FPGA before the transmission.

### 4.1.4 DPD optimization steps

The following steps allow fine-tuning the DPD as part of the proposed test and design optimization methodology:

**Step I:** A PA model with overdesigned parameters: T, CT, O, and LUT size was designed before adopting it in the test platform to minimize computation resources.

**Step II:** DPD architecture optimization: The overdesigned PA model was evaluated with the test platform for different values of T, CT, and O to minimize the nonlinearity of the PA output. Large LUT size and a high-precision data format are essential at this stage to avoid a loss of accuracy.

**Step III:** Predistorter optimization: The accuracy of the DPD heavily depends on the LUT size and LUT data format, which can be optimized prior to selecting an adaptation process and reducing the number of coefficients in the computations.

**Step IV:** Adaptation process selection: The test platform permits to program different adaptation processes in Matlab and characterize a specific PA for the selection of a process with favorable LUT complexity and convergence time trade-offs based on measurements.

**Step V:** Coefficient pruning: Coefficients with smaller magnitude have the least impact on the overall performance with DPD. The smallest number of coefficients that meets the specification requirement can be determined based on measurements.
4.1.5 Experimental results

Two off-the-shelf PAs having the relevant characteristics listed in Table 14 were tested with the optimization approach described in Section 4.1.4 using the ADI DPD algorithm with the parameters defined in Section 4.1.1.

4.1.5.1 DPD architecture optimization

The two PAs were evaluated for different values of taps (T), number of cross terms (CT) and order (O) of the model equation. The results are summarized in Table 14. With the higher values of (T = M, CT = 2 and O = 9), the measured ACLRs after DPD are approximately -58 dBc and -57 dBc for PA#1 and PA#2 respectively, which represents the best values that can be achieved but with a high computational cost overhead. Even though the number of taps T = M are not explicitly specified, the following results and discussions demonstrate the relative reduction of complexity that can be achieved with the presented test platform. Appropriate combinations of T, CT and O should be selected based on the ACLR requirement. In the discussed cases, targeted ACLRs of -57 dBc and -56 dBc were chosen for PA#1 and PA#2 respectively as demonstration examples with a combination of T = M - 1, CT = 1, and O = 7.

<table>
<thead>
<tr>
<th>PA #</th>
<th>Technology</th>
<th>Test Signal; Center Frequency; Peak-to-Average Ratio (PAR)</th>
<th>Average Output Power</th>
<th>Output 1-dB Compression Point</th>
<th>ACLR (dBc) w/o DPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Doherty LDMOS</td>
<td>LTE: 2c x 20 MHz; 2.14 GHz; 7.5 dB</td>
<td>8W</td>
<td>46 dBm</td>
<td>~ -33</td>
</tr>
<tr>
<td>2</td>
<td>AB GaN HEMT</td>
<td>2W</td>
<td>2W</td>
<td>42 dBm</td>
<td>~ -33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>CT</th>
<th>ACLR (dBc) with DPD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PA#1</td>
</tr>
<tr>
<td>O = 7</td>
<td>-58.1</td>
<td>-58.1</td>
</tr>
<tr>
<td>O = 8</td>
<td>-58.1</td>
<td>-58.1</td>
</tr>
<tr>
<td>O = 9</td>
<td>-58.1</td>
<td>-58.1</td>
</tr>
<tr>
<td>O = 7</td>
<td>-56.9</td>
<td>-56.9</td>
</tr>
<tr>
<td>O = 8</td>
<td>-57.0</td>
<td>-57.0</td>
</tr>
<tr>
<td>O = 9</td>
<td>-56.9</td>
<td>-56.9</td>
</tr>
</tbody>
</table>

Table 14. Power amplifier specifications, test signals and measured ACLR
4.1.5.2 Predistorter optimization

Fig. 29 shows absolute values of the measured ACLR with varying LUT size and LUT fraction bit-width for a fixed value of the LUT integer bit-width. Higher LUT parameters provide better results due to improved accuracy. The choice of LUT size = 2048 and LUT fraction bit-width = 15 gives the best results (ACLRs of approximately -58 dBc and -57 dBc for the two PAs). However, the targeted ACLR values (-57 dBc for PA#1 and -56 dBc for PA#2) can be achieved with a LUT size of 128 and a LUT fraction bit-width of 10, which represents the optimum solution when the goal is to minimize computation costs. Therefore, LUT size = 128 and LUT fraction bit-width = 10 were chosen for further analysis in the next subsections.

Figure 29. |ACLR| vs. LUT size and LUT fraction bit-width: (a) PA#1, (b) PA#2.
4.1.5.3 Adaptation process selection

Two different adaptation processes were evaluated with different LUT complexity, which in this context is defined as: LUT complexity (P) = LUT size \(2^n\) where \(n = 4\) to 12 \(\times\) LUT fraction bit-width (varying from 8 to 15).

The measured ACLR over LUT complexity curves are plotted in Fig. 30 for the two PAs with DPD using two different adaptation processes. The targeted ACLR values were achieved with the LUT complexities of \(>256\) and \(>768\) with adaptation process \#1 and \#2, respectively. Although the obvious choice is to select process \#1 over process \#2, the measured ACLR values in Section 4.1.5.2 for LUT complexities from 256 to 768 are lower than the targeted ACLRs. Therefore, a LUT size of 128 (LUT complexity: \(128 \times 10 = 1280\)) and a LUT fraction bit-width of 10 (for a low quantization error) along with adaptation process \#1 was selected.

4.1.5.4 Coefficient pruning

Pruning test results for the two PAs with different LUT sizes and with LUT fraction bit-width of 10 are shown in Fig. 31, where the measured ACLR is plotted against the number of coefficients that were eliminated to minimize the computational resource requirements. Given a total coefficient count of 'C', the first data point in Fig. 31 represents the measured ACLR with 'C-3' coefficients. With a LUT size greater than 128, the
targeted ACLRs were achieved with ‘C-66’ and ‘C-84’ coefficients for the two PAs respectively. Note that pruning of coefficients with a LUT size less than 128 significantly degrades performance (due to inaccuracies introduced into the PA model resulting in an unstable system performance), which supports the selection of 128 LUTs in Section 4.1.5.2. With the presented optimization platform, the effectiveness of the DPD system can be monitored over many iterations and over long durations of time before finalizing the decision to adopt the pruned coefficients. Generally, pruned coefficients under one test condition can be inadequate under a different test condition. For example, the performance difference between PA chips due to manufacturing variations presents a practical problem. A DPD architecture optimized for a PA sample with the best possible linearity profile will likely not deliver the desired performance for another sample of the same PA that exhibits the worst anticipated linearity due to manufacturing variations. When representative samples are available, the platform permits the pruned DPD performance using the worst cases to be verified, ensuring that vital coefficients are not
removed. Similarly, the DPD performance depends on the type of signals that the PA processes. Consequently, real LTE test signals were used in the examples discussed in this dissertation. In summary, the platform allows for rigorous testing and optimization of the integrated system (DUT + DPD) under various test conditions before actually deploying a pruned DPD in a particular application.

### 4.1.6 FPGA implementation results

To verify the practical feasibility of the test platform with different DUT characteristics such as PA class, technology and power level, a variety of PAs were tested with signals close to their 1dB compression points having up to 40 MHz bandwidth and a peak-to-average ratio (PAR) of 7.5 dB. Table 15 summarizes experimental results for two PAs, which were obtained with a DPD algorithm implemented on a Xilinx Virtex-6 FPGA. The FPGA results were measured with DPD implementations having LUT complexities down to 25% (i.e., 0.25·P) of the LUT complexity (P) compared to the Matlab-based solution. Furthermore, the coefficient pruning algorithm outlined in Section 4.1.4 was utilized to optimize the DPD system by reducing the number of coefficients from ‘N’ to ‘N/5’, for which results are also listed in Table 15. The optimized DPD implementation with LUT complexity of ‘0.25·P’ and with ‘N/5’ number of coefficients improves the measured ACLR of the GaAs HBT PA and the GaN SiC Doherty PA by 16 dB and 22 dB respectively, which is evident in Fig. 32 and in Fig. 33.

<table>
<thead>
<tr>
<th>PA Type, 1dB Compression Point</th>
<th>Test Signal (LTE)</th>
<th>ACLR (dBc) without DPD</th>
<th>ACLR (dBc) with DPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>GaAs HBT, 33 dBm</td>
<td>1c x 20 MHz</td>
<td>-41</td>
<td>-61.5</td>
</tr>
<tr>
<td></td>
<td>2c x 20 MHz</td>
<td>-38</td>
<td>-63.0</td>
</tr>
<tr>
<td>GaN SiC Doherty, 47 dBm</td>
<td>1c x 20 MHz</td>
<td>-35</td>
<td>-60.0</td>
</tr>
<tr>
<td></td>
<td>2c x 20 MHz</td>
<td>-33</td>
<td>-60.1</td>
</tr>
</tbody>
</table>

Table 15. Performance comparison of Matlab-based DPD results and measurements of the FPGA-based DPD implementation.
Figure 32. Measured ACLR using the optimized FPGA-based DPD system with LUT complexity of ‘0.25∙P’, with ‘N/2’ number of coefficients and with a 1c x 20 MHz LTE test signal for the GaAs HBT PA.

respectively. An error vector magnitude (EVM) of 7.33% without DPD and 7.03% with DPD was measured for the GaAs HBT PA, and the GaN SiC Doherty PA has a measured EVM of 7.39% without DPD and 7.15% with DPD. The non-optimized DPD implementation with LUT complexity ‘P’ and with ‘N’ number of coefficients utilizes about 500,000 application-specific integrated circuit (ASIC) gate equivalents of a Xilinx Virtex-6 FPGA, having a computation time and power consumption (depending on required number of iterations) in the ranges of 100-400 ms and 100-400 mW. Note that one Xilinx Virtex-6 logic cell is approximately 15 ASIC gate equivalents. On the other hand, the optimized DPD implementation example with LUT complexity of ‘0.25∙P’ and with ‘N/5’ number of coefficients has approximately five times less number of ASIC gates, computation time and power consumption. Note that this example was presented
only to demonstrate the hardware-software optimization technique introduced in this dissertation. The results do not represent the state-of-the-art of the ADI DPD algorithm performance, for which the further development efforts have led to improved results.

4.2 A simulation method for design and optimization of RF power amplifiers with digital predistortion

With the evolution of design methods and process technologies several power amplifier (PA) architectures such as Class A, B, AB and Doherty have been proposed to explore the trade-off between linearity and efficiency. To achieve better linearity and thus less distortion, PAs are often backed-off to operate at low output power levels, which degrades the efficiency. Other approaches involve the integration of intricate techniques such as envelope tracking and digital predistortion (DPD) with the PAs [10], [82]-[83]. Typically, PAs and DPD systems are designed independently, necessitating
that the integrated systems (PA + DPD) are tuned and configured either at run time or
during production testing to meet performance requirements. This approach implies
several cycles of design, fabrication, and optimization of the PA and DPD involving
hardware testing of the integrated system. Moreover, it is very likely that an integrated
system optimized for one application does not deliver best performance in a different
application. A simulation approach that provides new capabilities to designers for
customization and optimization of the integrated system is essential for more efficient
PA and DPD algorithm development and to circumvent hardware-based testing during
each design iteration. The simulation technique that was created for this purpose is
described in this chapter. Also, note that the DPD algorithm outlined in Section 4.1.1 is
adopted for PA linearization here. Fig. 34 displays the block diagram of the design and
simulation approach, where the complete transmitter chain except the power amplifier
(PA) was modeled with embedded Matlab in the Advanced Design System (ADS)
software. In the modeled system, a complex baseband (BB) signal \((I_{BB}, Q_{BB})\) for LTE
was generated in Matlab which after preprocessing and up-conversion to the real RF
signal was fed to the RF PA designed in the ADS environment. The real-valued RF
output of the PA was captured in Matlab where it was down-converted to a complex BB
signal \((I_{BB,O}, Q_{BB,O})\). The generated \((I_{BB}, Q_{BB})\) and the received \((I_{BB,O}, Q_{BB,O})\) BB
signals are fed to the digital predistortion (DPD) system modeled with embedded
Matlab. The output of the DPD system; i.e., the predistorted BB signal \((I_{BB,P}, Q_{BB,P})\), is
then up-converted to the RF frequency and applied to the PA in ADS. Ideally, the
modeled RF transceiver system should replicate the actual RF transceiver, but due to
tool limitations the signals were restricted only to the discrete domain. Nevertheless, the
signals were sampled at a relatively high sampling rate of 36 GS/s to emulate the actual
analog signals. This methodology offers flexibility to exchange the components of the
transceiver between Matlab and ADS to perform system-level co-simulations with
behavioral blocks and transistor-level designs. Furthermore, the designers have freedom
to fully customize/tune a transmitter or transceiver for estimation of resource require-
ments.
4.2.1 Inverted Doherty power amplifier design

To exemplify the proposed simulation approach, a 10 W, 2-way inverted Doherty PA (IDPA) was designed using Wavetek’s InGaP/GaAs Heterojunction Bipolar Transistor (HBT) technology in the ADS environment. The schematic of the designed IDPA is displayed in Fig. 35. The PA was primarily designed for small cell base station applications with a center frequency of 2.4 GHz. The simulated 1-dB compression point of the IDPA is 40 dBm with a gain of around 30 dB. The simulated Power Added Efficiency (PAE) is approximately 30% and 42% with and without the driver stage, respectively. Key specifications from simulations of the IDPA design are plotted vs. input power in Fig. 36.
Figure 35  Inverted Doherty power amplifier (IDPA).
4.2.2 Simulation method

A simulation method that integrates Matlab and ADS was developed to simultaneously optimize the DPD algorithm and IDPA design. The ADS setup of the simulation technique is displayed in Fig. 37. The methodology utilizes the Ptolemy feature of ADS to construct a co-simulation environment that allows designers to perform system-level simulations of RF amplifiers interfaced to signal processing elements modeled in Matlab [86]-[87].

4.2.2.1 Key aspects of the simulation technique

4.2.2.1.1 Matlab models

Embedded Matlab models, such as MatlabF_M, MatlabSinkF and MatlabFCx_M, provide an interface between ADS and Matlab [87]. These functions can only be used with the ‘numeric’ data type defined in ADS, requiring that the inputs and outputs of the models are of numeric type. In the setup in Fig. 37, the component ‘M1’ is the instance of model ‘MatlabF_M’. The function incorporated into M1 in Fig. 37 has no input as reflected by the grounded input in ADS, and its output is in the form of a floating-point real valued column matrix. Component ‘M2’ is the instance of the Matlab model ‘MatlabFCx_M’, and its input and output are in the form of floating-point complex matrices. The component ‘M3’ is a Matlab sink that takes a floating-point complex matrix and stores the computed floating-point complex output matrix in a Matlab variable. Data between components M1 and M3 is passed through shared Matlab variables (labeled ‘BasebandInputSignal’ and ‘BasebandPredistortedInputSignal’ in Fig. 37).
Figure 37. Simulation setup in ADS.
4.2.2.1.2 Data formatting

The restriction of Matlab models to numeric data type and the requirement of timed signals for the RF PA compels the formatting of the data at different stages in the simulation setup. The following components are used to format the data during simulation:

a. UnPk_M (U1): The floating-point real valued elements of a column vector at the output of M1 are converted into floating-point scalar outputs using ‘U1’. Parameters of U1, namely ‘NumRows’ and ‘NumColumns’, should match to the dimension of the matrix generated at the output of the component M1.

b. FloatToTimed (F1): Floating-point scalar output samples of U1 are converted into time-varying samples using ‘F1’. Parameter ‘TStep’ in the setting of F1 should be equal to the time step defined in the transient simulation controller of ADS as described in the next section. These time-varying samples are then applied to the RF PA (IDPA) shown in Fig. 35 that is included as subcircuit ‘RF_PZ_CKT:R1’ in Fig. 37.

c. TimedToCx (T1): Component ‘T1’ typecasts the floating-point real valued timed RF signal at the output of the IDPA to a floating-point complex valued signal of numeric type sample by sample.

d. PackCx_M (P1): Scalar floating-point complex valued samples at the output of T1 are packed into a complex matrix by ‘P1’. The dimension of the matrix can be configured by setting the parameters ‘NumRows’ and ‘NumColumns’.

4.2.2.1.3 Simulation controllers

a. Transient simulation controller (Tran1): The transient simulator was configured by setting the ‘Time step control method’ to ‘Fixed’ and ‘Max time step’ to ‘1/fsamp’, where $fsamp$ is the sampling frequency of the up-sampled baseband input signal before up-conversion to the RF frequency. A larger value for the parameter ‘Stop time’ is recommended to capture enough samples of the baseband input signal. A frequency of ‘2.4 GHz’ and an order of ‘3’ were used under the frequency tabs of the simulation controller to perform the simulation at the desired 2.4 GHz center frequency.
b. Data flow controller (DF): The Data flow (DF) controller is a mandatory element of ADS Ptolemy simulation. It controls the flow of mixed (timed and numeric) signals. The key parameters of the data flow controller are ‘DefaultNumericStop’ and ‘DefaultTimeStop’. The parameter ‘DefaultNumericStop’ defines the number of samples (R) of the RF input signal chosen for the simulation, and thereby sets the time required to capture ‘R’ samples to ‘R\cdot1/f_{samp}'. The maximum value permitted for ‘DefaultNumericStop’ is the length of the column vector generated by M1. A higher value of ‘R’ results in better accuracy with longer simulation time.

4.2.2.1.4 Simulation flow

At the start of the ADS simulation, component M1 in Fig. 37 loads a complex baseband input signal predefined in a shared Matlab variable ‘BasebandInputSignal’. The code in M1 up-samples and up-converts the initial complex baseband input signal ‘x’, and generates numeric samples of the RF input signal centered at the desired RF frequency. The output type of M1 is converted from numeric to timed by U1 and F1, after which it is fed to the RF PA. The time valued RF output samples of the PA are captured and converted back to numeric type by T1 and F1, after which they are passed to M2 for down-conversion and down-sampling to recover the complex baseband output signal. The up and down conversion schemes performed in M1 and M2 respectively are visualized in Fig. 38. The DPD algorithm (discussed in Section 4.1.1) defined in M3 processes the baseband output samples ‘y’ and baseband input samples ‘x’ from M2 and ‘BasebandInputSignal’, respectively. The first step in the DPD routine is to align ‘x’ and ‘y’ in time as shown in Fig. 39. After time alignment, the DPD algorithm uses ‘N’ number of samples of ‘x’ and ‘y’ along with the predefined values of i, j, and k to learn the forward PA model using equation (49). Finally, ‘x’ is predistorted using the inverse of the forward PA model and is saved to the shared variable ‘BasebandPredistortedInputSignal’. The predistorted baseband signal is fetched by M1 and is then applied to the PA as a new input during the second iteration. Ideally, the modeled signal path should mimic the actual RF transceiver nonlinearity and noise performance as well as the finite precision of the analog-to-digital conversion and digital-to-analog conversion, but due to
Figure 38. Up-conversion and down-conversion (implemented with Matlab code).

Figure 39. Example waveforms after time alignment of the scaled baseband input signal with the down-converted and scaled baseband output signal.

limitations of the tools, the signal was restricted to the discrete time domain with ideal up and down conversions in Matlab.

4.2.3 Simulation results

To validate the developed simulation technique, a modulated LTE test signal with a bandwidth (\# of carriers \times carrier bandwidth) of 40 MHz (2 carriers \times 20 MHz = 40 MHz) and a peak-to-average ratio (PAR) of 7.5 dB was applied to the RF Inverted Doherty PA (IDPA) operating with an average output power around 31 dBm. The RF signals captured at the input and output of the IDPA in Matlab are displayed in Fig. 40. A total of ‘\(N = M\)’ and ‘\(N = M/2\)’ samples were used to learn the PA model in the DPD, which is then used to predistort the baseband input signal ‘\(x\)’. The spectra of the RF output signal without and with the application of the DPD (with ‘\(N = M\)’ and ‘\(N = M/2\)’
Figure 40. RF signals at the input and the output of the PA captured in Matlab after a transient simulation with ADS.

samples) are shown in Fig. 41, where the x-axis represents the equivalent baseband frequency. An improvement of approximately 15 $dB$ and 17 $dB$ in the adjacent channel leakage ratio (ACLR) was observed with DPD using ‘$N = M/2$’ and ‘$N = M$’ samples, respectively. Note that the number of samples (N) is one of the factors that determine the DPD implementation complexity with regards to the required computational resources and power. The presented method not only permits designers to verify DPD algorithms more realistically (using transistor-level PA simulations instead of Matlab PA models) without hardware testing, but also to jointly optimize the DPD and PA during the design phase. Furthermore, it provides a foundation to investigate adaptive PA biasing schemes aimed at reducing the number of DPD learning iterations by introducing bias control signals based on results after the first DPD iteration.

4.2.4 Case study

The simulation platform described in this chapter allows designer to optimize different components of a transmitter or receiver chain. Some of the potential reconfigurable elements of a transmitter (such as the one in Fig. 16) are listed in Table 16. The simulation approach was adopted in a case study to select a DPD architecture (from among different alternatives) that meets the given requirements of an application with minimal resources. A class-AB PA similar to the one presented in Section 4.2.1 was designed
using Wavetek’s GaAs HBT technology. The PA operates at 2.4 GHz and has a gain, output 1-dB compression point, and power added efficiency (PAE) of 20 dB, 28 dBm, and 35% respectively.

4.2.4.1 Simulation results

DPD engines with different predistorter models [according to the definitions in Table 16: GMP (M = 2, O = 5, C = 1), MP (M = 2, O = 5), and ML (O = 5)] were adopted to linearize the designed PA. Fig. 42 shows the simulated power spectrum at the output of the PA when fed with a 2-carrier 40 MHz LTE test signal having 7.5 dB peak-to-average ratio (PAR). Thus, if an application requires an adjacent channel leakage ratio (ACLR) of 45 dBc, then the MP (with M = 2, O = 5) will be sufficient

<table>
<thead>
<tr>
<th>Modules</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPD</td>
<td>Predistorter models such as generalized memory polynomial (GMP), memory polynomial (MP), memoryless (ML), etc. with order (O), memory depth (M), and cross-terms (C).</td>
</tr>
<tr>
<td>Converters</td>
<td>DAC/ADC resolution, nonlinearity.</td>
</tr>
<tr>
<td>PA</td>
<td>PA architectures such as class-AB, Doherty PA, etc. under different operating conditions.</td>
</tr>
</tbody>
</table>
Figure 42. Simulated power spectrum at the output of the PA.

in this example case, which requires less resources than the corresponding GMP.

4.3 A tuning technique for temperature and process variation compensation of power amplifiers with digital predistortion

4.3.1 Introduction

Present day communication systems are very data intensive and demand high-performance low-cost radio frequency (RF) transceivers. One of the key components of RF transceivers is the power amplifier (PA). With the PA being inherently nonlinear and the most power hungry device, it dictates the overall performance and power budget of the transceiver system. A traditional way to improve the linearity of PAs is to operate them at output power levels much lower than the saturation point but at the expense of efficiency. Alternatively, PAs are often integrated with performance improvement techniques such as digital predistortion (DPD) techniques [1]. Modern process technologies facilitate the integration of a large number of analog/RF circuits to achieve high performance but with less control over process, voltage and temperature (PVT) variations. Consequently, techniques such as DPD usually provide significant improvement of linearity, which is typically measured in terms of Adjacent Channel Leakage Ratio (ACLR), defined as the amount of power leaking into the adjacent channel. However the performance of the DPD itself is governed by the region of operation of the PA, which in turn is highly influenced by PVT variations [88]-[90]. As visualized in Fig. 43(a), under normal operating conditions and with input power ($P_{IN}$), the PA is operating near the saturation point for which the DPD is capable to provide the required linearity improve-
Figure 43. Characteristics of a power amplifier: (a) under normal operating conditions; (b) under PVT variation showing drift of the 1-dB compression point into the deep saturation region for the same input power $P_{IN}$.

However, PVT variations can significantly influence the operating characteristics of the PA as shown in Fig. 43(b) where the 1-dB compression point of the PA is degraded. Hence, for the same input power ($P_{IN}$), the PA is now operating in deep saturation, for which the DPD is not qualified to compensate. In addition, the performance of the DPD and the linearity requirement of the PA (or the device under test – DUT) often vary for different communication standards, such as LTE and WCDMA, based on the bandwidth and the peak-to-average ratio (PAR) of the test signal [81]-[85]. For instance, in a system with wide signal bandwidth it is highly likely that even an overdesigned DPD system does not provide the desired linearity improvement for a DUT operating at its 1-dB compression point, whereas it is very likely that a low complexity DPD can improve the linearity significantly provided that the DUT is operating just below its 1-dB compression point. For instance, as can be seen in Fig. 44, the same DPD provides an additional 3-4dB improvement in the ACLR when the bias voltage of the DUT is raised from $V1$ to $V2$, which results in a slightly better 1-dB compression point. Thus, it is desirable to tune the DUT either at run-time or during production testing to achieve the required performance for a PA. Furthermore, digital circuits are often used to tune analog/RF circuits for performance optimization [91], which is a key motivation behind this work. This section presents a digitally-assisted tuning mechanism that employs a
Figure 44. Measured ACLR of an off-the-shelf PA with different bias conditions.

digital-to-analog converter (DAC) to tune the bias voltage of PAs with integrated DPD techniques for temperature and process variation compensation.

4.3.2 Digital predistortion overview

A conceptual representation of a conventional (or an indirect learning) digital predistortion (DPD) system is provided in Fig. 45. It consists of a ‘predistorter’ and a ‘postdistorter’ (also referred as an adaptation engine). The postdistorter is employed in the feedback path. The RF output of the DUT (PA in this case) is down-converted to baseband frequencies and is digitized using an ADC before it is applied to the postdistorter/adaptation engine. The postdistorter that mimics the inverse of the PA uses this down-converted digitized RF output to calculate the best estimate of the digital baseband input signal applied to the input of the PA (after up-convension). The postdistorter is generally modeled with a set of polynomial equations such as the Volterra series or its other variants such as a Generalized Memory polynomial (GMP) defined by the following equation [1]:

\[ y(t) = \sum_{i,j,k} a_{i,j,k} \cdot |x(t-i)|^k \cdot x(t-j), \]  

(50)
where $a_{i,j,k}$ are the complex coefficients, '$i$' is the memory depth, and '$k$' is the nonlinearity order (O) of the PA model. The terms in equation (50) for which '$i$' and '$j$' are different are referred to as cross terms (CT). The postdistorter function computes the values of the coefficients ($a_{i,j,k}$) by comparing the output signal of the DUT with the delayed version of the input signal. These coefficients are then fed to the predistorter, which is an exact copy of the postdistorter function and is placed in the main transmit path of the system to predistort the actual digital baseband input signal, such that the overall system response (Predistorter + PA) is linear as depicted in Fig. 46. In addition to the computation of the complex coefficients, the error (E) term that is the difference between the original digital baseband input and the estimated digital baseband signal from the PA output is also calculated during each iteration of the DPD engine. This error signal (E) has a strong correlation with the linearity of the PA. For example, if a PA is operating in the linear region, the distortions introduced by the PA are low and thus the PA output is just the amplified version of the input signal, yielding a small error term. Likewise, if the PA is operating in deep saturation, then the
output of the PA will be a nonlinear function of the input signal, implying that the error term will have a higher value. Also, as portrayed in Fig. 43, PVT variations change the 1-dB compression point of PAs and thereby have a strong impact on the error signal [88]-[90]. The proposed tuning technique exploits the error term (E) to monitor the variation of the PA’s 1-dB compression point. Furthermore, this work introduces a digitally-assisted tuning mechanism to compensate for the effects of PVT variations. A DPD technique developed by collaborators at Analog Devices Lyric Labs was employed to demonstrate the proposed tuning technique in this collaborative research project.

4.3.3 Tuning approach

The tuning technique was implemented in the Ptolemy design environment of Agilent’s Advanced Design System (ADS), which allows co-simulation of Matlab elements with transistor-level designs [86]-[87]. The block diagram of the approach is presented in Fig. 47. It consists of a PA designed in ADS that is integrated with digital processing elements, including the DPD algorithm in Matlab. The key advantage of using the co-simulation platform [31] is the ease of performing simulation and optimization at the system level. In the developed approach, the initial baseband test signal generation and up-conversion to RF is performed in Matlab. The RF test signal is then routed to the PA in the ADS design environment. The output of the RF PA is passed back to Matlab, where it is down-converted and digitized before being fed to the DPD engine. The postdistorter element of the DPD engine computes the complex coefficients that are used by the predistorter to predistort the original digital baseband input signal. The predistorter...
ed baseband signal is then up-converted to RF frequencies and is applied to the RF PA. The error signal (E) computed by the DPD engine is sent to the tuning algorithm implemented in Matlab, which generates the digital control codes. These digital codes are converted to a variable voltage signal by a circuit-level digital-to-analog converter design. Finally, the generated voltage is applied as a bias voltage to the PA to tune the operating point for linearity enhancement.

4.3.4 Inverted Doherty power amplifier

A 4 W inverted Doherty PA (IDPA) was designed using Wavetek’s InGaP/GaAs Heterojunction Bipolar Transistor (HBT) technology. The schematic of the 2.4 GHz IDPA is displayed in Fig. 48. It consists of a driver amplifier, a main amplifier and a peaking amplifier that are designed to operate in Class AB, Class B and Class C modes, respectively. Harmonic balance (HB) simulations were conducted to characterize the IDPA. It has a gain of 25.4 dB with a Power Added Efficiency (PAE) of 25.8% and an output 1-dB compression point of 36.0 dBm. These simulated characteristics of the IDPA are shown in Fig. 49. They were achieved by incorporating inductors with quality factor (Q) of 15 into the IDPA operating at a nominal temperature (T) of 25°C.

GaAs HBT RF PAs are sensitivity to PVT variations, which can significantly affect performance, resulting in several decibels of 1-dB compression point changes as well as gain and output power variations [92]-[93]. To investigate the impact of process and temperature variations on the performance of the IDPA under investigation, simulations were performed with ‘Q’ and ‘T’ varying from 5 to 30 and -25°C to 100°C using step sizes of 5 and 25°C respectively. Simulated output 1-dB compression points of the IDPA for various values of ‘Q’ and ‘T’ are tabulated in Table 17. For a lower value of Q = 5, the output 1-dB compression point degrades by 4.7 dB from 36.2 dBm with Q = 15, whereas it improves by around 1.2 dB with Q = 30. Likewise, the output 1-dB compression point drifts from 36.2 dBm at T = 25°C to 24.6 dBm at T = 75°C, exhibiting 11.6 dB degradation. In practice, the temperature of the PA typically rises up to higher values and the quality factor is restricted by the process technology [94]-[95], therefore the combined effect of both temperature and process variations on the performance of a PA could be even more severe.
Figure 48. Inverted Doherty power amplifier (IDPA).
Figure 49. Simulated characteristics of the standalone Inverted Doherty PA (IDPA) in the absence of the tuning algorithm and DAC.

| Table 17 Simulated output 1-dB compression point with varying quality factor (Q) and temperature (T) |
|-------------------------------------------------|---------|---------|---------|---------|---------|---------|
| Quality Factor (Q) | 5       | 10      | 15      | 20      | 25      | 30      |
| Output 1-dB Comp. Point (dBm) | 31.5    | 33.2    | 36.2    | 36.9    | 37.1    | 37.39   |
| Temperature (ºC) | -25     | 0       | 25      | 50      | 75      | 100     |
| Output 1-dB Comp. Point (dBm) | 36.7    | 36.5    | 36.2    | 35.8    | 24.6    | 19.4    |

Generally, PAs are operated near the compression point for efficiency reasons, and DPD techniques are used to compensate for the distortions introduced by the nonlinearities of the PAs, which improves ACLR. The drift of a PA’s compression point due to PVT variations can severely affect the overall performance of the DPD. For example, Table 18 summarizes the simulated performance of the DPD in terms of the ACLR and the normalized mean square error (NMSE) when the output 1-dB compression point of the example IDPA design varies from its nominal value of 36 dBm. A two-carrier 40 MHz LTE signal was used as a test signal. For the cases where the PA exhibits a higher
Table 18. Simulated ACLR and Normalized Mean Square Error (NMSE) of the IDPA with a 2c × 20 MHz test signal resulting in 36 dBm output power

<table>
<thead>
<tr>
<th>Region of IDPA Operation</th>
<th>Output 1-dB Compression Point</th>
<th>ACLR (dBc) Before DPD</th>
<th>ACLR (dBc) After DPD</th>
<th>NMSE (E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Saturation</td>
<td>34.5</td>
<td>-10</td>
<td>-12</td>
<td>19380.3</td>
</tr>
<tr>
<td>Saturation</td>
<td>35.5</td>
<td>-15</td>
<td>-20</td>
<td>12870.7</td>
</tr>
<tr>
<td><strong>Normal Operating Condition</strong></td>
<td><strong>36</strong></td>
<td><strong>-22</strong></td>
<td><strong>-41</strong></td>
<td><strong>567.54</strong></td>
</tr>
<tr>
<td>Near Saturation</td>
<td>36.5</td>
<td>-25</td>
<td>-45</td>
<td>365.18</td>
</tr>
<tr>
<td>Linear</td>
<td>38.0</td>
<td>-38</td>
<td>-55</td>
<td>104.44</td>
</tr>
</tbody>
</table>

1-dB compression point of >36 dBm and at 36 dBm output power, the PA is operating in a more linear region and thus less distortions are created by the PA, helping to boost the DPD performance as reflected by the lower values of ACLR after DPD and the low NMSE. Whereas, for the output 1-dB compression point of values <36 dBm, the PA is driven deeper into the saturation region which eventually degrades the DPD performance. The NMSE values are relatively high for the cases when the PA is operating in the deep saturation region compared to the values when the PA is operating near saturation or in the linear region. Hence, the NMSE computed by the DPD represents an indicator for the PA’s region of operation. As depicted in Fig. 47 and Fig. 48, the developed tuning technique utilizes NMSE values and employs a DAC to generate a variable bias voltage for the PA to bring it to near the saturation/linear region of operation.

4.3.5 Digital-to-analog converter (DAC)

The digital codes generated by the tuning algorithm (discussed in the next subsection) are provided to a DAC to generate a bias voltage that is applied as a drain voltage (V_{\text{DRAIN,P}}) of the peaking amplifier in the IDPA. For fine tuning of the bias voltage, and to keep the bias voltage within a safe limit, it is advisable to change the bias voltage with a small step size. Another key aspect is that a large change in the bias voltage can potentially lead to significant changes of the PA characteristics, which can offset the PA
model in the DPD engine and result in a large error term. Therefore, a 7-bit R-2R DAC (similar to the one in [96]) with an output buffer amplifier was designed using Wa-vetek’s InGaP/GaAs Heterojunction Bipolar Transistor (HBT) technology. The schematic of the DAC with the output buffer is displayed in Fig. 50. The DAC and the output buffer amplifier are tuned such that an output drain voltage ($V_{\text{DRAIN,P}}$) from 10.5 V to 13.15 V is generated when the digital code ($D_6D_5D_4D_3D_2D_1D_0$) is changed from (0000000) to (1111111). The simulated parameters of the IDPA with integrated DAC are listed in Table 19. For a lower bias voltage ($V_{\text{DRAIN,P}}$) of 10.5 V, the IDPA is backed off into a linear region with an output 1-dB compression point of 34.5 dBm and with a lower PAE of around 23.9%. In contrast, for a higher bias voltage ($V_{\text{DRAIN,P}}$) of 13.15 V, the IDPA exhibits a higher output 1-dB compression point of 37.4 dBm and also has a higher PAE of 28.5%. For the digital code ($D_6D_5D_4D_3D_2D_1D_0$) of (1000000), a bias voltage ($V_{\text{DRAIN,P}}$) of around 12 V is generated, which is equal to the original bias voltage of the peaking amplifier. The IDPA with DAC delivers slightly lower performance with $V_{\text{DRAIN,P}}$ of approximately 12 V compared to the IDPA without the DAC and

<table>
<thead>
<tr>
<th>Digital Code</th>
<th>$V_{\text{DRAIN,P}}$ (V)</th>
<th>Gain (dB)</th>
<th>PAE (%)</th>
<th>Output 1-dB Compression Point (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standalone IDPA without the DAC</td>
<td>-</td>
<td>12</td>
<td>25.402</td>
<td>25.843</td>
</tr>
<tr>
<td>IDPA integrated with the DAC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0</td>
<td>10.5</td>
<td>25.152</td>
<td>23.884</td>
<td>34.545 35.760</td>
</tr>
<tr>
<td>• • • • • • •</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0</td>
<td>12.06</td>
<td>25.384</td>
<td>25.781</td>
<td>35.992</td>
</tr>
<tr>
<td>1 0 0 0 0 0 1</td>
<td>12.09</td>
<td>25.388</td>
<td>25.814</td>
<td>36.002</td>
</tr>
<tr>
<td>• • • • • • •</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>• • • • • • •</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1</td>
<td>13.15</td>
<td>24.10</td>
<td>28.521</td>
<td>37.363</td>
</tr>
</tbody>
</table>
with $V_{\text{DRAIN,P}}$ of 12 V. Thus, according to the results in Table 19, the adopted DAC architecture allows fine tuning of the IDPA to compensate for the variations caused due to changes in $Q$ from 10 to 25 and for temperature changes of up to 75°C. Note that the proposed tuning technique is employed to tune the bias voltage of the peaking amplifier, but the same technique can be adopted to tune the main and the driver amplifier of the IDPA as well. Additionally, a lower resolution DAC can be used to reduce the convergence time of the IDPA.

### 4.3.6 Tuning algorithm

For a fast response time, the DPD is frequently realized on a Field Programmable Gate Array (FPGA), which provides sufficient resources to implement the proposed tuning algorithm. This algorithm to tune the IDPA design described in Section 4.3.4 is outlined in the flow chart of Fig. 51. During the power-up sequence the digital code is initialized to a value that corresponds to the recommended bias voltage ($V_{\text{DRAIN,P,Recom}}$). For this case, the digital code ($D_6D_5D_4D_3D_2D_1D_0$) is initialized to (1000000), which corresponds to the bias voltage of 12 V ($V_{\text{DRAIN,P,Recom}} = 12$ V). Two variables ($E_I$ and $E_F$) to hold the error (E) output from the DPD engine are also defined and initialized to ‘0’. After initialization, the original digital baseband input signal ($IN_{\text{DBB}}$) is up-converted to RF ($RF_{\text{IN,O}}$) and applied to the input of the IDPA as $RF_{\text{IN}}$. The amplified RF output ($RF_{\text{OUT}}$) is down-converted to digital baseband output signal ($OUT_{\text{DBB}}$) and is fed to the DPD engine along with the digital baseband input signal ($IN_{\text{DBB}}$). After ‘M’ iterations (required for convergence) of the DPD, the error signal ($E_F$) is fetched and saved in the
Figure 51. Flow chart of the tuning algorithm.

variable $E_I$. After these initial power-up steps, the tuning algorithm holds the value of the digital code until a change in the error signal ($E_F$) of the DPD is detected. Computational inaccuracies or slight changes in operating conditions could also cause minor variations in the error signal, which are neglected by the tuning algorithm. However, if the changes in error signal spread out of a certain range (which is user-defined, ±50 in this case), then the tuning algorithm becomes active. A positive change in the error signal ($E_F > E_I + 50$) reflects a case in which the DPD does not converge to the initial values due to the decrease of the output 1-dB compression of the IDPA (from PVT variations) or due to the increase in the input signal power that drives the PA into the deep saturation region. For such a case, the tuning algorithm will push the bias voltage to a higher value by
increasing the value of the digital code by 1 (or a user-defined step size) provided that
the digital code and the bias voltages are within the allowed limits. The increase in the
bias voltage will boost the PA’s performance. The loop is repeated until the DPD
converges to the initial error (E) value or until the bias voltage stays within the specified
limits. In contrast, if the computed error signal (E) from the DPD is below the desired
error span (E_F < E_I – 50), which could be because of low input signal power that places
the PA into the linear region and thus reduces the PAE, then the tuning algorithm will
decrease the value of the digital code in order to reduce the bias voltage of the PA. The
latter action will lower the 1-dB compression point and help to improve the PAE of the
PA.

4.3.7 Performance tuning simulation method

The ADS setup to evaluate the proposed tuning approach is displayed in Fig. 52. The
methodology utilizes the Ptolemy feature of ADS to construct a co-simulation environ-
ment that allows designers to perform system-level simulations of RF amplifiers
interfaced to signal processing elements modeled in Matlab [86]-[87].

Key aspects of the simulation technique:

1. Matlab models (M1 to M4): Embedded Matlab code, such as MatlabF_M, Matlab-
SinkF and MatlabFCx_M, provides an interface between ADS and Matlab. These
functions can only be used with the ‘numeric’ data type defined in ADS, requiring that
the inputs and outputs of these models are of numeric type. In the setup in Fig. 52, the
component ‘M1’ is the instance of model ‘MatlabF_M’. The function incorporated into
M1 in Fig. 52 has no input as reflected by the grounded input in ADS, and its output is in
the form of a floating-point real valued column matrix. Component ‘M2’ is the instance
of the Matlab model ‘MatlabFCx_M’, and its input and output are in the form of float-
ing-point complex matrices. The component ‘M3’ is a Matlab sink that takes a floating-
point complex matrix and stores the computed floating-point complex output matrix in a
Matlab variable. Data between components M1 and M3 is passed through shared Matlab
variables (labeled ‘BasebandInputSignal’ and ‘BasebandPredistortedInputSignal’ in Fig.
52). The component ‘M4’ is the instance of model ‘MatlabF_M’ and the tuning algo-

rithm described in Section 4.3.6 is embedded in it. It is a subroutine and is instantiated in
the M3 block. The output of M4 is a column vector that contains the digital code \((D_6D_5D_4D_3D_2D_1D_0)\). This digital code is in numeric floating point data format and
converted into a time-varying analog signal using the FloatToTimed (F2) block defined in point 2 below.

2. **Data formatting:** The restriction of Matlab models to numeric data type and the requirement of timed signals for the circuit level components such as the RF PA and DAC compels the formatting of the data at different stages in the simulation setup. The following components are used to format the data during simulation:

   a. UnPk_M (U1): The floating-point real valued elements of a column vector at the output of M1 are converted into floating-point scalar outputs using ‘U1’. Parameters of U1, namely ‘NumRows’ and ‘NumCoulmns’, should match to the dimension of the matrix generated at the output of the component M1.

   b. FloatToTimed (F1, F2): Floating-point scalar output samples of U1 and M4 are converted into time-varying samples using ‘F1’ and ‘F2’ respectively. Parameter ‘TStep’ in the setting of F1 should be equal to the time step defined in the transient simulation controller of ADS as described in point 3, below. These time-varying samples are then applied to the input (RF_IN) of the RF PA (IDPA) in Fig. 48 that is included as subcircuit ‘RF_PA_CKT:R1’ in Fig. 52.

   c. TimedToCx (T1): Component ‘T1’ typecasts the floating-point real valued timed RF signal at the output of the IDPA to a floating-point complex valued signal of numeric type sample by sample.

   d. PackCx_M (P1): Scalar floating-point complex valued samples at the output of T1 are packed into a complex matrix by ‘P1’. The dimension of the matrix can be configured by setting the parameters ‘NumRows’ and ‘NumCoulmns’.

3. **Simulation controllers:**

   a. Transient simulation controller (Tran1): The transient simulator was configured by setting the ‘Time step control method’ to ‘Fixed’ and ‘Max time step’ to ‘1/fsamp’, where \( fsamp \) is the sampling frequency of the up-sampled baseband input signal before up-conversion to the RF frequency. A larger value for the parameter ‘Stop time’ is recommended to capture enough samples of the baseband input signal. A frequency of ‘2.4 GHz’ and an order of ‘3’ were used under the frequency tabs of the simulation controller to perform the simulation at the desired 2.4 GHz center frequency.
b. Data flow controller (DF): The Data flow (DF) controller is a mandatory element of ADS Ptolemy simulation. It controls the flow of mixed (timed and numeric) signals. The key parameters of the data flow controller are ‘DefaultNumericStop’ and ‘DefaultTimeStop’. The parameter ‘DefaultNumericStop’ defines the number of samples (R) of the RF input signal chosen for the simulation, and thereby sets the time required to capture ‘R’ samples to ‘R\cdot1/fsamp’. The maximum value permitted for ‘DefaultNumericStop’ is the length of the column vector generated by M1. A higher value of ‘R’ results in better accuracy with longer simulation time.

4. Simulation flow:

At the start of the ADS simulation, component M1 in Fig. 52 loads a complex baseband input signal predefined in a shared Matlab variable ‘BasebandInputSignal’. The code in M1 up-samples and up-converts the initial complex baseband input signal ‘x’, and generates numeric samples of the RF input signal centered at the desired RF frequency. The output type of M1 is converted from numeric to timed by U1 and F1, after which it is fed to the RF PA. The time-valued RF output samples of the PA are captured and converted back to numeric type by T1 and P1, after which they are passed to M2 for down-conversion and down-sampling to recover the complex baseband output signal. The DPD algorithm defined in M3 processes the baseband output samples ‘y’ and baseband input samples ‘x’ from M2 and ‘BasebandInputSignal’, respectively. The first step in the DPD routine is to align ‘x’ and ‘y’ in time. After time alignment, the DPD algorithm uses ‘N’ number of samples of ‘x’ and ‘y’ along with the predefined values of i, j, and k to learn the forward PA model using equation (50). Finally, ‘x’ is predistorted using the inverse of the forward PA model and is saved to the shared variable ‘BasebandPredistortedInputSignal’. The predistorted baseband signal is fetched by M1 and then applied to the PA as a new input during the second iteration. The error signal (E) generated by the DPD engine is passed to an internal subroutine (M4) that defines the tuning algorithm. Ideally, the modeled signal path should mimic the actual RF transceiver nonlinearity and noise performance as well as the finite precision of the analog-to-digital conversion and digital-to-analog conversion, but due to limitations of the tools, the
signal was restricted to the discrete time domain with ideal up and down-conversions in Matlab.

4.3.8 Simulation results

Transient simulations were performed under different operating conditions (selected from Table 17 and Table 18) to verify the proposed tuning mechanism. A bias voltage of $V_{\text{DRAIN,P}} = 12 \, \text{V}$ was applied to the IDPA and a 40 $MHz$ LTE test signal with input power of 11 $dBm$ was provided at the input of the IDPA. With a gain of around 25 $dB$, a simulated output power of (11 $dBm$ + 25 $dB$ = 36 $dBm$) was observed. As the first test case, $Q = 15$ and $T = 25^\circ C$ were selected in the simulation settings such that IDPA operates close to saturation with an output 1-dB compression point of 36.3 $dBm$. With the DPD technique, the simulations revealed a 20 $dB$ improvement of ACLR for both the standalone IDPA and the IDPA with tuning technique as shown in Fig. 53(a). The digital code $(D_6 D_5 D_4 D_3 D_2 D_1 D_0) = (1000000)$ was automatically set by the tuning technique during the simulation. As the second test case, $Q = 15$ and $T = 50^\circ C$ were selected such that the output 1-dB compression point of the IDPA was reduced from 36.2 $dBm$ to 35.8 $dBm$. For the input power of 11 $dBm$, the IDPA was driven into the saturation region. Using the DPD technique, a simulated ACLR improvement of 10 $dB$ was obtained for the standalone IDPA. Whereas for the IDPA with the tuning technique, the automatic adaptation raised the bias voltage from 12 $V$ to 12.5 $V$ for the digital code $(D_6 D_5 D_4 D_3 D_2 D_1 D_0 = 1010000$, resulting in approximately 20 $dB$ ACLR improvement as can be observed in Fig. 53(b). For the third test case with $Q = 13$ and $T = 25^\circ C$, the output 1-dB compression point was further degraded to 34.5 $dBm$. With the same input power of 11 $dBm$, the IDPA was driven into the deep saturation region. Using the DPD technique, the simulated ACLR improvement was 2-3 $dB$ for the standalone IDPA. In comparison, the IDPA with tuning technique increased the bias voltage from 12 $V$ to 13.15 $V$ for the digital code $(D_6 D_5 D_4 D_3 D_2 D_1 D_0 = 1111111$, which improved the ACLR by around 10 $dB$ as shown in Fig. 53(c).
Figure 53. Simulated power spectrum at the output of the IDPA with and without the proposed tuning technique: (a) IDPA operating in the near-saturation region with $Q = 15$, and $T = 25^\circ C$, (b) IDPA operating in the saturation region with $Q = 15$, and $T = 50^\circ C$, and (c) IDPA operating in the deep saturation region with $Q = 13$, and $T = 25^\circ C$
5. Conclusion and Future Work

This dissertation research primarily focused on the development of digitally-assisted design, simulation and verification techniques for optimization of analog/RF circuits. An FFT-based spectral analysis technique was proposed for efficient on-chip spectral characterization of multi-tone signals. Simulation results showed that the simulated error for IM3 extraction from the output spectrum of the 16-point FFT is within 1.5 $dB$ for IM3 components $\leq 50$ $dBc$. In addition, the spectral analysis technique was employed to create an on-chip linearity calibration technique for RF amplifiers.

An approach to optimize DPD algorithms for best PA performance was demonstrated with a purpose-built integrated hardware-software test platform. This platform enables a comparison of different DPD implementations to meet target specifications with minimal computational resources. The optimization process was demonstrated with measurements of commercially available PAs using the Analog Devices DPD solution. A simulation method for design and optimization of RF PAs with DPD was also developed. The methodology was demonstrated through simulations of an inverted Doherty PA with an integrated DPD algorithm that was implemented with embedded Matlab in ADS. The simulation method enables designers to jointly optimize PAs and DPDs during design phases, such that costly hardware setups are circumvented during design iterations and are only required in final characterization test phases. Furthermore, a digitally-assisted tuning technique to compensate for temperature and process variations of RF PAs with DPD was introduced through this research. A co-simulation setup was developed in the ADS design environment to validate the tuning approach.

New efficient digital techniques to enhance the overall performance of an analog/RF circuits are always appealing and offer extensive research opportunities. Future research could involve on-chip integration of the presented tuning technique for PAs with DPD. The use of on-chip temperature sensors ([97]-[98]) as power detectors during built-in calibrations of PAs can be explored. Another aspect that could be addressed in future research is the development of design techniques that will assist dynamic optimizations of both the DPD and PAs to adapt to varying operating conditions. Furthermore, the co-
simulation platform with embedded Matlab can be employed with more circuit-level components for co-optimization of the complete transceiver signal chain. Other possibilities for future research include the hardware realization of the digitally-assisted calibration schemes presented in Chapter 3.
6. References


VLSI Computer Architecture Research Group, Oklahoma State University. Available online: http://vlsiarch.ecen.okstate.edu/flows/


