Design, Testing, and Analysis of Advanced Massive MIMO Transmitters

by

Jin Gyu Lim

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2026

© Jin Gyu Lim 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External-External Member: Prof. Patrick Roblin
Professor, Dept. of ECE, Ohio State University

Supervisor: Prof. Slim Boumaiza
Professor, Dept. of ECE, University of Waterloo

Internal Member: Prof. Peter Levine
Professor, Dept. of ECE, University of Waterloo

Internal Member: Prof. Lan Wei
Professor, Dept. of ECE, University of Waterloo

Internal-External Member: Prof. Jim Martin
Professor, Dept. of Physics, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Statement of Contributions

Work presented in Chapter 3:

Sec. 3.2 is based on the publication below. J. G. Lim developed the co-simulation framework and extracted simulation results of 2×2 MIMO transmitter system that H. Yu and E. Traore designed, fabricated, and tested. J. G. Lim performed further analysis on the effect of antenna mutual coupling using the developed co-simulation framework and the 2×2 MIMO transmitter system. J. Xia guided H. Yu and E. Traore with measurements and S. Boumaiza supervised the work. All authors contributed to the manuscript.

Citation:
1. H. Yu, E. Traore, M. Almoneer, J. G. Lim, J. Xia and S. Boumaiza, "Experimental Study of the Effects of Antenna Crosstalk on the Linearity and Efficiency of 5G Sub-6 GHz Wideband 2×2 Transmitter Arrays", 2021 IEEE MTT-S Int.l Microw. Symp. (IMS), Atlanta, GA, USA, June 2021, pp. 869-872.

2. J. G. Lim, H. Yu, E. Traore, M. Almoneer, J. Xia, and S. Boumaiza, "Modulated-Signal-Based EM/RF/DSP Co-Simulation Framework for Predictive Analysis of Fully Digital MIMO Transmitters," in 2021 51st Eur. Microw. Conf. (EuMC), pp. 454–457.

Work presented in Chapter 4:

Chapter 4 is based on the publication below. J. G. Lim developed the emulation framework and its theory, the 16-channel fully digital MIMO transmitter, and performed simulations and measurements. A. Ben Ayed performed calibration and measurements and provided the general PA model. S. Boumaiza supervised the work and all authors contributed to the manuscript.

Citation:
3. J. G. Lim, A. Ben Ayed, and S. Boumaiza, "Model-Assisted Emulation for Effective Design and Testing of Power Amplifiers in Massive MIMO Transmitters", to be submitted.

Work presented in Chapter 5:

Chapter 5 is based on the publications below. J. G. Lim developed the 16-channel fully digital MIMO transmitter and analyzed active PA output load modulation, A. Ben Ayed constructed the DSP architecture, and they performed measurements together. S. Boumaiza supervised the work and all authors contributed to the manuscript.

Citation:
4. J. G. Lim, A. Ben Ayed, and S. Boumaiza, "A Two-Tier Low-Complexity Linearization Architecture for Fully Digital mMIMO Transmitters," in IEEE Microw. Wirel. Technol. Lett. June 2026. [Accepted] (Presented in Appendix D)

5. J. G. Lim, A. Ben Ayed, and S. Boumaiza, "A Reduced-Dimensional Precoding-Aware MISO Digital Predistortion Framework for Fully Digital Massive MIMO Transmitters" to be submitted.

6. M. Almoneer, J. G. Lim, H. Yu, P. Mitran and S. Boumaiza, "Massive MIMO Precoding Methods That Minimize the Variation in Average Power and Active Impedance With Channel Conditions", in IEEE Trans. Microw. Theory and Techn., vol. 70, no. 8, pp. 4002-4016, Aug. 2022.

Work presented in Appendix E:
Appendix E is based on the publication below. J. G. Lim developed the 16-channel fully digital MIMO transmitter and performed the measurements. H. Barkhordar-pour and A. Ben Ayed developed the common DPD training formulation and also performed the measurements, and they performed measurements together. P. Mitran and S. Boumaiza supervised the work and all authors contributed to the manuscript.

Citation:
7. J. G. Lim, H. Barkhordar-Pour, A. Ben Ayed, P. Mitran and S. Boumaiza, "On the Viability of Using a Subset of Transmitter-Observation Receivers for Training a Common DPD in Fully Digital MIMO Transmitters", in IEEE Microw. Wireless Technol. Lett., vol. 33, no. 6, pp. 907-910, June 2023.

Other contributions:
For the publications below, the developed 16-channel fully digital MIMO transmitter system was utilized as a foundational tool for other research conducted in the group.

Citation:
8. H. Barkhordar-Pour, J. G. Lim, A. Ben Ayed, P. Mitran and S. Boumaiza, "Linearizability Assessment of a 3.5 GHz 16-Chain Fully Digital MIMO Transmitter Under Wideband Modulated Signals", 2024 103rd ARFTG Microw. Meas. Conf., Washington, DC, USA, 2024, pp. 1-4.

9. H. Barkhordar-Pour, J. G. Lim, A. Ben Ayed, P. Mitran and S. Boumaiza, "Active Calibration Approach Addressing Antenna Mutual Coupling and Power Amplifier Output Mismatch in Fully Digital MIMO Transmitters", in IEEE Microw. Wirel. Technol. Lett., vol. 34, no. 6, pp. 813-816, June 2024.

10. H. Barkhordar-Pour, J. G. Lim, M. Almoneer, P. Mitran and S. Boumaiza, "Real-Time FPGA-Based Implementation of Digital Predistorters for Fully Digital MIMO Transmitters", 2023 IEEE/MTT-S Int'l Microw. Symp. (IMS) June 2023, San Diego, CA, USA, 2023, pp. 263-266.

Abstract

The evolution of Fifth Generation (5G) wireless systems is driven by the need for higher data rates, lower latency, and improved coverage. Massive Multiple Input, Multiple Output (MIMO) transmitter front-ends have emerged as a key enabling technology, employing multiple parallel transmitter chains—each comprising Digital Signal Processing (DSP), Power Amplifiers (PAs), and antenna elements—to exploit spatial multiplexing and enable multi-user beamforming, at the cost of increased system complexity.

The PA is a critical component in each transmitter chain, as its efficiency dominates energy consumption and its output power determines coverage. However, PA performance is highly sensitive to load impedance. While isolators in 4G systems maintain a constant 50 Ω load, their use in massive MIMO arrays is impractical due to cost, size, bandwidth, and integration constraints. Consequently, antenna mismatches and mutual coupling introduce dynamic load variations that degrade PA and overall system performance, motivating a holistic system-level design approach.

The first objective of this thesis is to develop tools for system-level analysis of massive MIMO transmitters under realistic excitation. A multidisciplinary co-simulation framework integrating DSP, Radio Frequency (RF), and electromagnetic domains is proposed to capture signal processing, PA nonlinearities, and antenna coupling within a unified environment. Experimental validation using a four-channel fully digital MIMO transmitter demonstrates accurate prediction of system-level trends. A sixteen-channel testbed is further developed to validate design strategies and capture hardware-specific effects.

The second objective is to enable PA design under realistic system conditions. To mitigate the computational complexity of large-scale simulations, an emulation platform is developed that reproduces massive MIMO loading conditions using a single PA. This approach enables efficient characterization and optimization under dynamic impedance environments. Combined with the co-simulation framework, it supports PA design directly at the system level rather than under idealized 50 Ω assumptions.

The third objective is to investigate the impact of precoding on PA behavior. Conventional Digital Pre-Distortion (DPD) linearizes PAs using uncorrelated signals prior to precoding, implicitly assuming invariant load conditions. However, precoding alters signal correlation and power distribution, thereby modifying the load impedance seen by each PA in the presence of mutual coupling, which degrades linearization performance. To address this, an alternative architecture is explored in which precoding precedes linearization, enabling improved robustness and reduced DPD complexity under dynamic conditions.

Acknowledgements

I would like to first thank my supervisor Prof. Slim Boumaiza for taking me even though my master's degree was completed in a different field. I was also genuinely surprised that you still remembered me even after a few years had passed since taking your undergraduate courses. I would also like to express my sincere gratitude for all your guidance, patience, advice, and vision. My dream of designing my first electronic circuits came true in your group and I am getting closer to my next dream of becoming an independent researcher.

I would like to thank my committee members, Prof. Peter Levine, Prof. Lan Wei, and Prof. Jim Martin for reviewing this work and giving advice and constructive feedback. I also thank Prof. Patrick Roblin for agreeing to be my external examiner.

I would like to thank the members of the Emerging Radio Systems Group (EmRG) who helped me accomplish this journey. In particular, I would like to thank Dr. Jingjing (Peter) Xia for all his training and guidance. I used to get discouraged by big challenges in front of me but now I try to enjoy taking little steps every day to overcome them. Dr. Ahmed Ben Ayed helped me get out of my burnout phase so that I could sprint to the finish line. With his expertise and help, I could finish off the second and third objectives of this thesis. I also thank Dr. Mohammed Almoneer and Hoda Barkhordar-pour. All the discussions and collaboration with you helped me build and test the co-simulation framework and 16-channel transmitter system. I had a great experience of having the first publication from the group with Hang (Justin) Yu and Emile Traore. Your 4-channel transmitter system worked very well that I could finish the experimental verification of my co-simulation framework. The first conference I attended in Denver was also very memorable, sharing an AirBnB house with Bernard Tung, Ziran He, Huixin Jin, Ahmed Ben Ayed and Hoda Barkhordar-pour. I would like to thank all the office mates and EmRG members for all the laughs we shared over the past six years in no particular order: Mohammad A. Chalaki, Kyle Hong, Naved Anjoom, Zi Jun Su, Omar Morgan, Mehran Sahlabadi, Xiaofei Zhang, Yuxuan Chen, Yue Li, Nita Esfarayeni, Mahitab Eladwy, Bulent Ege Sancak, Justin Tung, Nizar Messaoudi, Alireza Salehiyan, Emma Li Gu, Yixuan Xie, Tom Chiu, Rio Liu, Mona Razmi.

I would like to thank NSERC (Postgraduate Scholarship Doctoral), Ontario (Ontario Graduate Scholarship), and the University of Waterloo for their financial support that made this work possible.

Last but not the least, I would like to thank my family for unconditional support that helped me finish this degree. It was a long journey and I would not have been able to complete this without them.

Dedication

This is dedicated to my family, including Louie the Bernese Mountain Dog.



Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract vii

Acknowledgements viii

Dedication ix

List of Figures xiv

List of Tables xxiii

List of Abbreviations xxv

1 Introduction 1

1.1 Motivation 1

1.2 Thesis Objectives 2

1.3 Thesis Outline 3

2 Background **5**
2.1 Power Amplifier in SISO Transmitter 5
2.1.1 Output Back-Off Efficiency Enhancement Power Amplifiers 8
2.1.2 Linearization 15
2.2 Massive MIMO Architectures and Their Implications on Power Amplifiers . 18
2.2.1 Massive MIMO Architectures 18
2.2.2 Antenna Mutual Coupling and Load Modulation 22
2.2.3 Linearization Complexity 23
2.3 Review on Power Amplifiers for MIMO Architectures 26

3 EM/RF/DSP Co-simulation Framework and Development Testbed **31**
3.1 Introduction 31
3.2 EM/RF/DSP Co-Simulation Framework 32
3.2.1 Baseband DSP 33
3.3 Development Transmitter Front-End Testbed 35
3.3.1 **2×2** Transmitter Front-End 35
3.3.2 **4×4** Transmitter Front-End 39
3.4 Conclusion and Outlook 74

4 Model-Assisted Emulation for Effective Design and Testing of Power Amplifiers in Massive MIMO Transmitters **75**
4.1 PA Modeling for Fast and Accurate Synthesis of Reflected Signals at PA Outputs 77
4.2 Proposed Model-Assisted Emulation Procedure 80
4.3 Validation of the Proposed PA model-assisted Emulation Approach 84
4.3.1 Application to PA-Array Co-Simulations 84
4.3.2 Application to PA Testing 87
4.4 Towards Modulated-signal-based PA Design in the Presence of Antenna Mutual Coupling using the Proposed PA Model-based Emulation 96
4.4.1 Design Methodology 96
4.5 Conclusion 101

5 Complexity-Reduced Precoding-Aware Multiple-Input Single-Output Digital Predistortion for Fully Digital Massive MIMO Transmitters **102**

5.1 Proposed Reduced-Dimensional Precoding-Aware MISO DPD Formulation 105

5.1.1 System Model . . . 105

5.1.2 Existing DPD Models . . . 106

5.1.3 Proposed Reduced-Dimensional MISO DPD Model . . . 107

5.1.4 Comparative DPD Complexity Analysis . . . 109

5.1.5 Proposed R-MISO DPD Training Approach . . . 110

5.2 Experimental Validation . . . 112

5.2.1 Measurement Setup and Signal Configuration . . . 112

5.2.2 R-MISO DPD Measurement Results . . . 113

5.2.3 Comparative DPD Training Approach Analysis . . . 116

5.2.4 Discussion . . . 117

5.3 Conclusion . . . 117

6 Conclusion and Future Work **118**

6.1 Future Work . . . 119

References **120**

APPENDICES **128**

A Analysis of Linearizability of the Balanced Power Amplifier in a MIMO Transmitter **129**

A.1 Application of the Co-Simulation Framework to the Analysis of Balanced PA-Based **2×2** and **4×4** MIMO Transmitter . . . 129

A.1.1 Background Theory . . . 130

A.1.2 Matching Network Analysis . . . 133

A.1.3 16-Channel Fully Digital MIMO Transmitter Co-Simulation . . . 137

B Massive MIMO Transmitter Front-End Analysis **138**

B.1 Introduction 138

B.2 Analysis of the **4×4** MIMO Transmitter Front-End Non-Linearity as a Function of the Level of Antenna Mutual Coupling 139

B.3 Analysis of the **4×4** MIMO Transmitter Front-End Active Load Modulation for Uncorrelated Modulated Input Signals as a Function of the Level of Antenna mutual coupling 140

B.4 Analysis of the Active Load Modulation for Precoded Modulated Input Signals 145

B.5 Conclusion 148

C Preliminary 2×2 Full Measurement-based Emulation **149**

D A Two-Tier Low-Complexity Linearization Architecture for Fully Digital mMIMO Transmitters **154**

E On the Viability of Using a Subset of Transmitter-Observation Receivers for Training a Common DPD in Fully Digital MIMO Transmitters **159**

F Digital Control Source Code **164**

List of Figures

2.1 **Diagram of general PA design.** V_{GS}: Gate Voltage, V_{DS}: Drain Voltage, IMN: Input Matching Network, OMN: Output Matching Network. 5

2.2 **DCIV Curves for Different PA Operating Classes.** 6

2.3 **Outphasing Power Amplifier.** 9

2.4 **Supply Modulation Efficiency Enhancement Power Amplifier.** . . 11

2.5 **Doherty PA Active Load Modulation.** 12

2.6 **Doherty PA Characteristics.** (a) Drain current profiles, (b) Drain voltage profiles, (c) Output impedance profiles. 12

2.7 **Doherty PA Characteristics** (a) Output Power, (b) Drain Efficiency. . . 14

2.8 **Load-Modulated Balanced Power Amplifier** 15

2.9 **Complex Baseband Equivalent RF Chain Block Diagram.** 16

2.10 **Analog Beamforming Architectures.** Reprinted from [1]. 19

2.11 **Hybrid Beamforming Architectures.** Reprinted from [1]. 19

2.12 **Digital Beamforming Architectures.** Reprinted from [1]. 20

2.13 **Detailed Fully Digital MIMO Transmitters Block Diagram.** Reprinted from [2]. 20

2.14 **MIMO Transmitter Front-End.** 22

2.15 **Switch-Based Reconfigurable Matching Network.** Reprinted from [3] 26

2.16 **Varactor-Based Reconfigurable Matching Network.** Reprinted from [4] 27

2.17 **Supply Modulation for Power Amplifier Output Power Correction.** Reprinted from [5] 27

2.18 **A Block Diagram of Balanced-to-Doherty Power Amplifier.** (a) Balanced mode, (b) Doherty mode. Reprinted from [6] 28

2.19 **A Schematic of Orthogonal Load-Modulated Balanced Power Amplifier.** Reprinted from [7]. 29

3.1 **Proposed Co-simulation Framework.** Figure reprinted from [8]. 32

3.2 **Envelope Simulation.** Reprinted from [9]. 34

3.3 **2×2 Fully Digital MIMO Transmitter Front-End** (a) Fully assembled modular fully digital MIMO Transmitter Front-End, (b) Class-AB PA, (c) Doherty PA. Reprinted from [10]. 35

3.4 **2×2 Array Measurement Setup.** Reprinted from [8] 36

3.5 **Spectra of Combined Signals for 2×2 Fully Digital MIMO Transmitter using class-AB PAs. (a) Without DPD, (b) With Single Input, Single Output (SISO) DPD, (c) With Dual Input, Single Output (DISO) DPD.** 38

3.6 **Spectra of Combined Signals for 2×2 Fully Digital MIMO Transmitter using Doherty PAs. (a) Without DPD, (b) With SISO DPD, (c) With DISO DPD.** 39

3.7 **Multi-stage PA Line-up.** QPA9127, HMC1122, and QPA9842 form a variable gain pre-driver stage and provide linear input power to a driver stage (first Wolfspeed 6 W transistor). The last Wolfspeed (now MACOM) 6 W transistor is a PA, designed to output a peak power of 38 dBm. 40

3.8 **Ideal Transistor Model.** Transistor operation mode is selected by setting gate bias (V_{GS}), drain bias (V_{DS}), and a loadline (Z_L). 40

3.9 **Ideal Transistor DCIV Curves.** Plots are normalized to I_{max} and V_{mas}. (a) IDS vs. VGS, (b) IDS vs. VDS 41

3.10 **Real Transistor Large-Signal Model.** 41

3.11 **Large-Signal G_m Simulation Setup.** By short-circuiting the drain terminal, the non-linearity of G_m can be characterized. C_{out} can be de-embedded later and used as a part of an output matching network. 42

3.12 **Commercial Packaged Transistor Large-Signal Model.** (a) Intrinsic plane large-signal transistor model, (b) Packaged transistor model. 42

3.13 **DCIV Loadline.** (a) DCIV loadline of Cree's CGHV1F006S GaN transistor, (b) Load values for each loadline in (a), (c) P_{out} and η for each loadline. 43

3.14 **Transistor Stabilization Network.** Transmission lines are for the transistor contacts. 50 Ω and 1.4 pF capacitor are for in-band stabilization and 100 Ω in the gate bias line is for low frequency stabilization. 44

3.15 **Stability Simulation Results.** (a) Stability Factor and Maximum Available Gain, (b) Source and Load Stability Circles. 45

3.16 **Packaged Transistor.** 46

3.17 **PA Output Matching Impedances at the Intrinsic Device Plane.** For $P_{out} \geq 38.0$ dBm and $\eta \geq 68\%$, the real impedances of the fundamental frequency bands is around 111 Ω. For Class-AB, harmonic frequency band impedances need to be short-circuited. However, the phases of second harmonic impedances affect both P_{out} and η so their phase values can be offset from 0 in real transistors. 47

3.18 **PA Output Matching Network Schematic.** 47

3.19 **PA Output Matching Network Schematic.** 48

3.20 **PA Input Matching Impedances at the Intrinsic Device Plane.** . . 49

3.21 **PA Input Matching Network Schematic.** 49

3.22 **PA Performance**. (a) Gain, (b) Phase distortion, (c) Drain Efficiency. . . 50

3.23 **Initial Ideal Driver Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step. 51

3.24 **Driver-PA Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step. 52

3.25 **Multistage-PA Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step. 53

3.26 **Final Multistage-PA Schematic.** 54

3.27 **Final Multistage-PA Layout.** 1. Qorvo QPA9127, 2. Analog Devices HMC1122, 3. Qorvo QPA9842, 4. Wolfspeed CGHV1F006S Driver Transistor, 5. Wolfspeed CGHV1F006S PA Transistor, 6. Anaren X4C40K1-30S 30 dB Directional Coupler. 55

3.28 **AMC7834 DAC/ADC Module for Gate Biasing and Drain Current Reading.** This chip is serially controlled via SPI protocol. VDD5V is 5 V supply for both digital and analog circuits. IOVDD is 3.3 V for setting digital logic levels. VSS5V is -5 V for negative analog supply and the PAD. These voltages are all provided from a single 6 V supply using a voltage inverter and low-dropout voltage regulators. 56

3.29 **Multi-stage PA Modulated-Signal-Based Co-simulation Results**. (a) AM/AM, (b) AM/PM, (c) Spectrum. Blue is before applying DPD and red is after applying DPD 58

3.30 **Multi-stage PA and AMC7834 DAC/ADC system.** (a) Standalone Multi-stage PA and AMC7834 DAC/ADC system, (b) Integrated Multi-stage PA and AMC7834 DAC/ADC system. 59

3.31 **Single-channel Multi-stage PA Small-signal Measurement Results.**. (a) S21, (b) S22. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results. HMC1122 attenuator was set to 12 dB. 60

3.32 **Multi-stage PA DPD Measurement Results @ 3.4 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results. 62

3.33 **Multi-stage PA DPD Measurement Results @ 3.5 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results. 62

3.34 **Multi-stage PA DPD Measurement Results @ 3.6 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results. 63

3.35 **1×4 PA Array Design for 16-Channel Fully Digital MIMO Transmitter Front-End.** Each PA is separated by 60 mm for 0.7 λ antenna spacing at 3.5 GHz. 64

3.36 **DC Power Distribution Board.** (a) Connections to 1×4 PA Array, (b) Connections to external DC power supplies and jumpers for reconfigurability. 65

3.37 **Digital Control Board.** NI USB-8452 is used as the digital control interface board. 66

3.38 **Reference Antenna Design** (a) top view, (b) trimetric view, (c) right view. Reprinted from [11]. 67

3.39 **New Antenna Design Dimensions.** (a) Driven patch antenna, (b) Parasitic patch antenna, (c) Bottom feed network, (d) Aperture between bottom and top feed networks, (e) Top feed network. 68

3.40 **Fabricated New Antennas.** (a) Driven patch antenna, (b) Parasitic patch antenna, (c) Feed network, (d) Assembled stacked patch antenna. 69

3.41 **Fabricated Antenna Characterization.** Port 1 is the bottom feed and port 2 is the top feed. 70

3.42 **Designed Mechanical Frame.** 72

3.43 **Designed Mechanical Frame.** 73

3.44 **Fully Asssembled** 4×4 **MIMO Transmitter.** 74

4.1 Block diagram of a fully digital mMIMO transmitter for PA linearization and output impedance analysis. 77

4.2 Block diagram of the proposed PA model-assisted emulation system for predictive PA design and testing. 80

4.3 Proposed PA model-assisted emulation procedure for PA simulation and testing. 81

4.4 Proposed PA model-assisted emulation procedure with linearizability assessment step in PA simulation and testing. 81

4.5 Comparison of simulated PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model. 85

4.6 Spectra of selected PAs after SISO and DISO DPD in the 4×4 MIMO transmitter front-end using a memoryless SISO PA model in simulation. 86

4.7 Comparison of simulated PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model. 87

4.8 Spectra of selected PAs after SISO and DISO DPD in the 4×4 MIMO transmitter front-end using a memoryless DISO PA model in simulation. 88

4.9 Fully assembled and building blocks of the 4 × 4 MIMO transmitter front-end. (a) single-chain multistage PA, (b) dual-polarization stacked patch antenna element, (c) 1 × 4 multistage PA module, and (d) 4 × 4 antenna array mounted on the mechanical frame. 89

4.10 Measured 4×4 stacked patch antenna array performance. Left: self-reflection; right: mutual coupling. 89

4.11 Full 4×4 MIMO transmitter system and proposed PA model-assisted emulation measurement setups. (a) full-system block diagram, (b) side view, (c) front view, (d) back view in anechoic chamber, (e) emulation setup block diagram, (f) emulation measurement setup. 90

4.12 Comparison of measured PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model. 92

4.13 Comparison of measured representative PA spectra in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model for PAs with good (PA 3) and poor (PA 14) agreement. 93

4.14 Comparison of measured PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model. 94

4.15 Comparison of measured representative PA spectra in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model for PAs with good (PA 3) and poor (PA 14) agreement. 95

4.16 **Peak output power load-pull plots using CW signals on the left and modulated signals on the right.** The powers are in the unit of dBm. 97

4.17 **Average output power contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD. 98

4.18 **Peak efficiency contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD. . . 98

4.19 **RNMSE contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD. 99

4.20 **Lower ACPR contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD. . . 99

4.21 **Upper ACPR contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD. . . 100

5.1 Block diagram of fully digital mMIMO transmitter with (a) DISO DPD (b) MISO DPD (c) precoding-aware two-tier SISO DPD 104

5.2 Block diagram of fully digital mMIMO transmitter with proposed reduced-dimensional precoding-aware MISO DPD 105

5.3 Comparative linearization complexity analysis. Linearization coefficients include both DPD and linear filters per PA for: (a) various number of transmit branches for a 2-user scenario, and (b) various number of users for 256 transmit branches 109

5.4 Reflection coefficients at each PA output in a 2×2 fully digital MIMO transmitter under 4-QAM, 100 MHz signals: (a) uncorrelated excitation, (b) conventional zero-forcing (ZF) precoding, (c) equal-power ZF (EP-ZF) precoding. 111

5.5 (a) 1×4 multistage PA module, (b) Full 4×4 MIMO transmitter front-end inside a 4×5×5 m^3 anechoic chamber, (c) Full 4×4 MIMO transmitter front-end inside a 4×5×5 m^3 anechoic chamber 112

5.6 Total radiated power measurement results before and after DPD using precoded signals for both DPD training and verification: (a) RNMSE, (b) ACPR. 114

5.7 Measured power spectral density before and after linearization for: (a) TRP, (b) User 1. 115

A.1 **Balanced PA Schematic.** 130

A.2 **Balanced PA Gain Curves.** (a) Top PA, (b) Bottom PA, (c) Total Ouptut. Different colours represent different output load impedances. . . . 131

A.3 **Balanced PA Impedances.** 132

A.4 **Balanced PA Schematic with Matching Networks.** 133

A.5 **Balanced PA Γ_{out} Comparison.** (a) Current profiles, and (b) Impedances. 135

A.6 **Effect of Choice of Γ_{out} for Balanced PA.** (a) $\Gamma_{out} = 0.52e^{j26.4^\circ}$, and (b) $\Gamma_{out} = 0.058e^{j112.4^\circ}$. 136

A.7 **DPD Applied on Class-AB PA and its balanced configuration in a 4×4 Fully Digital MIMO Transmitter for Various Maximum Antenna mutual coupling Magnitude.** (a) RNMSE and (b) ACPR. . . . 137

B.1 **DPD Applied on Class-AB PA in a 4×4 Fully Digital MIMO Transmitter for Various Maximum Antenna Mutual Coupling Magnitude.** (a) RNMSE and (b) ACPR. 139

B.2 **Time-Domain OFDM Symbol.** Reprinted from [12]. 140

B.3 **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average Voltage Standing Wave Ratio (VSWR) circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). 142

B.4 **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −25 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). 143

B.5 **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −35 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). 144

B.6 **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling with Conventional Zero-Forcinsg.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). As the modulation order increases, the impedance plots appear similar to those under uncorrelated signals. 146

B.7 **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling with Equal-Power Zero-Forcinsg.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). As the modulation order increases, the impedance plots appear similar to those under uncorrelated signals. 147

C.1 **Full 2×2 MIMO Transmitter Measurement Setup.** Red box: antenna array and directional couplers as transmitter observation receiver (TOR), Blue box: class AB PA and driver, Green box: signal generator and oscilloscope, White box: DC power supplies 150

C.2 **Active Load-Pull-Based Emulated MIMO Transmitter Measurement Setup.** 151

C.3 **PA Output Spectra Under 100 MHz, 8 dB PAPR OFDM, 256 QAM OFDM Signal.** 152

C.4 **PA Sub-Carrier Load Impedance Variation Across 100 MHz Bandwidth.** (a) Simulation, (b) Measurement. The results are normalized to 50 Ω. 153

List of Tables

2.1 **Characteristics of Different PA Operating Classes.** 8

3.1 **Comparison of System-Level Simulation and Measurement With and Without SISO & DISO DPD.** 37

3.2 **Multi-stage PA Linearization Results.** 58

3.3 **Prototype Multi-stage PA Linearization Results.** 100 MHz OFDM signal with 8 dB PAPR and an oversampling rate of 5 were used. 61

3.4 **Antenna PCB Stack-up Information.** Antenna feed and antenna patches are designed on this stack-up and they are stacked to make a 12-layer antenna. 67

4.1 Summary of the required number of measurements/simulations for the proposed PA model-assisted emulation approach and in [13]. K: total number of PAs; N: number of $v^{-}_{out,i}[n]$ convergence iterations (only needed for the DISO PA model in the proposed approach); M: number of DPD iterations. 83

4.2 PA model basis functions used for co-simulation and testing. SISO and DISO PA models are memoryless. 84

5.1 Comparison of DPD linearization results using different combinations of uncorrelated and precoded signals for DPD training and verification. [1]Starting TRP-RNMSE is 15.2% in [14]. 116

B.1 **Comparison of the effect of antenna mutual coupling factor for 16-chain fully digital MIMO transmitter, driven with uncorrelated 100 MHz, 8 dB PAPR OFDM signals.** 141

B.2 **Comparison of the effect of precoders for 16-chain fully digital MIMO transmitter, driven with 256-QAM, 100 MHz, 8 dB PAPR OFDM signals.** 145

C.1 **Comparison of Full MIMO Transmitter Performance and MIMO Emulation Configuration Employing Class AB PA under 100 MHz, 8 dB PAPR OFDM, 256 QAM OFDM Signal.** 151

List of Abbreviations

5G Fifth Generation vii, 1, 18, 19

ACPR Adjacent-Channel Power Ratio 23, 37, 59, 60, 84, 91–93, 100, 117, 137, 139, 141, 148–150

ADC Analog-to-Digital Converter 57, 59

ADS Advanced Design System 32–34, 139

AWG Arbitrary Wave Generator 33, 36, 37, 40, 53, 91

CBBE Complex Baseband Equivalent 16

CMOS Complementary Metal-Oxide Semiconductor 29

CRV Complexity-Reduced Volterra 18, 60, 149

CSI Channel State Information 21, 103, 119

CTMM Crosstalk and Mismatch 23, 106

CW Continuous-Wave 1, 2, 5, 8, 76, 97, 140

DAC Digital-to-Analog Converter 57, 59, 77, 88, 91

DDR Dynamic Deviation Reduction-Based Volterra 18

DISO Dual Input, Single Output xv, xxiii, 23, 37–39, 76, 79, 80, 82–86, 91–93, 103, 106, 138

DPA Dohery Power Amplifier 9, 14, 15, 28–30, 36, 37, 74

DPD Digital Pre-Distortion vii, xv, 15–18, 21, 23, 24, 26, 32, 33, 35, 37–39, 56, 59, 60, 71, 75–77, 82–87, 91–94, 100–103, 105, 106, 117–119, 129, 132, 137–141, 148–150

DSP Digital Signal Processing vii, 3, 20, 23, 26, 29, 31–34, 74, 119, 140

DUT Device Under Test 35, 91, 92, 94

EER Envelope Elimination and Restoration 9–11

EM Electromagnetic 3, 31, 32, 74, 76, 77, 84, 94, 117, 118

ET Envelope Tracking 9–11

FPGA Field-Programmable Gate Array 40, 53, 91

GPIO General Purpose Input/Output 65

HFSS High-Frequency Structure Simulator 32

IC Integrated Circuit 26, 29, 30

IMN Input Matching Network 5, 8

LMBA Load-Modulated Balanced Power Amplifier 14, 15, 29

MDDR Modified Dynamic-Deviation Reduction 59, 60

MIMO Multiple Input, Multiple Output vii, xv, 1–4, 18–22, 24–27, 31, 32, 35, 36, 38, 39, 57, 61, 63, 71, 74–78, 82, 84, 88, 91, 94, 101–103, 105, 106, 117–119, 129, 137, 138, 140, 141, 145, 148, 149, 151

MISO Multiple Input, Single Output 75, 117, 119, 137, 139, 141, 148

MMSE Minimum Mean Square Error 21

MP Memory Polynomial 16, 18

MRT Maximum Ratio Transmission 21

OBO Output Back-Off 9–11, 13–15, 27–30

OFDM Orthogonal Frequency-Division Multiplexing 8, 37, 57, 60, 76, 84, 100, 140, 141, 145

OMN Output Matching Network 5, 8, 26

PA Power Amplifier vii, xv, 1–11, 14–18, 21–33, 35–41, 43–50, 53, 56, 57, 59–61, 63–65, 70, 71, 74–80, 82–88, 90–94, 100–103, 105, 106, 117–119, 129, 130, 132–134, 137–141, 145, 148–151

PAPR Peak-to-Average Power Ratio 8, 9, 18, 37, 60, 77, 84, 87, 91, 141, 145, 149

RF Radio Frequency vii, 1–3, 5–7, 10, 15, 17–21, 24, 25, 31, 35, 36, 39–41, 48, 53, 65, 74–77, 84, 91, 95, 117, 118, 145, 149

RNMSE Root Normalized Mean Square Error 37, 84, 91–93, 100, 137, 139, 149

SISO Single Input, Single Output xv, 3, 5, 21, 23, 24, 26, 37–39, 79, 80, 82–86, 91–94, 102, 132, 137–139, 141, 148

SNR Signal-to-Noise Ratio 21

SPI Serial Peripheral Interface 57, 65

TOR Transmitter Observation Receiver 24, 119

VGA Variable Gain Amplifier 53

VSWR Voltage Standing Wave Ratio xxi, xxii, 23, 25–27, 30, 141–148

ZF Zero-Forcing 21

Chapter 1

Introduction

1.1 Motivation

Prior to the introduction of the 5G standard, communication systems predominantly operated with narrowband modulated signals (typically ≤ 20 MHz) and commonly required only a single RF transmit chain. In such architectures, PAs, which represent the most power-consuming components in the transmitter chain, were conventionally designed under fixed 50 Ω input and output terminations and characterized using Continuous-Wave (CW) excitations at discrete frequency points within the operating bandwidth. Although practical antennas do not present a constant 50 Ω impedance over frequency, this mismatch was traditionally mitigated by inserting a bulky external isolator between the PA and the antenna. The isolator ensured that the PA output experienced an approximately constant 50 Ω load, thereby preserving the validity of the conventional design assumptions.

In contrast, 5G and beyond communication systems rely on wideband modulated signals (typically ≥ 100 MHz) in conjunction with MIMO techniques to achieve higher data rates, reduced latency, and enhanced coverage. The implementation of MIMO systems necessitates a high degree of integration among multiple transmitter chains in order to satisfy size, cost, and efficiency constraints. This stringent integration requirement generally leads to the removal of external isolators following the PAs. Consequently, antenna impedance mismatches and mutual coupling effects directly influence the instantaneous output impedance seen by each PA.

Under these conditions, complex interactions arise among the various system building blocks, including signal correlation across transmitter paths, intrinsic PA nonlinear

behavior, antenna impedance mismatches, and mutual coupling within the array. These interactions collectively determine the overall performance of the massive MIMO transmitter. As a result, the conventional PA design methodology, which assumes fixed 50 Ω terminations and relies on CW characterization, is no longer sufficient to accurately predict system-level behavior. Therefore, a revised PA design methodology is required, in which the primary objective shifts from optimizing performance under idealized 50 Ω conditions to optimizing overall system-level performance within the realistic operating environment of a massive MIMO transmitter.

1.2 Thesis Objectives

In this thesis, a systematic methodology is developed to enhance the understanding of massive MIMO transmitter behavior, with particular emphasis on the fully digital MIMO architecture, and to improve the design strategies of individual transmitter building blocks for optimizing overall system-level performance under realistic signal excitations. The fully digital MIMO architecture is selected as the primary focus of this work because, while it offers the highest degree of flexibility among beamforming architectures, it also introduces additional challenges beyond those encountered in hybrid and RF beamforming systems.

To establish a comprehensive design framework, theoretical analysis, numerical simulations, and hardware prototype development are jointly considered. First, constructing a large-scale massive MIMO prototype solely for exploratory system-level investigations is both technically challenging and economically impractical, particularly as the number of transmitter chains increases. Therefore, an accurate and computationally efficient simulation environment for a MIMO transmitter must be developed to reliably predict system behavior during the design phase. Such a simulation platform enables controlled evaluation of signal correlation effects, load modulation, nonlinear distortion, and precoding strategies prior to hardware implementation.

Second, a MIMO transmitter testbed is developed to experimentally validate the proposed simulation framework. By comparing simulation results with measured data, the accuracy and limitations of the modeling assumptions can be assessed. In practice, discrepancies between simulations and measurements may reveal additional phenomena that are not captured in idealized models, since simulations typically assume identical PAs and antenna characteristics across all channels, whereas real hardware exhibits device-to-device variations and fabrication tolerances.

Finally, insights gained from both simulation and experimental investigations are utilized to formulate an improved transmitter building block design methodology. The objec-

tive is to move beyond isolated component-level optimization and instead achieve optimal system-level performance for fully digital massive MIMO transmitters operating under realistic wideband and precoded signal conditions.

The objectives of this thesis are the following:

1. Development of modulated-signal-based Electromagnetic (EM)/RF/DSP co-simulation framework for MIMO transmitter front-end and 4×4 fully digital MIMO transmitter front-end testbed
2. Testing and analysis of uncorrelated-signal-based and precoded-signal-based PA linearization architectures and techniques using various digital predistortion models on the developed 4×4 fully digital MIMO transmitter front-end testbed
3. Development of single PA-based MIMO transmitter emulation platform for realistic PA design and testing of massive MIMO transmitters

1.3 Thesis Outline

Chapter 2 provides the necessary theoretical and technical background to facilitate understanding of the subsequent chapters. It begins with an overview of various classes of PAs traditionally designed for SISO transmitters, highlighting their operating principles and performance trade-offs. This discussion establishes the foundation for explaining how PA behavior can degrade when deployed in massive MIMO transmitters, where dynamic load modulation and signal correlation effects are present. The chapter further reviews recent PA design methodologies proposed to mitigate performance degradation in massive MIMO systems, thereby positioning the contributions of this thesis within the current state of the art.

Chapter 3 introduces the development of the proposed modulated-signal-based EM/RF/DSP co-simulation framework and its experimental validation using a four-channel fully digital MIMO transmitter front-end. The structure, implementation details, and modeling assumptions of the co-simulation environment are described comprehensively. Building upon the validated framework, the chapter then presents a systematic design procedure for constructing a sixteen-channel dual-polarization fully digital MIMO transmitter front-end testbed. The verified co-simulation platform serves as a predictive tool throughout the hardware development process.

Chapter 5 focuses on the experimental evaluation and comparative analysis of different PA linearization architectures and techniques implemented on the developed 4×4 fully digital MIMO transmitter front-end testbed. The investigation begins with a conventional linearization procedure based on uncorrelated modulated signals. Subsequently, two alternative linearization approaches that incorporate precoded signals are proposed, experimentally validated, and critically analyzed. The impact of precoding on load modulation and linearization effectiveness is examined in detail.

Chapter 4 presents a novel platform for the design of system-level PAs optimized for massive MIMO transmitters. The proposed emulation platform utilizes a single transmitter chain in conjunction with two signal generators to replicate the PA operating conditions encountered in massive MIMO environments. By emulating load modulation and signal correlation effects without requiring the construction of a complete large-scale array, this approach significantly reduces development complexity and cost. The platform enables PA design that explicitly accounts for the non-idealities inherent in massive MIMO transmitters.

Finally, Chapter 6 summarizes the main contributions of this thesis and outlines potential directions for future research, including advancements in system-level PA optimization, improved linearization techniques, and scalable modeling methodologies for next-generation massive MIMO systems.

Chapter 2

Background

2.1 Power Amplifier in SISO Transmitter

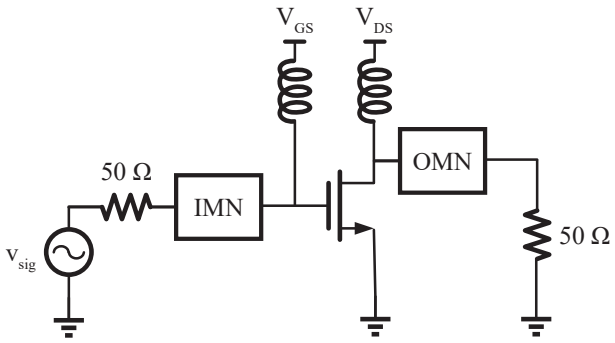


Figure 2.1: **Diagram of general PA design.** V_{GS}: Gate Voltage, V_{DS}: Drain Voltage, IMN: Input Matching Network, OMN: Output Matching Network.

RF PA is an integral nonlinear device at transmitters for emitting high-power RF signals for wider coverage. Fig. 2.1 shows a simplified diagram of a PA structure in a SISO transmitter. Both source and load impedances are fixed to be 50 Ω. For the given transistor selection, the PA is about the design of both Input Matching Network (IMN) and Output Matching Network (OMN) and of gate and drain bias lines, under Continuous-Wave (CW) excitations, to transform fixed 50 Ω to desired values at the input and output of the transistor. These are designed for three main design parameters: efficiency (η), linearity, and output power (P_{out}) and they all affect each other. If P_{out} is low, PA is linear but has

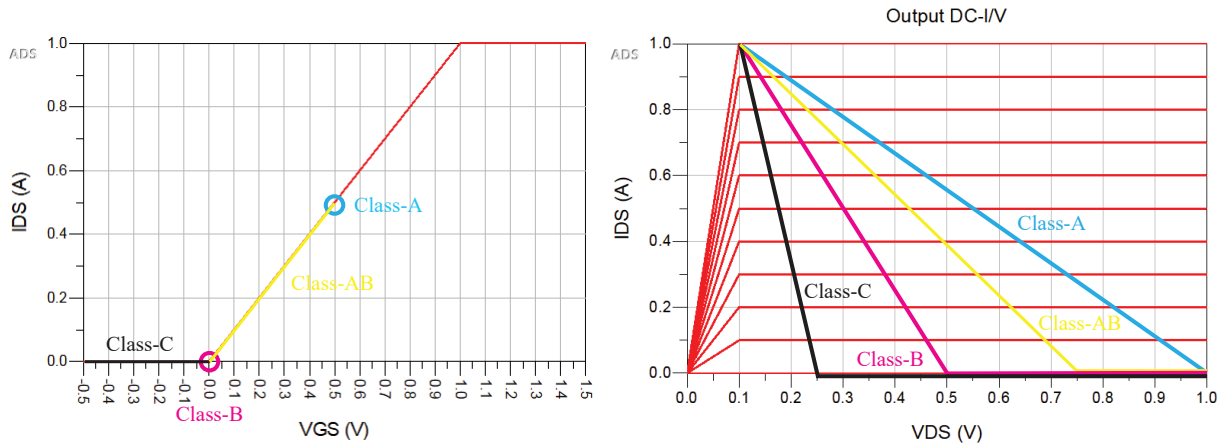


Figure 2.2: **DCIV Curves for Different PA Operating Classes.**

poor efficiency. On the other hand, if P_{out} is at or close to saturation, PA is nonlinear but has high efficiency. The trade-offs have to be considered for different applications.

Different PA classes of operations, such as class-A, -AB, -B, and -C, can be selected for the desired performances on the main parameters. Fig. 2.2 shows DCIV behavior of a transistor and that the choice of the gate voltage can determine the PA operating class. DC and RF powers, and drain efficiency can be calculated from fig. 2.2 as follows [15]:

$$P_{DC} = V_{DS,DC} I_{DS,DC} \tag{2.1}$$

$$P_{out,RF} = \frac{1}{2}\mathrm{Re}\{V_{DS,fund} I^*_{DS,fund}\} \tag{2.2}$$

$$\eta = \frac{P_{out,RF}}{P_{DC}} \tag{2.3}$$

where $V_{DS,fund}$ and $I_{DS,fund}$ are output voltage and current signals at the fundamental frequency, respectively.

For analyzing general classes of operations, time-domain $I_{DS}(\omega t)$ must be Fourier-transformed to calculate different harmonics components because it can be distorted. The time-domain signal can be represented as follows:

$$I_{DS}(\omega t) = \frac{I_{DS,max}}{1 - \cos(\theta/2)}(\cos(\omega t) - \cos(\theta/2)) \tag{2.4}$$

where $\cos(\theta/2) = -\frac{I_{DS,Q}}{I_{DS,peak}}$, $I_{DS,peak} = I_{DS,max} - I_{DS,Q}$, $I_{DS,max}$ is the maximum amplitude in time domain, $I_{DS,Q}$ is the quiescent point, and θ is the conduction angle of $I_{DS}(\omega t)$. Using this conduction angle, DC and fundamental current values can be calculated:

$$I_{DS,DC} = \frac{I_{DS,max}}{2\pi} \frac{2\sin(\theta/2) - \theta\cos(\theta/2)}{1 - \cos(\theta/2)} \tag{2.5}$$

$$I_{DS,fund} = \frac{I_{DS,max}}{2\pi} \frac{\theta - \sin(\theta)}{1 - \cos(\theta/2)} \tag{2.6}$$

Assuming $V_{knee} = 0$, Class-A PA has fixed $V_{DS,DC} = V_{DS,max}/2$ and $I_{DS,DC} = I_{DS,max}/2$ because gate and drain voltages are selected to be at the middle of voltage and current swings. On the other hand, the output RF voltage and current swings are dependent on the input drive level. With the conduction angle of 2π, the maximum voltage and current swings at the fundamental frequency are also $V_{max}/2$ and $I_{max}/2$, resulting in the maximum η of 50%.

Other classes of operations with reduced conduction angles in $I_{DS}(\omega t)$ have variable DC power consumptions because the conduction angle is a function of the input drive level. Thus, η can be improved compared to Class-A PA. However, these classes also produce harmonic distortions. These can be quenched by presenting short circuits to the transistor output at these harmonic frequencies, allowing only the fundamental I_{DS} to be delivered to the output. A general peak efficiency calculation for different classes of operations then becomes:

$$P_{DC} = \frac{V_{DS,max} I_{DS,DC}}{2} \tag{2.7}$$

$$P_{out,max} = \frac{1}{4} V_{DS,max} I_{DS,fund} \tag{2.8}$$

$$\eta = \frac{P_{out,RF}}{P_{DC}} = \frac{1}{2} \frac{I_{DS,fund}}{I_{DS,DC}} \tag{2.9}$$

The reduced conduction angle introduces a trade-off between linearity and efficiency. With the conduction angle smaller than 2π, the angle changes with the input drive level with the exception of Class-B (fixed at $\theta = \pi$). Thus, the output signal linearity is not preserved over the full signal swing. The inherent nonlinearity in different PA classes of operations can be beneficial to counteract intrinsic transistor nonlinearity. Table. 2.1

Table 2.1: **Characteristics of Different PA Operating Classes.**

PA Class	Linearity	Maximum Theoretical η	Gain	P_{out}
A	Best	50%	Best	Good
AB	Good	50−78.5%	Good	Best
B	Best	78.5%	Not Bad	Good
C	Worst	78.5−100%	Worst	Worst

summarizes characteristics of different classes. For $\eta > 50\%$, PA classes should be taken away from Class-A at the expense of sacrificing linearity, gain, and P_{out}.

After choosing the operating class, load-pull procedure is performed in simulations, under CW excitations, to find what impedance values should be presented at the transistor output for P_{out} and η. Typically, for the maximum P_{out}, the optimal resistance (R_{opt}) can be calculated as follows:

$$R_{opt} = \frac{V_{DS,max}/2}{I_{DS,fund}} \tag{2.10}$$

With the selected output impedances over a desired bandwidth, source-pull is performed to obtain impedances at the transistor input for finalizing gain and linearity. With the selected transistor input and output impedances, IMN and Output Matching Network (OMN) are designed to transform 50 Ω source and load impedances to the input and output impedances, respectively. For real transistors, input and output impedances at second and third harmonics can greatly affect performance. Thus, source- and load-pull are also performed at these frequencies as well. The bias lines are usually designed with matching networks to minimize very low frequency impedances for reducing long-term memory effects. Optimization is used to fine-tune the overall PA performance. Since this conventional PA design uses fixed 50 Ω at the source and the load, any impedance variation at the load can cause significant performance degradation.

2.1.1 Output Back-Off Efficiency Enhancement Power Amplifiers

Spectrally efficient modulation schemes, such as Orthogonal Frequency-Division Multiplexing (OFDM), for high data rate communications cause a high Peak-to-Average Power Ratio (PAPR) in time-domain signals due to a large number of sub-carriers and a high

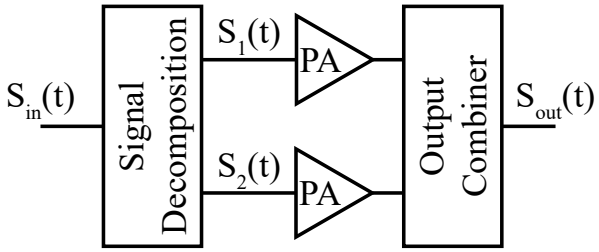


Figure 2.3: **Outphasing Power Amplifier.**

digital modulation order. Consequently, PA mainly operates at an average output power ($P_{avg} < P_{sat}$), causing a siginificant drop in η. Therefore, PA efficiency over the Output Back-Off (OBO) range equal to PAPR needs to be maintained close to the peak η. Four types of OBO efficiency enhancement PAs are popular: outphasing PA [16], Envelope Elimination and Restoration (EER) [17] and Envelope Tracking (ET) [15], Dohery Power Amplifier (DPA) [18], and load-modulated balanced PA [19]. These topologies either change the drain bias to lower P_{DC} or modulate the output impedance to modify P_{RF}.

Outphasing Power Amplifier

Fig. 2.3 shows a block diagram of outphasing PA topology. The signal decomposition block accepts amplitude- and phase-modulated signals and divide them into two signals of constant amplitude, phase-modulated signals as follows:

$$S_{in}(t) = A(t)\cos(\omega t + \phi(t)) = S_1(t) + S_2(t) \tag{2.11}$$

$$S_1(t) = \frac{A_{max}}{2}\cos\left(\omega t + \phi(t) + \cos^{-1}\left(\frac{A(t)}{A_{max}}\right)\right) \tag{2.12}$$

$$S_2(t) = \frac{A_{max}}{2}\cos\left(\omega t + \phi(t) - \cos^{-1}\left(\frac{A(t)}{A_{max}}\right)\right) \tag{2.13}$$

The encoded amplitude modulation information is converted to differential phase modulation after the signal decomposition block. Each PA sees a constant amplitude signal, so it can be set to operate at the maximum output power for the highest efficiency at all times. As a result, a nonlinear, highly efficient PA class of operation can be selected. However, it has a bandwidth limitation. The differential phase modulation is a nonlinear

operation so the signal decomposition block has to process much wider bandwidth than the input signal bandwidth.

The output combiner can be either isolated or non-isolated. The isolated output combiner provides constant resistive loads to both PAs at all input power levels. This allows the output signal, S_{out}, to be linear but the final efficiency can be significantly degraded at back-off levels. For example, if the input signal amplitude is zero, $S_1(t)$ and $S_2(t)$ are 180° out of phase. When combined, S_{out} is also zero while both PAs are still operating at the maximum output powers. Thus, the power have to all dissipate in the output combiner block. Using the isolated output combiner has similar back-off efficiency characteristics as conventional PA classes of operations.

For improving OBO efficiency, a non-isolated output combiner should be implemented. It allows the output impedances of each PA to be modulated as follows:

$$Z_1 = \frac{R_L}{2}\left(1 - j\tan\left(\frac{A(t)}{A_{max}}\right)\right) \tag{2.14}$$

$$Z_2 = \frac{R_L}{2}\left(1 + j\tan\left(\frac{A(t)}{A_{max}}\right)\right) \tag{2.15}$$

With the modulated impedances, each PA no longer outputs maximum power at all input power levels while a high efficiency is maintained at back-off levels. However, the reactive components in Z_1 and Z_2 cause the loadlines to deviate from ideal resistive ones. Thus, parallel reactive components $\mp jX$ are added to make the impedances real-valued at a specific back-off level. Although OBO efficiency enhancement can be achieved with the non-isolated output combiner, linearity is compromised because PA are subject to time-varying output impedances that lead to distortion.

Envelope Elimination and Restoration and Envelop Tracking

EER and ET enhance OBO efficiency by drain supply voltage modulation. The supply modulation method dynamically changes the drain bias of the PA according to the input signal envelope. This can minimize the DC power consumption for the given RF input drive level to keep the peak drain efficiency over all input power range.

Fig. 2.4 shows a block diagram of supply modulation efficiency enhancement PA. EER removes the envelope of the input signal entirely and, similar to the outphasing PA, feeds constant amplitude, phase-modulated signal to the PA. Thus, a nonlinear, highly efficient

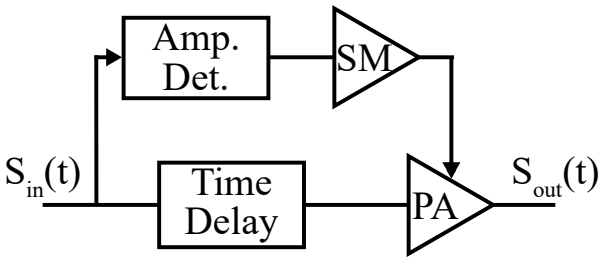


Figure 2.4: **Supply Modulation Efficiency Enhancement Power Amplifier.**

PA can be employed for maximizing efficiency. The extracted envelope is supplied to the supply modulation block to control the drain bias of the PA for restoring the envelope of the input signal. EER technique is sensitive to time delays between two paths for proper alignment, and the dynamic range of the supply modulation block should be large enough to cover all possible envelope values. These limit the bandwidth of the overall PA performance and are potential sources of nonlinearity. On the other hand, ET does not remove the envelope of the input signal to the PA block. It only extracts the envelope information for the supply modulation block and the dynamic range of supply modulation does not have to be as dynamic as the case in EER. However, time alignment issue still exists.

While the OBO efficiency of the PA block is improved, the efficiency of the overall PA is $\eta_{total} = \eta_{PA}\eta_{SM}$. Unless $\eta_{SM} = 100\%$, η_{total} is always degraded. In addition, the bandwidth of the supply modulation block needs to be wider than the input signal bandwidth because the envelop is $\sqrt{I^2 + Q^2}$, where I and Q are in-phase and quadrature components of the input signal. Therefore, the overall PA performance is limited by the supply modulation block and inherently suffers from bandwidth issues.

Doherty Power Amplifier

Doherty PA achieves OBO efficiency enhancement by its robust design without the need for real-time processing. Fig. 2.5 shows the Doherty PA active load modulation scheme at its output. It utilizes two transistors (Class-B main PA and Class-C auxiliary PA) to perform active load modulation on each other to produce the current, voltage, and impedance profiles shown in fig. 2.6.

The analysis of the active load modulation can start from the ABCD matrix of the quarter-wavelength ($\lambda/4$) transmission line and the voltage across the load (R_L):

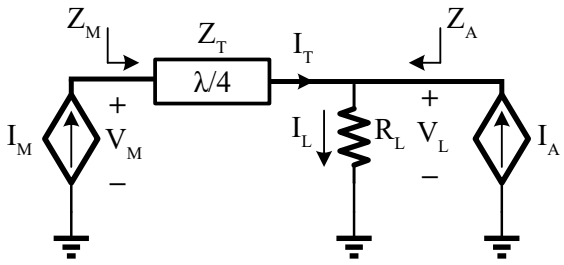


Figure 2.5: **Doherty PA Active Load Modulation.**

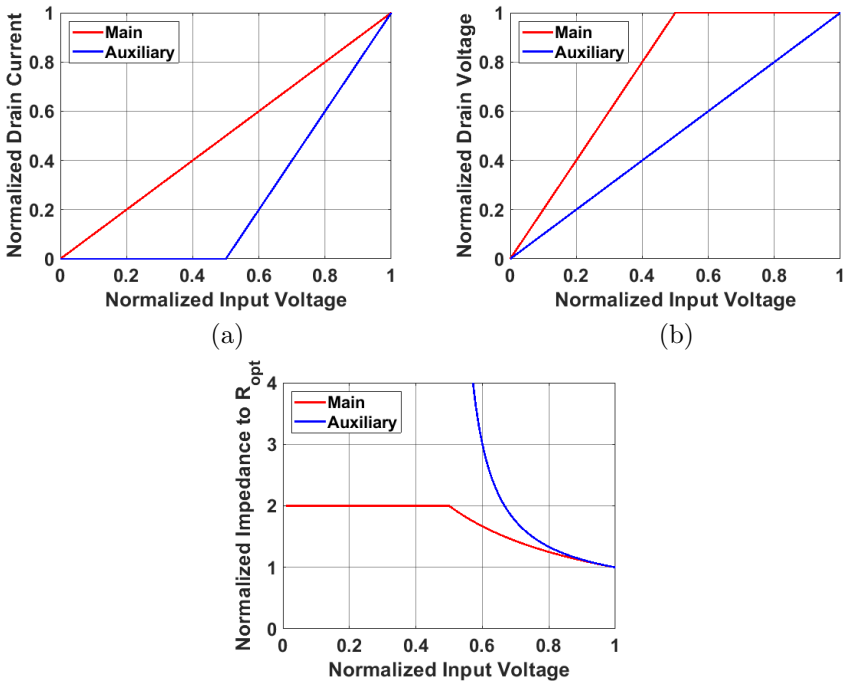


Figure 2.6: **Doherty PA Characteristics.** (a) Drain current profiles, (b) Drain voltage profiles, (c) Output impedance profiles.

$$\begin{bmatrix} \mathbf{V_M} \\ \mathbf{I_M} \end{bmatrix} = \begin{bmatrix} 0 & jZ_T \\ j/Z_T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V_T} \\ \mathbf{I_T} \end{bmatrix} \tag{2.16}$$

$$V_L = R_L(I_T + I_A) \tag{2.17}$$

The effective impedances seen by the transistors can then be calculated as follows:

$$I_T = \frac{Z_T}{jR_L} I_M - I_A \tag{2.18}$$

$$Z_M = \frac{V_M}{I_M} = \frac{jZ_T I_T}{I_M} = \frac{jZ_T(\frac{Z_T}{jR_L} I_M - I_A)}{I_M} = \frac{Z_T^2}{R_L} - jZ_T \frac{I_A}{I_M} \tag{2.19}$$

$$Z_A = \frac{V_L}{I_A} = \frac{R_L(I_T + I_A)}{I_A} = \frac{R_L(\frac{Z_T}{jR_L} I_M - I_A + I_A)}{I_A} = \frac{-jZ_T I_M}{I_A} \tag{2.20}$$

Since the main transistor has a quarter-wavelength transmission line, the ratio between the main and auxiliary current values have 90−degree phase offset. Thus, another Z_o quarter-wavelength transmission line is inserted at the input of the auxiliary transistor to remove j term in Z_M and Z_A. With the added phase offset compensation, the impedances become

$$Z_M = \frac{Z_T^2}{R_L} - Z_T \frac{I_A}{I_M} \tag{2.21}$$

$$Z_A = Z_T \frac{I_M}{I_A} \tag{2.22}$$

The values of Z_T and R_L are design parameters for achieving the impedance profiles in fig. 2.6. At the input voltage level below OBO,

$$I_A = 0 \tag{2.23}$$

$$Z_{M,OBO} = \frac{Z_T^2}{R_L} = 2R_{opt} \text{ for 6 dB back-off} \tag{2.24}$$

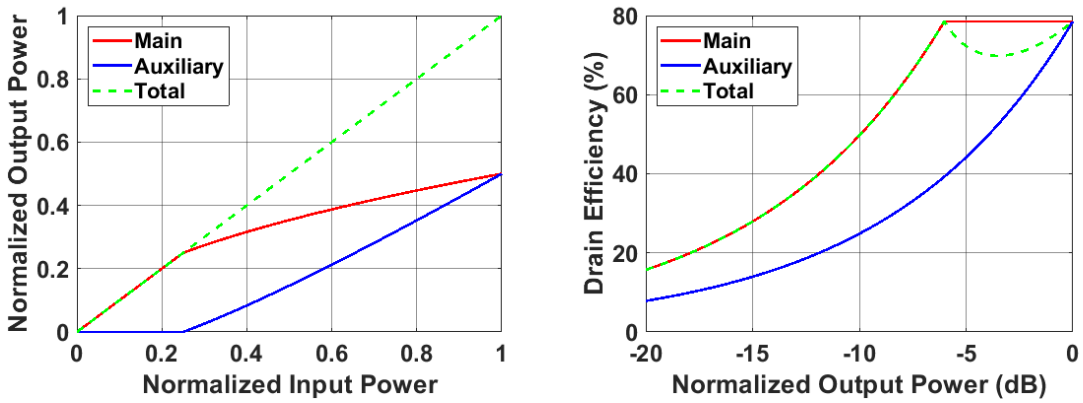
$$Z_{A,OBO} = \infty \tag{2.25}$$

At the peak input voltage level, $I_M = I_A = 1$ for 6 dB back-off.

$$I_M = I_A \tag{2.26}$$

$$Z_{M,sat} = \frac{Z_T^2}{R_L} - Z_T = R_{opt} \tag{2.27}$$

$$Z_{A,sat} = Z_T = R_{opt} \tag{2.28}$$



(a) (b)

Figure 2.7: **Doherty PA Characteristics** (a) Output Power, (b) Drain Efficiency.

These give $Z_T = R_{opt}$ and $R_L = R_{opt}/2$. With proper selection of the design parameters, load modulation behavior of Doherty PA can be achieved, giving the output power and drain efficiency characteristics shown in fig. 2.7. Fig. 2.7(a) shows that Doherty PA is inherently linear and fig. 2.7(b) confirms that a high OBO efficiency can be maintained. The ratio between I_M and I_A can be tuned for a different OBO value but the drop in η between two peak points decreases further in return. Doherty PA also has a narrow bandwidth problem due to the existence of quarter-wavelength transmission line.

Load-Modulated Balanced Power Amplifier

Load-Modulated Balanced Power Amplifier (LMBA) is another load-modulation-based OBO efficiency enhancement topology by adding a control source at the isolated port of the output hybrid-90 coupler of the balanced PA architecture [19]. Fig. 2.8 shows a block diagram of LMBA. When the control source, I_c, is added at the isolated port, the impedances seen by each PA is modulated, similar to DPA. Eq. 2.29 shows that the impedances seen by each PA can be modulated to decrease or increase by setting appropriate phase settings of the control source. This feature is different from DPA, which can only decrease from its back-off impedances.

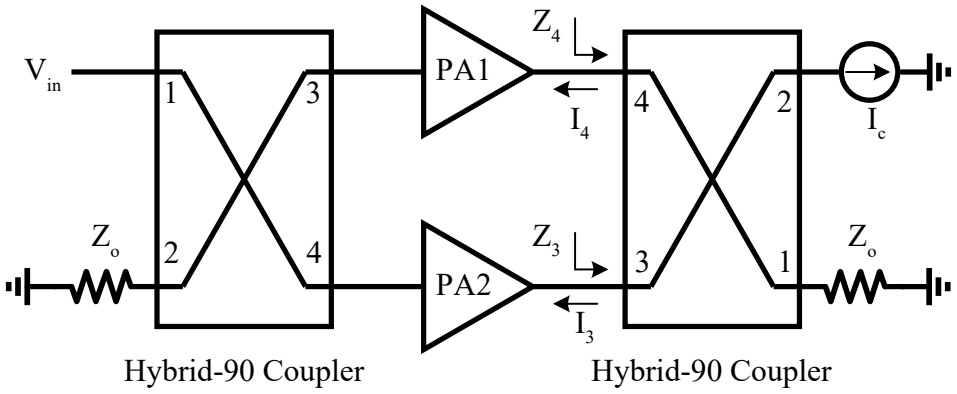


Figure 2.8: **Load-Modulated Balanced Power Amplifier**

$$Z_3 = Z_4 = Z_o \left(1 - \frac{\sqrt{2} I_c e^{j\phi_c}}{I_b}\right) \tag{2.29}$$

$$I_3 = -jI_b \qquad I_4 = I_b \tag{2.30}$$

LMBA also has a flexibility to choose OBO range, similar to three-way DPA [20], by the following relationship [21]:

$$OBO = 1 + 2\left(\frac{I_{b,MAX}}{I_{c,MAX}}\right)^2 \left(1 - \frac{\sqrt{2} I_c}{I_b}\cos(\phi_c)\right) \tag{2.31}$$

LMBA is inherently wideband because the hybrid-90 coupler can be utilized as a wideband matching network. This topology can be modified to operate as a DPA [7], pseudo-DPA [21], and sequential PA [22] by modifying the PA class of operations and current profiles to further improve efficiency and linearity.

2.1.2 Linearization

As linearity and efficiency are trade-offs when designing a PA, it is designed for high efficiency because it has the highest power consumption at the transmitter. Consequently, for a linear output behavior, a DPD module is placed before a PA to counteract its non-linearity. Fig. 2.9 shows an RF chain block diagram that incorporates a DPD module

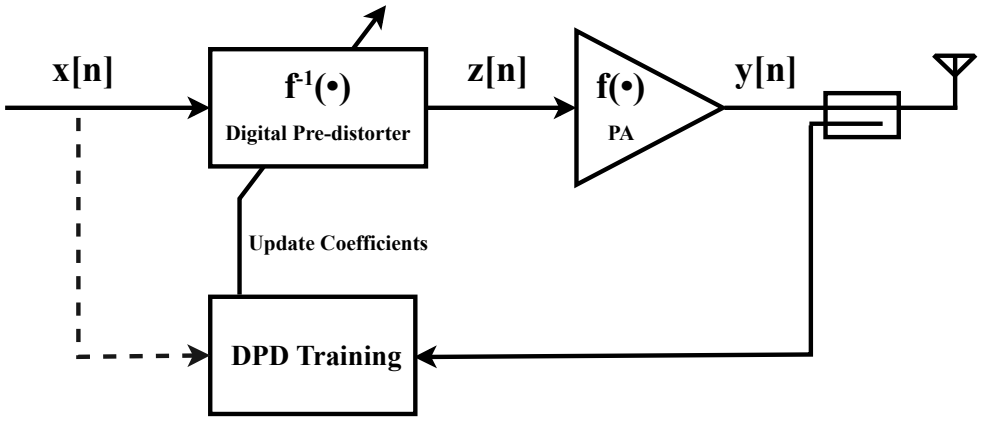


Figure 2.9: **Complex Baseband Equivalent RF Chain Block Diagram.**

for PA linearity. The input ($x[n]$) and output ($y[n]$) signals are used to model the DPD module as the inverse function of the PA behavior such that $y[n] \approx x[n]$. Before applying DPD module, the nonlinear PA behavior is modeled using Volterra series as it can include the memory effect. In continuous-time domain, the Volterra series has the following representation [23]:

$$y(t) = \sum_{n=1}^{\infty} \int_{\mathbb{R}} \int_{\mathbb{R}} \dots \int_{\mathbb{R}} h(\tau_1, \dots, \tau_n) \prod_{j=1}^{n} z(t - \tau_j) d\tau_j \tag{2.32}$$

where $h(\tau_1, \dots, \tau_n)$ is the n-th order Volterra kernel. This can also be represented as the sampled Complex Baseband Equivalent (CBBE) Volterra series in discrete-time domain for digital processing as follows [24]:

$$y[n] = \sum_{k=0}^{P} \sum_{q_1=0}^{M} \dots \sum_{q_k=q_{k-1}}^{M} \sum_{q_{k+1}=0}^{M} \sum_{q_{k+2}=q_{k+1}}^{M} \dots \sum_{q_{2k+1}=q_{2k}}^{M} f(\mathbf{q_{2k+1}}) \prod_{i=1}^{k} z^*[n - q_i] \prod_{j=k+1}^{2k+1} z[n - q_j] \tag{2.33}$$

where $f(\mathbf{q_{2k+1}}) = h[q_1, q_2, \dots, q_{2k+1}]$ is the CBBE Volterra kernel, P the polynomial order, and M the memory depth order.

In empirical settings, the sampled CBBE Volterra series in Eq. 2.33 are difficult to implement. Thus, they require a further simplification. The simplest model is called Memory Polynomial (MP) model with the following formulation [25]:

$$y[n] = \sum_{p=0}^{P-1} \sum_{m=0}^{M} \alpha_{pm} z[n-m]|z[n-m]|^p = \sum_{p=0}^{P-1} \alpha_p \psi_p(\mathbf{z}[n]) \tag{2.34}$$

where $\mathbf{z}[n] = [z[n], \ldots, z[n-m]]$, α_{pm} is complex coefficient representing the Volterra kernel, $\psi(\cdot)$ the basis function, P the polynomial order, and M the memory depth. With this nonlinear PA forward modeling, the DPD model can be of the same form.

$$z[n] = \sum_{p=0}^{P-1} \sum_{m=0}^{M} c_{pm} x[n-m]|x[n-m]|^p = \sum_{p=0}^{P-1} c_p \psi_p(\mathbf{x}[n]) \tag{2.35}$$

This model then simplifies the implementation of DPD to calculating complex coefficients α_p and to properly choosing P and M for the basis functions.

Direct and indirect learning are two techniques for calculating the coefficients. For the former, the DPD training block in fig. 2.9 accepts both $\mathbf{x}$ and $\mathbf{y}$ and utilizes the error between the signals at each iteration for estimating the coefficients [26, 27]:

$$\mathbf{e} = \frac{\mathbf{y}}{G} - \mathbf{x} \tag{2.36}$$

$$\mathbf{c}^{l+1} = \mathbf{c}^l - \mu(\Psi^H \Psi)^{-1} \Psi^H \mathbf{e} \tag{2.37}$$

$$\Psi = \begin{pmatrix} \psi_1(x[n]) & \cdots & \psi_P(x[n]) \\ \psi_1(x[n-1]) & \cdots & \psi_P(x[n-1]) \\ \vdots & \vdots & \vdots \\ \psi_1(x[n-M]) & \cdots & \psi_P(x[n-M]) \end{pmatrix} \tag{2.38}$$

where $\mathbf{c} = (c_0, c_1, \ldots, c_{P-1})$, $0 < \mu \leq 1$ is the update step size, l the iteration step, and G the linear gain of the RF lineup. For the initial step, $\mathbf{c}^1 = (1, 0, \ldots, 0)$.

While the direct learning uses the error between the input and output signals, the indirect learning DPD and PA forward models as $f^{-1}(\cdot)$ and $f(\cdot)$, respectively. It assumes that the pre-inverse $f^{-1}(x[n])$ from the DPD module would give the same result as the post-inverse $f^{-1}(y[n])$ from the DPD training block:

$$z[n] = f_{pre}^{-1}(x[n]) = f_{post}^{-1}(y[n]) \tag{2.39}$$

The coefficients can then be calculated as the following:

$$\mathbf{c} = (\Psi^H \Psi)^{-1} \Psi^H \mathbf{x} \tag{2.40}$$

$$\Psi = \begin{pmatrix} \psi_1(y[n]) & \cdots & \psi_P(y[n]) \\ \psi_1(y[n-1]) & \cdots & \psi_P(y[n-1]) \\ \vdots & \vdots & \vdots \\ \psi_1(y[n-M]) & \cdots & \psi_P(y[n-M]) \end{pmatrix} \tag{2.41}$$

These calculated coefficients are valid only for the input signals that have the same characteristics, including power, PAPR, and bandwidth, for the given *fixed* PA output load impedance. If anything changes in the setup, the coefficients have to be trained again. In addition to MP model, other DPD models, such as Complexity-Reduced Volterra (CRV) [28] and Dynamic Deviation Reduction-Based Volterra (DDR) [29], are also available for implementation.

2.2 Massive MIMO Architectures and Their Implications on Power Amplifiers

2.2.1 Massive MIMO Architectures

With ever increasing data traffic demands, 5G communication networks employ spatial multiplexing to enhance spectral efficiency and overall system capacity. Instead of adopting a conventional broadcasting approach, in which data are radiated uniformly in all directions, a *beamforming* technique concentrates the transmitted energy toward a specific spatial direction. Furthermore, multiple beams can be simultaneously generated to support multiple users within the same frequency band, where each beam can sustain a data rate comparable to that of the broadcasting scheme. Figures 2.10, 2.11, and 2.12 illustrate the three principal beamforming architectures available for constructing MIMO transceivers, namely RF (analog), hybrid, and fully digital beamforming.

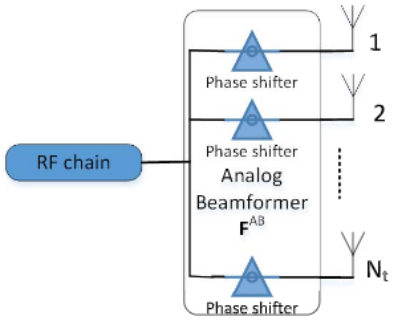


Figure 2.10: **Analog Beamforming Architectures.** Reprinted from [1].

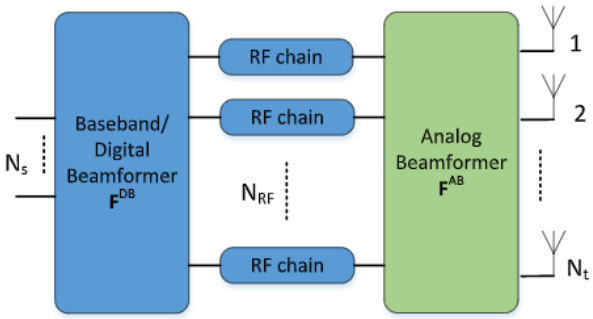


Figure 2.11: **Hybrid Beamforming Architectures.** Reprinted from [1].

An RF beamformer primarily supports line-of-sight communication scenarios. In this architecture, a single input signal is equally divided among N_t antenna elements, and phase shifters are employed to impose constant phase offsets between adjacent antennas in order to generate a single data stream. By appropriately adjusting these phase shifts, the main beam can be steered within approximately $\pm 60°$ around the boresight direction. For two-dimensional antenna arrays, beam steering can also be achieved in both azimuth and elevation. Compared to other beamforming architectures, the RF approach offers reduced hardware complexity; however, multiple RF beamformers are required to generate multiple simultaneous beams. This architecture has been extensively investigated in the millimeter-wave frequency spectrum. Fully digital MIMO transmitters, also referred to as digital beamforming architectures, are actively researched for sub-10 GHz 5G communication systems and constitute the architecture considered in this work. Among the available beamforming solutions, the fully digital architecture provides the highest degree

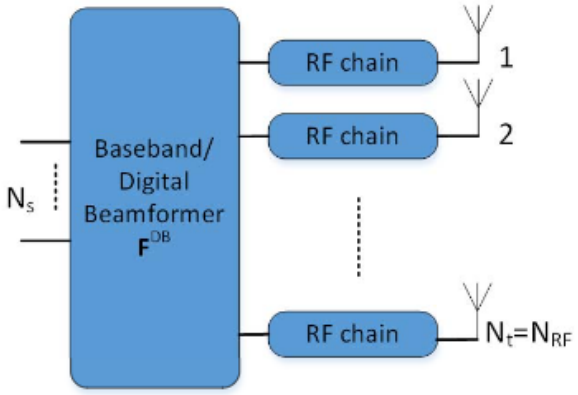


Figure 2.12: **Digital Beamforming Architectures.** Reprinted from [1].

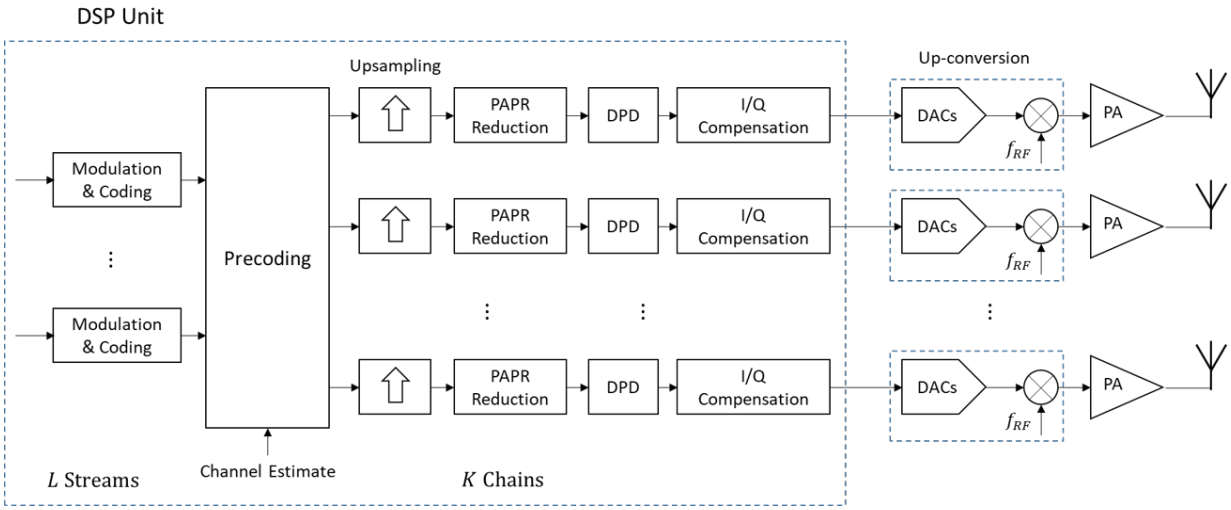


Figure 2.13: **Detailed Fully Digital MIMO Transmitters Block Diagram.** Reprinted from [2].

of flexibility at the expense of increased hardware resources per RF chain. Figure 2.13 presents a detailed block diagram of a fully digital MIMO transmitter. This architecture can simultaneously support L users using K ($L < K$) RF chains. The DSP unit executes the required signal processing algorithms to ensure correct signal reception at the user terminals, whereas the RF front-ends upconvert and amplify the signals to the desired carrier frequency while providing high output power and efficiency. At baseband, the data

streams for individual users are first modulated and subsequently precoded collectively to compensate for the channel characteristics. Thereafter, a dedicated DPD module within each RF chain linearizes its associated PA.

In a conventional SISO transmitter, an isolator is typically placed between the PA and the antenna to ensure a constant 50 Ω load impedance at the PA output. In contrast, fully integrated MIMO transmitters require a high degree of compactness, and therefore bulky isolators are generally omitted between the PAs and the antenna array. The removal of isolators allows antenna mutual coupling to influence the output load impedances seen by each PA, since the reflected waves now include leaked signal components from adjacent RF chains. As a consequence, the effective load impedance varies with the transmitted signals, giving rise to the load modulation phenomenon. These load variations induce fluctuations in PA characteristics, rendering conventional SISO DPD insufficient for accurate linearization. In the presence of antenna crosstalk, the reflected waves must therefore be explicitly considered to properly model and compensate for load modulation effects. Thus, a seemingly minor architectural modification introduces significant new challenges in transmitter linearization.

For the fully digital MIMO architecture, precoding is implemented to mitigate multiuser interference and/or to maximize the received Signal-to-Noise Ratio (SNR). Prior to transmission, instantaneous Channel State Information (CSI) is estimated in the form of a channel matrix $\hat{H}$, where $\hat{H} \in \mathbb{C}^{L \times K \times N}$ and N denotes the number of subcarriers [30]. Based on $\hat{H}$, linear precoding matrices can be computed. Three widely adopted linear precoders are Maximum Ratio Transmission (MRT) [31], Zero-Forcing (ZF), and Minimum Mean Square Error (MMSE) [32]. The MRT precoder maximizes the received SNR and is particularly suitable in noise-limited conditions, but it does not explicitly suppress inter-user interference. Conversely, the ZF precoder eliminates user interference by inverting the channel matrix, at the cost of reduced SNR due to the associated power penalty. The MMSE precoder provides a balanced tradeoff between MRT and ZF, achieving acceptable interference suppression while maintaining satisfactory SNR performance. The corresponding mathematical formulations are derived in [2]. At a carrier frequency of 3.5 GHz, the 90% channel correlation time ranges from 1−10 ms for user velocities between 3−30 km/h [33], implying that precoding matrices must be updated frequently to track channel variations.

The hybrid beamforming architecture aims to combine the advantages of RF and fully digital beamforming in order to reduce system complexity while retaining a degree of signal generation flexibility. In line-of-sight scenarios, hybrid beamforming can generate separate beams from each sub-RF beamformer. In non-line-of-sight environments, digital precoding can be incorporated to mitigate multipath and multiuser interference effects.

Consequently, hybrid beamforming represents a compromise between hardware complexity and beamforming capability.

2.2.2 Antenna Mutual Coupling and Load Modulation

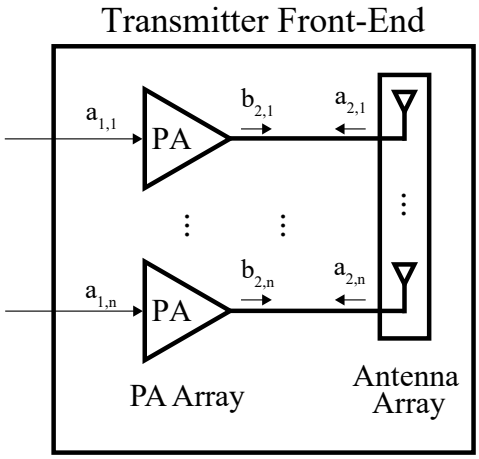


Figure 2.14: **MIMO Transmitter Front-End.**

Figure 2.14 illustrates the MIMO transmitter front-end and the corresponding power-wave flow. When isolators are inserted between the PA outputs and the antenna array, the reflected wave vector $\mathbf{a_2}[n] = [a_{2,1}[n], \ldots, a_{2,n}[n]]^{\mathrm{T}}$ becomes ideally zero due to perfect isolation. However, when the isolators are removed to achieve a high level of integration, $\mathbf{a_2}[n]$ is no longer negligible and is related to the outgoing waves through the antenna scattering matrix as

$$\begin{bmatrix} a_{2,1}(f) \\ \vdots \\ a_{2,n}(f) \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ S_{n1} & \cdots & \cdots & S_{nn} \end{bmatrix} \begin{bmatrix} b_{2,1}(f) \\ \vdots \\ b_{2,n}(f) \end{bmatrix}, \tag{2.42}$$

where S_{ij} denotes the antenna S-parameters and $b_{2,i}$ represents the outgoing wave from the ith PA.

The reflection coefficient Γ_i seen at the output of the ith PA can therefore be expressed as

$$\Gamma_i = \frac{\sum_{j=1}^{n} S_{ij} a_{2,j}(f)}{b_{2,i}(f)}. \tag{2.43}$$

Equation 2.43 indicates that the effective output load impedance, equivalently Γ_i, depends jointly on the antenna mutual coupling levels and the instantaneous signal conditions across all transmit branches. Since conventional PA designs assume a nominal 50 Ω load impedance, any increase in the magnitude of the numerator in (2.43) results in greater impedance deviation and consequently more pronounced performance degradation. In extreme cases, Γ_i may exceed unity, corresponding to an apparent negative resistance condition at the PA output.

It has been shown in [34] that the nonlinear distortion metric Adjacent-Channel Power Ratio (ACPR) can vary by approximately 10 dB when the antenna mutual coupling level changes from −14 dB to −28.4 dB. In addition to degraded linearity, the output power P_{out} and drain efficiency η are also adversely affected under severe load modulation. Furthermore, results reported in [35, 36] demonstrate that for −20 dB antenna crosstalk, Γ_i can vary within a VSWR circle of 2:1 depending on the transmitted signals. Conventional SISO DPD cannot fully compensate for this behavior because the predistorted signal is generated solely as a function of its own input $a_{1,i}$, whereas Γ_i is influenced by the coupled wave components $a_{2,j}$ from adjacent branches.

The resulting load modulation problem is therefore multifaceted and has been addressed from three primary perspectives: advanced DSP algorithms, robust PA design methodologies, and improved antenna and array design techniques aimed at mitigating mutual coupling.

2.2.3 Linearization Complexity

DPD

For addressing the load modulation problem at the behavioral modeling level, two principal DPD based approaches have been investigated. The first approach extends the conventional SISO formulation by explicitly incorporating the reflected wave component $\mathbf{a_2}$ into the model structure. In this context, a dual-input model, referred to as DISO DPD [37], is employed to capture the interaction between the incident wave $\mathbf{a_1}$ and the reflected wave arising from antenna mutual coupling. Prior to applying the DISO DPD, a Crosstalk and Mismatch (CTMM) module is utilized to estimate $\mathbf{a_2}$ from $\mathbf{a_1}$ and the measured antenna S-parameters. The DPD model then processes both the input signal $\mathbf{a_1}$ and the predicted reflected wave $\hat{\mathbf{a}}_\mathbf{2}$, resulting in the following dual-input formulation, which assumes that $\mathbf{a_2}$ remains sufficiently small such that only linear terms are required for its representation:

$$z[n] = \sum_{p=0}^{P_1-1} \sum_{m=0}^{M_1} \alpha_{pm} x[n-m]|x[n-m]|^p \tag{2.44}$$

$$+ \sum_{p=0}^{P_2-1} \sum_{m_1=0}^{M_2} \sum_{m_2=0}^{M_3} \beta_{pm_1m_2} \hat{a}_2[n-m*1]|x[n-m_2]|^p \tag{2.45}$$

$$+ \sum_{p=0}^{P_3-1} \sum_{m_1=0}^{M_4} \sum_{m_2=0}^{M_5} \beta_{pm_1m_2} \hat{a}_2^*[n-m*1]x^2[n-m_2]|x[n-m_2]|^p. \tag{2.46}$$

This model has been shown to provide improved linearization performance compared to conventional SISO models in fully digital MIMO transmitters. Nevertheless, its practical implementation requires the same number of Transmitter Observation Receiver (TOR) units as the number of RF chains in order to accurately predict $\mathbf{a_2}$ for each branch. As a consequence, the hardware complexity increases significantly with array size, which limits its scalability to large antenna arrays.

The second approach is based on load-dependent DPD [38]. Instead of identifying DPD coefficients at a single nominal impedance, multiple impedance points are selected in the vicinity of the expected antenna load region for coefficient extraction. During operation, when the effective load impedances vary due to precoding and changes in the input signal statistics, the set of DPD coefficients corresponding to the trained impedance that is closest to the instantaneous load condition is selected for PA linearization. This strategy enables the continued use of SISO DPD structures, thereby reducing implementation complexity while partially compensating for load modulation effects.

Precoder

The incorporation of precoding introduces an additional layer of complexity in the analysis of load modulation. In the absence of precoders, uncorrelated signals are typically employed for training DPD coefficients and characterizing PA behavior. These signals are designed to have equal average power levels across all RF chains, resulting in similar load modulation conditions for each branch. However, when conventional precoders are applied, disparities in the average power levels among RF chains can reach 6−12 dB [39]. Under such conditions, the severity of load modulation becomes highly dependent on the relative power distribution across neighboring chains. For instance, an RF chain operating at high

output power and surrounded by low-power chains may experience negligible load modulation. Conversely, a low-power chain adjacent to high-power chains can be subjected to substantial load modulation, potentially forcing the effective output impedance outside the Smith chart. Consequently, precoders can induce significant, channel-dependent performance variations among RF chains.

To mitigate these variations, equal-power precoders were proposed in [39]. These precoding schemes enforce identical average power levels across all RF chains while preserving the fundamental characteristics of the three linear precoders discussed earlier. Although the total transmitted power summed over all RF chains remains unchanged compared to conventional precoding, the peak power levels are reduced by approximately $2-3$ dB when equal-power precoders are employed. Experimental results obtained from a 2×2 fully digital MIMO transmitter demonstrate that channel-dependent performance variations among RF chains are substantially reduced using equal-power precoders, albeit at the expense of increased computational complexity.

Antenna

As mentioned earlier, a VSWR circle of 2:1 corresponds to approximately -20 dB antenna mutual coupling [35, 36]. If the mutual coupling level can be further reduced, the severity of active load modulation may be significantly alleviated or even eliminated. Several antenna-level decoupling techniques have been proposed in the literature to address this challenge. For instance, the neutralization line approach introduces additional transmission lines between adjacent antenna elements to compensate for the mutual impedance components Z_{ij} [40, 41]. Although effective for specific configurations, this technique becomes increasingly difficult to scale for large arrays and to maintain broadband performance.

Alternatively, a directional-coupler-based decoupling network can be inserted between the PAs and the antenna elements [42]. This approach is supported by rigorous theoretical analysis and can, in principle, achieve substantial isolation improvement. However, its practical implementation becomes complex and hardware intensive for array dimensions beyond 2×2. Other approaches include the use of stacked decoupling surfaces positioned above the antenna array [43, 44] and specially engineered antenna geometries designed to inherently suppress mutual coupling [45]. While these methods can further reduce mutual coupling, their performance is often sensitive to fabrication tolerances and mechanical constraints. Consequently, the development of a scalable and practically realizable antenna structure that provides sufficiently low mutual coupling for large-scale arrays remains an open research problem.

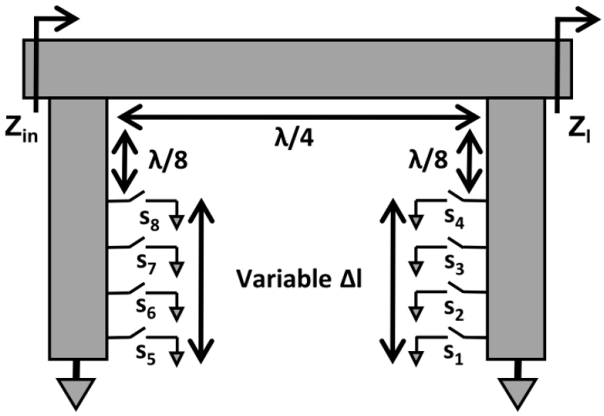


Figure 2.15: **Switch-Based Reconfigurable Matching Network.** Reprinted from [3]

2.3 Review on Power Amplifiers for MIMO Architectures

The main goal for the PA design in the MIMO architecture is to maintain similar performance (linearity, output power, efficiency) under load modulation. In this case, SISO DPD may be able to model the PA, reducing the complexity and power consumption of the DSP unit.

One design approach for load-insensitive PA is to develop reconfigurable matching networks. Authors in [3] designed a π-shaped matching networks, as shown in fig. 2.15, with four switches connected to each parallel stub to map load impedances within a $2:1$ VSWR circle to the values that would give the same output power. On the other hand, in Integrated Circuit (IC) design, fig. 2.16 shows that varactors can be implemented to directly control capacitances of each parallel stubs [4]. In addition, in IC design, it is possible to integrate impedance detection module and the tunable OMN together for establishing an automatic output impedance correction system [46, 47]. PA output power can be controlled very well but the reconfigurable matching networks can add additional loss that reduces the overall efficiency. Furthermore, studies on frequency-dependent antenna loads should be conducted to verify if it can handle a spectrum of load impedances.

Another method is supply modulation for correcting PA output power. The idea is to apply supply drain voltage modulation to provide the same loadline as the 50 Ω load at the intrinsic transistor plane [5, 48]. Fig. 2.17 shows that the impedance measurement system is connected to the drain bias control unit for dynamically correcting the load presented to

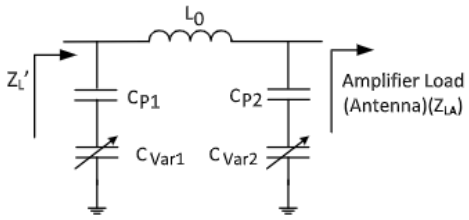


Figure 2.16: **Varactor-Based Reconfigurable Matching Network.** Reprinted from [4]

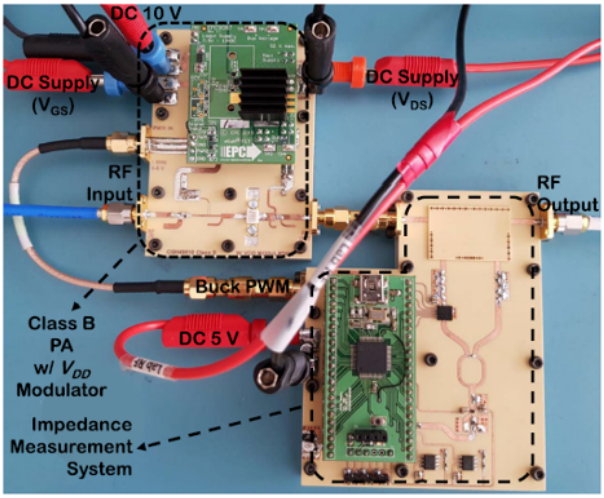


Figure 2.17: **Supply Modulation for Power Amplifier Output Power Correction.** Reprinted from [5]

the PA. This approach can transmit the same output power within 2 : 1 VSWR circle but the impedance measurement processing time may be the bottleneck and the efficiency is not well compensated for. By the time the techniques are applied to adapt to the new output load impedance, it is likely to be changed already to a different value. Furthermore, this method can only choose one load value at a single frequency for correction. Thus, under modulated signal excitation, it may struggle for frequency-dependent antenna loads and when precoders are applied in MIMO architectures.

The above-mentioned methods focused on correcting PA output power but when modulated signals are used for communications, OBO efficiency is also important. Thus, PA mode switching method was proposed to switch between two operating modes depending on the output load impedance to achieve both linearity and OBO efficiency under load modulation. Authors in [6] proposed balanced-to-Doherty mode switching PA. Balanced

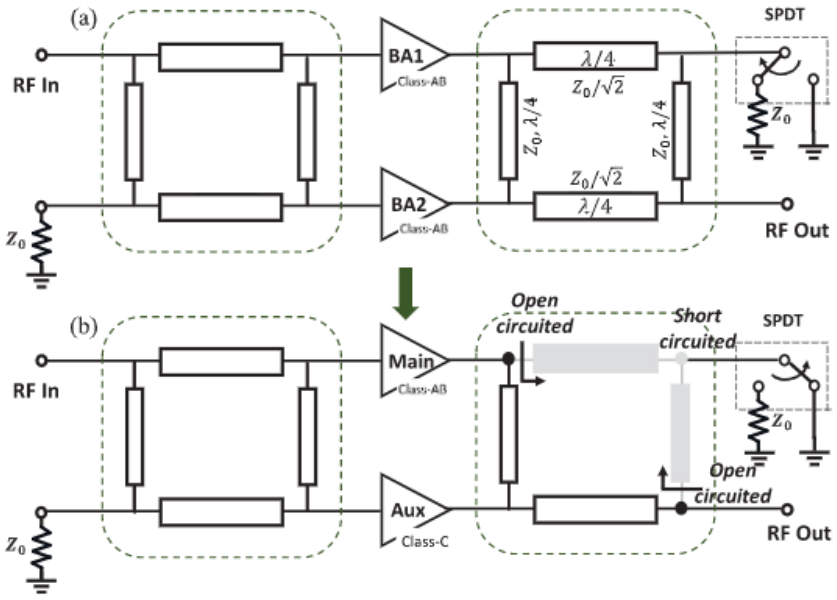


Figure 2.18: **A Block Diagram of Balanced-to-Doherty Power Amplifier.** (a) Balanced mode, (b) Doherty mode. Reprinted from [6]

PA is linear and performs similarly across a frequency spectrum, making it inherently resilient against load modulation, but it suffers from a low back-off efficiency. Fig. 2.18 shows a block diagram of the balanced-to-Doherty mode switching PA. By changing the load connected to the isolated port of the output hybrid-90 coupler, the balanced PA architecture can be changed to DPA. This design was further improved to use a transistor at the isolated port for the mode-switching control [49]. For full OBO efficiency enhancement, series/parallel DPA mode switching PA was also proposed [50]. If the output load is connected between the main and auxiliary PAs, the DPA is in series mode. If the output load is connected at the output of both PAs, it is in parallel mode. By connecting a short- or an open-circuit load at the isolated port of the output hybrid-90 coupler in 2.18(b), the mode switches between series or parallel DPA, respectively. The authors also showed that connecting small reactance and susceptance loading instead of short- and open-circuit loads can improve linearity of series/parallel mode switching DPA. One of the advantages of mode-switching PA is that when the performance of one mode begins to degrade, that of the other mode does not, so the best of both modes can be obtained. However, the mode switching is determined by a single impedance value at a single frequency component. Similar to the supply modulation technique, under modulated signals and/or precoders,

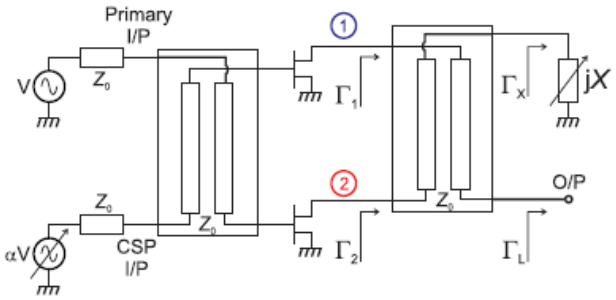


Figure 2.19: **A Schematic of Orthogonal Load-Modulated Balanced Power Amplifier.** Reprinted from [7].

frequency-dependent antenna loads can span across both mode regions. This technique should be tested and simulated further for load impedances that would be presented in real deployment.

Load modulation can be handled from input signal sources in digital DPA [51]. Instead of designing matching networks and choosing biasing points for achieving desired current and impedance profiles, digitally driven DPA can achieve DPA behavior by sending separately controlled input signals to each PA. Depending on the presented output load impedance, the amplitude and phases of the input signals can be modified for maintaining optimal DPA performance. This method can give the most flexibility but it would put more burden on the DSP unit. Since the goal of this thesis is to reduce the complexity of DSP unit by designing a PA that can tolerate load modulation, this would be unfavored.

Multiple methods can be combined for a PA topology. Orthogonal LMBA was proposed to use isolated port loading from mode-switching PA and input signal control similar to digital DPA [52]. Fig. 2.19 shows a schematic of orthogonal LMBA. Unlike the conventional LMBA, shown in fig. 2.8, the control source is connected at the isolated port of the input hybrid-90 coupler, instead of the output coupler. This has the advantage of having low-power control source and the load connected to the isolated port of the output coupler can be utilized for reconfigurability. An extensive analysis of load modulation on orthogonal LMBA [7] showed that it can tolerate wider range of load modulation compared to balanced PA. In addition, authors in [53] developed a balanced DPA by preserving the inherent load resiliency of the balanced structure and boosting OBO efficiency from DPA. Authors in [54] developed a Complementary Metal-Oxide Semiconductor (CMOS) IC PA design that utilizes series/parallel DPA and load switching at the isolated port of the output coupler, in the same way done by [50]. In addition, the IC chip controls the drain bias and

the number of active transistor cells for making the PA load resilient. Although this PA can work well for 3 : 1 VSWR circle, the authors have only applied modulation signal of less than 1% fractional bandwidth.

From the PA designers' perspective, the best load-insensitive PA should be robust against load modulation by its inherent structure and provide a high OBO efficiency. Balanced PA is a candidate for being robust against load modulation but it has low OBO efficiency. Appendix A shows the analysis on how the balanced PA is robust. Reconfigurable matching networks showed great performance improvements in linearity but their application could be more suitable for adapting to the channel conditions (precoders) instead of switching for every instantaneous loads. Supply modulation and mode-switching PA can be effective but their performance are uncertain for wider bandwidth with frequency-dependent loads. The development of a new PA design that can operate similar to balanced DPA with three transistors would be the final goal of this study. The design in IC may be necessary incorporating multiple features in a small footprint.

Chapter 3

EM/RF/DSP Co-simulation Framework and Development Testbed

3.1 Introduction

Research works in DSP, PA, and antenna are mostly conducted individually and separately. In spite of the solutions in each part giving great results, system-level performance may yield sub-optimal results due to the additional interactions between different parts of the system. A full system, which requires much more resources than a single channel RF front-end, needs to be constructed to understand these interactions.

As the first step, a multi-channel DSP/RF/EM co-simulation framework that incorporates compact transistor modeling for PA is proposed for predicting a holistic system-level behavior prior to any circuit and antenna fabrication. The development of such framework can help designers account for all interactions between each building block for achieving optimal fully digital MIMO transmitter front-end performance. DSP is already handled by software and an antenna array can be modeled in S-parameters. The remaining challenge is to predict PA performance in MIMO configurations interconnecting DSP, PA, and an antenna array during the design process.

In parallel, 2×2 and subsequently 4×4 development transmitter front-end testbeds were designed and developed to study the fully digital MIMO transmitter front-end. The 2×2 development testbed incorporated two different PA topologies (Doherty PA and Class-AB) and was utilized to experimentally validate the proposed co-simulation framework.

4×4 development testbed was designed and fabricated for the analysis of the massive MIMO fully digital front-end system-level behavior in practice. 16 channels are a minimum number of channels needed, due to a limited amount of resources, for understanding the performance of a *massive* MIMO transmitter that consists of at least 64 channels. The testbed includes a multistage PA to allow direct connections to the signal sources at the input and to the antenna array at the output.

3.2 EM/RF/DSP Co-Simulation Framework

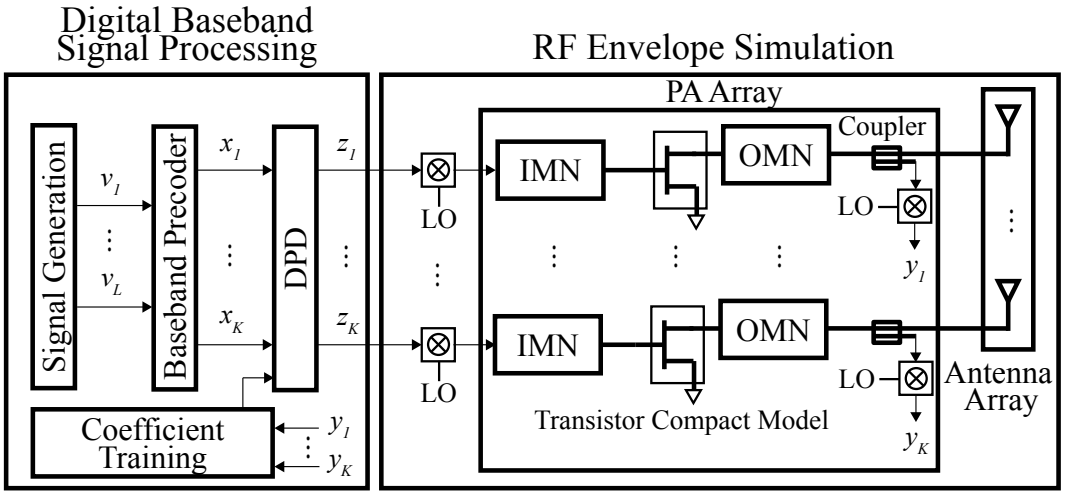


Figure 3.1: **Proposed Co-simulation Framework.** Figure reprinted from [8].

Fig. 3.1 shows the proposed co-simulation framework. Baseband DSP is performed in MATLAB, PA design in Keysight Advanced Design System (ADS), and antenna design in Ansys High-Frequency Structure Simulator (HFSS). Thus, a co-simulation framework integrates cross-platform designs for system-level simulations. Baseband DSP generates modulated signals, applies precoding for fully digital beamforming, and trains DPD coefficients for PA linearization. The generated baseband modulated signals are transmitted to ADS for upconversion then to the PAs for envelope simulation. PA transistors are compact models, provided by manufacturers, and matching networks are EM-simulated S-parameters. PAs are also terminated with antenna array S-parameters, extracted from HFSS, to account for antenna mismatches and crosstalk. The output modulated signals from the PAs are coupled back to baseband DSP for DPD coefficient training.

Algorithm 1 Baseband DSP in Experiments

Configure signal, transmitter (Arbitrary Wave Generator (AWG)), receiver (Oscilloscope), DPD parameters settings
Generate Modulated Signal using the settings
Generate DPD Basis
for every DPD iteration **do**
    Send input modulated signal to AWG
    Receive output modulated signal at oscilloscope
    Update DPD coefficients
    Apply predistortion to the modulated signal using the new DPD coefficients
end for
Apply a precoding matrix, if needed

Algorithm 2 Baseband DSP in Co-simulation

Configure signal, transmitter (AWG), receiver (Oscilloscope), DPD parameters settings
Generate Modulated Signal using the settings
Generate DPD Basis
for every DPD iteration **do**
    Send input modulated signal to ADS
    Run ADS simulation [system('hpeesofsim netlist.txt')]
    Receive output modulated signal from ADS
    Update DPD coefficients
    Apply predistortion to the modulated signal using the new DPD coefficients
end for
Apply a precoding matrix, if needed

3.2.1 Baseband DSP

Algorithm 1 summarizes how it is applied for characterizing PAs in experiments. First, modulated signal, AWG, oscillscope, and DPD settings are configured. Using these configurations, modulated signals are then generated. Subsequently, DPD basis functions are formulated for modeling a PA under test. For each iteration, the baseband modulated signal is uploaded to the AWG and the output signal is captured at the oscilloscope. By analyzing the input and output signals, the coefficients of DPD basis functions are updated. With the updated set of coefficients, the input modulated signal is pre-distorted for linearizing the PA under test.

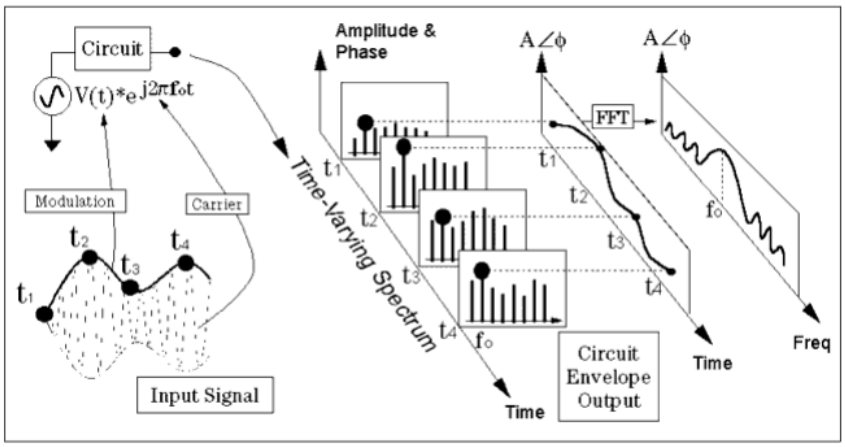


Figure 3.2: **Envelope Simulation.** Reprinted from [9].

For the co-simulation framework, as algorithm 2 shows the modified version of the baseband DSP algorithm, only the transmission and retrieval of the modulated signal need to be modified while the rest can stay unchanged. For running ADS simulation from MATLAB, ADS circuit netlist file needs to be generated and the following system environment variables need to be set up for calling ADS simulator *hpeesofsim* via MATLAB operating system commands [55]:

1. Variable Name: 'HPEESOF_DIR', Variable value: 'ADS install directory'
2. Variable Name: 'COMPL_DIR', Variable value: '%HPEESOF_DIR%'
3. Variable Name: 'SIMARCH', Variable value: 'win32_64'
4. Add %HPEESOF_DIR%\bin to the variable 'PATH'
5. Add %HPEESOF_DIR%\adsptolemy\lib.win32_64 to the variable 'PATH'
6. Add %HPEESOF_DIR%\lib\win32_64 to the variable 'PATH'

MATLAB now solely controls the flow of the co-simulation processing, given that the ADS circuit netlist is properly generated. The next subsection discusses how to set up the ADS schematics.

ADS Ptolemy signal processing simulator [56] provides features to handle time-domain modulated signals from MATLAB DSP algorithm. The processed time-domain signal can

be applied to a Device Under Test (DUT) for frequency-domain analysis using envelope simulation. The envelope simulator [9] extracts the envelope of the time-domain signal and performs harmonic balance (frequency-domain) simulation at the carrier frequency for the amplitude and phase at each sampled time. Memory effects can also be captured because each harmonic balance simulation result is dependent on past values of the envelope. Harmonic balance simulation produces results at DC, fundamental frequency, and harmonic frequencies. Thus, for the output time-domain signal, Ptolemy has an envelope output selector model (EnvOutSelector) to extract only the fundamental frequency results over the entire input signal length for generating the output time-domain signal. This signal is fed back to MATLAB for DPD training. Fig. 3.2 summarizes how Ptolemy can be combined with RF envelope simulation to perform time-domain analysis.

3.3 Development Transmitter Front-End Testbed

3.3.1 2×2 Transmitter Front-End

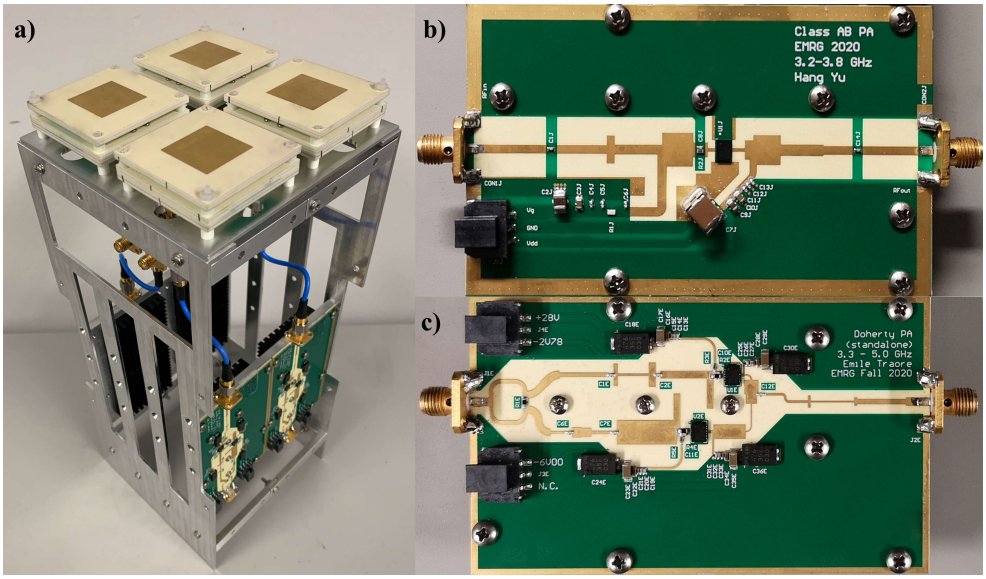


Figure 3.3: **2×2 Fully Digital MIMO Transmitter Front-End** (a) Fully assembled modular fully digital MIMO Transmitter Front-End, (b) Class-AB PA, (c) Doherty PA. Reprinted from [10].

The 2×2 development transmitter front-end was designed and developed by Hang

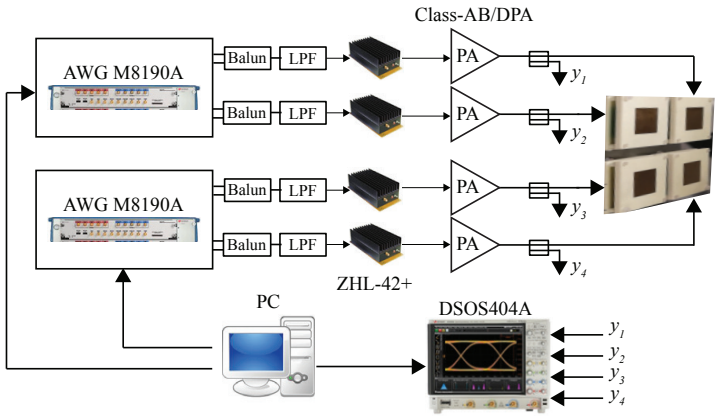


Figure 3.4: **2×2 Array Measurement Setup.** Reprinted from [8]

(Justin) Yu and Emile Traore from our research group. A 2×2 array is the smallest MIMO system for studying the effects of antenna crosstalk on PA performance. Fig. 3.3 shows the complete assembly of the 2×2 modular fully digital MIMO transmitter front-end, and custom-built Doherty PA and Class-AB PA that were designed and fabricated. This transmitter front-end was used for the experimental verification of the proposed co-simulation framework and as a reference design for the next 4×4 transmitter front-end.

Application of the Proposed Co-simulation Framework to the Study of the Antenna Crosstalk effects on 2×2 Transmitter Front-End Performance and Experimental Validation

For the experimental validation of the proposed co-simulation framework in Sec. 3.2, a four-channel fully digital MIMO transmitter was assembled, as shown in fig. C.2. For providing sufficient input power and converting the differential signal to the single-ended one, each RF chain consists of a balun (Marki Microwave BAL-0010), a low-pass filter (LPF; Mini-Circuits VLF-5500+), a driver amplifier (Mini-Circuits ZHL-42+), a custom-built PA (4-watt class-AB PA or 7.6-watt DPA shown in fig. 3.3), and a directional coupler (Anaren X4C40K1-20S). For the antenna array, modular dual-polarized stacked patch antennas with 0.7λ spacing are configured in a 2×2 grid. Modulated signals are generated by two dual-channel AWGs (Keysight M8190A) and captured by a four-channel oscilloscope (Keysight DSOS404A).

The PAs and the antennas were designed to cover N78 and/or N79 operating bands (3.3−3.8 GHz and 4.4−5.0 GHz). Both Class-AB and DPA were designed using 6 W

Table 3.1: **Comparison of System-Level Simulation and Measurement With and Without SISO & DISO DPD.**

		Simulation					Measurement					# of DPD Coeff
		RNMSE (%)	ACPR L (dB)	ACPR R (dB)	η_{avg} (%)	P_{avg} (dBm)	RNMSE (%)	ACPR L (dB)	ACPR R (dB)	η_{avg} (%)	P_{avg} (dBm)	
Class AB	w/o DPD	7.0	−34.3	−32.5	25.0	33.3	9.5	−34.4	−32.6		33.4	
	SISO	1.24	−47.2	−46.6	23.7	33.0	2.9	−46.5	−47.0	21.3	33.4	31
	DISO	0.69	−52.8	−52.5	23.7	33.0	1.3	−49.2	−49.9	21.0	33.4	43
DPA	w/o DPD	7.77	−35.1	−32.2	34.0	35.7	9.5	−31.6	−31.7		34.7	
	SISO	1.43	−46.2	−45.2	32.9	35.6	2.2	−45.8	−46.1	30.5	34.7	68
	DISO	0.93	−49.5	−50.0	32.9	35.6	1.2	−49.6	−50.0	31.2	34.7	84

transistors (Wolfspeed CGHV1006S) that operate with the drain voltage of 28 V and the main transistor's quiescent current of 20 mA. Class-AB PA has the small-signal gain of 13.0 ± 0.5 dB for 3.2−3.8 GHz, while DPA has 8.75 ± 0.95 dB for 3.3−5.0 GHz with the peaking transistor biased at −6.4 V in class-C mode. The stacked patch antenna achieved an excellent cross-polarization isolation of 40 dB, and the 2×2 antenna configuration with 0.7λ spacing gives the worst case cross-coupling level of −24 dB.

For the system-level measurements, four uncorrelated OFDM signals at $f_0 = 3.4$ GHz having the characteristics of 8 dB PAPR, a modulation bandwidth of 100 MHz, and an oversampling factor of 5 were generated by the AWGs. The co-simulation framework emulated the input signal generation process by scaling it to the AWG's output level and by inserting S-parameters of passive components between the AWGs and the PAs in cascade. Without any external isolators between the PAs and the antenna array, dual-input DPD architecture in [37] was employed to mitigate the combined effects of PA nonlinearity and antenna crosstalk.

Table 3.1 compares simulation and measurement results. Root Normalized Mean Square Error (RNMSE) measures the discrepancy between the input and output signals within the band, while ACPR measures the ratio between the in-band and out-of-band powers of the same bandwidth. For estimating over-the-air system-level behavior at boresight, the results of 2×2 array are vector-weighted summed. Before applying DPD, both Class-AB and DPA show excellent spectra overlap between simulation and measurement, as shown figs. 3.5a and 3.6a. A small discrepancy in the RNMSE may be due to noise and linear distortion produced by the measurement equipment. When SISO and DISO DPDs were applied to both PAs, simulation and measurement results follow a similar trend, as shown

in figs. 3.5b, 3.5c and 3.6b. The proposed co-simulation framework can predict the general trend of the system-level performance very well.

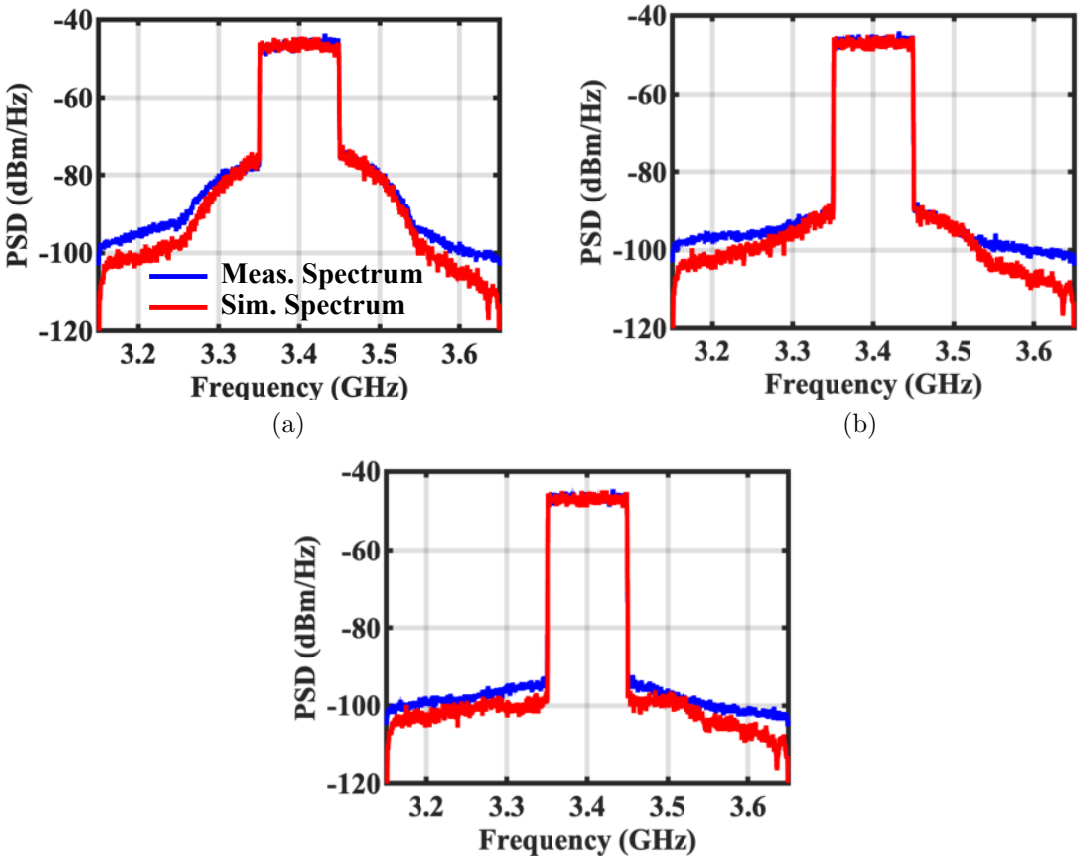


Figure 3.5: **Spectra of Combined Signals for 2×2 Fully Digital MIMO Transmitter using class-AB PAs. (a) Without DPD, (b) With SISO DPD, (c) With DISO DPD.**

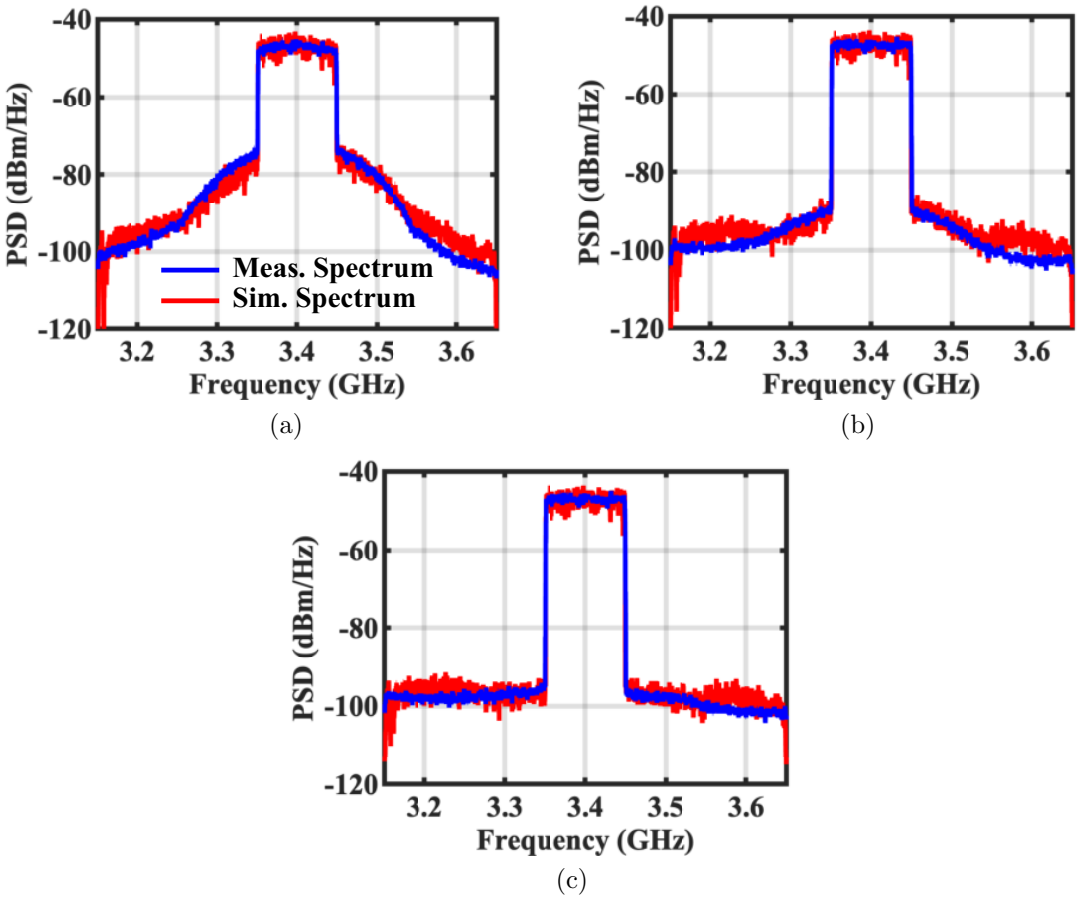


Figure 3.6: **Spectra of Combined Signals for 2×2 Fully Digital MIMO Transmitter using Doherty PAs. (a) Without DPD, (b) With SISO DPD, (c) With DISO DPD.**

3.3.2 4×4 Transmitter Front-End

Multi-stage PA

In addition to expanding a 4-channel fully digital MIMO transmitter system to a 16-channel system, RF front-end system needs to be fully integrated to remove any external components such that signal sources and antennas can be directly connected to the RF inputs and outputs, respectively. As the array size becomes larger, having additional

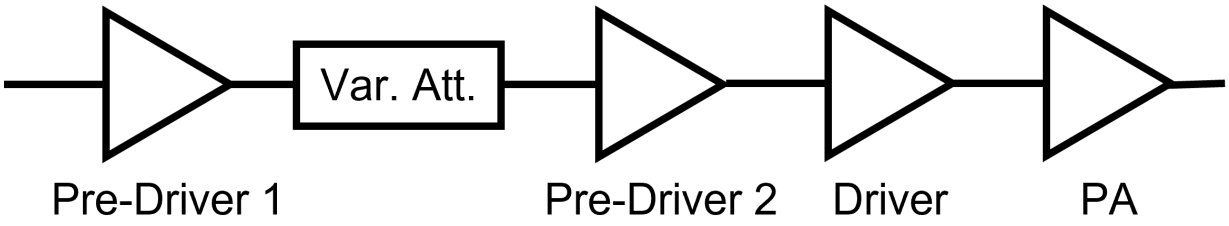


Figure 3.7: **Multi-stage PA Line-up.** QPA9127, HMC1122, and QPA9842 form a variable gain pre-driver stage and provide linear input power to a driver stage (first Wolfspeed 6 W transistor). The last Wolfspeed (now MACOM) 6 W transistor is a PA, designed to output a peak power of 38 dBm.

external components may introduce variations between channels and is not cost-effective. Available signal sources are AWG and Field-Programmable Gate Array (FPGA) (Xilinx RFSoC and MPSoC), so the RF front-end system must have a variable gain feature for -20 to -7 dBm RF input power range.

Figure 3.7 shows an RF line-up with a variable gain feature for this project. The driver and PA stages, using Wolfspeed's (now MACOM's) 6 W transistors, were first designed separately and then re-tuned as a multistage PA. The components for the pre-driver stage were then selected to provide a fixed RF power to the driver input while accepting a required range of input power levels from signal sources.

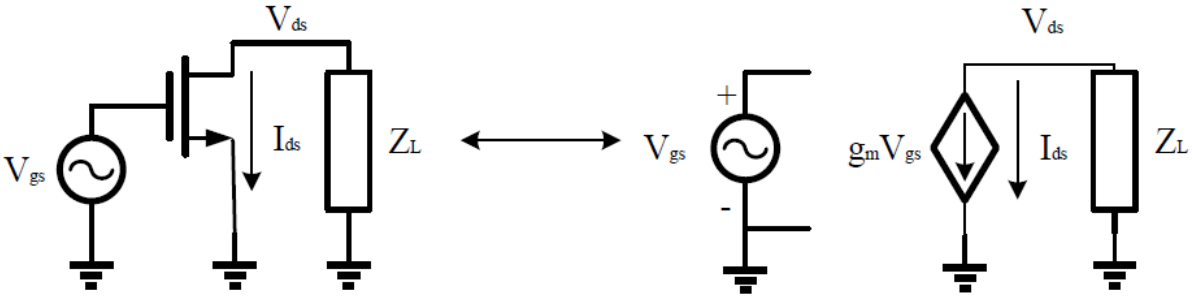
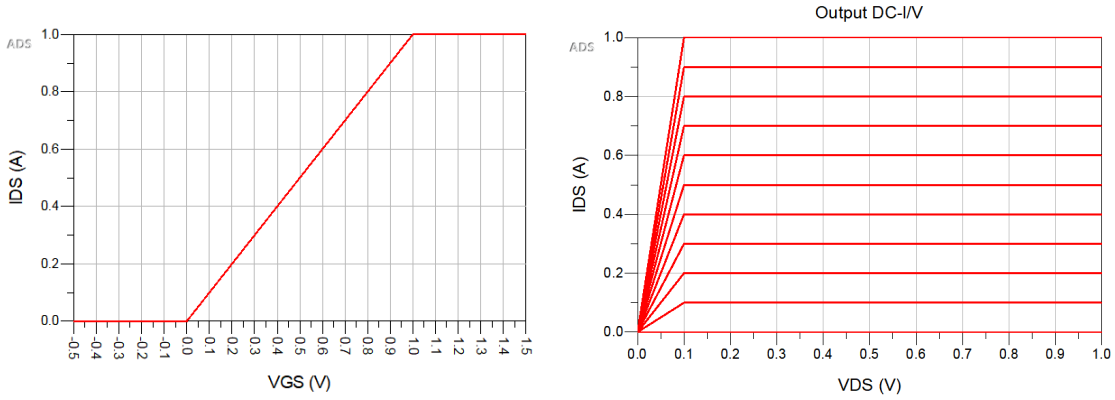


Figure 3.8: **Ideal Transistor Model.** Transistor operation mode is selected by setting gate bias (V_{GS}), drain bias (V_{DS}), and a loadline (Z_L).

Figure 3.8 shows an ideal transistor model. For the RF PA design, the transistor is typically connected in the common-source configuration, in which the gate is connected to an input source, the drain to an output impedance, and the source to a signal ground.



(a) (b)

Figure 3.9: **Ideal Transistor DCIV Curves.** Plots are normalized to I_{max} and V_{mas}. (a) IDS vs. VGS, (b) IDS vs. VDS

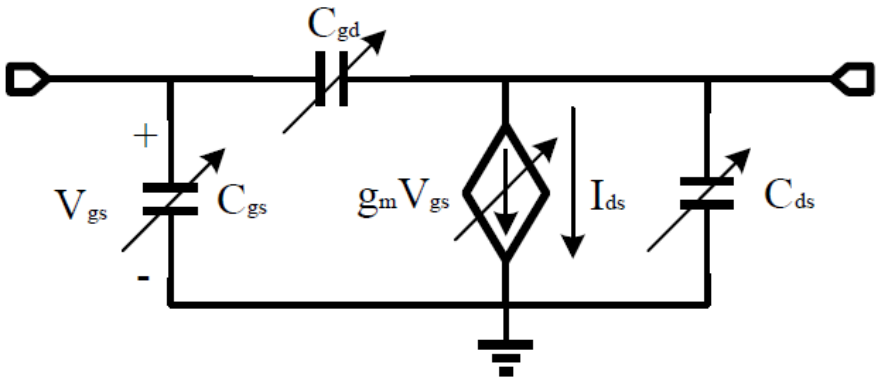


Figure 3.10: **Real Transistor Large-Signal Model.**

For designing a specific PA class of operation, gate bias (V_{GS}), drain bias (V_{DS}), and loadline (Z_L) are typically set such that the input RF signal stays in the linear region in fig. 3.9a ($0 \leq V_{GS} \leq 1$) and the saturated region in fig. 3.9b ($0.1 \leq V_{DS} \leq 1$). For the given V_{GS} and V_{DS}, the optimal output impedance (R_{opt}) for the maximum output power can be calculated by $R_{opt} = (V_{DS,max} - V_{knee})/I_{DS,max}$, where V_{knee} is 0.1 in fig. 3.9b and $I_{DS,max} = 1$.

Commercial transistors, unlike ideal ones, have non-linear transconductance and parasitic capacitances, as shown in fig. 3.10. Because of these non-linearities, the real tran-

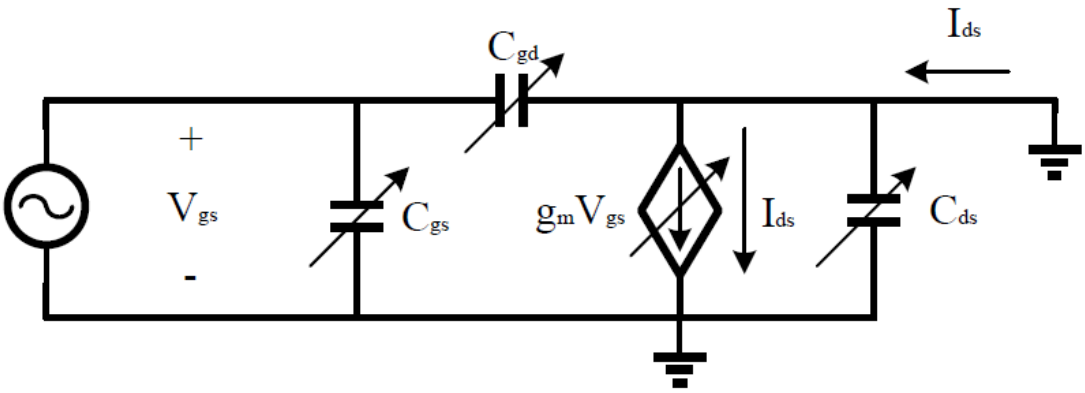
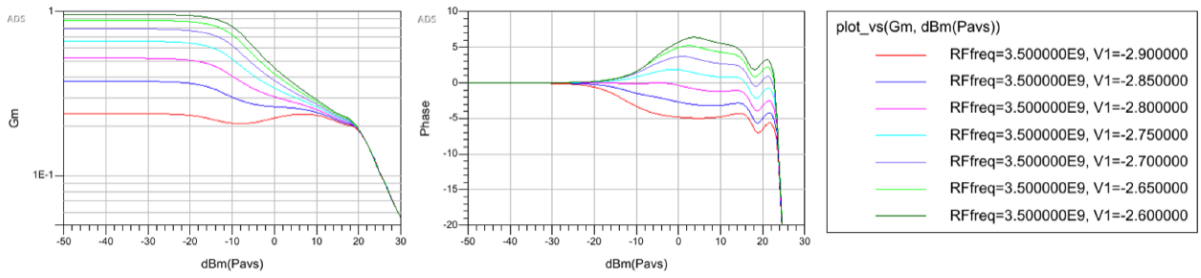
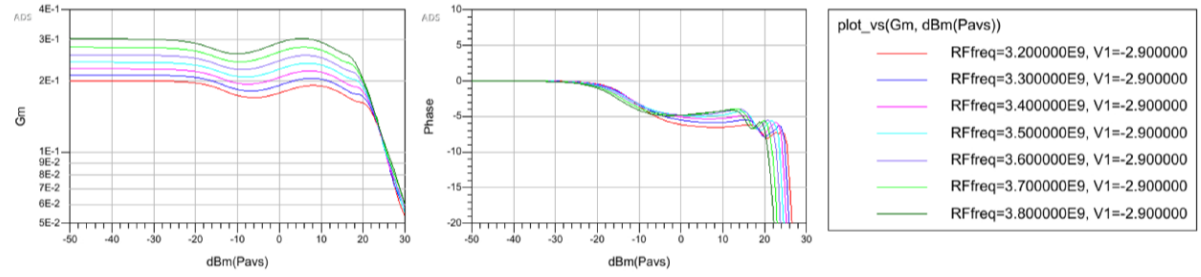


Figure 3.11: **Large-Signal G_m Simulation Setup.** By short-circuiting the drain terminal, the non-linearity of G_m can be characterized. C_{out} can be de-embedded later and used as a part of an output matching network.



(a)



(b)

Figure 3.12: **Commercial Packaged Transistor Large-Signal Model.** (a) Intrinsic plane large-signal transistor model, (b) Packaged transistor model.

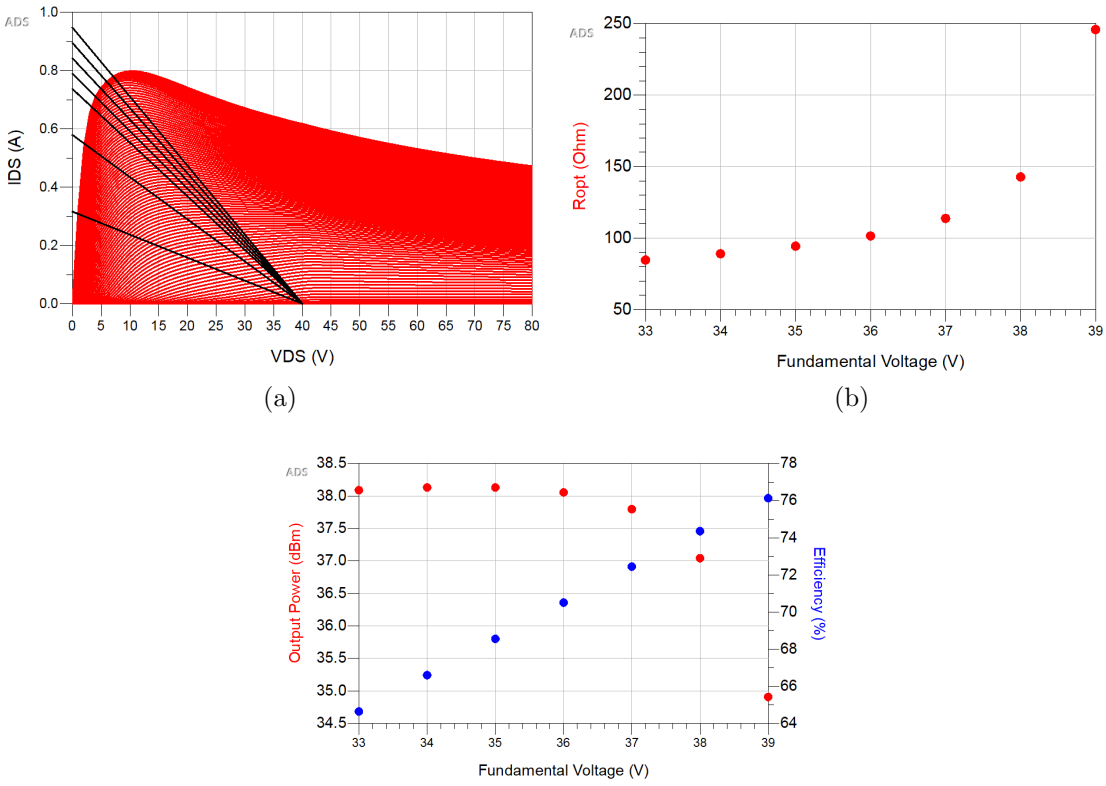


Figure 3.13: **DCIV Loadline.** (a) DCIV loadline of Cree's CGHV1F006S GaN transistor, (b) Load values for each loadline in (a), (c) P_{out} and η for each loadline.

sistor's behavior needs to be scanned over the entire input and output impedance design in simulation to choose proper input and output impedances for the desired behavior with respect to gain, linearity, efficiency, and output power. This procedure is called source-pull and load-pull and prior to these source- and load-pull, a few design steps should be executed.

First, proper biasing points need to be established. The drain bias of Wolfspeed's CGHV1F006S GaN transistor is designed to be between 20 V and 40 V [57]. For maximizing the output power, 40 V is chosen for the PA stage. For the driver stage, a lower drain bias can be used for improving its efficiency, but 40 V was also chosen as simulations showed that the higher the drain bias, the larger its small-signal gain. A high gain is needed to relax the maximum linear output power requirements on the pre-driver stages.

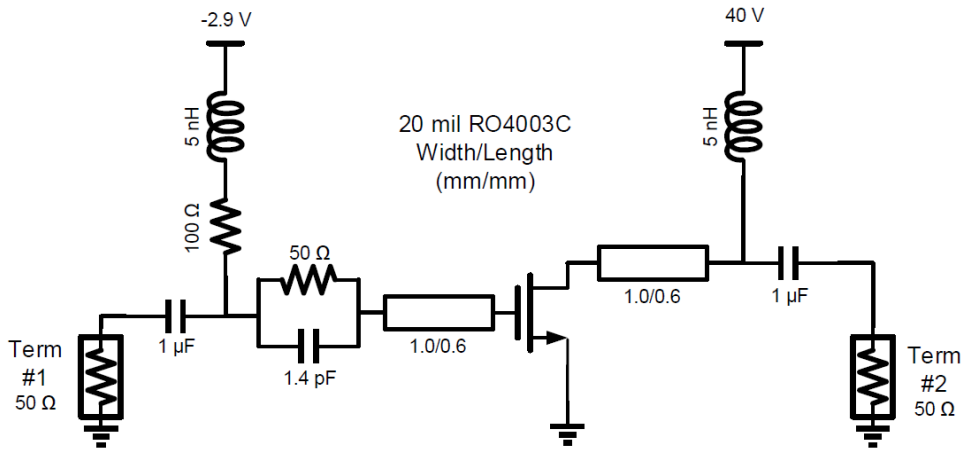
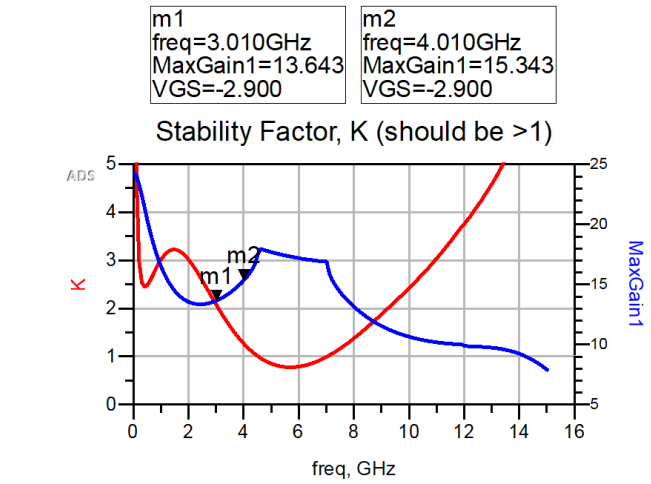


Figure 3.14: **Transistor Stabilization Network.** Transmission lines are for the transistor contacts. 50 Ω and 1.4 pF capacitor are for in-band stabilization and 100 Ω in the gate bias line is for low frequency stabilization.

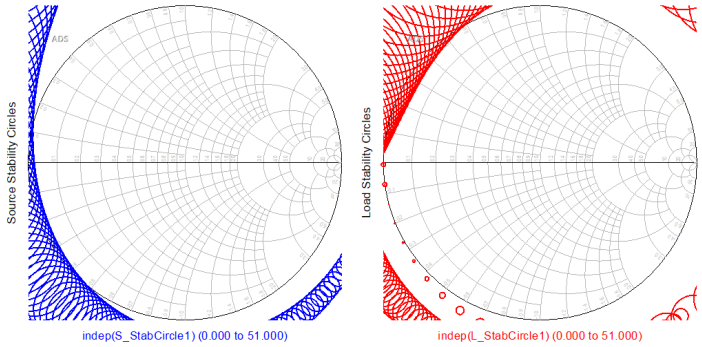
Second, the gate bias needs to be selected. Although a real transistor's transconductance (G_m) is non-linear, by carefully choosing the gate bias, G_m can be linearized as much as possible. Figure 3.11 shows how the large-signal G_m can be simulated and its value can be found by calculating $|I_{ds,fund}/V_{gs,fund}|$. Figure 3.12a shows that $-2.9\ V$ gives the most linear G_m at 3.5 GHz and fig. 3.12b shows that similar linearity characteristic is preserved for the whole bandwidth of interest. $-2.9\ V$ gate bias for deep Class-AB operation is chosen for both driver and PA stages to make the combined stages as linear as possible.

With the selected V_{GS} and V_{DS}, a loadline can be selected based on realizability, output power (P_{out}), and drain efficiency (η). Various loadlines are shown in fig. 3.13a for Wolfspeed's CGHV1F006S transistor. For each loadline, its corresponding load value is plotted in fig. 3.13b. For a wideband matching network, a smaller ratio between Z_L and 50 Ω is beneficial so 250 Ω loadline was ignored. Figure 3.13c shows how P_{out} and η change for each loadline. As the load value becomes larger, V_{knee} and I_{max} decrease. When V_{knee} decreases, the fundamental voltage swing becomes larger. When I_{max} decreases, the fundamental current swing becomes smaller, but η improves as DC current also decreases. There is an optimal load value that gives both high P_{out} and η. Load values between 100 Ω and 110 Ω are such values, as shown in fig. 3.13c.

Stability is a measure of how resilient a transistor is against spontaneous oscillations. As stability depends on the transistor scattering matrix, biasing points can change how the stabilizing network should be designed. After selecting proper biasing points, stability



(a)



(b)

Figure 3.15: **Stability Simulation Results.** (a) Stability Factor and Maximum Available Gain, (b) Source and Load Stability Circles.

analysis can then be performed before designing matching networks. Stability circles can be plotted for both input and output nodes but it is sufficient to design a stabilizing network at one side. For PA design, the network is designed at the input for maximizing output power and efficiency. Figure 3.14 shows the stabilizing network for CGHV1F0006S transistor biased at -2.9 V at the gate and 40 V at the drain. The parallel 50 Ω and 1.4 pF capacitor are for stabilizing in-band (3.2−3.8 GHz), while 100 Ω in the gate bias line is for low frequency stabilization. Figure 3.15 shows stability simulation results for the schematic. The schematic is unconditionally stable, except for 5.0−7.0 GHz. When

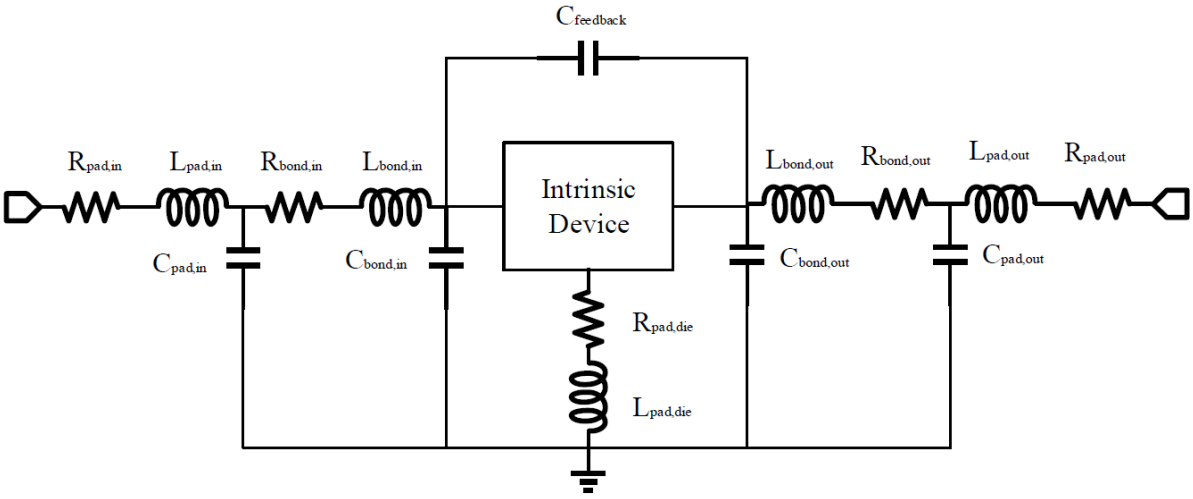


Figure 3.16: **Packaged Transistor.**

the stability circles are checked, both source and load sides are unstable near the edge of the smith chart on the left. This design strategy is used to maximize available gain as much as possible. If impedances in 5.0−7.0 GHz band are designed to avoid the unstable regions, the transistor will still be stable. When matching networks with transmission line models are added to the PA design, signal loss adds to the overall PA behavior, pushing the stability circles outside of Smith chart.

Input and output matching networks for the driver and PA were initially designed separately, and they were combined and re-tuned for the overall multistage PA performance. The driver was designed for high gain and linear performance, while the PA was designed for high efficiency and high output power. During the design process, four frequency bands need careful attention: low frequency, fundamental, second harmonic and third harmonic. The low frequency band (up to 1 GHz) impedances need to be as low and flat as possible; otherwise, during modulated signal test, it can be a significant source of slow memory effect. This can cause the linearization step more complex. The fundamental and harmonic frequency band impedances need to be matched together as harmonic impedances in real transistors affect an amplifier's performance and determine a class of operation.

Source- and load-pull simulations need to be performed to find exact impedances required for the input and output, respectively. Prior to the simulations, transistor package parasitics, shown in fig. 3.16, should be extracted first to find impedances required at the intrinsic device plane such that the intrinsic device's C_{in} and C_{out} as well as the package parasitics can be included in the matching network designs.

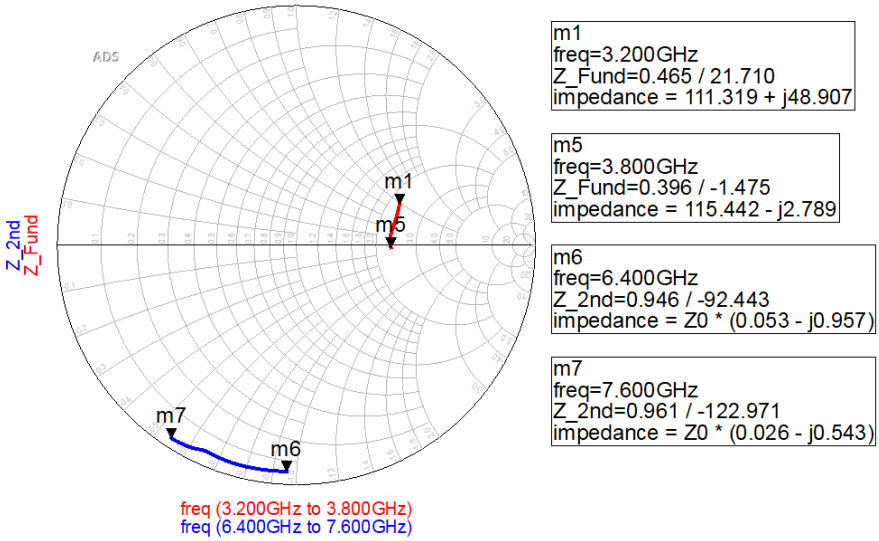


Figure 3.17: **PA Output Matching Impedances at the Intrinsic Device Plane.** For $P_{out} \geq 38.0$ dBm and $\eta \geq 68\%$, the real impedances of the fundamental frequency bands is around 111 Ω. For Class-AB, harmonic frequency band impedances need to be short-circuited. However, the phases of second harmonic impedances affect both P_{out} and η so their phase values can be offset from 0 in real transistors.

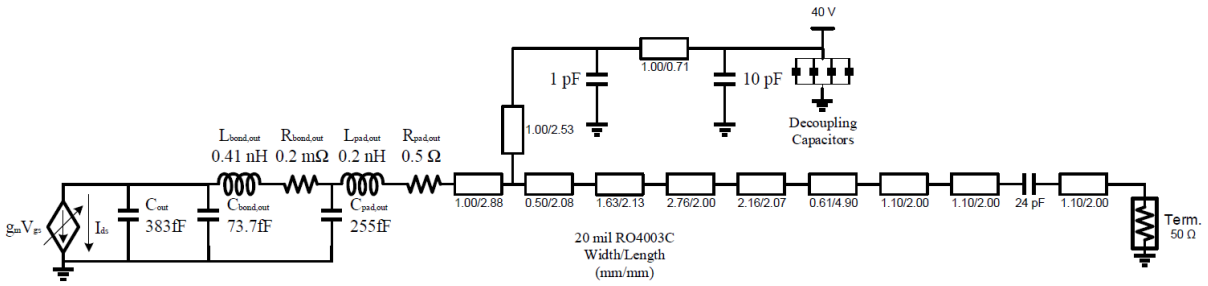


Figure 3.18: **PA Output Matching Network Schematic.**

Through load-pull and optimizations, the optimal impedances for the PA output for $P_{out} \geq 38.0$ dBm and $\eta \geq 68\%$ are shown in fig. 3.17. The real impedances of the fundamental frequency bands are between 111 Ω and 115 Ω, close to DCIV predictions made in fig. 3.13. For Class-AB, harmonic frequency band impedances need to be short-circuited. However, the phases of second harmonic impedances in real transistors affect both P_{out}

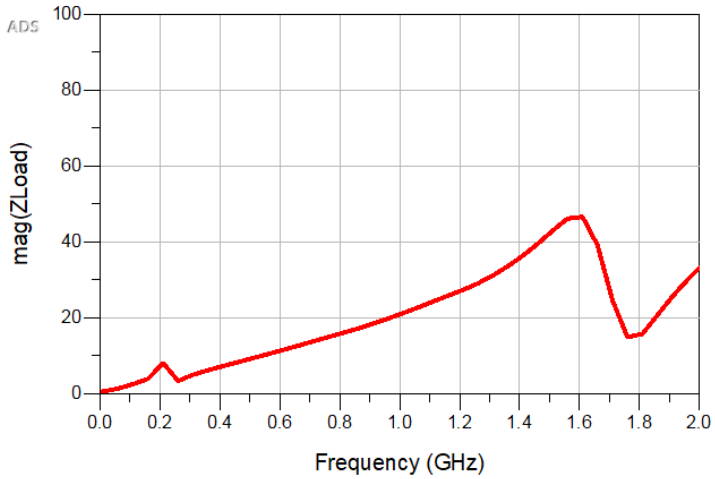


Figure 3.19: **PA Output Matching Network Schematic.**

and η so their phase values can be offset from 0.

Figure 3.18 shows the final schematic of the PA output matching network to match the loads showin in fig. 3.17 to 50 Ω. The matching networks were designed with short transmission line segments to apply Richard's transformation to save layout space for future fabrication. The drain bias line and the choice of RF block capacitor (24 pF) were carefully designed to maintain very low impedances for low frequency band, as shown in fig. 3.19. Up to 1 GHz, it does not change abruptly and is kept below 20 Ω.

The input matching network is responsible for gain and linearity but for the PA design, the main focus is linearity. The gain curve should not compress too early and the phase distortion needs to be minimized. A wide input power range was swept for source-pull and the input impedances that give the minimum compression were chosen for the matching network. Figure 3.20 shows that input impedances need to be around 16 Ω and $-j19.5$ Ω in the real and imaginary parts, respectively.

Figure 3.21 shows the final schematic of PA input matching network design. Package parasitics and C_{in} were added to the overall matching network design. RC parallel tank and 100 Ω in the bias line are for stabilizng the transistor. This matching network is later combined with the output matching network of the driver transistor.

Figure 3.22 shows the overall standalone PA performance. The small signal gain is 14.0 ± 0.5 dB and the 3 dB compression point occurs at $P_{out} = 38.0$ dBm. The drain efficiencies at the peak P_{out} and at 8 dB back-off are 63−75% and 25−31%, respsectively.

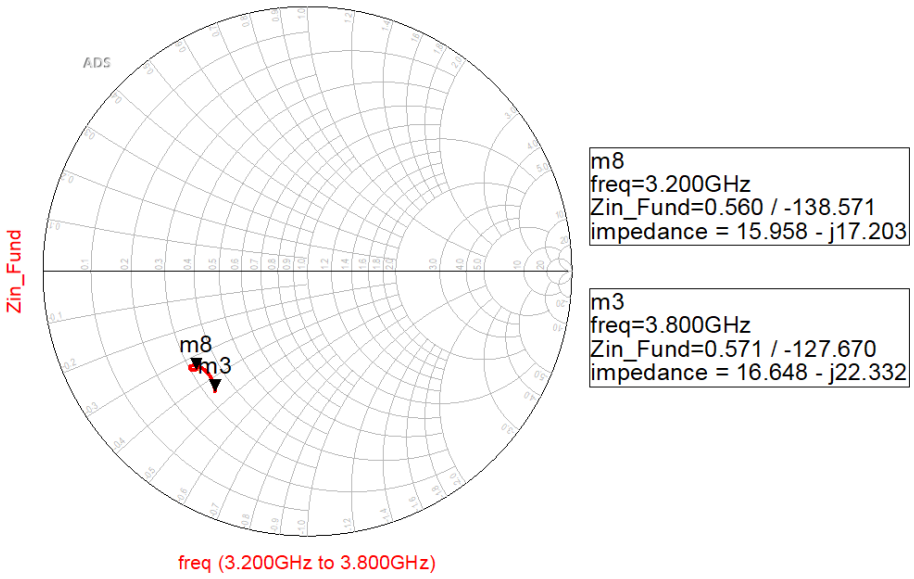


Figure 3.20: **PA Input Matching Impedances at the Intrinsic Device Plane.**

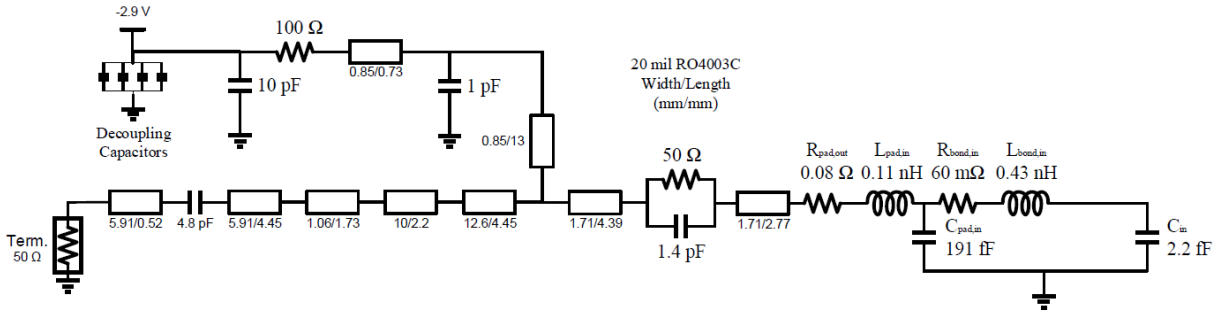


Figure 3.21: **PA Input Matching Network Schematic.**

Having the 3 dB gain compression at the peak P_{out} and the phase distortion of less than 20° show that the PA is very linear and has reached its maximum P_{out}. Both P_{out} and η have reached the values obtained from the load-pull simulation and from the DCIV prediction.

For the driver stage, the P_{out} does not need to be as high as that of the PA stage, so for better overall efficiency, a smaller transistor size should be chosen. However, CGHV1F006S transistor is the smallest transistor available. As a result, the same transistor was also used for the driver stage. The driver can be designed for high gain or for high linearity

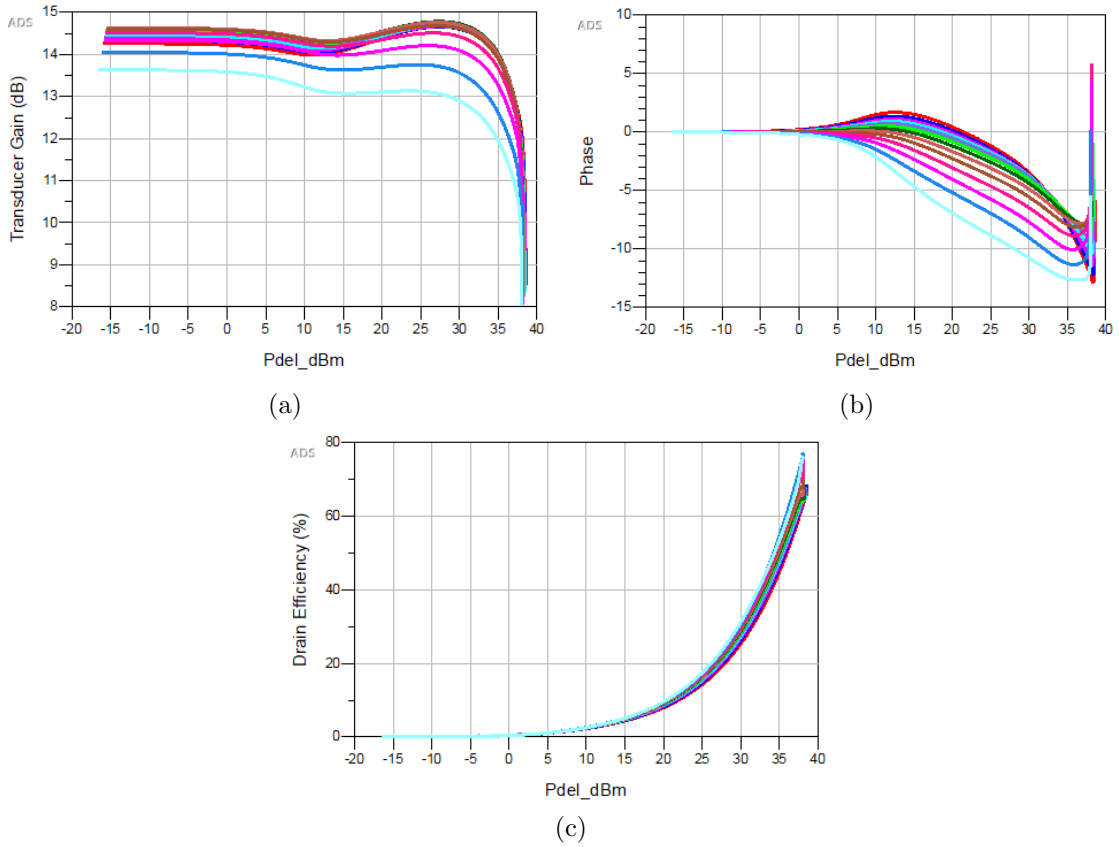
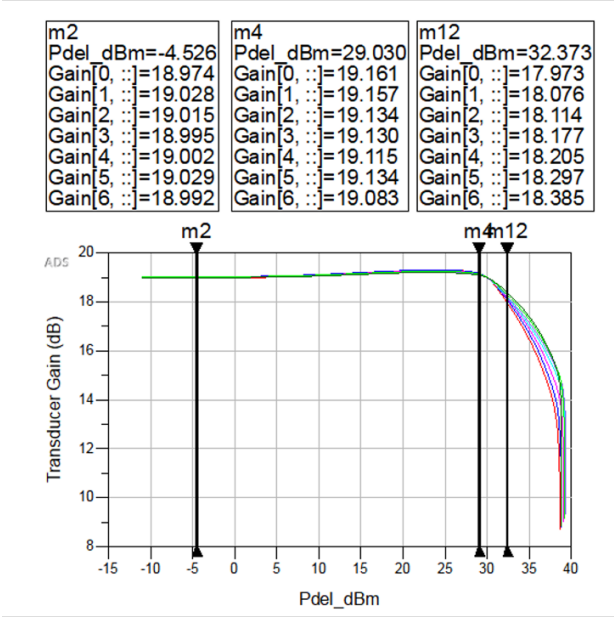


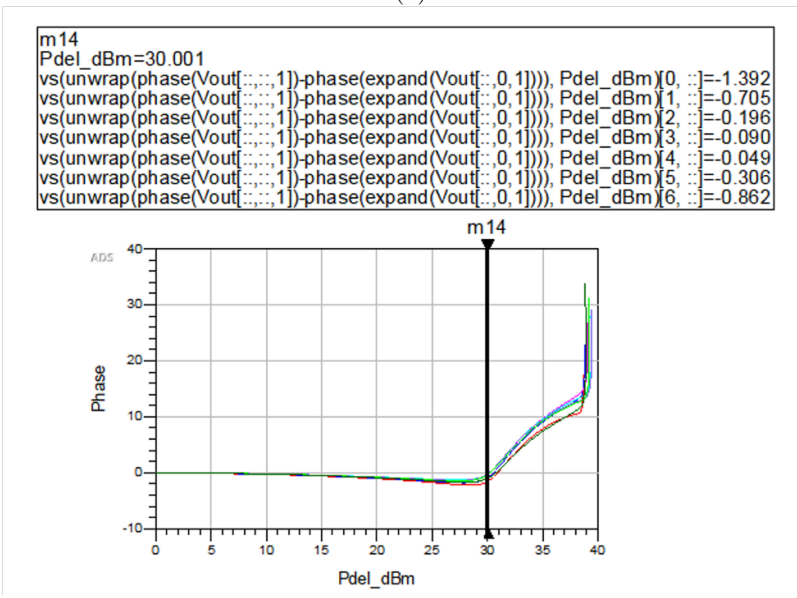
Figure 3.22: **PA Performance**. (a) Gain, (b) Phase distortion, (c) Drain Efficiency.

by manipulating the input matching network. For allowing a range of P_{in}, high gain pre-driver stages can be added, so the driver stage was designed for linearity as well. For the maximum PA P_{out}, P_{in} should be 27 dBm, as shown in fig. 3.22a. For linear driver-PA performance, the driver was initially designed to have a linear P_{out} at 30 dBm. Figure 3.23 shows the initial driver gain and phase characteristics using schematic transmission line modeling. It has linear gain and phase behavior up to 29−30 dBm.

The driver was then placed before the PA stage, connected by a DC block capacitor, and its output matching network as well as the input matching network were re-tuned to keep the same overall P_{out} at 38 dBm and linear behavior in both gain and phase characteristics. Figure 3.24 shows the gain and phase characteristics of driver-PA stages for 3.2−3.8 GHz with 50 MHz step. 3 dB compression point is at the maximum P_{out} and the gain curve is

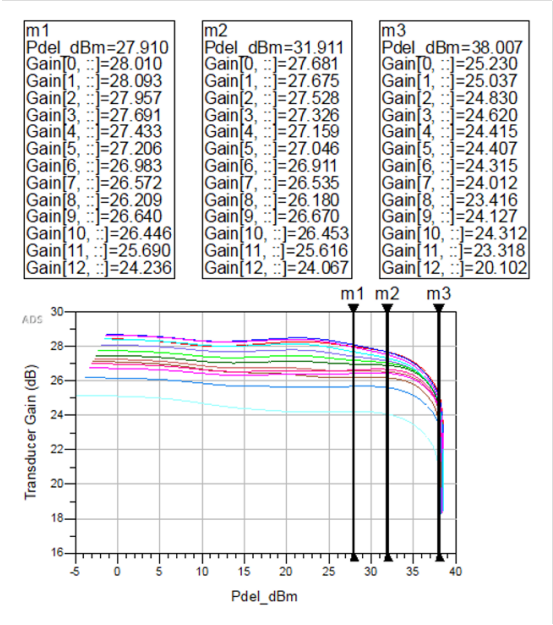


(a)

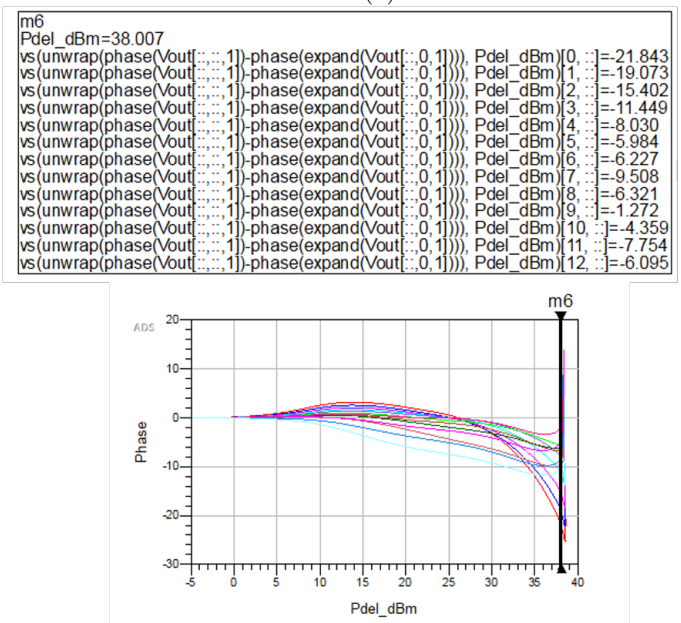


(b)

Figure 3.23: **Initial Ideal Driver Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step.

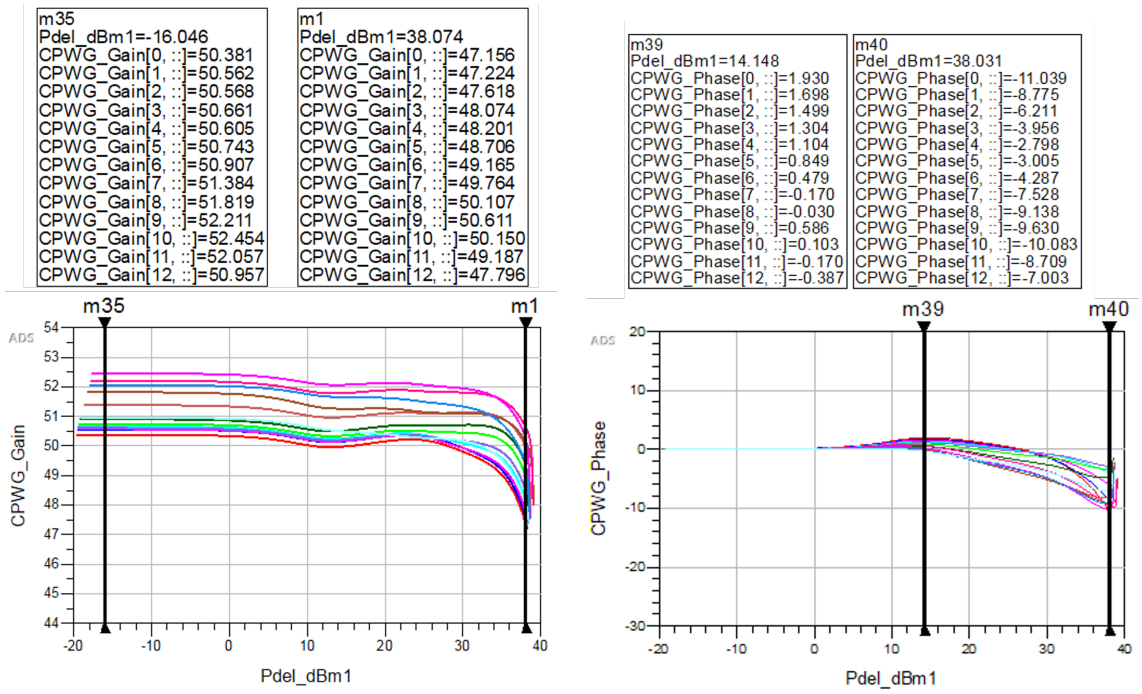


(a)



(b)

Figure 3.24: **Driver-PA Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step.



(a) (b)

Figure 3.25: **Multistage-PA Performance**. (a) Gain, (b) Phase distortion for 3.2−3.8 GHz with 50 MHz step.

very flat, showing linear behavior and the phase distortion do not exceed 20° for most of the frequency components.

This multi-stage PA requires ≈13−15 dBm P_{in} for the maximum P_{out} for 3.2−3.8 GHz. For supporting P_{in} from both AWG- and FPGA-based signal sources of −20 dBm to −7 dBm, pre-driver stages need to provide 20−35 dB variable gain. A typical Variable Gain Amplifier (VGA) contains a variable attenuator between two fixed amplifiers. Pin layouts on commercial products make routing complicated and the second amplifier stage must also be linear for P_{out} greater than 15 dBm for making the full RF chain linear. Thus, two amplifiers and a variable attenuator were selected separately. QPA9842 [58] operates in 2.7−3.8 GHz and has a typical small signal gain of 19.5 dB. Its output power at which 1 dB compression (OP1dB) occurs is 25 dBm, providing a sufficient margin for linearity. QPA9127 [59] operates in 1.0−6.0 GHz and has a typical small signal gain of 20 dB. Its OP1dB is 19.5 dBm, allowing enough margin for the highest Pin of −7 dBm. HMC1122 [60] is a 6-bit digital attenuator that varies from 0 dB to 31.5 dB in 0.5 dB step.

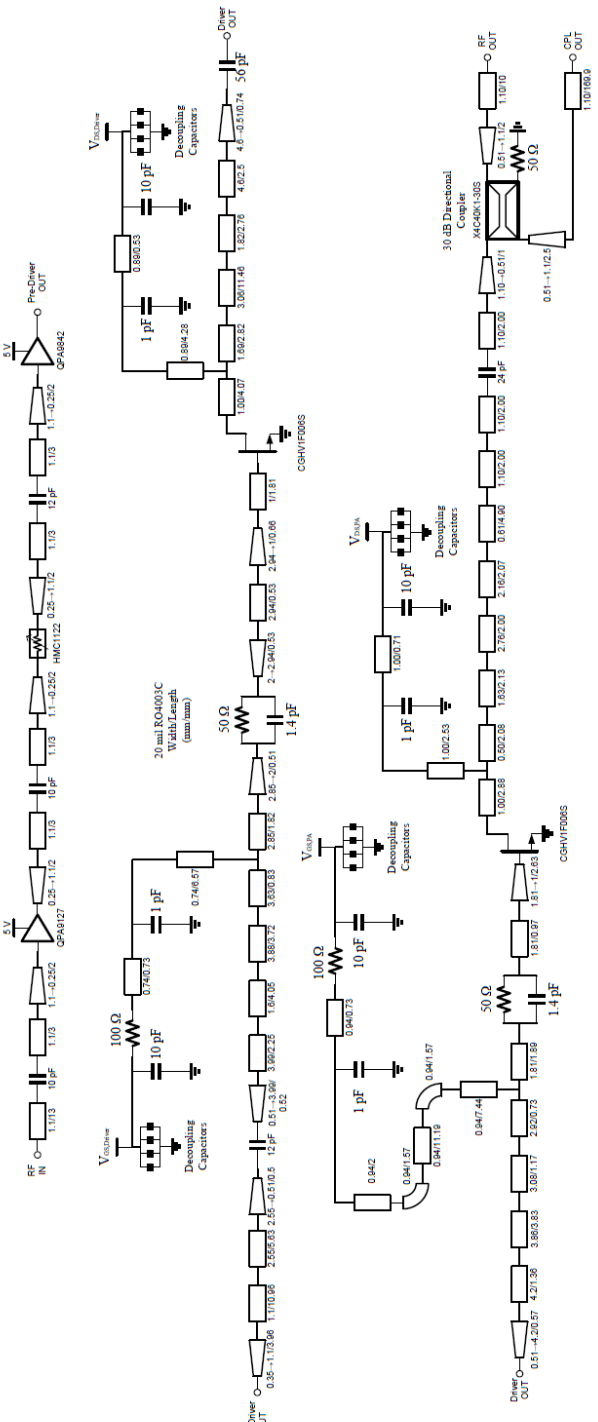


Figure 3.26: Final Multistage-PA Schematic.

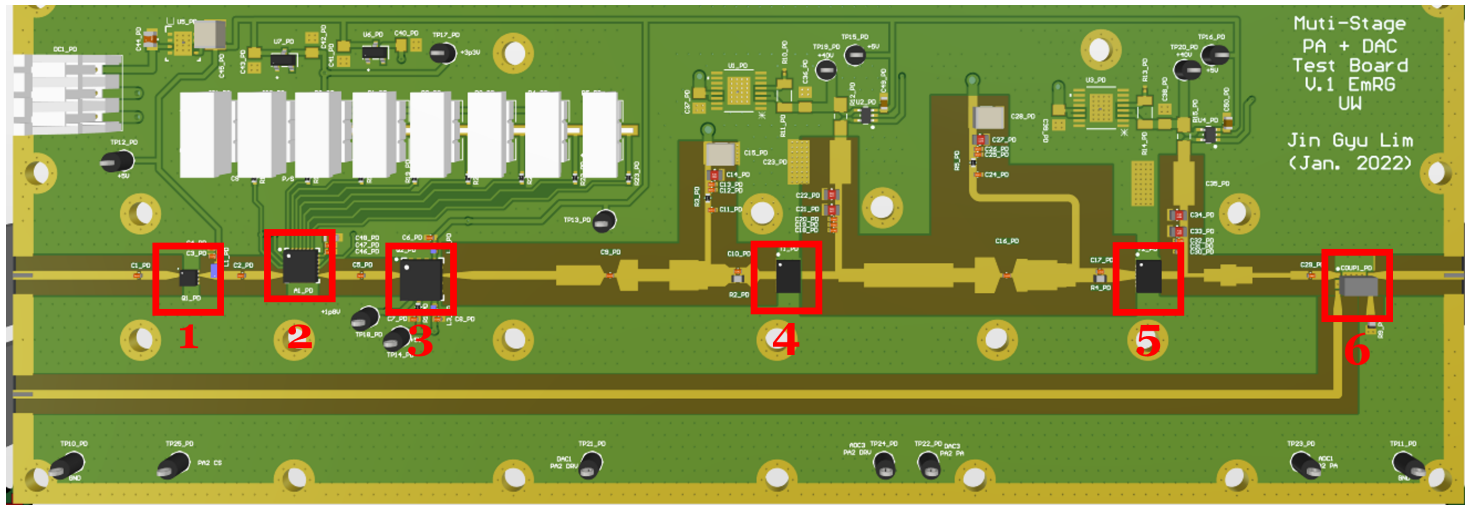


Figure 3.27: **Final Multistage-PA Layout.** 1. Qorvo QPA9127, 2. Analog Devices HMC1122, 3. Qorvo QPA9842, 4. Wolfspeed CGHV1F006S Driver Transistor, 5. Wolfspeed CGHV1F006S PA Transistor, 6. Anaren X4C40K1-30S 30 dB Directional Coupler.

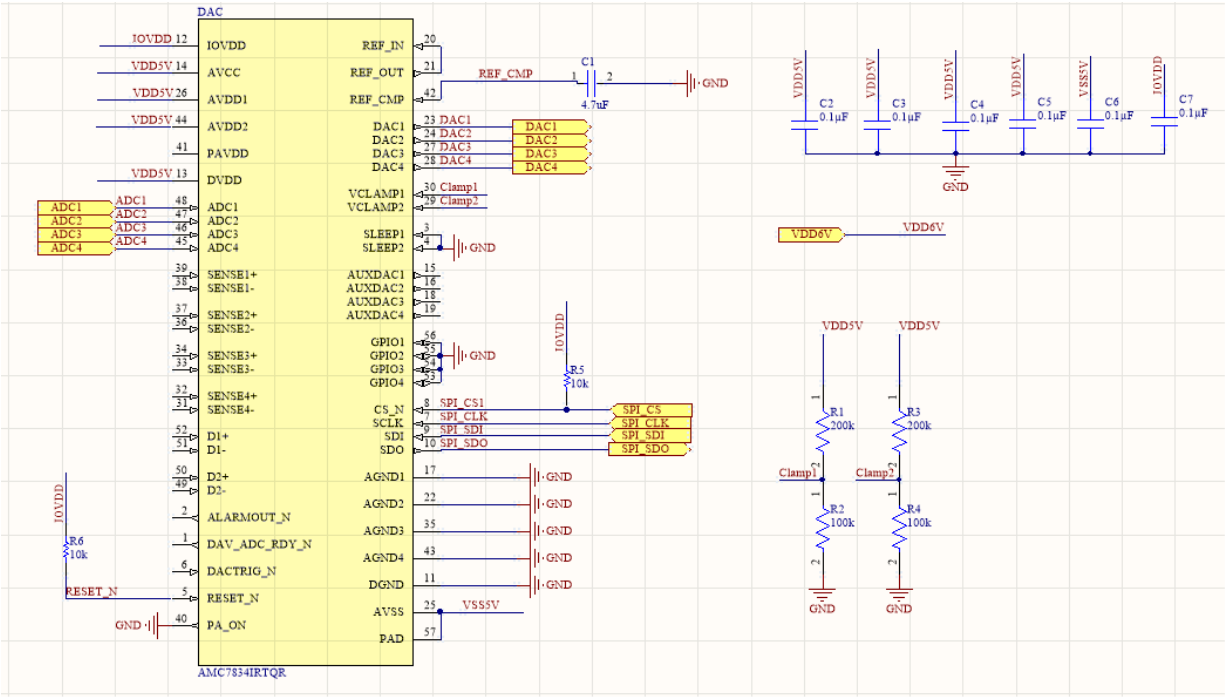


Figure 3.28: **AMC7834 DAC/ADC Module for Gate Biasing and Drain Current Reading.** This chip is serially controlled via SPI protocol. VDD5V is 5 V supply for both digital and analog circuits. IOVDD is 3.3 V for setting digital logic levels. VSS5V is -5 V for negative analog supply and the PAD. These voltages are all provided from a single 6 V supply using a voltage inverter and low-dropout voltage regulators.

Its worst insertion loss for 3.2−3.8 GHz is 2.4 dB. This variable gain pre-driver stages can provide 8−37 dB of gain for the following multi-stage PA.

Using QPA9842's S-parameters, provided by the manufacturer, the input matching network of the driver stage was re-tuned to account for the frequency response of QPA9842. The variable gain of 23−35.5 dB in the final design can allow the peak input signal power to fall between −19.5 dBm and −7 dBm. After the five-stage PA design was completed, a 30 dB directional coupler (X4C40K1-30S) was added after the last PA stage to provide 0 dBm coupled output power for DPD training. The final gain and phase behavior of the multi-stage PA is shown in fig. 3.25. The final schematic and PCB layout of the PA for fabrication are shown in figs. 3.26 and 3.27.

In practice, the gate biases of transistors are controlled by separate DC power supply channels because each transistor requires a different voltage for the same bias current due

to process variation. For the designed multi-stage PA, two gate biases are needed: one for the driver and the other for the PA. This expands to 32 gate bias lines for a 4×4 fully digital MIMO transmitter system. For solving such wiring issue, Texas Instruments AMC7834 [61], an integrated Digital-to-Analog Converter (DAC)/Analog-to-Digital Converter (ADC) module for PA application, is used. It has four bipolar (-5–0 V) DAC channels and four external ADC (0–2.5 V) channels. Thus, one AMC7834 chip can support two channels of the designed multi-stage PA for supplying gate biases and measuring corresponding drain bias currents. Figure 3.28 shows a schematic of how AMC7834 is connected. It accepts a single 6 V DC supply, which is inverted and regulated to -5 V, 3.3 V, and 5 V. DAC pins are connected to transistor gate bias lines and ADC pins are connected to buffered voltage outputs of transistor drain bias lines. 0.1 Ω sensing resistor and a 50 V/V operational amplifier is implemented to scale output voltages to 0–2.5 V ADC ranges and not to disturb the drain bias voltages when the drain current increases. It is digitally controlled via Serial Peripheral Interface (SPI) protocol and algorithm 3 shows its initialization procedure. Clamp voltages (Clamp1 and Clamp2) are set to force output DAC voltages to -5 V upon reset/start-up to protect the transistors.

Algorithm 3 AMC7834 Initialization and Settings

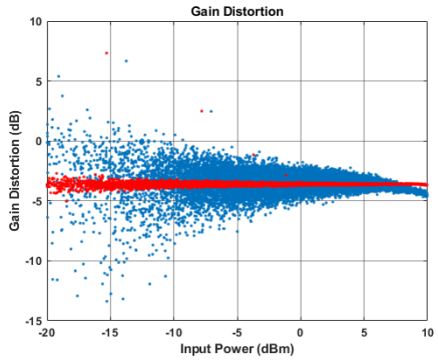
1: Software Reset (Send 0xC to register 0x1D)
2: Set DAC ranges to -5 V to 0 V (Send 0x5 to register 0x16).
3: Set ADC MUX for 4 external, 4 internal (DAC readings), and 1 local temperature pins (Send 0x40FF to register 0x12).
4: Set AMC Config 0 for Direct-Mode ADC Conversion (Send 0x1300 to register 0x10).
5: Activate ADC and DAC (Send 0x2 to register 0x2).
6: Release DAC clamps (Send 0x3C to register 0x17).
7: Set DAC voltages.
8: Check if ADC conversion is ready.
9: Read drain currents, DAC voltages, and chip temperature.

Modulated-Signal-Based Co-Simulation Results

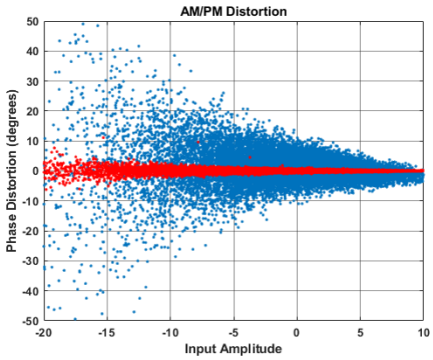
As a final verification step, two- or multi-tone simulation used to be performed to predict how a PA would behave under modulated signal excitation. However, with the development of modulated-signal-based co-simulation framework, explained in Section 3.2, the designed multi-stage PA in fig. 3.27, terminated with 50 Ω, was simulated under 100 MHz, 8 dB PAPR OFDM signal with an oversampling ratio of 5. The PA was subsequently linearized

Table 3.2: **Multi-stage PA Linearization Results.**

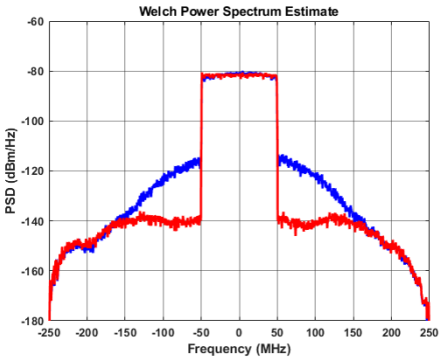
	NMSE (%)	ACPR L (dB)	ACPR R (dB)	P_{avg} (dBm)
Before DPD	7.49	−39.6	−38.3	30.3
After DPD	0.68	−58.0	−58.3	30.2



(a)



(b)

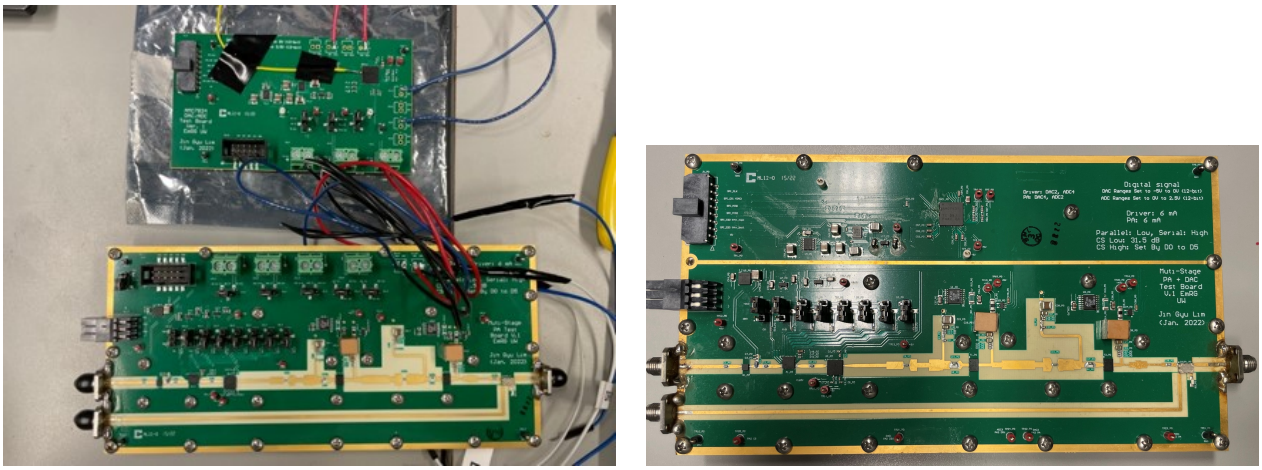


(c)

Figure 3.29: **Multi-stage PA Modulated-Signal-Based Co-simulation Results**. (a) AM/AM, (b) AM/PM, (c) Spectrum. Blue is before applying DPD and red is after applying DPD

by Modified Dynamic-Deviation Reduction (MDDR) DPD with 68 coefficients to verify its linearizability. Table 3.2 summarizes multi-stage PA co-simulation results. NMSE calculates how different output in-band signal is from that of the input while ACPR is the power ratio between the in-band 100 MHz signal and the left and right adjacent 100 MHz signals. The multi-stage PA has very good linearity before applying DPD as it reaches only 1.8 dB gain compression, shown in fig. 3.29a, instead of typical 3 dB compression at the peak P_{out}. It can be linearized very well as the adjacent channel powers are more than 50 dB lower than that of the in-band with the peak P_{out} of 38 dBm. This completes the design procedure and verification of a multi-stage PA.

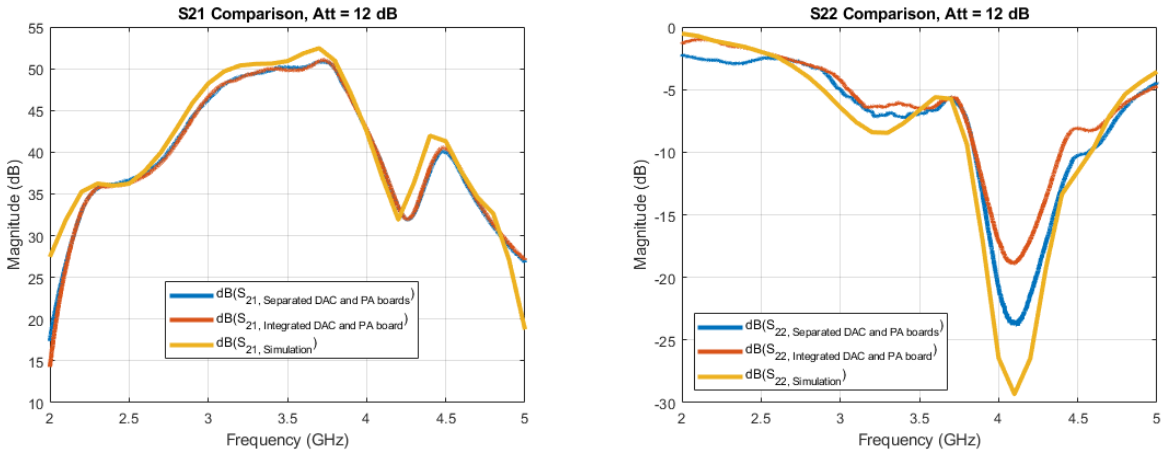
Single-channel Measurement Results



(a) (b)

Figure 3.30: **Multi-stage PA and AMC7834 DAC/ADC system.** (a) Standalone Multi-stage PA and AMC7834 DAC/ADC system, (b) Integrated Multi-stage PA and AMC7834 DAC/ADC system.

For prototyping, two versions of multi-stage PA and AMC7834 DAC/ADC controller were fabricated. One is separate, standalone PA and DAC/ADC controller boards (fig. 3.30a, and the other is the two boards integrated together (fig. 3.30b). Both versions were first tested under small signal excitation. Figure 3.31 shows that simulation predicted the multi-stage PA behavior very well and that two versions behave almost exactly the same. As long as the same drain bias current is applied to both driver and PA transistors, the multi-stage PA performances under small signal excitation are similar.



(a) (b)

Figure 3.31: **Single-channel Multi-stage PA Small-signal Measurement Results.**. (a) S21, (b) S22. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results. HMC1122 attenuator was set to 12 dB.

As both versions perform similarly, only the integrated version was used for the subsequent modulated-signal-based DPD measurements. Since the designed operating frequency range is 3.2−3.8 GHz, 3.4 GHz, 3.5 GHz, and 3.6 GHz were selected as carrier frequencies for 100 MHz and 200 MHz wide OFDM signal with 8 dB PAPR and an oversampling rate of 5. CRV Pruned and MDDR DPD algorithms were applied for all cases.

Table 3.3 summarizes DPD results at 3.4 GHz, 3.5 GHz, and 3.6 GHz carrier frequencies for 100 MHz OFDM signal with 8 dB PAPR and an oversampling rate of 5. ACPR levels before DPD were notably different between simulation and measurement results. Due to the presence of noise in the measurement equipment, there is a limit. In addition, the pre-drivers and attenuator could only be simulated with their S-parameters. In practice, they also change behavior upon input power.

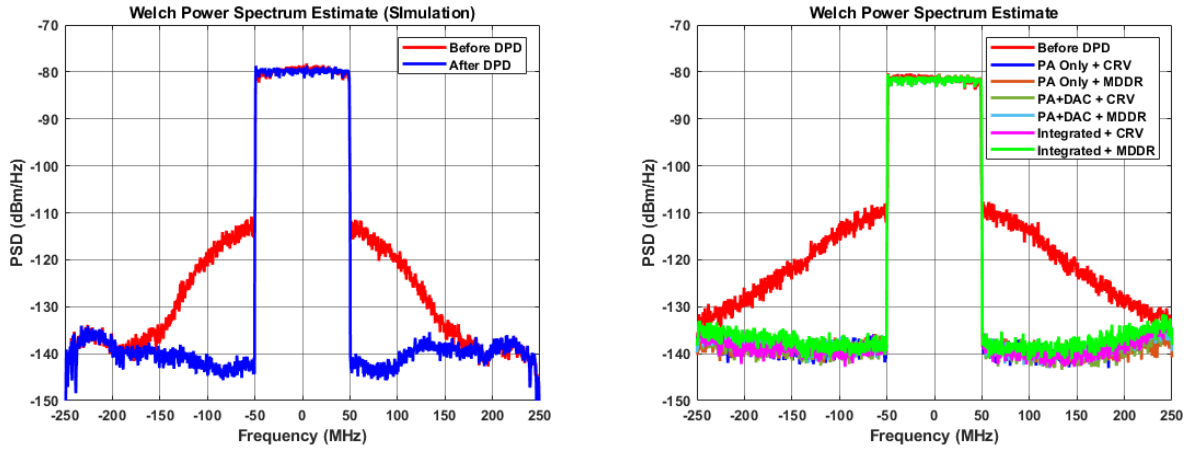
Despite the differences in the starting ACPR values before applying DPD algorithms, simulation predicted well that CRV Pruned DPD linearization algorithm performs better than the MDDR by ≈1 dB in ACPR levels. Measurement results also show that the designed multi-stage PA shows very good linearizability. Figures 3.32 to 3.34 show visually why 3.4 GHz carrier frequency has better ACPR levels after applying DPD. For future measurements, 3.6 GHz may be replaced by 3.3 GHz because 3.6 GHz carrier frequency

Table 3.3: **Prototype Multi-stage PA Linearization Results.** 100 MHz OFDM signal with 8 dB PAPR and an oversampling rate of 5 were used.

			Simulation					Measurement (PA Only + DAC/ADC board)					Measurement (ALL INTEGRATED)				
			NMSE (%)	ACPR L (dB)	ACPR R (dB)	DE (%)	P_{avg} (dBm)	NMSE (%)	ACPR L (dB)	ACPR R (dB)	DE (%)	P_{avg} (dBm)	NMSE (%)	ACPR L (dB)	ACPR R (dB)	DE (%)	P_{avg} (dBm)
3.4 GHz	CRV_Pruned	Before DPD	6.30	-37.9	-38.1	14.3	30.2	6.86	-32.4	-31.5	12.0	29.1	6.84	-32.4	-31.1	11.9	29.1
		After DPD	0.38	-61.9	-60.6	14.1	30.1	0.37	-57.4	-58.1	12.0	29.1	0.38	-57.3	-57.9	11.9	29.1
	MDDR	Before DPD	6.29	-37.9	-38.1	14.3	30.2	6.87	-32.3	-31.4	12.0	29.1	6.82	-32.3	-31.1	11.9	29.1
		After DPD	0.40	-59.5	-59.2	14.1	30.1	0.42	-56.7	-57.4	12.0	29.1	0.47	-56.4	-56.8	11.9	29.1
3.5 GHz	CRV_Pruned	Before DPD	6.61	-39.2	-37.3	14.6	30.4	8.54	-32.9	-30.7	11.8	29.0	7.88	-33.4	-30.9	12.2	29.2
		After DPD	0.64	-59.6	-58.6	14.5	30.3	0.61	-53.9	-54.6	11.8	29.0	0.61	-54.3	-54.8	12.2	29.2
	MDDR	Before DPD	6.61	-39.2	-37.3	14.6	30.4	8.82	-32.4	-30.4	11.8	29.0	7.90	-33.4	-30.8	12.2	29.2
		After DPD	0.65	-58.4	-57.5	14.5	30.3	0.53	-53.5	-55.5	11.8	29.0	1.00	-54.4	-55.5	12.2	29.2
3.6 GHz	CRV_Pruned	Before DPD	7.00	-38.1	-35.0	15.4	30.7	9.82	-32.2	-28.2	11.8	28.9	10.4	-32.2	-27.7	11.1	28.8
		After DPD	0.85	-53.4	-52.5	15.2	30.7	0.64	-53.0	-49.6	11.8	28.9	0.63	-53.2	-49.8	11.1	28.8
	MDDR	Before DPD	7.00	-38.1	-35.0	15.4	30.7	9.88	-32.2	-28.1	11.8	28.9	10.4	-32.2	-27.7	11.1	28.8
		After DPD	0.85	-53.4	-52.5	15.2	30.7	0.59	-53.0	-50.6	11.8	28.9	0.58	-53.1	-50.6	11.1	28.8

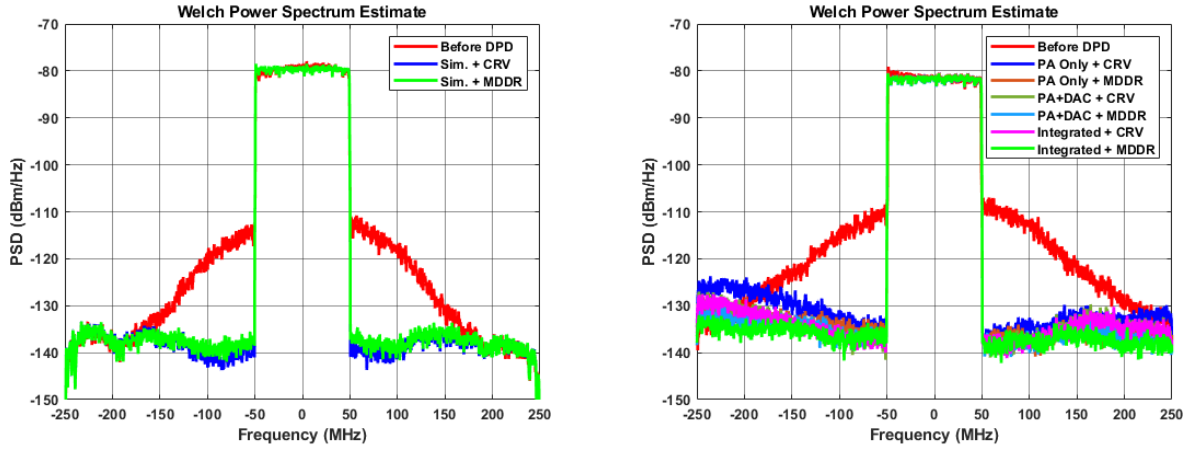
suffered from the upper frequency linearization because of the sudden PA gain drop after 3.8 GHz.

Overall, simulation predicted the general trends well and the measurement results showed great linearizability. This single-channel prototype multi-stage PA design is good enough to be extended to build 16-channel fully digital MIMO transmitter front-end.



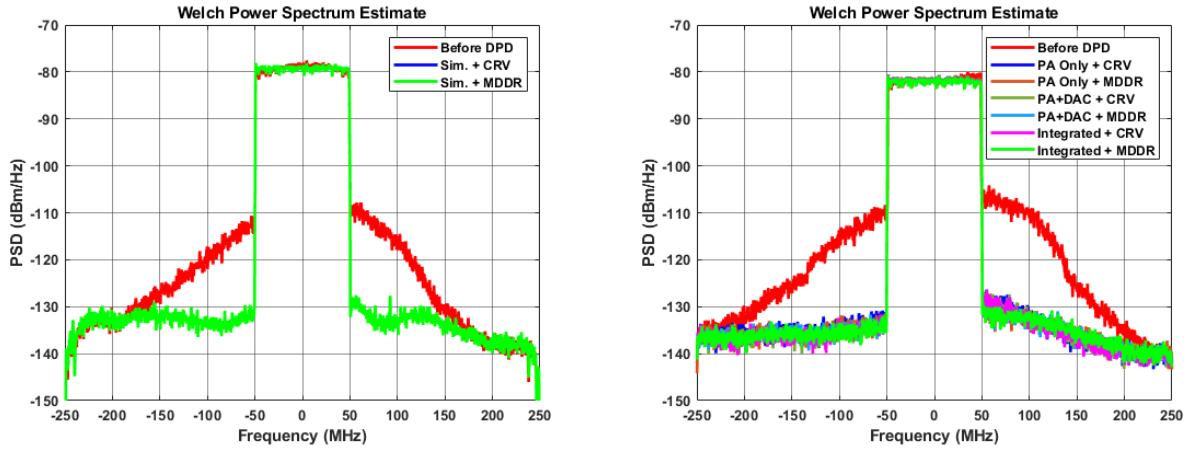
(a) (b)

Figure 3.32: **Multi-stage PA DPD Measurement Results @ 3.4 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results.



(a) (b)

Figure 3.33: **Multi-stage PA DPD Measurement Results @ 3.5 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results.



(a) (b)

Figure 3.34: **Multi-stage PA DPD Measurement Results @ 3.6 GHz Carrier Frequency.** (a) Simulation, (b) Measurements. Measurement results of both versions of multi-stage PA and AMC7834 DAC/ADC controller are compared to simulation results.

1x4 PA Array Design

For building 16-channel fully digital MIMO transmitter front-end, 4×4 configuration is selected. The design goal is to allow any configurations within 4×4 structure to operate. Since AMC7834 gate biasing chip can handle two multi-stage PAs and each PA layout dimension is limited to antenna array spacing, four PA channels are fabricated on the same PCB with two AMC7834 chips, placed at top and bottom of the PCB. In addition, the height of the PCB, mounted on a metal base and heat sinks, is carefully designed to use both polarization ports of the antennas. Figure 3.35 shows the final 1×4 PA Array Design for fabrication submission. It is an extended version of fig. 3.30b. Four (eight) rows of 1×4 PA Array is needed for 16-channel single (dual) polarization 4×4 fully digital MIMO transmitter front-end. This PCB is submitted for fabrication.

DC Power Distribution Board

In addition to transistor gate biasing, each PA requires one channel of 5.5 V and two channels of 40 V. For 16-channel dual polarization setup, 32 DC power connections are needed. For reducing the complexity and number of wire connections and for preventing

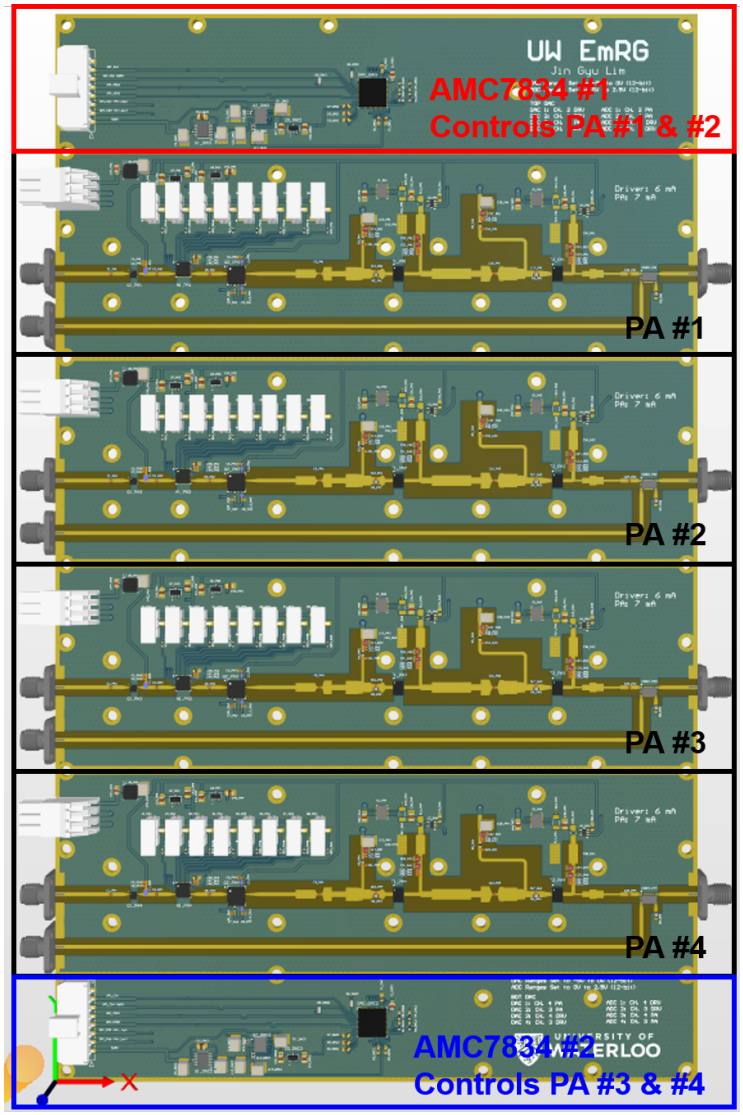
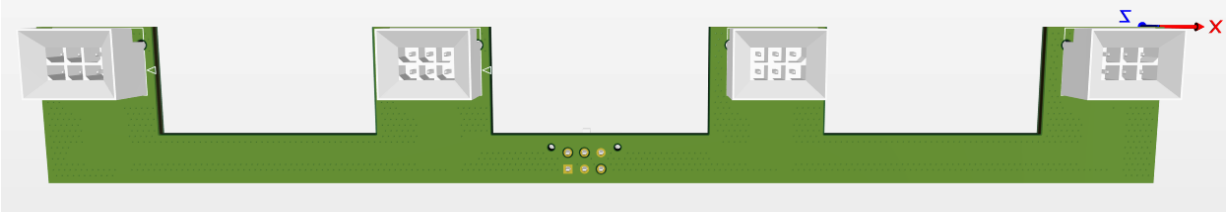
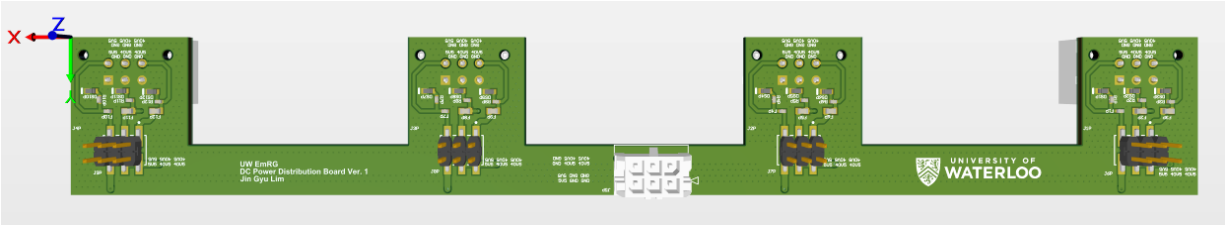


Figure 3.35: **1×4 PA Array Design for 16-Channel Fully Digital MIMO Transmitter Front-End.** Each PA is separated by 60 mm for 0.7 λ antenna spacing at 3.5 GHz.

any possible human errors, a DC power distribution board for the 1×4 PA Array is designed. Figure 3.36 shows the top and bottom view of the designed DC power distribution board. Figure 3.36a is for the connections to 1×4 PA Array, and fig. 3.36b is for the external DC supply connection and jumpers for reconfigurability. LEDs and fuses are added to



(a)



(b)

Figure 3.36: **DC Power Distribution Board.** (a) Connections to 1×4 PA Array, (b) Connections to external DC power supplies and jumpers for reconfigurability.

each DC channel for checking and protecting PAs, respectively. Openings between each DC connector is to provide enough room for input and coupled RF ports of each PA. This PCB is also submitted for fabrication.

Digital Control Board

For SPI digital communication, 1×4 PA array requires six chip select signals (two for AMC7834 transistor gate biasing chips and four for HMC1122 attenuators). For 16-channel single polarization setup, a total of 24 chip select signals are needed. However, National Instruments' USB-8452 I^2C/SPI interface device has eight chip select signals and eight digital input/output signals. Thus, a digital control board, shown in fig. 3.37, is designed to provide more chip select signals, to buffer SPI digital signals, and to supply 5.5 V to AMC7834 chips. SN74HC244 is an SPI-based buffer, which is responsible for taking care of two connectors in the same row. Two connectors in the same row should be connected to the top and bottom AMC7834 connectors of the 1×4 PA array. TCA9534A is an I^2C-based General Purpose Input/Output (GPIO) chip for providing chip select signals to PAs. For controlling the PAs, the digital control board should activate one buffer at a time with its own chip select signal and then apply proper chip select signals from TCA9534A chips for sequentially setting up top and bottom AMC7834 chips and multi-stage PAs controlled by them. This PCB is also submitted for fabrication.

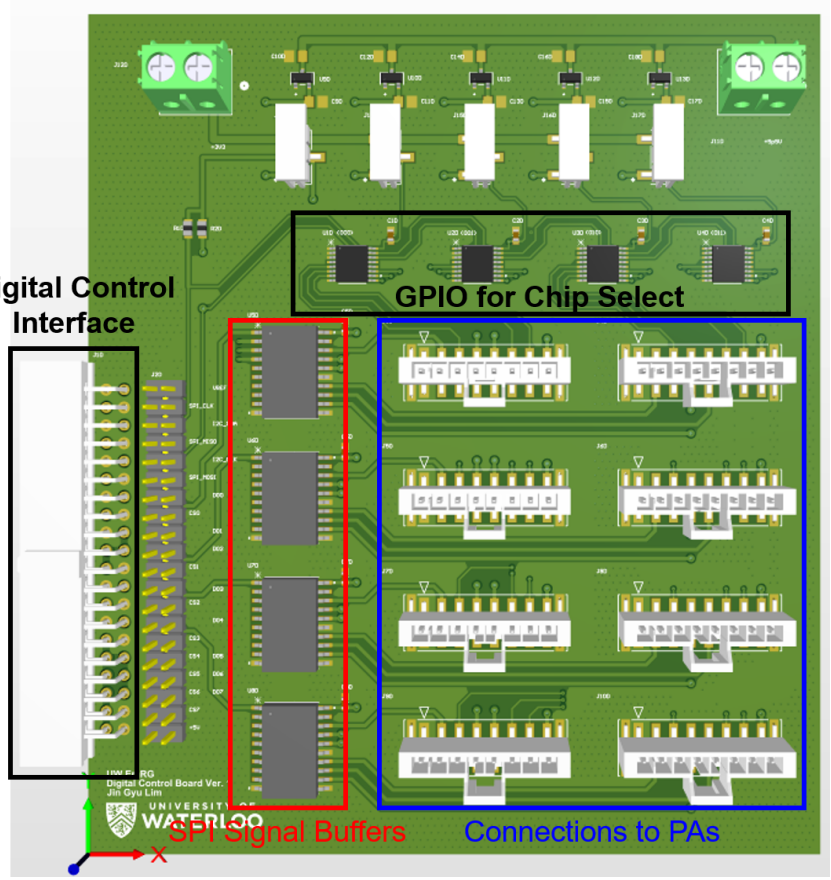


Figure 3.37: **Digital Control Board.** NI USB-8452 is used as the digital control interface board.

Dual-Polarization Patch Antenna Design

The dual-polarization patch antenna design is based on the design shown in fig. 3.38. The reference antenna design consists of three PCBs: antenna feed network, drive patch antenna, and parasitic patch antenna with each PCB having the stack-up information shown in Table 3.4. For the new antenna design, the design goal is to improve antenna matching from -10 dB and remove spacers between patch antennas for mechanical stability. However, due to Bode-Fano criterion, bandwidth has to be reduced at the expense of improved matching. Patch antennas and feed networks dimensions are modified to reach the design goals. Figure 3.39 shows the final dimensions of the newly designed stacked patch antennas submitted for fabrication and fig. 3.40 shows the fabricated antennas.

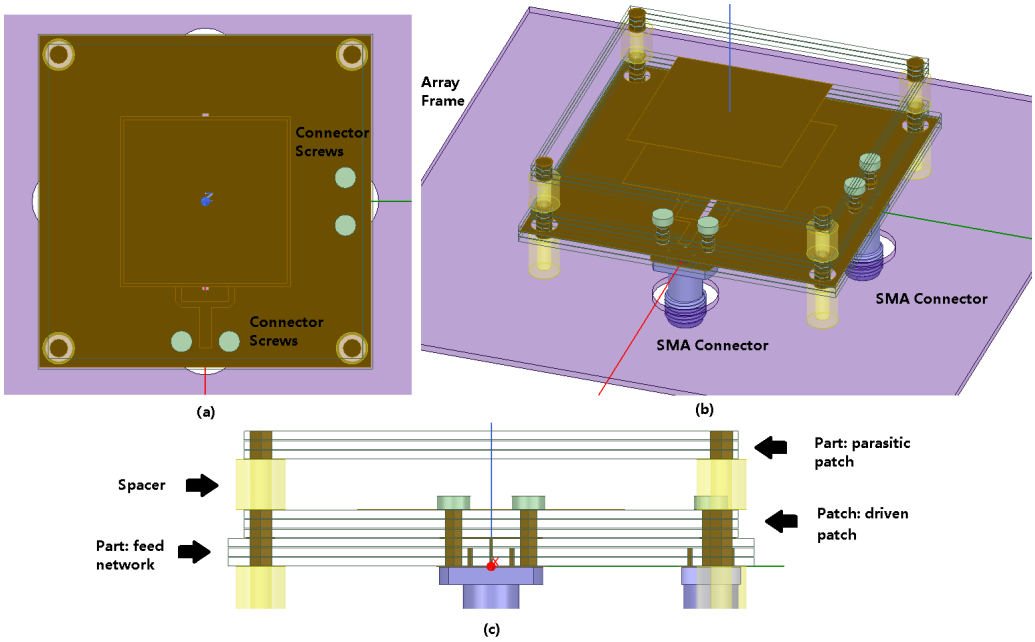


Figure 3.38: **Reference Antenna Design** (a) top view, (b) trimetric view, (c) right view. Reprinted from [11].

Table 3.4: **Antenna PCB Stack-up Information.** Antenna feed and antenna patches are designed on this stack-up and they are stacked to make a 12-layer antenna.

	6L build up	Copper Thickness
L1 L2	0.813mm RO4003C core	Finished 1.0 oz 0.5 oz
	0.189mm RO4450F prepreg	
L3 L4	0.813mm RO4003C core	0 oz 0.5 oz
	0.189mm RO4450F prepreg	
L5 L6	0.813mm RO4003C core	0 oz Finished 1.0 oz
	Finished board thickness: 3 mm+/-10%	
	Pressed board thickness: 2.892 mm+/-2%	

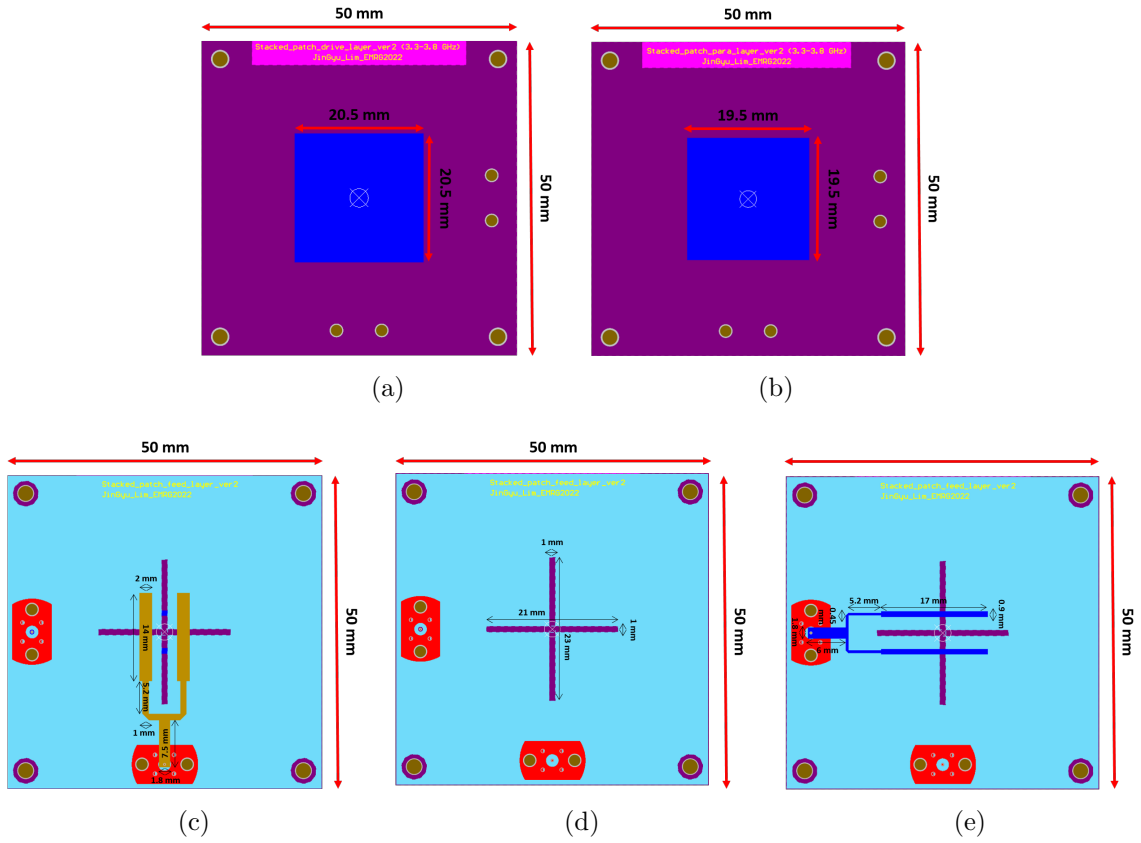
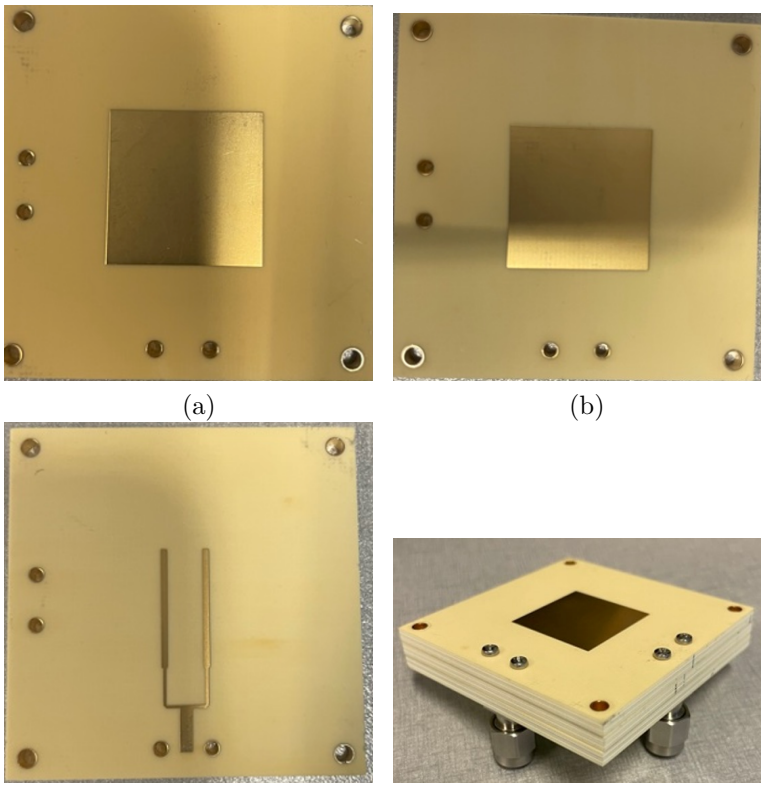
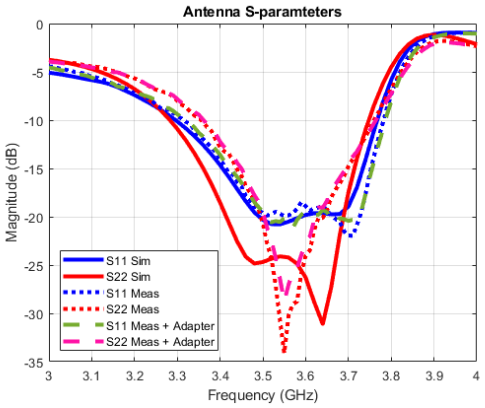


Figure 3.39: **New Antenna Design Dimensions.** (a) Driven patch antenna, (b) Parasitic patch antenna, (c) Bottom feed network, (d) Aperture between bottom and top feed networks, (e) Top feed network.

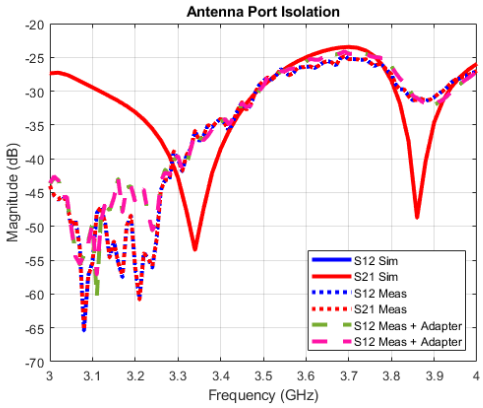


(a) (b) (c) (d)

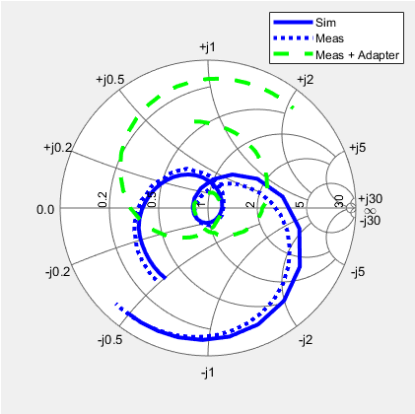
Figure 3.40: **Fabricated New Antennas.** (a) Driven patch antenna, (b) Parasitic patch antenna, (c) Feed network, (d) Assembled stacked patch antenna.



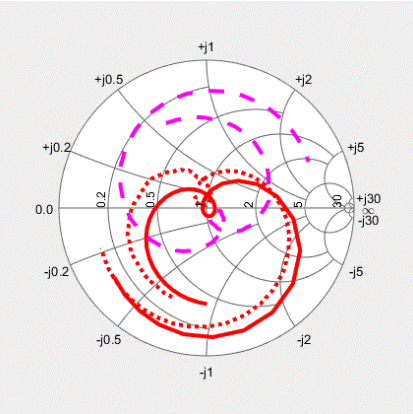
(a) Antenna Matching



(b) Port Isolation



(c) Bottom Feed



(d) Top Feed

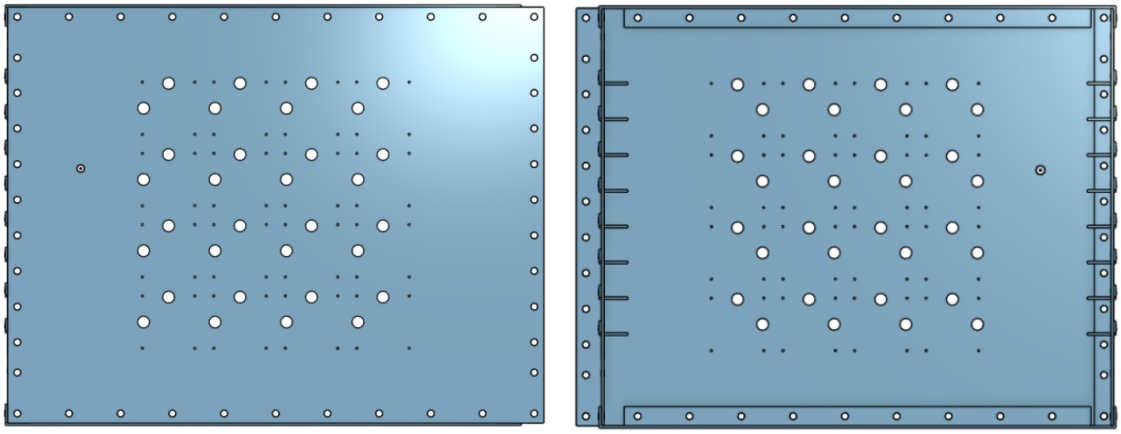
Figure 3.41: **Fabricated Antenna Characterization.** Port 1 is the bottom feed and port 2 is the top feed.

Figure 3.41 summarizes new antenna simulation and measurement results. Port 1 is the bottom feed and port 2 is the top feed. Figure 3.41a shows that for −15 dB or better matching, the bottom feed network operates in 3.4−3.75 GHz and the top feed network 3.34−3.7 GHz. Thus, 3.5 GHz centre frequency would be the best choice for both antenna and PA behaviors. After fabrication, the bottom fed antenna shows almost the same behavior as that of simulation while the top fed antenna behavior changed drastically. Figures 3.41c and 3.41d show that the bottom fed antenna tracks the simulation results on Smith chart but the top fed antenna moved away from the centre of Smith chart. This

caused the top fed matching to become narrowband. The top feed network dimensions were twice as narrow as those of the bottom feed network. This could explain its higher sensitivity to the fabrication tolerances, and any offsets during the assembly process for altering the final characteristics. Figure 3.41b shows port isolation between the bottom and top feeding networks. It is the worst at 3.7 GHz for −25 dB but it improves to −30 dB at 3.5 GHz, where the centre frequency would be located. Unfortunately, this parameter is worsened by 15 dB from the previous version due to the removal of the spacers so for dual-polarization MIMO transmitter behavior, linearization performed by DPD may be challenging. The adapter used in the measurements is Pasternack's push-on adapter (PE91336) for directly connecting antennas and PAs without any cables. For future single polarization measurements, the bottom fed antennas will be mainly used.

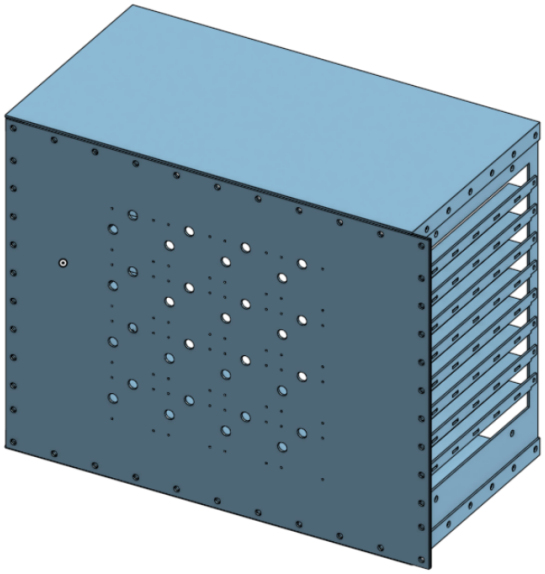
Mechanical Frame Design

Lastly, a mechanical frame to hold dual-polarization 16-channel antennas and multi-stage PAs was designed. The antennas are configured in 4×4 arrangement and 1×4 multi-stage PAs are placed row by row. The main challenges were that the height of the final PA module with a metal base and heat sinks should fit in 21 mm vertical separation between two antenna ports and that the same metal base should work for both rows of antenna ports with 21 mm horizontal separation. Figure 3.42 shows the assembled mechanical frame design. The antennas will be placed at the front view, while the PAs on metal bases will be placed on the shelves on the side frames. For one polarization, the PAs will face up, and for the other, they will face down to fit in the 21 mm separation. Finally, fig. 3.43 shows fully assembled 4×4 fully digital MIMO transmitter front-end. Fans will also be placed at the sides for the temperature control and the extra PCB at the bottom of the frame is the digital control PCB.



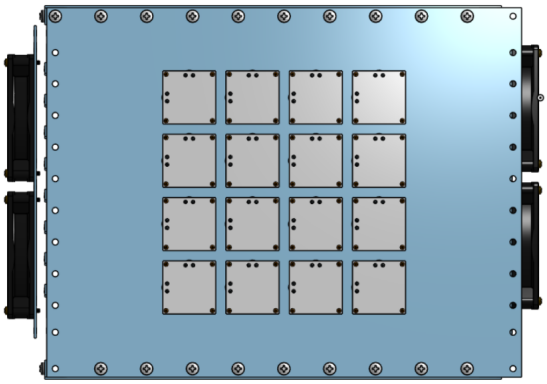
(a) Front View

(b) Back View

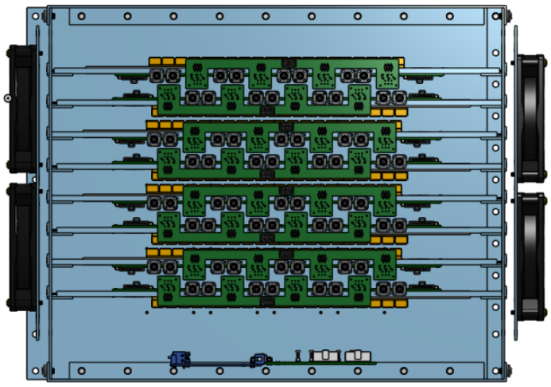


(c) Trimetric View

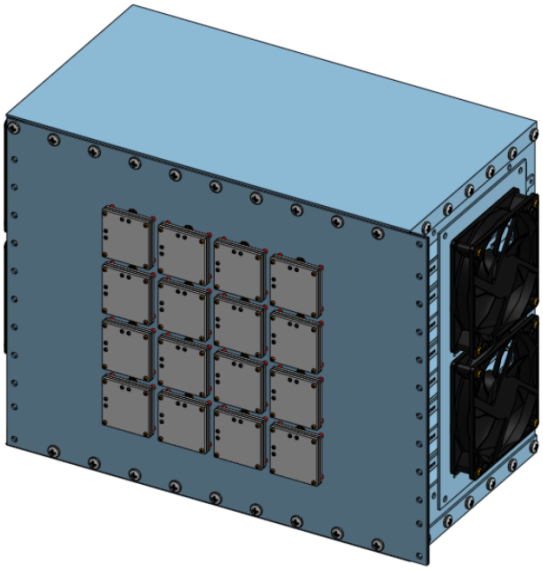
Figure 3.42: **Designed Mechanical Frame.**



(a) Front View



(b) Back View



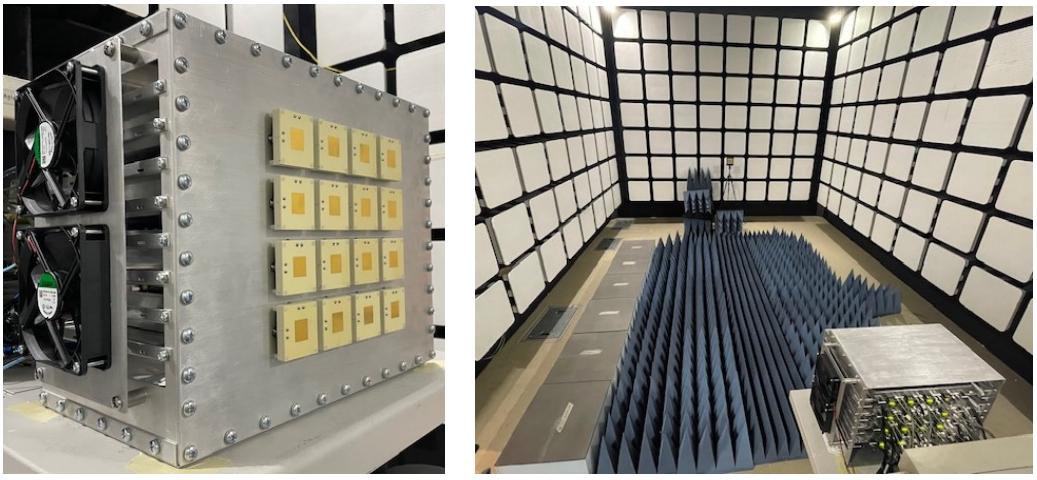
(c) Trimetric View

Figure 3.43: **Designed Mechanical Frame.**

3.4 Conclusion and Outlook

The proposed EM/RF/DSP co-simulation framework can be utilized for predicting system-level MIMO transmitter performance during the design process without any device fabrication. An experimental validation of the framework using a 2×2 fully digital MIMO transmitter, employing class-AB PA and DPA, showed that it can estimate the general trends of the measurements very well. Thus, the framework can be used to co-design different building blocks of the system for the optimal system-level performance without realizing any physical system.

A complete 16-channel dual-polarization fully digital MIMO transmitter testbed was designed. The multistage test PA with a variable gain control can be directly connected to signal sources for better integration. The small signal simulation and measurement results almost overlap each other and the modulated signal simulation and measurement results showed a similar trend before and after linearization. For the dual-polarization antenna, one port exactly matched the simulation results, while the other shifted from the design due to fabrication variation. The properly functioning antenna port can be used for testing 16-channel operation. Figure 3.44 shows the assembled 16-channel testbed inside a 4×5×5 m^3 anechoic chamber. This testbed can be used for experimentally validating both DSP algorithms and transmitter front-end designs for solving problems arising from MIMO architectures.



(a) (b)

Figure 3.44: **Fully Asssembled** 4×4 **MIMO Transmitter.**

Chapter 4

Model-Assisted Emulation for Effective Design and Testing of Power Amplifiers in Massive MIMO Transmitters

Beamforming and massive MIMO architectures are key enablers of emerging terrestrial and non-terrestrial networks (TN–NTN). Their practical realization introduces stringent physical-layer constraints, including wide modulation bandwidths, operation at high carrier frequencies, and tight integration of RF front-ends. These constraints are strongly influenced by the chosen beamforming architecture (digital, analog, or hybrid) and ultimately determine the trade-offs among system complexity, power consumption, and link performance.

Previous studies on fully digital massive MIMO transmitters have shown that antenna mutual coupling substantially degrade system performance [34, 62, 63]. In particular, mutual coupling induces precoder-dependent active load modulation at the PA outputs, which reduces linearity, efficiency, and delivered output power. Since the resulting active loading depends on the propagation channel and the beamforming weights [39, 38], the effective PA behavior becomes channel dependent. Although isolators can mitigate coupling effects, they are impractical in large arrays and at higher frequencies due to size and integration constraints, insertion loss, and cost.

To address mutual-coupling-induced PA performance degradation, two main research directions have emerged. From a DPD perspective, Multiple Input, Single Output (MISO)

and DISO DPD techniques [37] have been proposed to compensate for nonlinearities under active load modulation. In parallel, PA design methodologies have been developed to improve robustness to load variations [6, 4, 48, 5, 46, 47, 50, 52]. However, these methods are commonly validated using CW excitation and a limited set of loading conditions, which do not fully represent the wideband, modulated, and precoder-dependent operating regimes encountered in practical fully digital massive MIMO transmitters. At the antenna level, several approaches aim to reduce mutual coupling [40, 41, 42, 43, 44, 45], but their scalability to large arrays, wide bandwidths, and high frequencies remains challenging. Consequently, existing solutions do not provide a complete and scalable framework for system-aware PA design and testing in beamforming massive MIMO transmitters.

Alternatively, authors in [8] explored EM/RF/DPD co-simulation to predict PA behavior under array loading and to support system-aware PA design and testing. Close agreement between measured and simulated results was reported for four 6-W GaN PAs terminated with a 2×2 antenna array and driven by 100 MHz OFDM signals. Extending this approach to larger scale systems, however, reveals major limitations. On the simulation side, increasing the number of PAs and circuit complexity (e.g., multi-transistor topologies) significantly increases computational cost and can lead to unreliable numerical convergence. On the measurement side, building large-scale coherent wideband MIMO testbeds with accurate feedback paths becomes prohibitively complex and costly. Thus, EM/RF/DPD co-simulation alone is not a practical solution for scalable system-aware PA design and testing.

To reduce measurement complexity, the authors in [13] proposed an emulation system that characterizes PAs in an massive MIMO transmitter using a single PA while emulating coupling-induced active loading. The approach relies on a sequential iterative procedure involving repeated measurement, synthesis, and re-application of reflected signals until convergence. Experimental validation was limited to three PAs driven by a 20-MHz modulated signal, with coupling emulated using power dividers, couplers, and variable attenuators. While effective at a small scale, the iterative nature fundamentally limits scalability and complicates wideband operation in massive MIMO transmitters.

This work introduces a PA model-assisted emulation methodology for effective design and testing of power amplifiers in massive MIMO transmitters. The proposed approach synthesizes the reflected signals at a given PA output by modeling the contributions of the remaining PAs, thereby avoiding their explicit measurement or simulation. Compared to [13], the proposed method significantly reduces the number of required measurements while preserving prediction accuracy. The methodology supports both forward-behavior characterization and linearizability assessment using DPD. The proposed model-assisted emulation is validated through simulations and measurements on a 4×4 fully digital MIMO

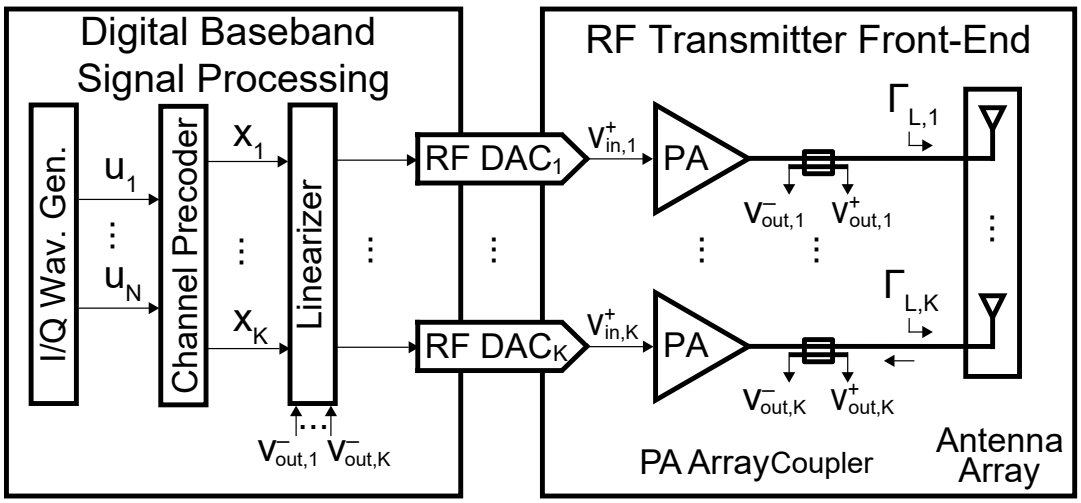


Figure 4.1: Block diagram of a fully digital mMIMO transmitter for PA linearization and output impedance analysis.

transmitter prototype, demonstrating close agreement with full array results and accurate prediction of distortion and DPD performance. The proposed approach therefore enables scalable and system-aware PA design and testing for next-generation massive MIMO transmitters.

4.1 PA Modeling for Fast and Accurate Synthesis of Reflected Signals at PA Outputs

Fig. 4.1 shows the reference fully digital massive MIMO transmitter used to analyze the impact of antenna mutual coupling on PA behavior in simulation and measurement. The digital baseband processing comprises an IQ waveform generator, a channel precoder, and a DPD block. The waveform generator produces L user signals with prescribed modulation format, bandwidth, PAPR, and power level. Precoding is represented by the $K \times L$ matrix, $\mathbf{W}$, where K denotes the number of PAs. The DPD block compensates PA nonlinearities prior to frequency upconversion, while the RF DAC translates the predistorted signals to the RF band. The antenna array is represented using EM-simulated or measured S-parameters.

In the presence of antenna mutual coupling, the output of the ith PA, denoted by

$\mathrm{v}^{-}_{out,i}[n]$, depends on the incident signals at the PA input and output ports, denoted $\mathrm{v}^{+}_{in,i}[n]$ and $\mathrm{v}^{+}_{out,i}[n]$, respectively. This relationship is expressed as

$$\mathrm{v}^{-}_{out,i}[n] = \sum_{p=1}^{P} \alpha_{i,p}\psi_p(\bar{\mathbf{v}}^{+}_{in,i}[n], \bar{\mathbf{v}}^{+}_{out,i}[n]) \tag{4.1}$$

$$= f_i(\bar{\mathbf{v}}^{+}_{in,i}[n], \bar{\mathbf{v}}^{+}_{out,i}[n]), \tag{4.2}$$

where $\bar{\mathbf{v}}^{+}_{in,i}[n] = (\mathrm{v}^{+}_{in,i}[n], \ldots, \mathrm{v}^{+}_{in,i}[n - ML_1])^T$, $\bar{\mathbf{v}}^{+}_{out,i}[n] = (\mathrm{v}^{+}_{out,i}[n], \ldots, \mathrm{v}^{+}_{out,i}[n - ML_2])^T$, and ML_1 and ML_2 denote the memory depths. The functions $\psi_p(\cdot)$ denote nonlinear basis functions and $\alpha_{i,p}$ are the corresponding complex coefficients. According to (4.1), accurate characterization of the ith PA requires knowledge of both $\mathrm{v}^{+}_{in,i}[n]$ and $\mathrm{v}^{+}_{out,i}[n]$. Since $\mathrm{v}^{+}_{in,i}[n]$ is known, the main challenge lies in *synthesizing* $\mathrm{v}^{+}_{out,i}[n]$.

The reflected signals $\mathrm{v}^{+}_{out,i}[n]$ can be expressed as a linear combination of the transmitted signals at K PA outputs through the antenna coupling network:

$$\mathrm{v}^{+}_{out,i}[n] = \sum_{j=1}^{K} \sum_{q=0}^{Q-1} \lambda_{ij}[q]\, \mathrm{v}^{-}_{out,j}[n - q], \tag{4.3}$$

where $\lambda_{ij}[q]$ are coefficients of finite impulse response (FIR) filters derived from the antenna array S-parameters, and Q denotes the filter order. These filters capture both self-reflection and mutual coupling among antenna elements.

Combining (4.1) and (4.3), the PA output can be written as

$$\mathrm{v}^{-}_{out,i}[n] = \sum_{p=1}^{P} \alpha_{i,p}\psi_p\left(\bar{\mathbf{v}}^{+}_{in,i}[n], \sum_{j=1}^{K} \sum_{q=0}^{Q-1} \lambda_{ij}[q]\, \bar{\mathbf{v}}^{-}_{out,j}[n - q]\right). \tag{4.4}$$

Thus, characterizing the ith PA under active load modulation reduces to estimating the remaining $K-1$ PA output signals $\mathrm{v}^{-}_{out,j}[n]$ for known antenna S-parameters. In principle, for an massive MIMO transmitter with K PAs, the vectors $\bar{\mathbf{v}}^{+}_{in}[n]$ and $\bar{\mathbf{v}}^{-}_{out}[n]$ can be obtained by full-array co-simulation or by constructing a full coherent massive MIMO testbed. Both options rapidly become impractical as K grows due to co-simulation burden and convergence risk, or testbed cost and measurement time.

To reduce the experimental burden, [13] proposed an emulation system that sequentially measures $\mathrm{v}^{-}_{out,j}[n]$ to form $\bar{\mathbf{v}}^{-}_{out}[n]$, and then synthesizes $\bar{\mathbf{v}}^{+}_{out}[n]$ using (4.3). Convergence typically required 3–4 iterations of K sequential measurements. Using [13], the required

number of measurements and total characterization time scale with array size, limiting applicability to large systems.

This work reduces the measurements required to obtain $\mathrm{v}^{+}_{out,i}[n]$ in (4.1) for characterizing the ith PA. In contrast to [13], the proposed method requires only a single measurement of $\mathrm{v}^{-}_{out,i}[n]$. The remaining $K-1$ outputs $\mathrm{v}^{-}_{out,j}[n]$ are generated using a PA model.

Regarding the PA model, each PA is treated as a weakly nonlinear system where the dominant behavior can be captured by a memoryless nonlinearity with additional memory effects of decreasing contribution. Two models are considered to trade off complexity and emulation accuracy: a SISO model and a DISO model.

The SISO model, assumes that $K-1$ PAs in the array can be modeled as SISO function removing the dependence their incident signals at the PA output. Accordingly, $\mathrm{v}^{-}_{out,j}[n]$ for $j \neq i$ can be expressed as follows:

$$\mathrm{v}^{-}_{out,j}[n] = \sum_{p=1}^{P} \hat{\alpha}_{i,p} \psi_p \left(\bar{\mathbf{v}}^{+}_{in,j}[n], \mathbf{0} \right) \tag{4.5}$$

$$= f_i \left(\bar{\mathbf{v}}^{+}_{in,j}[n], \mathbf{0} \right), \tag{4.6}$$

where $\mathbf{0}$ represents zero vectors for basis functions that include $\bar{\mathbf{v}}^{+}_{out,j}[n]$ and $\hat{\alpha}_{i,p}$ denotes an estimation of the PA coefficients in (4.1).

Using (4.5) the incident signal at the ith PA output can then be obtained as follows:

$$\begin{aligned} \mathrm{v}^{+}_{out,i}[n] &= \sum_{j=1}^{N} \sum_{q=0}^{Q-1} \lambda_{ij}[q] \left(\sum_{p=1}^{P} \hat{\alpha}_{i,p} \psi_p (\bar{\mathbf{v}}^{+}_{in,j}[n-q], \mathbf{0}) \right) \\ &= \sum_{j=1}^{N} \sum_{q=0}^{Q-1} \sum_{p=1}^{P} \hat{\alpha}_{i,p} \lambda_{ij}[q] \psi_p \left(\bar{\mathbf{v}}^{+}_{in,j}[n-q], \mathbf{0} \right), \end{aligned} \tag{4.7}$$

where $\alpha_{i,p}$ are trained using $\mathrm{v}^{+}_{in,i}[n]$ and $\mathrm{v}^{-}_{out,i}[n]$. With the SISO model, explicit measurement or simulation of $\mathrm{v}^{-}_{out,j}[n]$ for $j \neq i$ is not required.

Similarly, for the DISO model, $\mathrm{v}^{-}_{out,j}[n]$ for $j \neq i$ can be expressed as follows:

$$\mathrm{v}^{-}_{out,j}[n] = \sum_{p=1}^{P} \hat{\alpha}_{i,p} \psi_p (\bar{\mathbf{v}}^{+}_{in,j}[n], \bar{\mathbf{v}}^{+}_{out,j}[n]) \tag{4.8}$$

$$= f_i (\bar{\mathbf{v}}^{+}_{in,j}[n], \bar{\mathbf{v}}^{+}_{out,j}[n]), \tag{4.9}$$

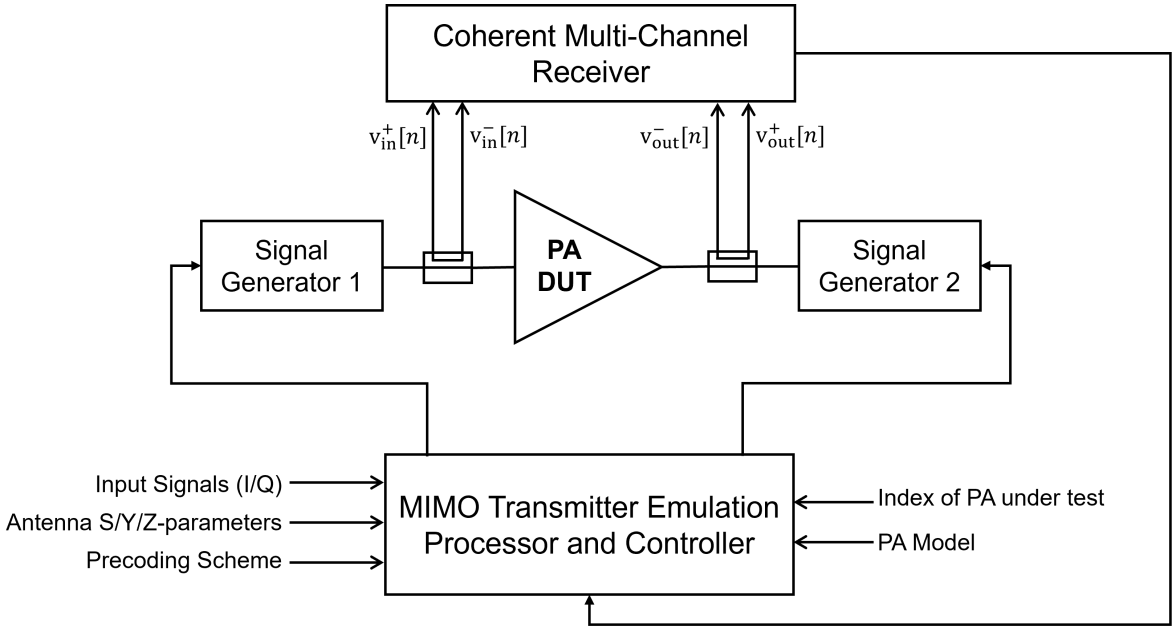


Figure 4.2: Block diagram of the proposed PA model-assisted emulation system for predictive PA design and testing.

Using (4.8) the incident signal at the ith PA output can then be obtained as follows:

$$\begin{aligned} v_{out,i}^{+}[n] &= \sum_{j=1}^{N}\sum_{q=0}^{Q-1}\lambda_{ij}[q]\left(\sum_{p=1}^{P}\hat{\alpha}_{i,p}\psi_p\left(\bar{\mathbf{v}}_{in,j}^{+}[n],\bar{\mathbf{v}}_{out,j}^{+}[n]\right)\right) \\ &= \sum_{j=1}^{N}\sum_{q=0}^{Q-1}\sum_{p=1}^{P}\hat{\alpha}_{i,p}\lambda_{ij}[q]\psi_p\left(\bar{\mathbf{v}}_{in,j}^{+}[n],\bar{\mathbf{v}}_{out,j}^{+}[n]\right), \end{aligned} \tag{4.10}$$

where $\alpha_{i,p}$ are trained using $v_{in,i}^{+}[n]$, $v_{out,i}^{+}[n]$, and $v_{out,i}^{-}[n]$. The basis set includes SISO terms and additional terms capturing interaction between $\bar{\mathbf{v}}_{in,i}^{+}[n]$ and $\bar{\mathbf{v}}_{out,i}^{+}[n]$. Unlike the SISO model, estimating $v_{out,i}^{+}[n]$ in the DISO case generally requires $v_{out,j}^{+}[n]$ for accurate synthesis, which motivates the convergence procedure described next.

4.2 Proposed Model-Assisted Emulation Procedure

Fig. 4.2 summarizes the proposed PA model-assisted emulation procedure for simulation and measurement. Inputs include: i) user-defined test waveforms; ii) the index of the PA

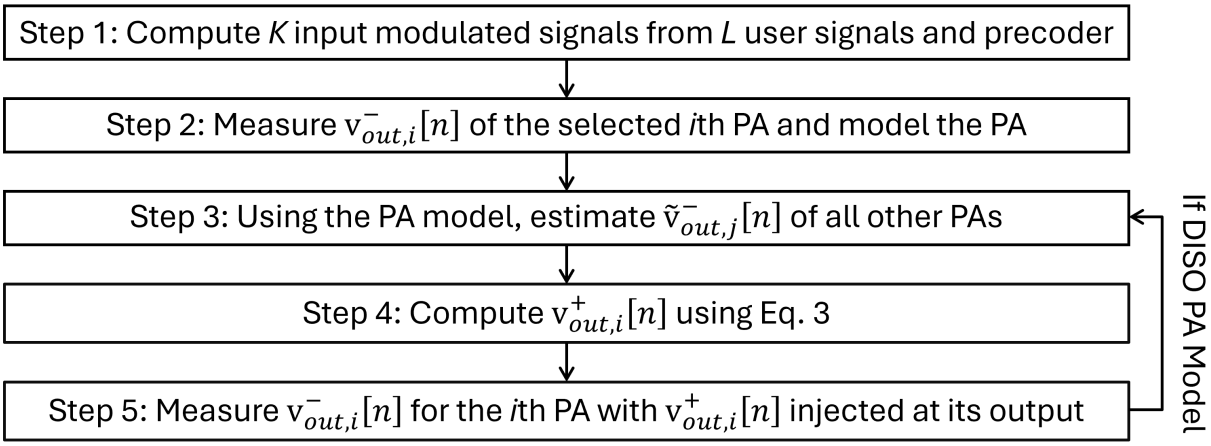


Figure 4.3: Proposed PA model-assisted emulation procedure for PA simulation and testing.

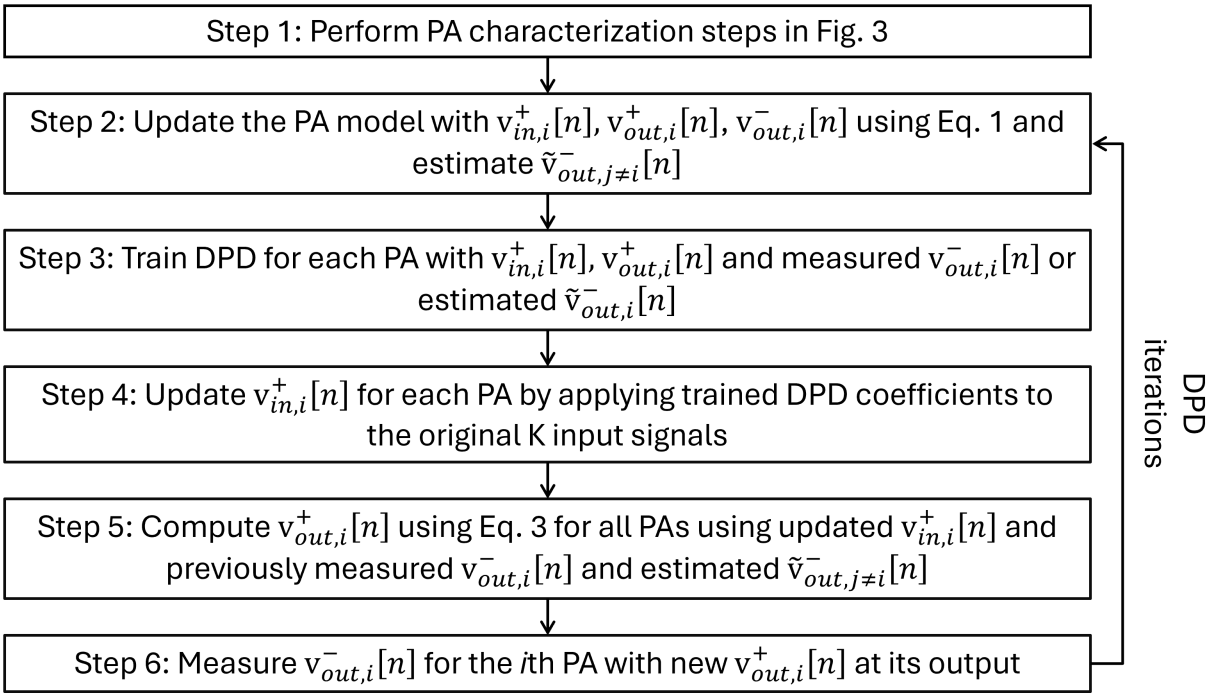


Figure 4.4: Proposed PA model-assisted emulation procedure with linearizability assessment step in PA simulation and testing.

under test; iii) a PA behavioral model; iv) the precoding scheme; and v) antenna array S-, Y-, or Z-parameters. The procedure applies precoding to generate the incident input signal of the ith PA, $\mathrm{v}^{+}_{in,i}[n]$, uploaded to signal generator #1. It then synthesizes the reflected signal incident at the PA output, $\mathrm{v}^{+}_{out,i}[n]$, and uploads it to signal generator #2. The same process is repeated to characterize different PAs within the array.

Fig. 4.3 details the procedure for PA simulation and testing. Compared to [13], the proposed approach adds a pre-characterization stage to extract a PA model, enabling prediction of the remaining $K-1$ PA outputs and avoiding their explicit measurement when synthesizing $\mathrm{v}^{+}_{out,i}[n]$. The procedure is:

1. Compute $\mathrm{v}^{+}_{in,i}[n]$ for all K PAs using the L user waveforms, and the precoding scheme.
2. Measure $\mathrm{v}^{-}_{out,i}[n]$ of the selected ith PA and identify the corresponding PA model.
3. Using the identified model, estimate $\tilde{\mathrm{v}}^{-}_{out,j}[n]$ for the remaining $K-1$ PAs driven by $\mathrm{v}^{+}_{in,j}[n]$.
4. Synthesize $\mathrm{v}^{+}_{out,i}[n]$ using (4.3) and the estimated $\tilde{\mathrm{v}}^{-}_{out,j}[n]$.
5. Measure $\mathrm{v}^{-}_{out,i}[n]$ with $\mathrm{v}^{+}_{out,i}[n]$ incident at the output for characterization under active load modulation.

If the PA model is SISO, Step 2 terminates the PA with the corresponding antenna input reflection coefficient while neglecting mutual coupling, providing an efficient approximation. If a DISO model is used, the procedure in Fig. 4.3 is repeated until convergence, thereby accounting for mutual coupling effects in the synthesized reflected signal.

The procedure above characterizes the forward PA behavior. When assessing PA linearizability within the massive MIMO transmitter, the process is extended as shown in Fig. 4.4. Step 1 is identical to the five-step procedure above. The additional steps iteratively train DPD coefficients, update excitations, and measure the linearized output:

1. Perform the five-step characterization procedure in Fig. 4.3.
2. Update the PA model in (4.1) using measured $\mathrm{v}^{-}_{out,i}[n]$, $\mathrm{v}^{+}_{in,i}[n]$, and synthesized $\mathrm{v}^{+}_{out,i}[n]$ to refine $\tilde{\mathrm{v}}^{-}_{out,j}[n]$.
3. Train DPD coefficients for each PA using the latest $\mathrm{v}^{+}_{in,i}[n]$, $\mathrm{v}^{+}_{out,i}[n]$, and $\mathrm{v}^{-}_{out,i}[n]$ (measured or estimated).

Table 4.1: Summary of the required number of measurements/simulations for the proposed PA model-assisted emulation approach and in [13]. K: total number of PAs; N: number of $\mathrm{v}^{-}_{out,i}[n]$ convergence iterations (only needed for the DISO PA model in the proposed approach); M: number of DPD iterations.

Emulation Mode	PA Model	Proposed Approach	[22]
Forward Modeling	SISO	$K+1$	$K\times N$
	DISO	$K\times N$	
Linearization Assessment	SISO	$K\times M+1$	$K\times N\times M$
	DISO	$K\times\max\{M,N\}+1$	

4. Update $\mathrm{v}^{+}_{in,i}[n]$ for all PAs using the predistorted signals.
5. Synthesize $\mathrm{v}^{+}_{out,i}[n]$ for the ith PA using (4.3) with updated $\mathrm{v}^{+}_{in,i}[n]$ and estimated $\tilde{\mathrm{v}}^{-}_{out,j}[n]$.
6. Measure $\mathrm{v}^{-}_{out,i}[n]$ with the updated $\mathrm{v}^{+}_{out,i}[n]$ incident at the output.
7. Repeat Steps 2–6 for M DPD iterations.

According to Table 4.1, when the proposed approach employs the SISO PA model, the number of required measurements/simulations is reduced from $K\times N$ in [13] to $K+1$. This includes one step for PA pre-characterization/modeling, followed by one emulation run per PA. When the DISO PA model is used, both the proposed approach and [13] require $K\times N$ steps to achieve convergence. However, when linearizability assessment is included, the proposed approach reduces the required steps from $K\times N\times M$ to $K\times M+1$ for the SISO PA model and to $K\times\max\{M,N\}+1$ for the DISO PA model. For the linearziation assessment using the DISO PA model, the number of simulations or measurements is reduced to one run per PA. For the initial $\mathrm{v}^{+}_{out,i}[n]$ estimation, $\mathrm{v}^{-}_{out,j}[n]$ from the SISO PA model are used to skip convergence iterations. Subsequently, the DISO PA model and $\mathrm{v}^{+}_{out,i}[n]$ are updated with every DPD iteration to simultaneously achieve both model convergence and DPD linearization. Therefore, a few more DPD iterations may be needed for more accurate model convergence.

Table 4.2: PA model basis functions used for co-simulation and testing. SISO and DISO PA models are memoryless.

PA Model	Basis Functions $\psi_p(\cdot)$
SISO	$v^+_{in,i}[n]$, $v^+_{in,i}[n]\|v^+_{in,i}[n]\|^2$, $v^+_{in,i}[n]\|v^+_{in,i}[n]\|^4$
DISO	$v^+_{in,i}[n]$, $v^+_{in,i}[n]\|v^+_{in,i}[n]\|^2$, $v^+_{in,i}[n]\|v^+_{in,i}[n]\|^4$, $v^+_{out,i}[n]$

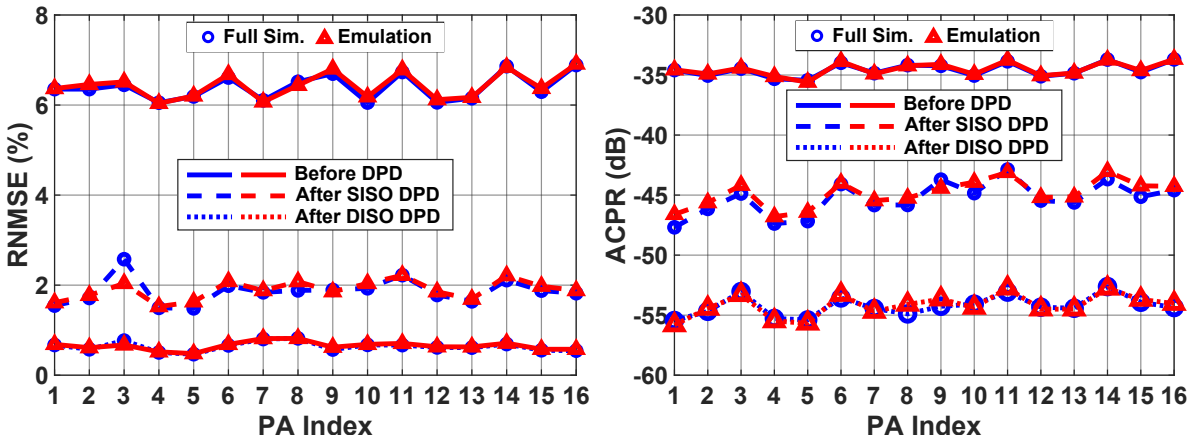
4.3 Validation of the Proposed PA model-assisted Emulation Approach

4.3.1 Application to PA-Array Co-Simulations

Motivated by the demonstrated ability of the EM/RF/DPD co-simulation approach in [8] to predict measured behavior of PAs in 2×2 MIMO transmitters, the proposed emulation procedure is applied to address key challenges in co-simulation of large-scale arrays, including divergence in harmonic-balance/envelope simulators and excessive runtime even when convergence is achieved.

Sixteen 6-W class-AB GaN PAs and a 4×4 antenna array, designed for 3.5 GHz operation, are used to form a 4×4 fully digital MIMO transmitter under test. Sixteen uncorrelated OFDM signals with 100 MHz bandwidth and 8 dB PAPR are used as excitations. Of 20000 signal samples, half is used for training DPD coefficients and the other half is used as the verification samples. Full-array co-simulation results are used as a reference to evaluate the proposed emulation procedure using the PA models in Table 4.2. A 4×4 configuration is selected to balance practical implementation constraints with manageable co-simulation time and reduced divergence risk.

With the SISO PA model, Fig. 4.5 shows that, before DPD, predicted RNMSE, ACPR, gain compression, and power spectral density (PSD) closely match full array co-simulation. When linearizability assessment is required, the DISO PA model emulation yields improved agreement with full-array co-simulation after SISO DPD in both RNMSE and ACPR in Fig. 4.7. In particular, the SISO PA model underestimates ACPR by 0.5 dB. It indicates that higher emulation fidelity is needed once intrinsic nonlinearity is reduced by SISO DPD. However, after DISO DPD linearization, both SISO and DISO PA models show the



(a) (b)

Figure 4.5: Comparison of simulated PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model.

same estimation capacity.

When the SISO PA model is used to synthesize $\mathrm{v}^{+}_{out,i}[n]$ using (4.7), the impact of *active load modulation* on the remaining $K-1$ PA outputs $\mathrm{v}^{-}_{out,j}[n]$ is not explicitly represented. The strong agreement before DPD in Fig. 4.5 indicates that the ith PA response is dominated by its intrinsic nonlinearity in this regime. After SISO DPD, intrinsic nonlinearity is reduced, and residual distortion becomes more sensitive to coupling-induced effects on the remaining $K-1$ PAs. The DISO PA model, on the other hand, incorporates the coupling-dependent reflected signal when synthesizing $\mathrm{v}^{+}_{out,i}[n]$ and thus provides accurate linearizability prediction using SISO DPD. For DISO DPD linearization prediction, both PA models give similar results because they are affected by the predistorted signals. The predistorted signal from the DISO DPD is as follows:

$$\mathrm{z}_i[n] = g_i(\mathrm{x}_i[n], \tilde{\mathrm{v}}^{+}_{out,i}[n]), \tag{4.11}$$

where $\mathrm{x}_i[n]$ is the original uncorrelated signal and $\tilde{\mathrm{v}}^{+}_{out,i}[n]$ is estimated using $\tilde{\mathbf{v}}^{-}_{\mathbf{out}}[\mathbf{n}] = (\tilde{\mathrm{v}}^{-}_{out,1}[n], \ldots, \tilde{\mathrm{v}}^{-}_{out,K}[n])^T$ and antenna S-parameters in the baseband. As a result, $\bar{\mathbf{v}}^{+}_{in,j}[n]$ in (4.5) becomes a function of $\mathrm{x}_i[n]$ and $\tilde{\mathrm{v}}^{+}_{out,i}[n]$, transforming the SISO PA model into the DISO PA model. Therefore, the estimation of $K-1$ PA outputs $\mathrm{v}^{-}_{out,j}[n]$ using the SISO PA model becomes accurate by incorporating the impact of active load modulation. Furthermore, $\bar{\mathbf{v}}^{+}_{out,j}[n]$ in (4.8) becomes redundant. Fig. 4.6 and Fig. 4.8 show that spectra

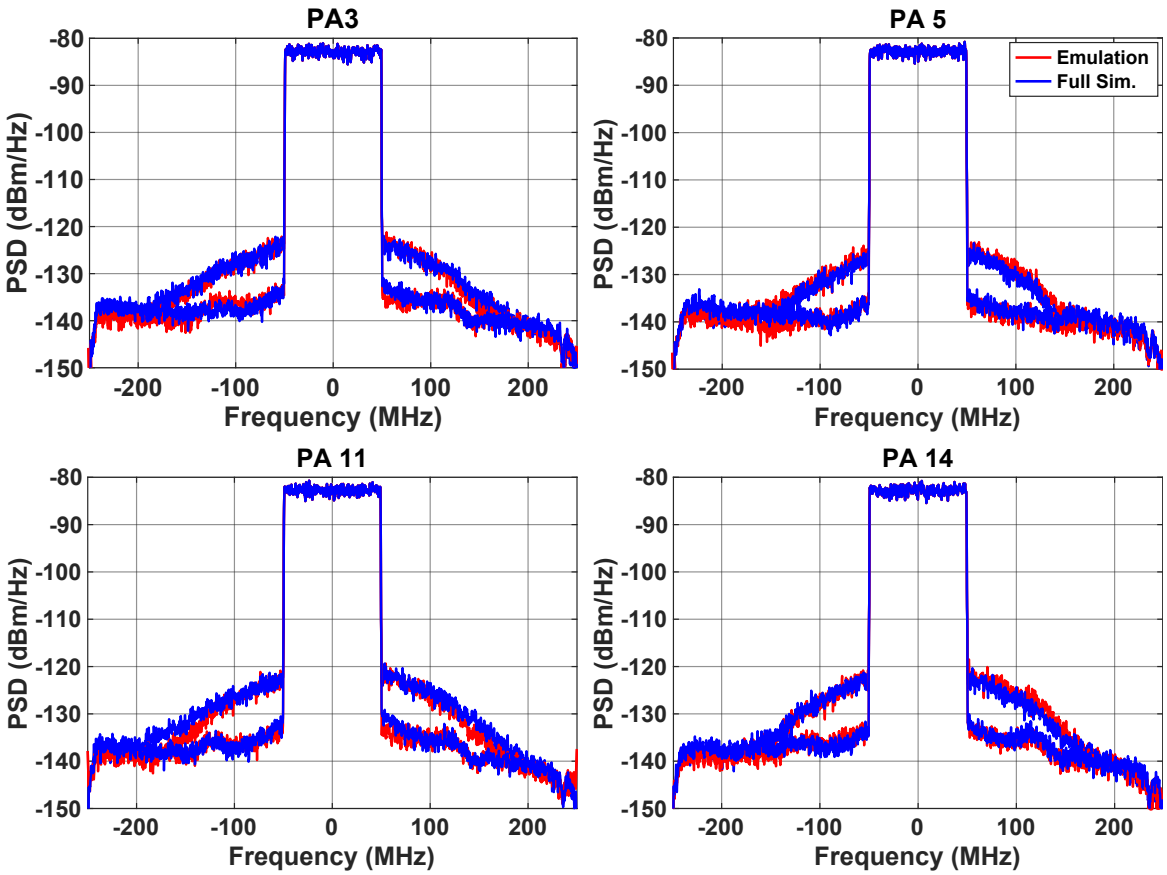
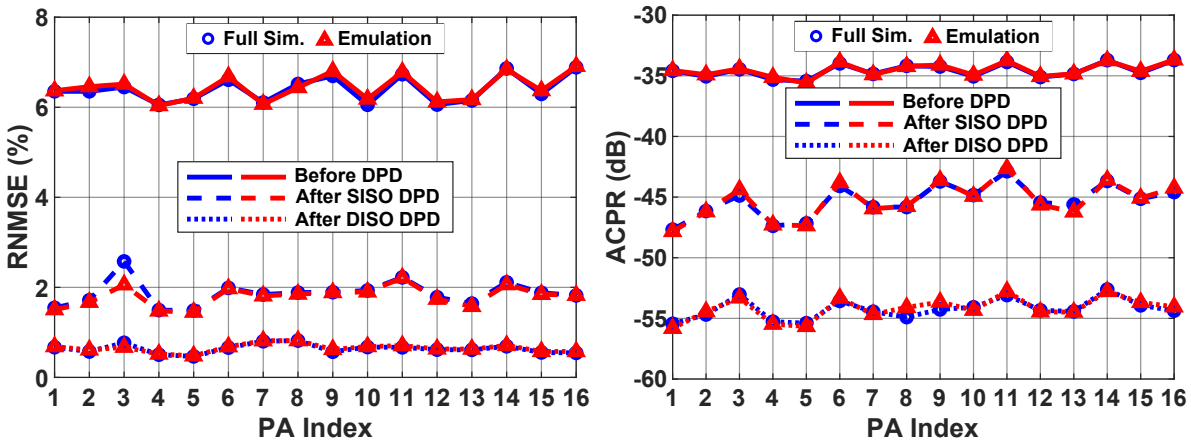


Figure 4.6: Spectra of selected PAs after SISO and DISO DPD in the 4×4 MIMO transmitter front-end using a memoryless SISO PA model in simulation.

obtained with SISO and DISO DPD are nearly identical for both PA models, supporting the conclusion that combining the proposed emulation procedure with the co-simulation framework in [8] enables PA design, characterization, and linearization under realistic coupling-induced active loading prior to fabrication. The trained SISO and DISO DPD should perform well when different input uncorrelated signals are applied. The SISO PA model can be used both DPD models, but for achieving both linearization prediction accuracy and reduced overall modeling complexity, the use of following pairs are recommended: {SISO DPD, DISO PA} and {DISO DPD, SISO PA}.



(a) (b)

Figure 4.7: Comparison of simulated PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model.

4.3.2 Application to PA Testing

To experimentally validate the proposed PA model-assisted emulation approach, measurements from the full 4×4 MIMO transmitter front-end are compared against those obtained using the proposed PA model-assisted emulation setup. The transmitter building blocks are described first, followed by measurement configurations and results.

Transmitter Front-End Building Blocks

Fig. 4.9(a) shows the custom multistage PA, comprising a variable-gain pre-driver, a driver stage, and a final PA stage, enabling direct connection between signal sources and PA inputs. The pre-driver uses an attenuator (Analog Devices HMC1122) between pre-drivers (Qorvo QPA9127 and QPA9842), providing approximately 25 dB gain over 2.7−3.8 GHz. The driver and final PA stages use 6 W GaN transistors (MACOM CGHV1F006S) and were co-designed to improve overall linearity. Over 3.2−3.8 GHz, the small-signal gain ranges from 25 to 28 dB, and at a saturated output power of 38 dBm, the gain compression varies between 1.4 and 3 dB. A 30 dB directional coupler (Anaren X4C30F1-30) is placed at the PA output to provide coupled signals for DPD training. Linearizability was previously verified within the co-simulation framework of [8] using 100 MHz, 8 dB PAPR modulated

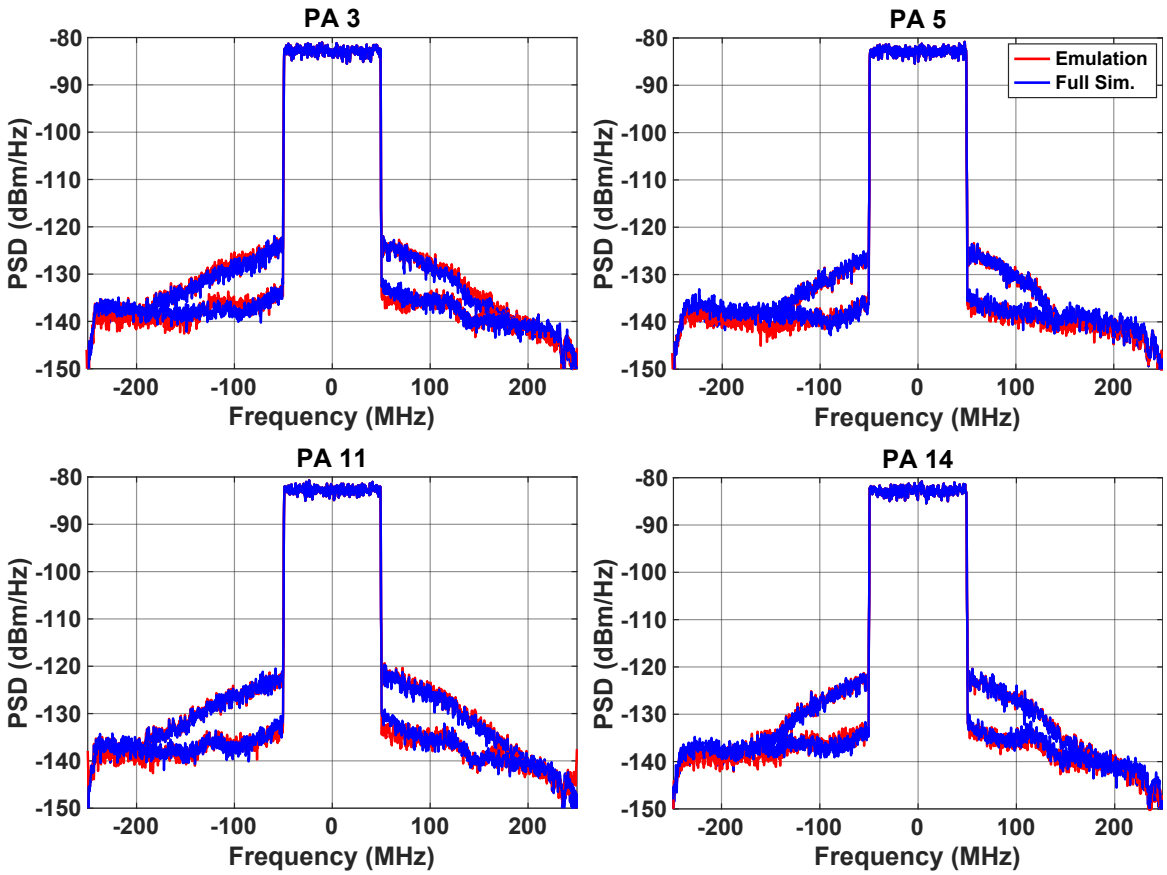


Figure 4.8: Spectra of selected PAs after SISO and DISO DPD in the 4×4 MIMO transmitter front-end using a memoryless DISO PA model in simulation.

signals.

For the 4×4 MIMO front-end, four multistage PAs are integrated on a single printed circuit board (PCB), as shown in Fig. 4.9(c), enabling direct connection to the antenna array. The PA height matches the antenna element spacing. DACs (Texas Instruments AMC7834) provide gate biases for the driver and final stage transistors. DAC #1 and DAC #2, placed at the top and bottom of the board in Fig. 4.9(c), control PAs 1–2 and PAs 3–4, respectively. Push-on SMA adapters (Pasternack PE91336) are used at the antenna interfaces to enable direct PA-to-antenna connections and to avoid additional impedance transformations.

Fig. 4.9(b) shows the dual-polarization stacked patch antenna designed for compact size

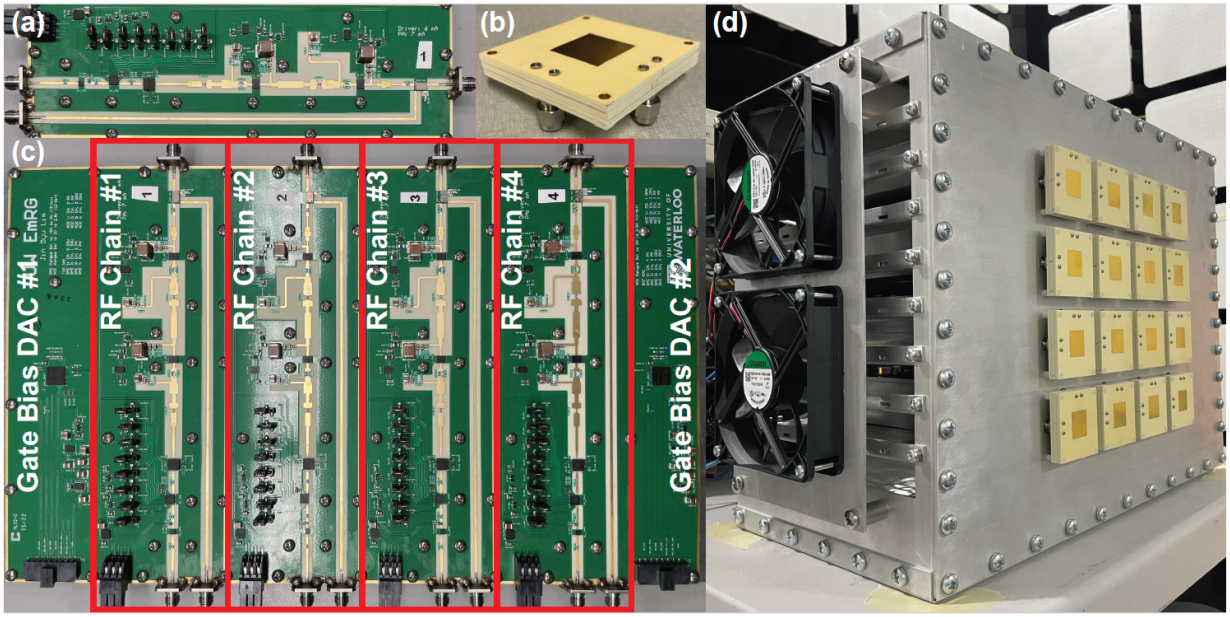


Figure 4.9: Fully assembled and building blocks of the 4×4 MIMO transmitter front-end. (a) single-chain multistage PA, (b) dual-polarization stacked patch antenna element, (c) 1×4 multistage PA module, and (d) 4×4 antenna array mounted on the mechanical frame.

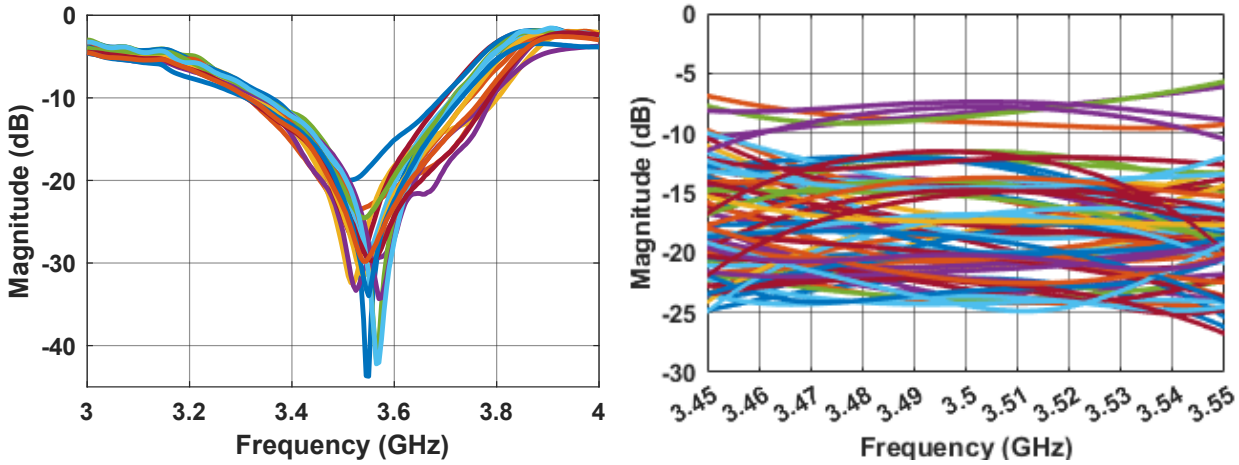


Figure 4.10: Measured 4×4 stacked patch antenna array performance. Left: self-reflection; right: mutual coupling.

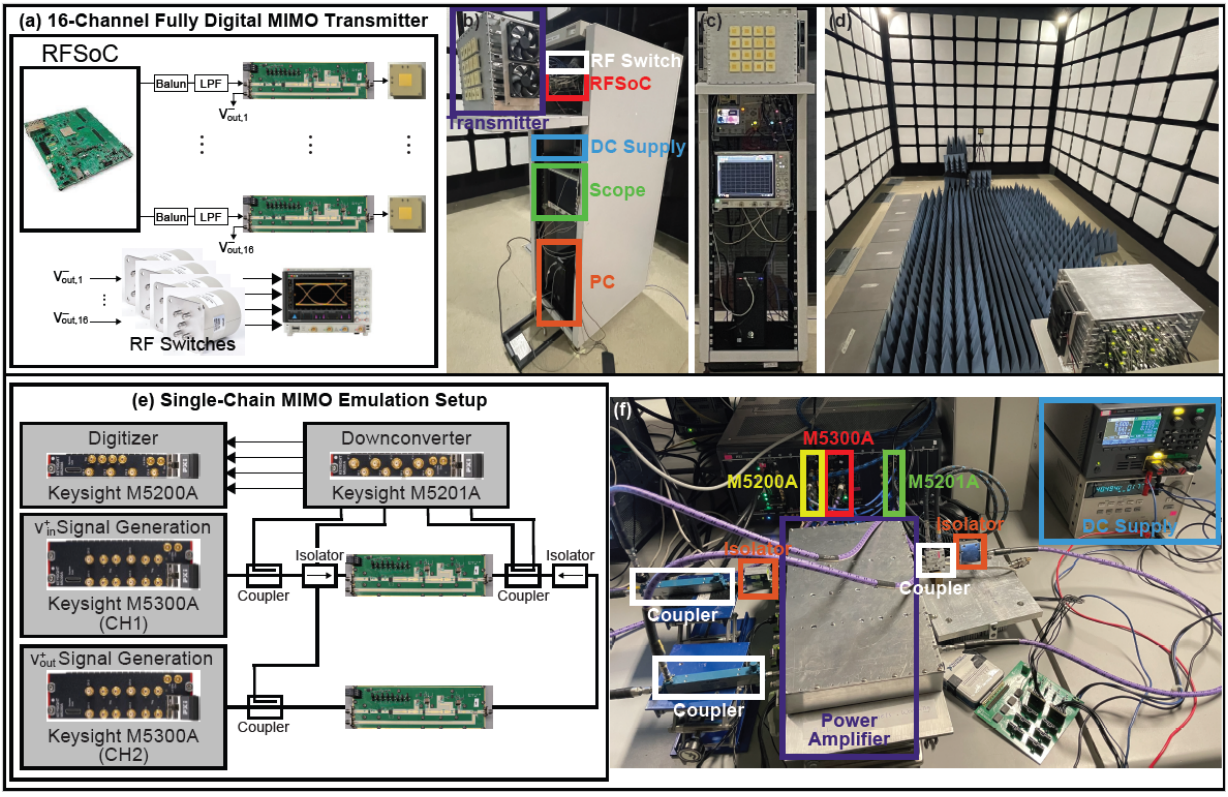


Figure 4.11: Full 4×4 MIMO transmitter system and proposed PA model-assisted emulation measurement setups. (a) full-system block diagram, (b) side view, (c) front view, (d) back view in anechoic chamber, (e) emulation setup block diagram, (f) emulation measurement setup.

and wide bandwidth. Aperture- and coupled-feeding techniques [64] are used to improve port isolation. Both ports target 3.3−3.7 GHz operation with −10 dB or better matching. Port isolation exceeds −25 dB across the band; only one polarization is used in this work. Port 2 is selected due to improved matching of approximately −20 dB for the 75 MHz modulated signals centered at 3.5 GHz.

Fig. 4.9(d) shows the assembled 4×4 antenna array mounted on the mechanical frame, and Fig. 4.10 summarizes measured array performance. All sixteen elements exhibit similar matching, achieving −18 dB or better return loss at 3.5 GHz. Dominant coupling levels relevant to PA behavior are approximately −8 dB, −12 dB, and −15 dB.

Measurement Setups

Fig. C.2(a)–(d) shows the assembled 16-channel digital MIMO transmitter system inside a 4×5×5 m^3 anechoic chamber. A radio-frequency system-on-chip (RFSoC) FPGA board (Xilinx ZCU216) generates 75 MHz bandwidth modulated signals centered at 3.5 GHz with oversampling factor 8 and PAPR of 8 dB. DAC outputs directly drive PA inputs, while coupled PA outputs are routed to a four-channel 8 GHz oscilloscope (Keysight MSOS804A) through 4-to-1 mechanical RF switches.

The proposed model-assisted emulation setup is shown in Fig. C.2(e)–(f). Arbitrary waveform generators (AWGs; Keysight M5300A) generate the PA input signal $v^{+}_{in,i}[n]$ and the synthesized reflected signal $v^{+}_{out,i}[n]$, matching the full-system signal characteristics. On the receiver side, downconverters (Keysight M5201A) with 2 GHz intermediate frequency (IF) and digitizers (Keysight M5200A) are used.

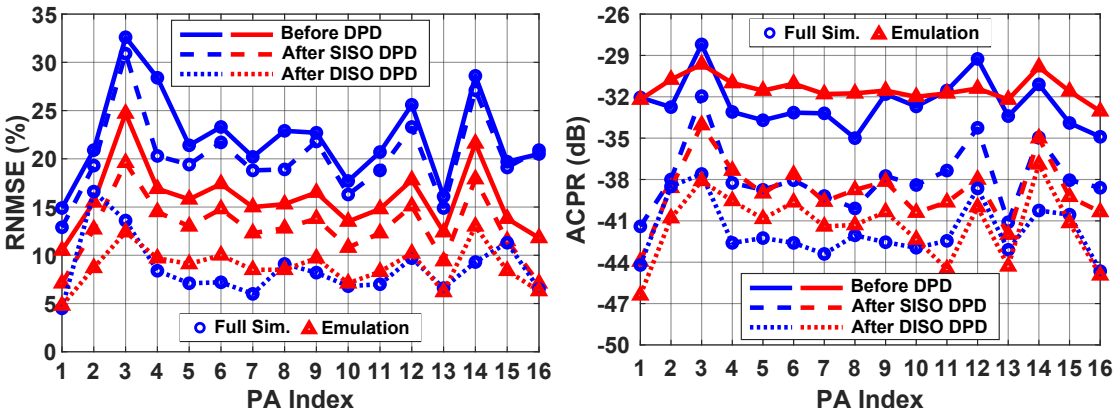
At the PA input, a 10 dB directional coupler is used for transmitter calibration, followed by an isolator to ensure accurate delivery of $v^{+}_{in,i}[n]$ to the PA. At the PA output, a 40 dB directional coupler measures both the output signal $v^{-}_{out,i}[n]$ and the reflected signal $v^{+}_{out,i}[n]$ required for the emulation procedure in Sec. 4.2. An additional isolator after the coupler reduces unintended reflections that could corrupt $v^{+}_{out,i}[n]$.

A critical requirement is magnitude and phase alignment between $v^{+}_{in,i}[n]$ and $v^{+}_{out,i}[n]$. Three calibration steps are performed: receiver calibration using a comb generator, transmitter calibration at the PA input, and magnitude/phase alignment of $v^{-}_{out,i}[n]$ and $v^{+}_{out,i}[n]$ with respect to $v^{-}_{out,i}[n]$.

Measurement Results

For experimental verification, measured SISO and DISO linearization results obtained using the proposed emulation approach using both SISO and DISO PA models are compared with those from the full 4×4 MIMO transmitter front-end. In the emulation configuration, PA #1 is used as the DUT, while PA #4 generates the reflected signal $v^{+}_{out,i}[n]$ at the required power level. Both SISO and DISO DPD employ a complexity-reduced Volterra model [28], with 88 and 104 coefficients, respectively, trained using direct learning over eight iterations.

Fig. 4.12 and 4.14 summarize the results. Before DPD, RNMSE and ACPR of PAs in the full transmitter vary between 14.9% and 32.6% (22.3% average) and between −28.2 dB and −35 dB (−32.5 dB average), respectively. Using the proposed SISO PA model-assisted emulation setup, RNMSE ranges from 10.5% to 24.7% (15.8% average), while ACPR ranges



(a) (b)

Figure 4.12: Comparison of measured PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model.

from −29.7 dB to −33.1 dB (−31.4 dB average). Although absolute RNMSE values differ (due to different signal sources and receiver chains), the PA-to-PA performance variations are reproduced. For ACPR, values for PA #1 are nearly identical in both setups since it is used as the DUT, while variations across the remaining PAs are somewhat compressed.

Next, both PA models are used to assess linearization using SISO and DISO DPD. After SISO DPD, RNMSE decreases to 12.9−30.9% (20.3% average) in the full system and to 7.1−19.6% (12.8% average) for the SISO PA model and 6.8−18.8% (12.2% average) for the DISO PA model in emulation. ACPR improves to −32 to −41.4 dB (−37.9 dB average) in the full system and to −34.1 to −44.0 dB (−38.9 dB average) for the SISO PA model and −34.0 to −43.7 dB (−38.9 dB average) for the DISO PA model in emulation. The ACPR trends across PAs closely match, and results largely overlap. With DISO DPD, RNMSE further decreases to 4.5−16.6% (8.6% average) in the full system and to 4.8−13.0% (8.8% average) for the SISO PA model and 5.2−13.1% (9.0% average) for the DISO PA model in emulation, while ACPR improves to −37.6 to −44.7 dB (−41.8 dB average) in the full system and −36.9 to −46.4 dB (−41.4 dB average) for the SISO PA model and −36.7 to −45.6 dB (−40.9 dB average) for the DISO PA model in emulation.

Representative spectra are shown in Fig. 4.13 and 4.15. PA #3 exhibits similar ACPR values before and after both DPD schemes, yielding overlapping spectra in both setups. After DISO DPD, the upper sideband in the emulated spectra begins to deviate beyond 100 MHz, which is consistent with mutual coupling estimated over a 75 MHz band around

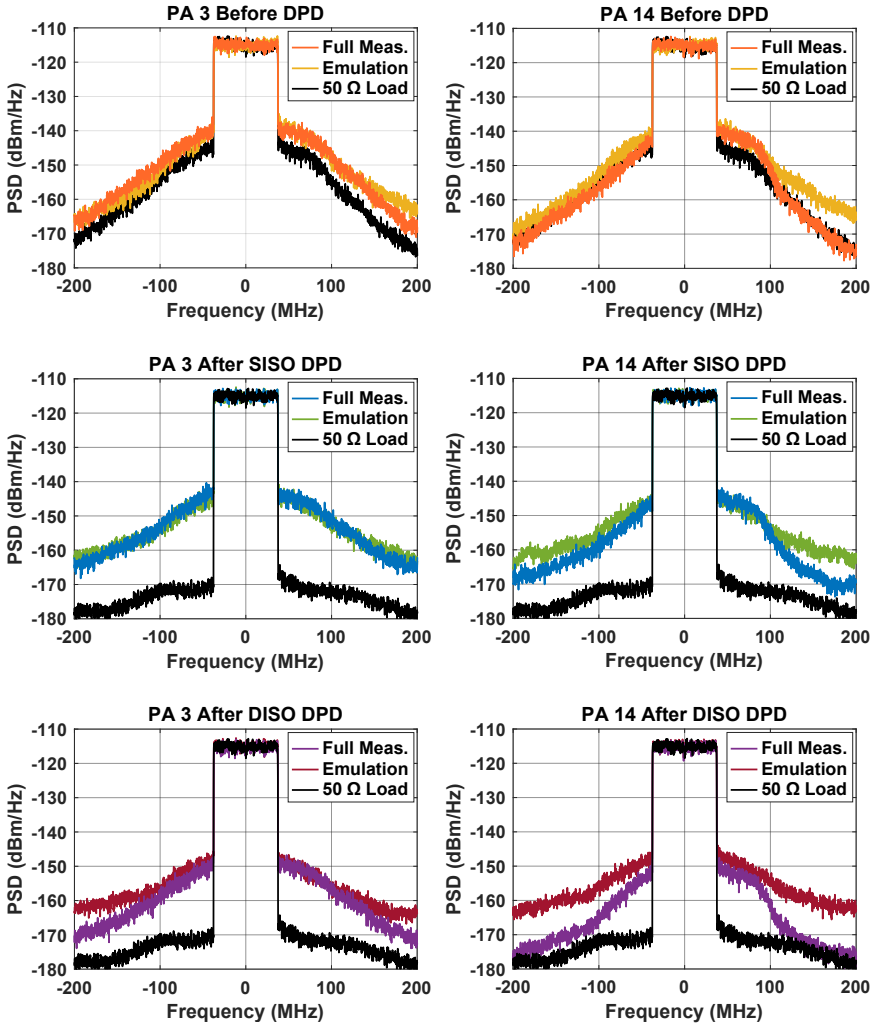
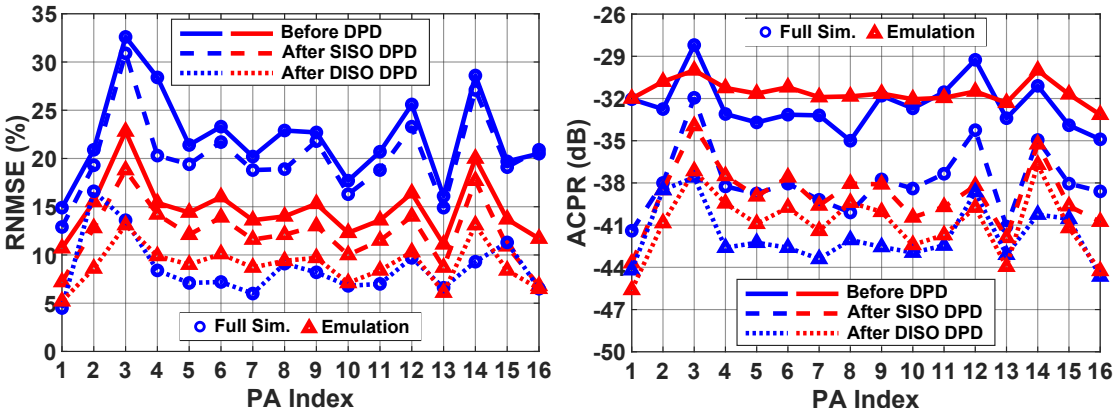


Figure 4.13: Comparison of measured representative PA spectra in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless SISO PA model for PAs with good (PA 3) and poor (PA 14) agreement.

the center frequency. In addition, PA #14 shows overestimated ACPR before DPD, which is also visible in the spectra. Overall, the emulation results closely reproduce the full-system measurement trends before and after DPD using SISO and DISO PA models. While absolute RNMSE differs due to distinct signal-generation/receiver chains, relative PA variations and linearization improvements remain consistent. For ACPR, initial values



(a) (b)

Figure 4.14: Comparison of measured PA performance in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model.

are nearly identical for the DUT (PA #1), and after both DPD schemes the two setups yield nearly indistinguishable results. The SISO PA model can be mainly used for reducing the total number of needed simulations or measurements in forward modeling and linearization assessment.

Discussion

Measurement results demonstrate that the proposed PA model-assisted emulation approach reproduces key PA performance trends in a MIMO transmitter under active load modulation, both with and without DPD. Compared to co-simulation, agreement is degraded primarily due to design, fabrication, and assembly imperfections in the 4×4 hardware prototype. For example, the PAs in the 1×4 module in Fig. 4.9(c) were not initially shielded, and post-fabrication shielding did not fully eliminate coupling, particularly given the high lineup gain (≈50 dB). With all PAs DC biased and only one PA driven, leakage of approximately −25 dB into adjacent PAs was observed. In addition, the modular array topology in Fig. 4.9(d) introduced assembly tolerances that limited the ability to predict the realized antenna-array S-parameters via EM simulation. This was further aggravated by mounting on the mechanical frame. With PAs placed only a few centimeters behind the antenna backplane, radiation from PA outputs likely perturbed the effective array behavior. The resulting change could be partially estimated using the procedure in [2]; however,

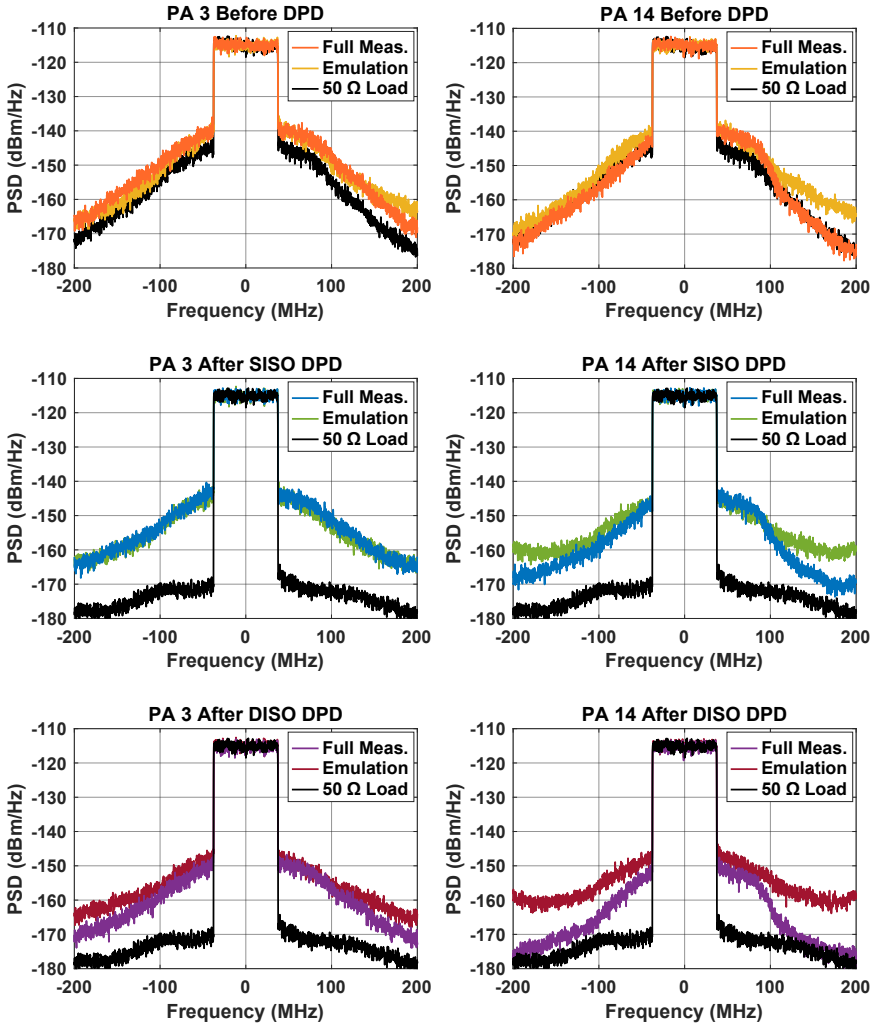


Figure 4.15: Comparison of measured representative PA spectra in the full 4×4 MIMO transmitter front-end and in the proposed PA model-assisted emulation using a memoryless DISO PA model for PAs with good (PA 3) and poor (PA 14) agreement.

that method targets transmission coefficients rather than reflection coefficients. These effects could be mitigated through a more rigorous mechanical/RF co-design and improved shielding and integration in a subsequent prototype, but cost constraints limited further fabrication iterations and, in turn, the achievable agreement.

4.4 Towards Modulated-signal-based PA Design in the Presence of Antenna Mutual Coupling using the Proposed PA Model-based Emulation

This section discusses how the developed PA model-based emulation setup can be utilized for a new PA design methodology for optimizing the system-level performance. As discussed in Sec. 2.1, the conventional PA design performs source-pull and load-pull process to find source and load impedances for the desired output power, efficiency, gain, and linearity using single-tone CW signals. Source-pull and load-pull process scan potential source and load impedances, respectively, in a design space and a transistor is characterized for the output power, efficiency, gain, and linearity. After the process is completed, a contour map is plotted for each parameter at each single-tone frequency. One source and load impedance at each single-tone frequency are selected for the desired performance and input and output matching networks are designed to transform fixed 50 Ω at the input and output to the selected values across the frequency spectrum. In a realistic MIMO transmitter system, modulated signals with a single carrier frequency and a wide bandwidth and 8−10 dB PAPR are used as input signals to the PAs so the signal characteristics are completely different from single-tone CW signals. In addition, due to the presence of antenna mismatches and mutual coupling, the PAs see time-varying output load impedances. Thus, a PA for a transmitter system should be designed with modulated signals in the presence of antenna mutual coupling of a given antenna array behavior.

4.4.1 Design Methodology

The developed model-based emulation procedures in Fig. 4.3 and Fig. 4.4 can be modified slightly to find optimal source and load impedances. In these procedures, the source impedance is set to a fixed 50 Ω and the load impedances are set by antenna mismatches so they need to add a feature to change to any source and load impedances inside Smith chart. At the source, the input modulated signals have to be scaled according to chosen source impedances, while at the load, antenna mismatches are replaced by selected load impedances. Furthermore, the input modulated signals before scaling should be used for computing $\mathrm{v}^{+}_{out,i}[n]$ at the output.

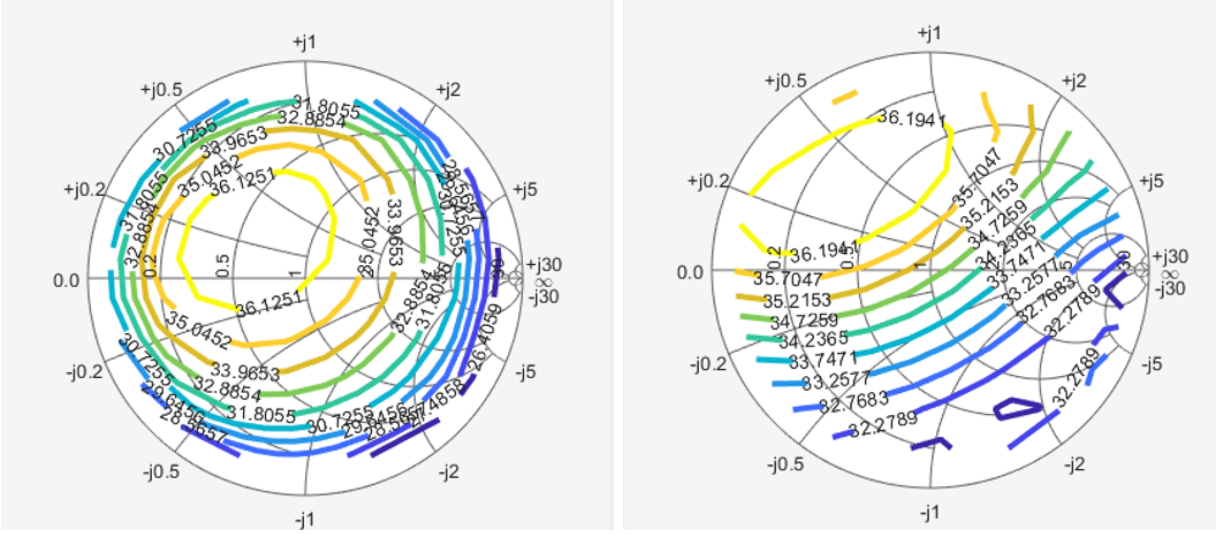


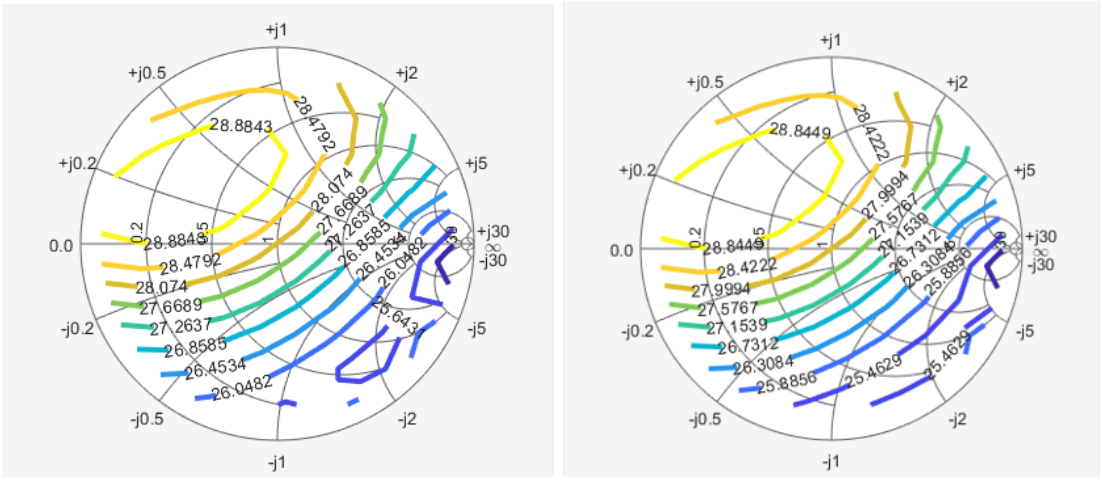
Figure 4.16: **Peak output power load-pull plots using CW signals on the left and modulated signals on the right.** The powers are in the unit of dBm.

Comparing CW signal-based and Modulated signal-based load-pull

Authors in [65] showed that the selection of source and load impedances becomes considerably different when the modulated signals are used instead of the single-tone CW signals. Using the modified model-based emulation procedure in Fig. 4.3, the entire design space inside Smith chart was scanned. The antenna mutual coupling amount remains the same to capture the extent of PA output load impedance variations but the antenna mismatches (S_{ii}) are swept across the entire Smith chart similar to how PA output load impedances are swept for the conventional PA design procedure with CW excitations.

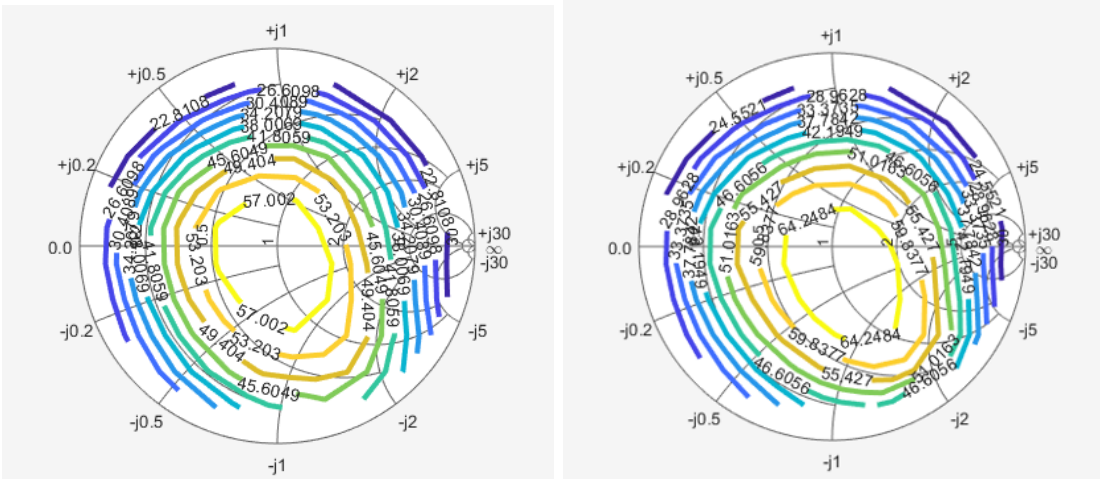
Fig. 4.16 compares the peak output power contour maps of a custom-design Class-AB PA using 100 MHz and 8 dB PAPR modulated signals at 3.5 GHz center frequency and a single-tone CW signal at 3.5 GHz. The general shape of the contours is different and the modulated signal results vary only about 4 dB whereas those of CW signal vary about 8 dB. Modulated signal result at 50 Ω, at which the PA operates, is about 1 dB lower than that of CW signal result as well. Overall, the use of modulated signals result in different design contour maps and require different selections of impedances compared to the conventional PA design with CW excitations.

Modulated-signal-based load-pull with linearization



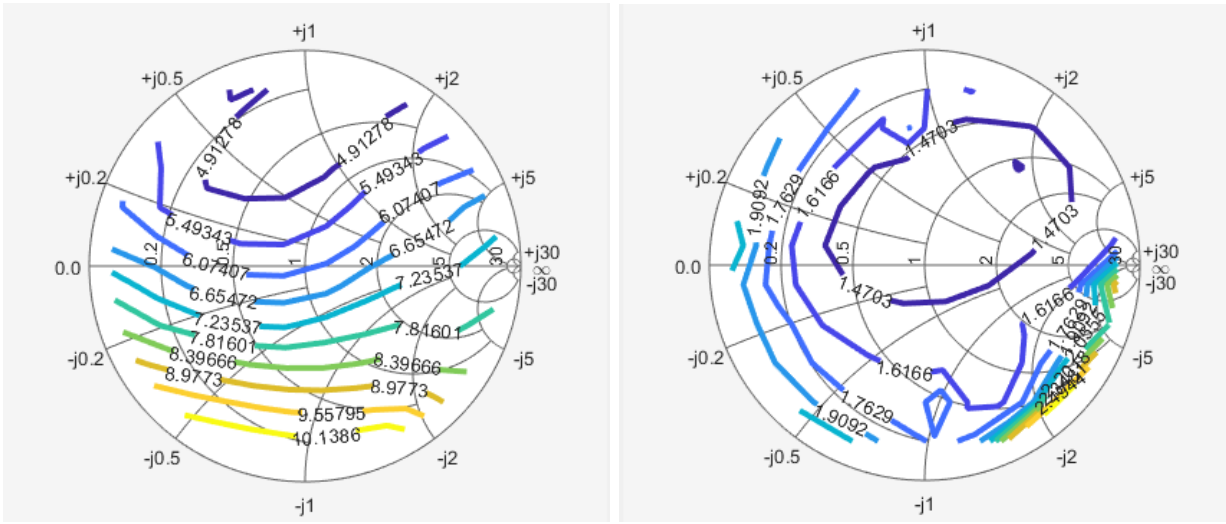
(a) (b)

Figure 4.17: **Average output power contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD.



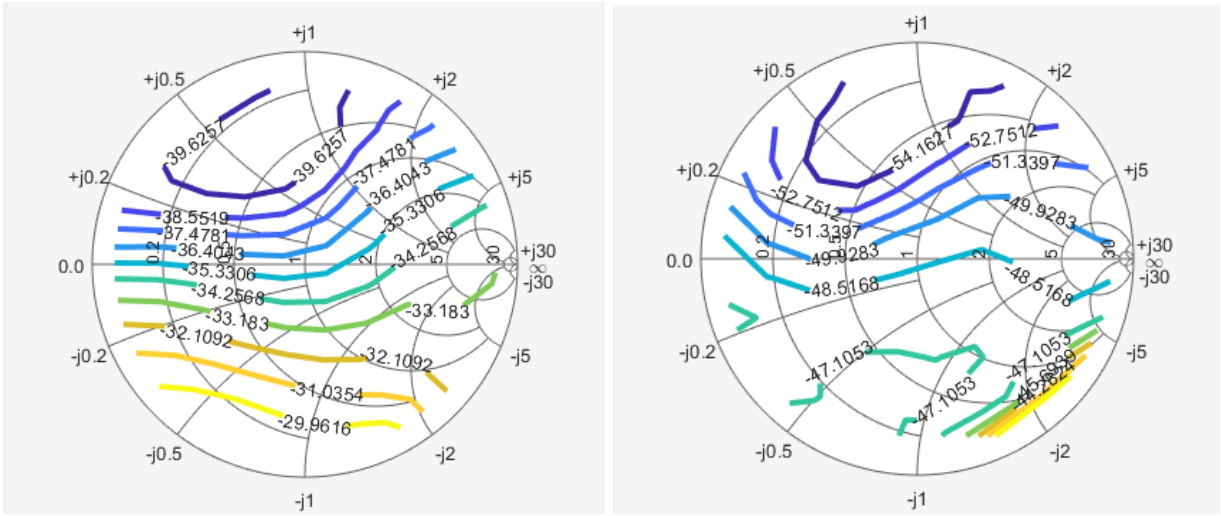
(a) (b)

Figure 4.18: **Peak efficiency contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD.



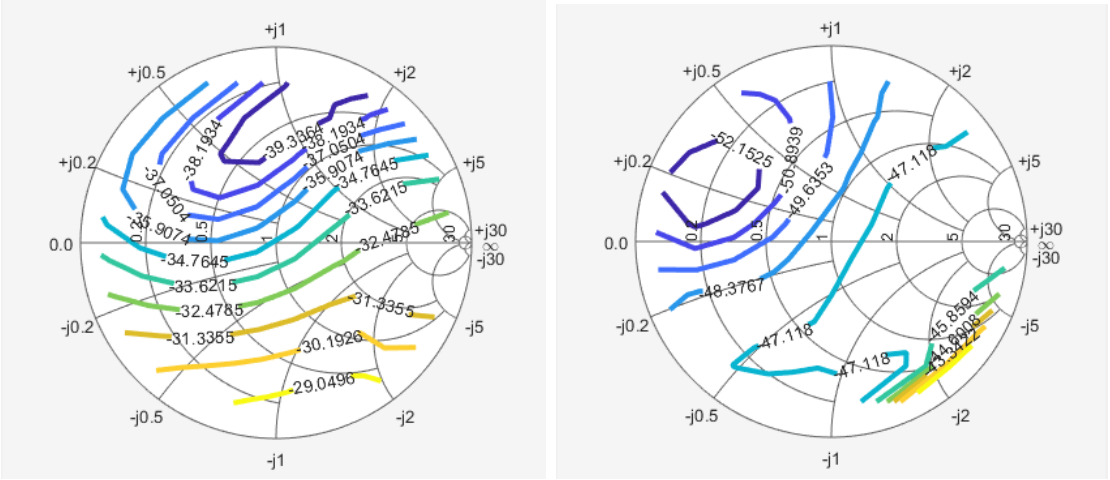
(a) (b)

Figure 4.19: **RNMSE contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD.



(a) (b)

Figure 4.20: **Lower ACPR contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD.



(a) (b)

Figure 4.21: **Upper ACPR contour maps using modulated signals in the presence of antenna mutual coupling.** (a) Before DPD, (b) After DPD.

In the presence of antenna mutual coupling, it is also important to design a PA that meets the radiation specifications. Using the developed model-based emulation procedure with linearization assessment in Fig. 4.4, design contour maps can be plotted before and after DPD to analyze their behaviors. Fig. 4.17-4.21 show contour plots for average output power, peak efficiency, RNMSE, and lower and upper ACPR before and after SISO DPD. The contour map shapes remain the same for the average output power and peak efficiency before and after DPD. For the peak efficiency contours, after DPD contour has an increase in the peak efficiency. In OFDM signals, only $2-3\%$ of the signals reside in the peak power level. Although the average output power remains the same before and after DPD, the peak output power experiences nonlinear compression before DPD. DPD training recovers this peak power level so the maximum efficiency value at the peak power can be recovered as well. For RNMSE and ACPR, before and after DPD contours are quite different. The contour maps show that impedance values that give the most linear PA performance do not necessarily give the best linearization after DPD. Linear PA and DPD linearizability are two different aspects of the PA design that need to be considered for the optimal system-level performance. The load impedance should be chosen to meet all the linearity requirements while maximizing output power and efficiency. The same procedure can be applied to the source side after fixing the load impedance with the selected value.

It is also feasible to verify the influences of different antenna design candidates and precoders because they change PA output impedances together. The overall system design

can be centred around the PA design because it is affected by the building blocks at its both input and output.

4.5 Conclusion

This chapter presented a model-assisted emulation methodology for effective design and testing of power amplifiers in beamforming and massive MIMO transmitters. By synthesizing reflected waveforms that reproduce precoder-dependent active load modulation, the proposed approach enables realistic system-aware PA characterization without requiring a full multi-channel array implementation or computationally intensive co-simulation. The method combines PA behavioral modeling with antenna array S-parameters and a precoding scheme to reconstruct dynamic operating conditions using a single measurement chain. Compared with prior iterative coupling emulation approaches, the proposed formulation significantly reduces the number of required measurements while maintaining high prediction accuracy. Simulation and measurement results on a 4×4 fully digital MIMO transmitter prototype demonstrate close agreement with full-array results and accurate prediction of nonlinear distortion and DPD performance. The proposed emulation framework provides a scalable and cost-effective tool for evaluating PA designs under realistic beamforming conditions, enabling system-level optimization of linearity, efficiency, and output power in next-generation wideband beamforming and massive MIMO transmitters.

Chapter 5

Complexity-Reduced Precoding-Aware Multiple-Input Single-Output Digital Predistortion for Fully Digital Massive MIMO Transmitters

PAs account for a dominant portion of the power consumption in modern radio communication transmitters, making efficiency a central design objective. Achieving high efficiency typically requires operation near saturation, which introduces nonlinear distortion and spectral regrowth that may violate spectral emission requirements. DPD is therefore widely adopted to compensate for PA nonlinearities, enabling efficient operation while maintaining acceptable in-band and out-of-band linearity.

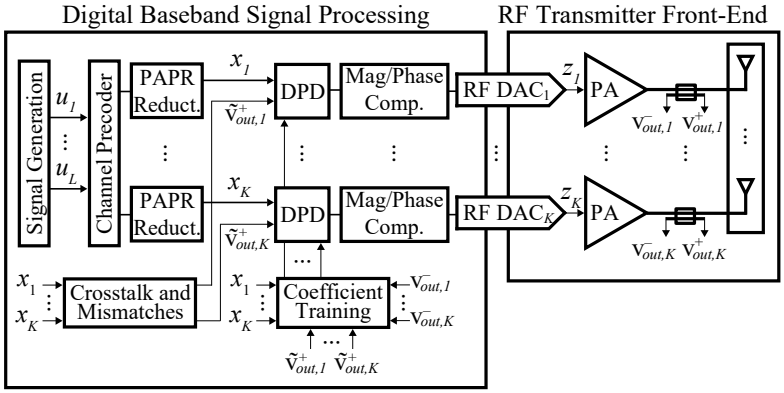
Most DPD techniques [66, 28, 29] have been developed for single-antenna transmitters, where each PA operates under a fixed load impedance. However, in massive MIMO transmitters, closely spaced antenna elements introduce mutual coupling, resulting in active load modulation at the PA outputs. Although isolators can mitigate coupling, they are often avoided due to cost, insertion loss, and integration constraints. Consequently, the effective load impedance seen by each PA varies dynamically with the transmitted signals across the array, challenging conventional per-PA SISO DPD approaches.

To address coupling-induced distortion, several DPD architectures have been proposed. In RF beamforming systems, user-direction SISO DPD techniques [67, 68, 69, 70, 71,

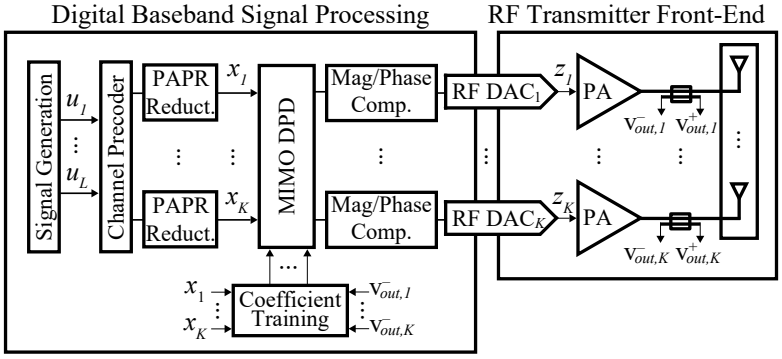
72] linearize the received signal in a target spatial direction. However, in hybrid and fully digital MIMO transmitters, multiple data streams are simultaneously precoded and transmitted through parallel transmit branches, rendering such approaches insufficient. Practical implementations therefore employ one DPD block per PA, but independent SISO DPD remains suboptimal under mutual coupling due to time-varying loading conditions. This limitation has motivated the development of DISO DPD [62] and MIMO DPD [73, 74], which explicitly incorporate coupling effects. DISO DPD is a low-complexity architecture that accounts for antenna mutual coupling; however, it assumes that the reflected signal $\mathrm{v}_{out}^{+}[n]$ from the antenna array is small. In this approach, $\mathrm{v}_{out}^{+}[n]$ is estimated using all input signals in conjunction with a linear filter representing the antenna array S-parameters, and only its linear component is considered in the DPD model. Consequently, the nonlinear behavior at the PA outputs is not fully captured. In contrast, MIMO DPD enables more accurate modeling of individual mutual coupling effects during the identification stage but its computational complexity increases significantly with the number of antennas in the system.

Despite these advances, the impact of precoding is typically not explicitly incorporated into DPD identification. In massive MIMO systems, precoding based on CSI modifies both the correlation structure and the power distribution of the transmitted signals, thereby altering the active load modulation experienced by each PA. Since DPD coefficients are inherently dependent on excitation statistics and loading conditions, such variations introduce a mismatch between training and operational regimes. As reported in [39], applying conventional precoding after DPD can significantly degrade linearization performance. While equal-power precoding mitigates average power imbalance, it does not eliminate correlation-induced effects nor the resulting precoding-dependent loading.

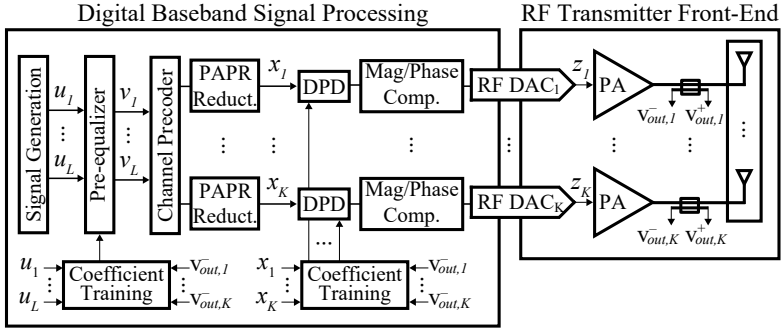
This chapter addresses this limitation by proposing a reduced-dimensional, precoding-aware MISO DPD framework for fully digital massive MIMO transmitters. The key idea is to perform DPD identification directly using realistic precoded signals and to exploit the low-dimensional structure of multi-user MIMO transmission. Specifically, although the transmitter may employ a large number of transmit branches, the transmitted signals reside in a subspace spanned by a smaller number of user streams. This observation enables the representation of coupling-induced nonlinear behavior using a reduced set of user-domain inputs, avoiding the excessive complexity of full MIMO DPD. Furthermore, the linearization performance of existing DPD models is evaluated under both uncorrelated and precoded training conditions, emphasizing the importance of using realistic training signals for accurate DPD identification in practical operation.



(a)



(b)



(c)

Figure 5.1: Block diagram of fully digital mMIMO transmitter with (a) DISO DPD (b) MISO DPD (c) precoding-aware two-tier SISO DPD

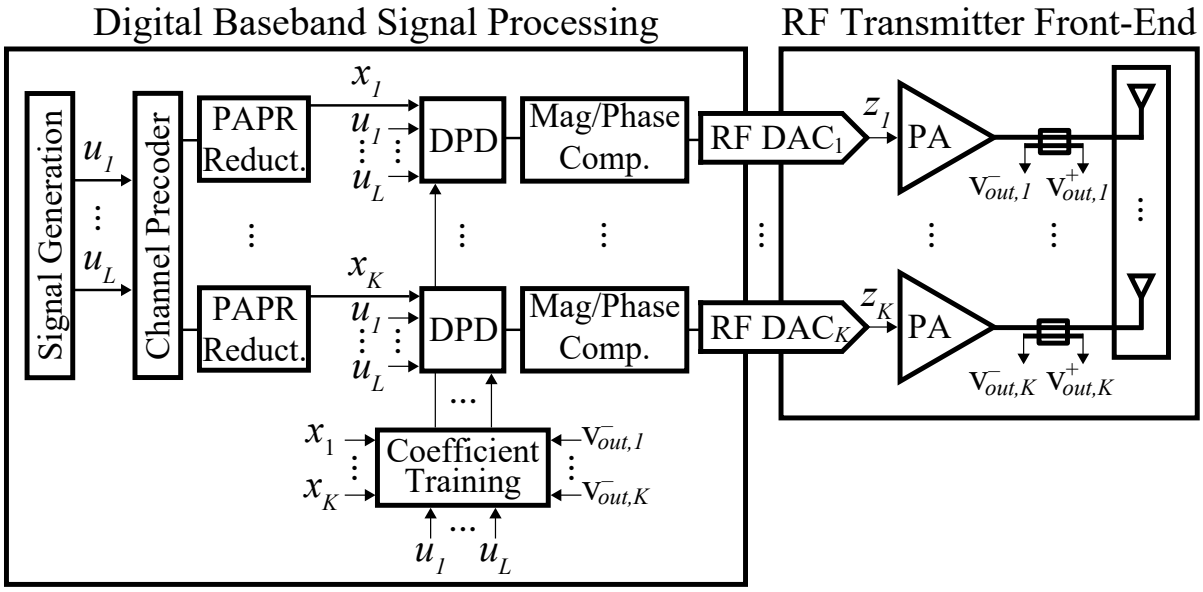


Figure 5.2: Block diagram of fully digital mMIMO transmitter with proposed reduced-dimensional precoding-aware MISO DPD

5.1 Proposed Reduced-Dimensional Precoding-Aware MISO DPD Formulation

Under practical operation with precoded signals, signal correlation and power imbalance modify the effective loading conditions of each PA. Therefore, DPD identification must be performed under representative excitation conditions in order to accurately capture the underlying nonlinear behavior. This section presents a reduced-dimensional precoding-aware multiple-input single-output (R-MISO) DPD formulation that incorporates precoded signals during both training and operation, while maintaining scalable model complexity with number of array antenna elements.

5.1.1 System Model

Consider a fully digital MIMO transmitter with K transmit branches serving L users, where typically $L \ll K$. All signals are represented in the discrete-time complex baseband domain.

Let $\tilde{u}_l[n]$ denote the data stream for user $l \in \{1, \ldots, L\}$, $\tilde{x}_k[n]$ the precoded signal

driving the kth branch, $\tilde{z}_k[n]$ the predistorted signal applied to the kth PA, and $\tilde{y}_k[n]$ the PA output.

The precoded signal at the kth branch is given by

$$\tilde{x}_k[n] = \sum_{l=1}^{L} w_{k,l}\, \tilde{u}_l[n], \tag{5.1}$$

where $w_{k,l}$ denotes the precoding coefficient.

In the presence of antenna coupling, the PA output depends not only on the intended input signal but also on signals coupled from other branches. This behavior can be expressed as

$$\tilde{\mathrm{v}}^{-}_{out,k}[n] = f_k\big(\tilde{\mathbf{x}}_k[n],\, \tilde{\mathbf{v}}^{+}_{out,k}[n]\big)\,, \tag{5.2}$$

where $\tilde{\mathbf{x}}_k[n]$ and $\tilde{\mathbf{v}}_k[n]$ denote memory vectors associated with the direct and coupled inputs, respectively. The coupled signal is given by

$$\tilde{\mathrm{v}}^{+}_{out,k}[n] = \sum_{m=1}^{K} s_{k,m}\, \tilde{x}_m[n], \tag{5.3}$$

with $s_{k,m}$ denoting the coupling coefficient from branch m to branch k.

This formulation highlights that the nonlinear behavior of each PA is governed by both local excitation and coupling-induced interactions.

5.1.2 Existing DPD Models

Fig. 5.1a and 5.1b show block diagrams of existing DISO and MIMO DPD structures.

The DISO DPD structure estimates the reflected signals ($\mathrm{v}^{+}_{out,k}$) coming from the antenna array using all K input signals ($[x_1[n], \cdots, x_K[n]]$) and a linear filter block called antenna CTMM that represents the antenna array. The estimated reflected signals ($\tilde{\mathrm{v}}^{+}_{out,k}$) are utilized as the second input to each DPD block. Based on (5.2), the DISO DPD structure can be defined as

$$\tilde{z}_k[n] = g_k\big(\tilde{\mathbf{x}}_k[n],\, \tilde{\mathbf{v}}^{+}_{out,k}[n]\big)\,, \tag{5.4}$$

where $g_k(\cdot)$ denotes the predistorter function. This approach explicitly accounts for coupling effects but requires real-time computation of $\tilde{v}_k[n]$, which depends on all K branch signals and coupling coefficients.

Alternatively, coupling effects can be captured implicitly by defining a full MIMO predistorter as

$$\tilde{z}_k[n] = g_k^{\text{MIMO}}(\tilde{\mathbf{x}}_1[n], \tilde{\mathbf{x}}_2[n], \ldots, \tilde{\mathbf{x}}_K[n]) \,, \tag{5.5}$$

which eliminates the need for explicit reconstruction of $\tilde{v}_k[n]$. However, the input dimensionality of this formulation scales with K, leading to rapidly increasing complexity for large antenna arrays.

Fig. 5.2 shows the block diagram of two-tier SISO (TT-SISO) DPD architecture [14]. Unlike the DISO and MIMO DPD that are trained using uncorrelated signals, the two-tier SISO DPD performs linearization after applying a precoding scheme. The MIMO pre-equalizer and SISO DPD blocks jointly minimize user error vector magnitude (EVM) and transmitter total radiated power adjacent channel power ratio (TRP–ACPR), where the pre-equalizer focuses on suppressing the in-band distortion of the recevied user signals and the SISO DPD only corrects out-of-band non-linearity.

The SISO predistorter can be defined as

$$\tilde{z}_k[n] = \mathcal{B}\{g_k(\tilde{\mathbf{x}}_k[n])\}, \tag{5.6}$$

where $\mathcal{B}\{\cdot\}$ denotes a linear bandstop filter and the precoded signals in (5.7) are modified to

$$\tilde{x}_k[n] = \sum_{l=1}^{L} w_{k,l}\, \tilde{v}_l[n] \tag{5.7}$$

5.1.3 Proposed Reduced-Dimensional MISO DPD Model

While the two-tier SISO DPD fixed in-band distortion of received user signals to 0.7% EVM, the TRP–ACPR still needs further improvement to satisfy the emission mask requirements. The proposed R-MISO DPD begins from the existing MIMO DPD architecture but a key observation is obtained by substituting (5.7) into (5.3), yielding

$$\tilde{v}_k[n] = \sum_{m=1}^{K} s_{k,m} \sum_{l=1}^{L} w_{m,l}\, \tilde{u}_l[n]. \tag{5.8}$$

Reordering the summations leads to

$$\tilde{v}_k[n] = \sum_{l=1}^{L} \left(\sum_{m=1}^{K} s_{k,m} w_{m,l} \right) \tilde{u}_l[n]. \tag{5.9}$$

Equation (5.9) shows that the coupled signal $\tilde{v}_k[n]$ lies in the subspace spanned by the L user streams. Therefore, the nonlinear behavior observed at each PA can be represented using the local branch signal together with the user-domain signals, rather than all K branch signals.

This observation motivates the reduced-dimensional DPD structure, shown in Fig. 5.2, and it is defined by

$$\tilde{z}_k[n] = g_k^{\text{red}}(\tilde{\mathbf{x}}_k[n],\, \tilde{\mathbf{u}}_1[n],\, \tilde{\mathbf{u}}_2[n],\, \ldots,\, \tilde{\mathbf{u}}_L[n])\,, \tag{5.10}$$

where $g_k^{\text{red}}(\cdot)$ denotes the reduced-dimensional predistorter.

In this formulation, each predistorter capture self-induced nonlinearities through the local branch signal $\tilde{x}_k[n]$ and coupling-induced nonlinearities through the user-domain signals $\tilde{u}_l[n]$. This structure is consistent with the physical mechanism of active load modulation, where reflected waves are linear combinations of user-domain signals.

Each predistorter is implemented as a nonlinear function of its input regressors, expressed as

$$\tilde{z}_k[n] = \tilde{x}_k[n] + \psi_k^T(n)\alpha_k, \tag{5.11}$$

where α_k is the coefficient vector and $\psi_k(n)$ contains nonlinear basis functions.

The regressor vectors are defined as

$$\tilde{\mathbf{x}}_k(n) = [\tilde{x}_k[n], \ldots, \tilde{x}_k[n - M_1]]^T, \tag{5.12}$$

$$\tilde{\mathbf{u}}_l(n) = [\tilde{u}_l[n], \ldots, \tilde{u}_l[n - M_2]]^T. \tag{5.13}$$

The nonlinear basis functions are constructed as

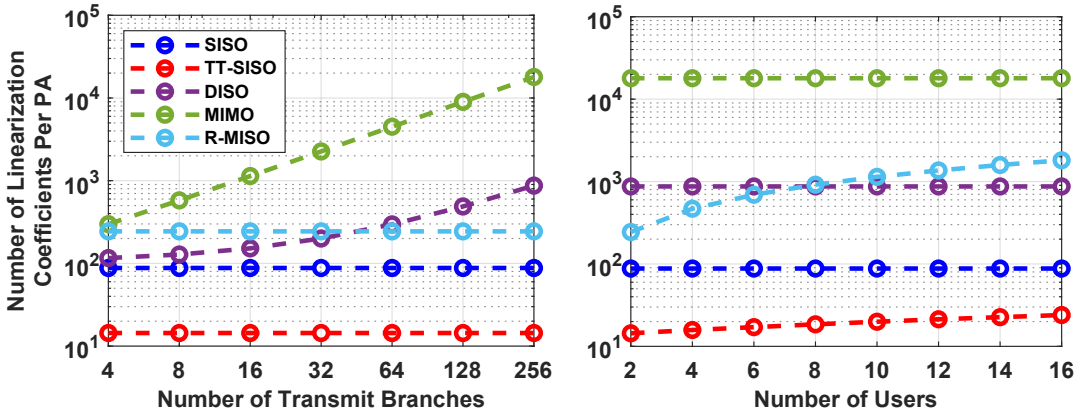
$$\psi_p(\tilde{\mathbf{x}}_k(n), \tilde{\mathbf{u}}(n)), \tag{5.14}$$

where

$$\tilde{\mathbf{u}}(n) = [\tilde{\mathbf{u}}_1[n], \ldots, \tilde{\mathbf{u}}_L[n]]\,. \tag{5.15}$$

The DPD coefficients are obtained by minimizing the squared error between the PA output and the desired linear response:

$$J = \sum_{k=1}^{K} \sum_{m=0}^{M} |\tilde{y}_k[n-m] - \tilde{x}_k[n-m]|^2\,. \tag{5.16}$$



(a) (b)

Figure 5.3: Comparative linearization complexity analysis. Linearization coefficients include both DPD and linear filters per PA for: (a) various number of transmit branches for a 2-user scenario, and (b) various number of users for 256 transmit branches

Stacking regressors over $M+1$ samples yields

$$\boldsymbol{\Psi}_k = [\psi_k(n), \ldots, \psi_k(n-M)]^T . \tag{5.17}$$

The coefficient update follows a least-squares-based iterative procedure:

$$\hat{\alpha}_k^{(j+1)} = \hat{\alpha}_k^{(j)} - \mu \boldsymbol{\Psi}_k^{\dagger} \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{x}}_k\right), \tag{5.18}$$

where $(\cdot)^{\dagger}$ denotes the Moore–Penrose pseudoinverse and μ is a step-size parameter.

5.1.4 Comparative DPD Complexity Analysis

Fig. 5.3 compares the model complexity of various DPD architectures for different transmitter sizes and numbers of users. Fig. 5.3a illustrates the impact of varying the number of transmit branches for 2 users, whereas Fig. 5.3b examines the effect of the number of users for a 256-chain transmitter. The reported number of linearization coefficients per PA includes those associated with both the DPD and the linear filtering blocks required to realize each linearization model.

The complexity of SISO DPD depends solely on its input signal and is therefore independent of both the number of transmit branches and users. In contrast, TT-SISO DPD

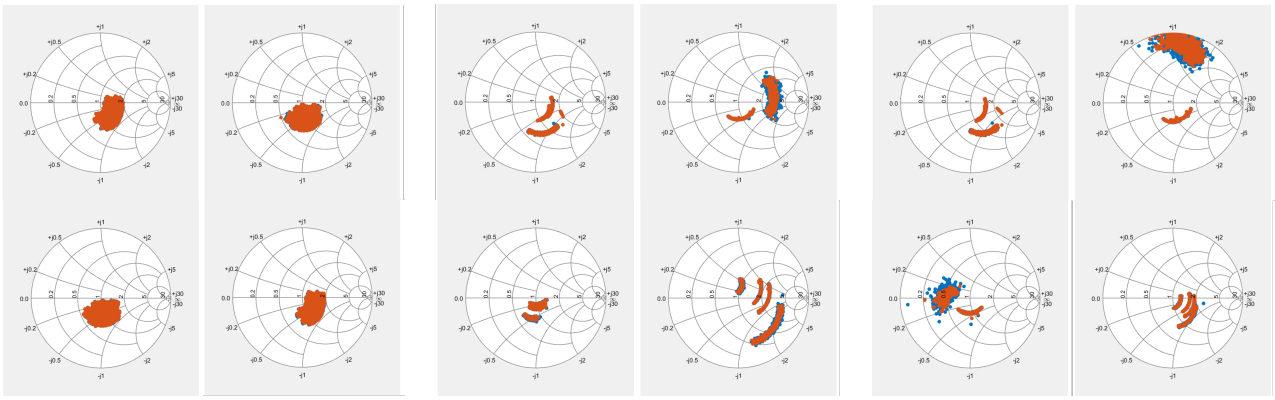
employs a linear pre-equalizer preceding the precoder, resulting in complexity that scales linearly with the number of users, while the SISO DPD block itself remains unchanged. Similarly, the complexity of DISO DPD is unaffected, as it utilizes only the estimated reflected signal ($\tilde{v}^{+}_{\text{out},k}$) as an additional input. However, the associated CTMM block used for reflected signal estimation scales linearly with the number of transmit branches.

In conventional MIMO DPD, each predistorter incorporates regressors from all K transmit branches, resulting in exponential growth in model dimensionality as K increases. In contrast, the proposed formulation exploits the fact that precoded signals reside in an L-dimensional subspace. As a result, the number of input signals required for modeling scales with the number of users rather than the number of transmit branches. Since $L \leq \sqrt{K}$ for good received signal quality at the users [39], this leads to a substantial reduction in model complexity. For instance, for a 256 transmit branches serving 2 and 16 users, the proposed approach reduces the number of linearization coefficients by factors of 73.5 and 10, respectively, compared to conventional MIMO DPD, while still capturing the dominant coupling-induced nonlinear effects.

5.1.5 Proposed R-MISO DPD Training Approach

This subsection proposes R-MISO DPD training approach based on how the statistical properties of excitation signals influence the active load modulation experienced by PAs in fully digital MIMO transmitters. In conventional DPD training in fully digital MIMO transmitters, uncorrelated signals are utilized to train and verify the DPD models. During operation, the precoder is frequently updated while the trained DPD coefficients using uncorrelated signals remain unchanged and this can cause significant linearity variations in PAs [39]. Since precoding reshapes both the correlation structure and the power distribution of these signals, the resulting impedance trajectories are inherently precoding-dependent. The EM/RF/DSP co-simulation framework developed in [75], which jointly captures antenna mutual coupling, nonlinear PA behavior, and digital signal processing, is utilized to analyze how the precoder affects the PA output load impedances.

Fig. 5.4 shows the co-simulated reflection coefficients at each PA output for a 2×2 fully digital MIMO transmitter driven by 4-QAM signals with 100 MHz bandwidth. Each marker represents the load reflection coefficient at the ith PA output ($\Gamma_i(\omega_q)$) evaluated at an individual OFDM subcarrier ω_q and a different color represents a different OFDM symbol such that the distribution of markers characterizes the impedance trajectory experienced by the ith PA over both time and frequency. A 2×2 fully digital MIMO configuration with 4-QAM excitation is considered to clearly illustrate the impact of different precoding schemes on the PA output impedances.



(a) (b) (c)

Figure 5.4: Reflection coefficients at each PA output in a 2×2 fully digital MIMO transmitter under 4-QAM, 100 MHz signals: (a) uncorrelated excitation, (b) conventional zero-forcing (ZF) precoding, (c) equal-power ZF (EP-ZF) precoding.

Under uncorrelated excitation, Fig. 5.4a, the impedance trajectories of all PAs cluster symmetrically around the antenna self-reflection coefficient s_{ii}. The PA output load impedances exhibit statistically indistinguishable distributions, showing that mutual coupling alone does not introduce systematic branch-to-branch asymmetry when the excitation signals are independent.

The picture changes markedly once precoding is applied. With conventional zero-forcing (ZF) precoding, Fig. 5.4b, the precoder assigns unequal power across PAs to satisfy the inter-user interference constraint. The top two PAs deliver maximum average output powers and the bottom left and right PAs deliver -4 dB and -11 dB back-off powers, respectively. This power imbalance manifests as asymmetric impedance trajectories: the high-power PA sees a tightly concentrated impedance cloud with limited modulation depth, while the low-power PA undergoes substantially wider excursions and shifted away from s_{ii}.

Equal-power ZF precoding [39], Fig. 5.4c, restores a uniform power allocation and thereby reduces the asymmetry between branches. However, because the precoded signals remain correlated linear combinations of the user streams, the impedance trajectories still deviate from the self-reflection point and retain a structured, precoding-dependent shape. This result highlights that power balancing alone is insufficient to replicate the load conditions of the uncorrelated case; the residual signal correlation continues to drive systematic load modulation that must be accounted for in PA design and linearization.

The above analysis demonstrates that PA load impedance distributions are strongly de-

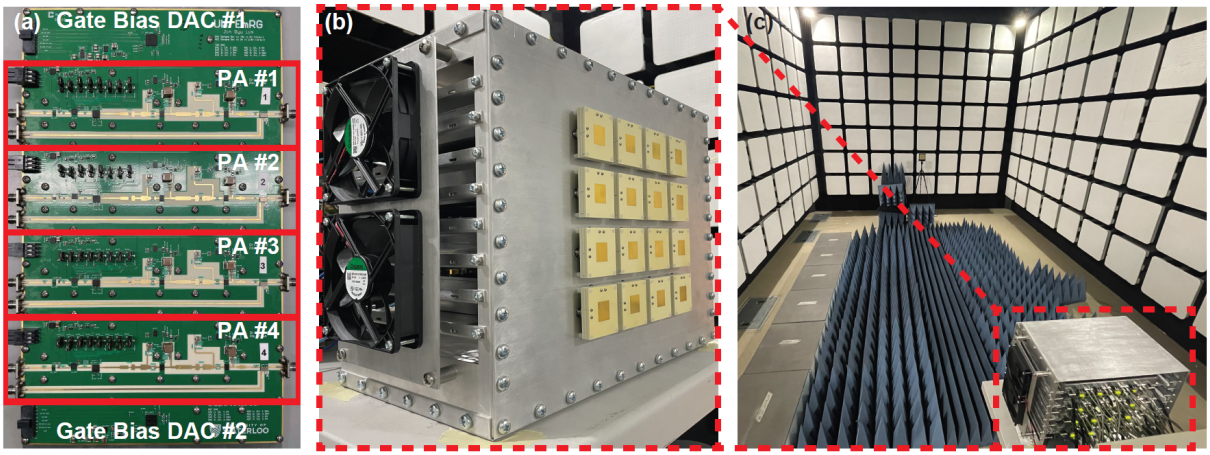


Figure 5.5: (a) 1×4 multistage PA module, (b) Full 4×4 MIMO transmitter front-end inside a 4×5×5 m^3 anechoic chamber, (c) Full 4×4 MIMO transmitter front-end inside a 4×5×5 m^3 anechoic chamber

pendent on excitation statistics. Under uncorrelated excitation, all PAs experience statistically similar loading conditions, whereas precoded excitation introduces branch-dependent loading due to signal correlation and power imbalance. If DPD models are trained on uncorrelated signals but deployed under precoded operation, the resulting mismatch in impedance trajectories degrades linearization performance. Since PA nonlinear behavior depends on loading conditions, accurate DPD modeling requires identification under representative excitation statistics. These observations motivate the proposed R-MISO DPD training approach to explicitly incorporate the statistical properties of precoded signals during model extraction and operation.

5.2 Experimental Validation

5.2.1 Measurement Setup and Signal Configuration

The proposed R-MISO DPD scheme is experimentally validated using a 16-channel (4×4) fully digital transmitter operating at 3.5 GHz. The measurement setup, shown in Fig. 5.5, comprises a multi-stage PA array integrated with an antenna system and is evaluated inside a 4×5×5 m^3 anechoic chamber. The input signals are 256-QAM OFDM waveforms with 75 MHz bandwidth and 8 dB peak-to-average power ratio (PAPR). The 16-channel

transmitter serves L=2 users using maximum-ratio transmission (MRT) precoding. Ideal channel state information (CSI) and the corresponding precoder are assumed, and the received user signals are obtained using the ideal CSI and the measured output signals from the transmitter observation receivers (TORs).

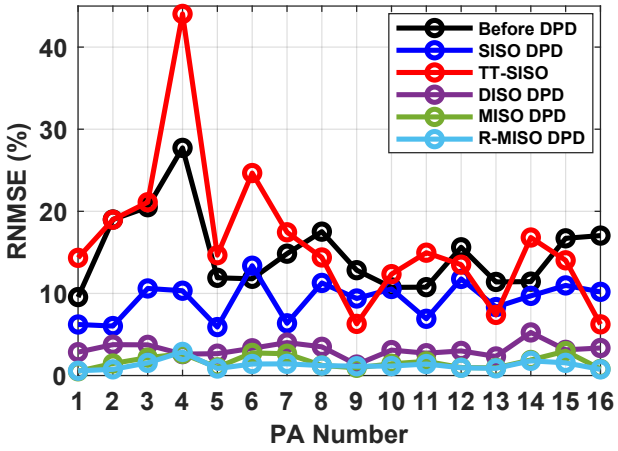
The proposed R-MISO DPD is trained and validated using precoded signals to ensure consistency between training and operational conditions. This enables the predistorter to capture precoding-dependent loading effects observed in practice. For a fair comparison, all benchmark DPD models are also trained and validated using precoded signals. Each PA is modeled using 88, 13, 104, 1138, and 244 coefficients for SISO, TT-SISO, DISO, MIMO, and R-MISO DPDs, respectively. TT-SISO and DISO DPDs include additional linear filtering stages: TT-SISO employs an 11-tap filter per user, whereas DISO requires a 48-tap filter per PA.

In addition to the experimental validation, the impact of DPD training and verification signal selection is investigated across all benchmark models. Conventional DPD development typically employs uncorrelated input signals for both training and verification, whereas in practical operation, uncorrelated signals are used for training and precoded signals are applied during deployment. These scenarios are compared with the proposed approach, which utilizes precoded signals for both training and verification.

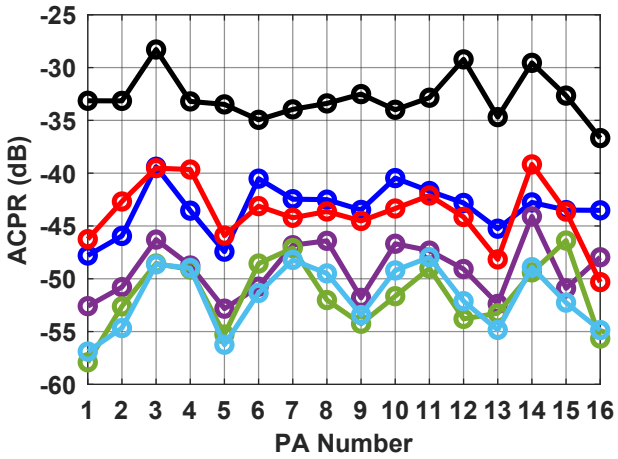
5.2.2 R-MISO DPD Measurement Results

The performance of the proposed R-MISO DPD scheme is evaluated using both branch- and user-domain metrics. Root normalized mean square error (RNMSE) is used to quantify in-band linearization performance, while adjacent-channel power ratio (ACPR) evaluates out-of-band distortion. In addition, total radiated power ACPR (TRP-ACPR) is employed to assess system-level emissions. Together, these metrics provide a comprehensive evaluation of user-domain signal quality and radiated spectral compliance.

The measured performance before and after DPD linearization is presented in Fig. 5.6, while the measured power spectral density (PSD) is shown in Fig. 5.7. The results demonstrate that the proposed R-MISO DPD achieves substantial improvements in both in-band and out-of-band performance across branch- and user-level metrics. Specifically, the TRP-RNMSE across branches is reduced to 0.9%, and the TRP-ACPR improves to −50.9 dB. Moreover, significant suppression of user-domain spectral regrowth is observed, with the ACPR of User 1 improving from −34.0 dB to −56.4 dB and that of User 2 from −34.2 dB to −57.0 dB. The RNMSE at the user locations is reduced from 7.6% to 0.9% and from



(a)

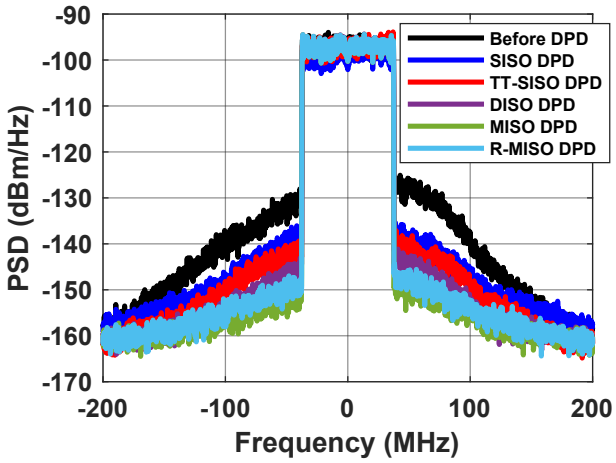


(b)

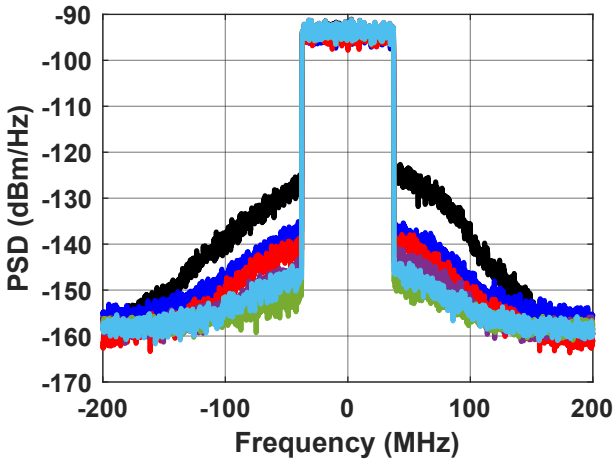
Figure 5.6: Total radiated power measurement results before and after DPD using precoded signals for both DPD training and verification: (a) RNMSE, (b) ACPR.

6.4% to 0.9%, corresponding to <1% error vector magnitude (EVM). These results confirm that the proposed R-MISO DPD effectively mitigates both in-band and out-of-band distortion under realistic precoded excitation, achieving emission-compliant performance while preserving beamforming integrity.

Compared with benchmark DPD models, the proposed R-MISO DPD achieves performance comparable to MIMO DPD while requiring approximately one-fifth the number of



(a)



(b)

Figure 5.7: Measured power spectral density before and after linearization for: (a) TRP, (b) User 1.

coefficients. DISO DPD uses 90 fewer coefficients than the proposed approach but exhibits degraded performance, with TRP-RNMSE and TRP-ACPR worsening by 0.9% and 2.6 dB, respectively. Both SISO and TT-SISO DPD struggle to satisfy the emission mask requirement of −45 dB TRP-ACPR. Although SISO DPD reduces both in-band and out-of-band distortion, the resulting user-domain RNMSE remains in the range of 3–4%. In contrast, TT-SISO DPD separates the linearization tasks, where the DPD block mitigates out-of-band distortion and a linear filter preceding the precoder compensates in-band distortion

Table 5.1: Comparison of DPD linearization results using different combinations of uncorrelated and precoded signals for DPD training and verification. [1]Starting TRP-RNMSE is 15.2% in [14].

Training Signal	Verification Signal	DPD	TRP-RNMSE (%)	TRP-ACPR (dB)	DPD/Linear Filter Coeff. per PA
Uncorrelated	Uncorrelated	Without DPD	17.4	−31.3	N/A
		With SISO	14.4	−38.6	88/0
		With DISO	4.8	−43.0	104/48
		With MIMO	2.0	−47.1	1138/0
Uncorrelated	Precoded	Without DPD	14.0	−31.2	N/A
		With SISO	10.8	−38.0	88/0
		With DISO	4.8	−41.6	104/48
		With MIMO	2.3	−43.6	1138/0
Precoded	Precoded	Without DPD	13.4	−31.9	N/A
		With SISO	5.3	−42.7	88/0
		With TT-SISO [14]	16.3[1]	−42.7	13/11 per user
		With DISO	1.8	−48.3	104/48
		With MIMO	0.9	−50.5	1138/0
		With R-MISO	0.9	−50.9	244/0

at the users, achieving approximately 0.8% RNMSE.

5.2.3 Comparative DPD Training Approach Analysis

Table 5.1 summarizes the performance of different DPD training and verification approaches under uncorrelated and precoded excitation conditions. When uncorrelated signals are used for both training and verification, both TRP-RNMSE and TRP-ACPR degrade significantly for DPD models compared to the case where precoded signals are used for both stages. Specifically, SISO DPD exhibits degradations of 8.9% in TRP-RNMSE and 4.1 dB in TRP-ACPR, while DISO DPD shows degradations of 3.0% and 5.3 dB, respectively. MIMO DPD is least affected, with degradations of 1.1% in TRP-RNMSE and 3.4 dB in TRP-ACPR.

In practical operation, the precoder is updated with changes in CSI, while DPD coefficients trained using uncorrelated signals remain fixed, leading to further performance degradation across all models. In terms of TRP-ACPR, SISO, DISO, and MIMO DPD degrade by 0.6 dB, 1.4 dB, and 3.5 dB, respectively. Since DPD models are trained for specific input power levels and PA loading conditions, performance cannot be maintained when both vary. The results further indicate that DISO and MIMO DPD, which explicitly account for PA loading conditions during training, are more sensitive to precoder updates.

5.2.4 Discussion

The experimental results validate the key premise of this work, namely that DPD identification must account for the statistical properties of precoded signals. In practical operation, PA input signals are inherently correlated due to precoding, whereas many existing DPD models for massive MIMO transmitter systems are developed using uncorrelated training signals.

When realistic precoded signals are employed, SISO and DISO DPD achieve linearization performance comparable to DISO and MIMO DPD trained using uncorrelated signals, respectively. By incorporating precoded excitation during training and exploiting the reduced-dimensional structure of multiuser transmission, the proposed R-MISO DPD achieves effective linearization with practical model complexity. However, the sensitivity of trained DPDs to precoder variations requires further investigation to ensure robust and optimal linearization performance across varying operating conditions.

5.3 Conclusion

This chapter presented a precoded-signal-based user-focused DPD framework for fully digital massive MIMO transmitters and experimentally demonstrated its advantages over conventional uncorrelated-signal-based training. An EM/RF/DPD co-simulation study first showed that multi-user precoding changes both the correlation structure and the per-branch power distribution, which in turn produces precoding-dependent active load modulation at the PA outputs. As a result, DPD models trained with uncorrelated excitations can be significantly mismatched during operation.

To address this issue, a complexity-reduced precoding-aware MISO DPD architecture was proposed in which each branch predistorter operates on its own precoded waveform and a reduced set of user-domain inputs. Compared with full MIMO DPD, the proposed formulation leverages the fact that precoded signals are linear combinations of the user streams and therefore capture key coupling-induced interactions with substantially lower complexity. Measurements on a 4×4 fully digital transmitter verified that the proposed approach improves both user quality and radiated emissions, achieving TRP–ACPR of -50.9 dB and user ACPR better than -56 dB with 224 coefficients per PA.

Chapter 6

Conclusion and Future Work

The EM/RF/DPD co-simulation framework was developed to predict the PA behavior in fully digital MIMO transmitter during the PA design phase prior to fabrication, and it showed great accuracy. The co-simulation framework was then used to design the PA for a 4×4 fully digital MIMO transmitter. The full 4×4 transmitter was designed, fabricated, tested, and analyzed because it is the smallest array size that displays all non-idealities present in a MIMO transmitter. Linearization performed on the 4×4 MIMO transmitter using both uncorrelated and precoded signals showed that precoded signals can greatly simplify the DPD models while achieving better performance.

Model-assisted emulation procedures were developed to characterize PAs in a MIMO transmitter using a single PA by injecting proper signals at both input and output. In addition, linearization assessment can also be added on to evalute how good DPD models perform in the presence of antenna mutual coupling. Compared to previously developed method in [13], the new procedures implemented the use of PA modeling to significantly reduce the number of necessary measurements. The procedures were also combined with the EM/RF/DPD co-simulation framework to perform the same characterization and linearization in simulations. With all the foundational tools developed in this work, a new PA design methodology should be developed to achieve optimal system-level performance.

The influence of the precoder block in the fully digital MIMO transmitter was analyzed. It introduces signal correlation and provides unequal power distirbution to the PAs in the transmitter and when combined with the antenna mutual coupling effect, the PA output load impedances change every time the precoder is updated. The conventional way of training DPD with uncorrelated signals and subsequently updating the precoder can significantly degrade the overall system-level performance. The proposed reduced-dimensional

precoding-aware MISO DPD utilized the precoded signals for both DPD training and verification procedures and the precoded signals being linear combincations of the user streams made the reduction in the DPD architecture possible. The proposed R-MISO DPD linearizes the 16-channel fully digital MIMO transmitter better than the MISO DPD trained and verified using uncorrelated signals with five times fewer number of DPD coefficients.

6.1 Future Work

Regards to the development of a practical testbed, a larger scale transmitter system can be developed with more advanced PA topologies to study their load modulation resilience in a practical setup. In addition, as the system scales, it is difficult to maintain a TOR per PA. Thus, a novel method needs to be developed to reduce the required number of TORs while maintaining a similar level of linearization performance.

For both co-simulation framework and model-assisted emulation procedures, it is assumed that the performance of the PA under test does not vary due to process, temperature, fabrication, and so on. Thus, future simulation and modeling need to account for potential performance variations. The work in Appendix E shows that the PA performance variations can be handled through DSP but the effect of antenna mutual coupling needs to be considered for future work.

The development of precoding-aware DPD models is inevitable for practical operation of fully digital massive MIMO transmitters. Although the proposed linearization method performed well with reduced complexity, frequent precoder update needs to be investigated to study the sensitivity of the proposed method. Thus, future work should investigate the robustness of precoded-signal-based DPD under time-varying CSI and changing precoding matrices, and should quantify the required re-training rate and tracking strategies for dynamic multi-user scenarios.

In order to perform the precoder sensitivity study, a real and controlled channel condition needs to be established first. The development of R-MISO DPD utilized ideal CSI and PA output signals measured at the TORs to estimate the received user signals. Under the effect of realistic CSI, the proposed R-MISO DPD model can be validated further using actual received user signals. With the addition of realistic CSI and receivers, a full system-level MIMO communications performance can be studied.

References

[1] I. Ahmed, H. Khammari, A. Shahid, A. Musa, K. S. Kim, E. De Poorter, and I. Moerman, "A survey on hybrid beamforming techniques in 5g: Architecture and system model perspectives," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3060–3097, 2018.

[2] M. Almoneer, *Hardware-Aware Precoding and Digital Predistortion for Massive MIMO Transmitters*. PhD thesis, 2022.

[3] C. F. Gonçalves, F. M. Barradas, P. M. Cabral, and J. C. Pedro, "Switch-Based Variable Length Stubs Network for PA Load Sensitivity Reduction," *IEEE ACCESS*, vol. 7, pp. 152576–152584, 2019.

[4] M. Gilasgar, A. Barlabé, and L. Pradell, "A 2.4 ghz cmos class-f power amplifier with reconfigurable load-impedance matching," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 31–42, 2019.

[5] C. F. Gonçalves, F. M. Barradas, L. C. Nunes, P. M. Cabral, and J. C. Pedro, "Quasi-load insensitive doherty pa using supply voltage and input excitation adaptation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 1, pp. 779–789, 2022.

[6] H. Lyu and K. Chen, "Balanced-to-Doherty Mode-Reconfigurable Power Amplifier with High Efficiency and Linearity Against Load Mismatch," *IEEE Trans. Microw. Theory Techn.*, vol. 68, pp. 1717–1728, Mar. 2020.

[7] R. Quaglia, J. Powell, K. Chaudhry, and S. Cripps, "Effect of hemt output characteristics on the load mismatch resilience in balanced power amplifiers," in *2022 International Workshop on Integrated Nonlinear Microwave and Millimetre-Wave Circuits (INMMiC)*, pp. 1–3, 2022.

[8] J. G. Lim, H. Yu, E. Traore, M. Almoneer, J. Xia, and S. Boumaiza, "Modulated-Signal-Based EM/RF/DSP Co-Simulation Framework for Predictive Analysis of Fully Digital MIMO Transmitters," in *2021 51st Eur. Microw. Conf. (EuMC)*, pp. 454–457, 2022.

[9] "Circuit Envelope Simulation." http://edadownload.software.keysight.com/eedl/ads/2011/pdf/cktsimenv.pdf. Accessed: 2022-11-07.

[10] H. Yu, E. Traore, M. Almoneer, J. G. Lim, J. Xia, and S. Boumaiza, "Experimental Study of the Effects of Antenna Crosstalk on the Linearity and Efficiency of 5G Sub-6 GHz Wideband 2 × 2 Transmitter Arrays," in *2021 IEEE MTT-S Int'l Microw. Symp.*, pp. 869–872, 2021.

[11] H. Yu, "Efficiency and Linearity Improvement of Sub-6 GHz Digital Beamforming Transmitters," Master's thesis, 2021.

[12] "The basics of 5G's modulation, OFDM." https://www.eeworldonline.com/the-basics-of-5gs-modulation-ofdm. Accessed: 2022-12-12.

[13] D. Nopchinda and K. Buisman, "Measurement technique to emulate signal coupling between power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 4, pp. 2034–2046, 2018.

[14] J. G. Lim, A. Ben Ayed, and S. Boumaiza, "A Two-Tier Low-Complexity Linearization Architecture for Fully Digital mMIMO Transmitters," *IEEE Microw. Wirel. Technol. Lett.*, pp. 1–4, 2026.

[15] S. Cripps, *RF Power Amplifiers for Wireless Communications*. Artech House microwave library, Artech House, 1999.

[16] H. Chireix, "High power outphasing modulation," *Proceedings of the Institute of Radio Engineers*, vol. 23, no. 11, pp. 1370–1392, 1935.

[17] L. R. Kahn, "Single-sideband transmission by envelope elimination and restoration," *Proceedings of the IRE*, vol. 40, no. 7, pp. 803–806, 1952.

[18] W. Doherty, "A new high efficiency power amplifier for modulated waves," *Proceedings of the Institute of Radio Engineers*, vol. 24, no. 9, pp. 1163–1182, 1936.

[19] D. J. Shepphard, J. Powell, and S. C. Cripps, "An efficient broadband reconfigurable power amplifier using active load modulation," *IEEE Microwave and Wireless Components Letters*, vol. 26, no. 6, pp. 443–445, 2016.

[20] H. Golestaneh, F. A. Malekzadeh, and S. Boumaiza, "An extended-bandwidth three-way doherty power amplifier," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 9, pp. 3318–3328, 2013.

[21] Y. Cao and K. Chen, "Pseudo-doherty load-modulated balanced amplifier with wide bandwidth and extended power back-off range," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 7, pp. 3172–3183, 2020.

[22] J. Pang, Y. Li, M. Li, Y. Zhang, X. Y. Zhou, Z. Dai, and A. Zhu, "Analysis and design of highly efficient wideband rf-input sequential load modulated balanced power amplifier," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 5, pp. 1741–1753, 2020.

[23] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. Krieger Pub., 2006.

[24] E. G. Lima, T. R. Cunha, H. M. Teixeira, M. Pirola, and J. C. Pedro, "Base-band derived volterra series for power amplifier modeling," in *2009 IEEE MTT-S International Microwave Symposium Digest*, pp. 1361–1364, 2009.

[25] J. Kim, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, pp. 1417–1418(1), November 2001.

[26] R. N. Braithwaite, "Memory correction of a doherty power amplifier with a wcdma input using digital predistortion," in *2006 IEEE MTT-S International Microwave Symposium Digest*, pp. 1526–1529, 2006.

[27] L. Ding, F. Mujica, and Z. Yang, "Digital predistortion using direct learning with reduced bandwidth feedback," in *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, pp. 1–3, 2013.

[28] F. Mkadem, M. C. Fares, S. Boumaiza, and J. Wood, "Complexity-reduced Volterra Series Model for Power Amplifier Digital Predistortion," *Analog Integr. Circ. Sig. Process.*, vol. 79, pp. 331–343, Feb. 2014.

[29] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based volterra behavioral modeling of rf power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 12, pp. 4323–4332, 2006.

[30] B. S. Mysore, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Precoding for satellite communications: Why, how and what next?," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2453–2457, 2021.

[31] T. Lo, "Maximum ratio transmission," *IEEE Transactions on Communications*, vol. 47, no. 10, pp. 1458–1461, 1999.

[32] M. Joham, W. Utschick, and J. Nossek, "Linear transmit processing in mimo communications systems," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2700–2712, 2005.

[33] T. A. Levanen, J. Pirskanen, T. Koskela, J. Talvitie, and M. Valkama, "Radio interface evolution towards 5g and enhanced local area communications," *IEEE Access*, vol. 2, pp. 1005–1029, 2014.

[34] C. Fager, X. Bland, K. Hausmair, J. Chani Cahuana, and T. Eriksson, "Prediction of smart antenna transmitter characteristics using a new behavioral modeling approach," in *2014 IEEE MTT-S International Microwave Symposium (IMS2014)*, pp. 1–4, 2014.

[35] X. Chen, S. Zhang, and Q. Li, "A Review of Mutual Coupling in MIMO Systems," *IEEE Access*, vol. 6, pp. 24706–24719, 2018.

[36] L. Savy and M. Lesturgie, "Coupling Effects in MIMO Phased Array," in *2016 IEEE Radar Conference*, (Philadelphia, PA, USA), pp. 1–6, 2016.

[37] K. Hausmair, P. N. Landin, U. Gustavsson, C. Fager, and T. Eriksson, "Digital Predistortion for Multi-Antenna Transmitters Affected by Antenna Crosstalk," *IEEE Trans. Microw. Theory Techn.*, vol. 66, pp. 1524–1535, Mar. 2018.

[38] M. Almoneer, P. Mitran, and S. Boumaiza, "Load and power indexed predistortion for improved robusteness to mimo channel conditions," *IEEE Microwave and Wireless Technology Letters*, pp. 1–4, 2022.

[39] M. Almoneer, J. G. Lim, H. Yu, P. Mitran, and S. Boumaiza, "Massive MIMO Precoding Methods That Minimize the Variation in Average Power and Active Impedance With Channel Conditions," *IEEE Trans. Microw. Theory Tech.*, vol. 70, no. 8, pp. 4002–4016, 2022.

[40] M. Li, L. Jiang, and K. L. Yeung, "A general and systematic method to design neutralization lines for isolation enhancement in mimo antenna arrays," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6242–6253, 2020.

[41] R. Liu, X. An, H. Zheng, M. Wang, Z. Gao, and E. Li, "Neutralization line decoupling tri-band multiple-input multiple-output antenna design," *IEEE Access*, vol. 8, pp. 27018–27026, 2020.

[42] C. Volmer, J. Weber, R. Stephan, K. Blau, and M. A. Hein, "An eigen-analysis of compact antenna arrays and its application to port decoupling," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 2, pp. 360–370, 2008.

[43] M. Jiang, Z. N. Chen, Y. Zhang, W. Hong, and X. Xuan, "Metamaterial-based thin planar lens antenna for spatial beamforming and multibeam massive mimo," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 2, pp. 464–472, 2017.

[44] K.-L. Wu, C. Wei, X. Mei, and Z.-Y. Zhang, "Array-antenna decoupling surface," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6728–6738, 2017.

[45] D. Manteuffel and R. Martens, "Compact multimode multielement antenna for indoor uwb massive mimo," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 7, pp. 2689–2697, 2016.

[46] D. Ji, J. Jeon, and J. Kim, "A novel load mismatch detection and correction technique for 3g/4g load insensitive power amplifier application," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 5, pp. 1530–1543, 2015.

[47] Y. Yoon, H. Kim, H. Kim, K.-S. Lee, C.-H. Lee, and J. S. Kenney, "A 2.4-ghz cmos power amplifier with an integrated antenna impedance mismatch correction system," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 3, pp. 608–621, 2014.

[48] J. Guo, Y. Cao, and K. Chen, "1-D reconfigurable pseudo-doherty load modulated balanced amplifier with intrinsic vswr resilience across wide bandwidth," *IEEE Transactions on Microwave Theory and Techniques*, pp. 1–14, 2023.

[49] N. B. Vangipurapu, H. Lyu, Y. Cao, and K. Chen, "Intrinsically mode-reconfigurable load-modulation power amplifier leveraging transistor's analog-digital duality," in *2022 IEEE/MTT-S International Microwave Symposium - IMS 2022*, pp. 418–421, 2022.

[50] H. Lyu, Y. Cao, and K. Chen, "Linearity-enhanced quasi-balanced doherty power amplifier with mismatch resilience through series/parallel reconfiguration for massive mimo," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 4, pp. 2319–2335, 2021.

[51] S. Hu, S. Kousai, and H. Wang, "Antenna impedance variation compensation by exploiting a digital doherty power amplifier architecture," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 2, pp. 580–597, 2015.

[52] D. J. Collins, R. Quaglia, J. R. Powell, and S. C. Cripps, "The orthogonal lmba: A novel rfpa architecture with broadband reconfigurability," *IEEE Microwave and Wireless Components Letters*, vol. 30, no. 9, pp. 888–891, 2020.

[53] M. Pashaeifar, L. C. N. de Vreede, and M. S. Alavi, "A millimeter-wave mutual-coupling-resilient double-quadrature transmitter for 5g applications," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 12, pp. 3784–3798, 2021.

[54] N. S. Mannem, M.-Y. Huang, T.-Y. Huang, and H. Wang, "A reconfigurable hybrid series/parallel doherty power amplifier with antenna vswr resilient performance for mimo arrays," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 12, pp. 3335–3348, 2020.

[55] "ADS Simulator Input Syntax." https://edadocs.software.keysight.com/display/ads2009/ADS+Simulator+Input+Syntax. Accessed: 2022-11-07.

[56] "ADSPtolemy Simulation." http://edadownload.software.keysight.com/eedl/ads/2011_01/pdf/ptolemy.pdf. Accessed: 2022-11-07.

[57] "CGHV1F006S." https://assets.wolfspeed.com/uploads/2020/12/CGHV1F006S.pdf. Accessed: 2022-12-12.

[58] "QPA9842." https://www.qorvo.com/products/p/QPA9842. Accessed: 2022-12-12.

[59] "QPA9127." https://www.qorvo.com/products/p/QPA9127. Accessed: 2022-12-12.

[60] "HMC1122." https://www.analog.com/en/products/hmc1122.html. Accessed: 2022-12-12.

[61] "AMC7834." https://www.ti.com/lit/ds/symlink/amc7834.pdf?ts=1670854362148&ref_url=https%253A%252F%252Fwww.google.ca%252F. Accessed: 2022-12-12.

[62] K. Hausmair, S. Gustafsson, C. Sánchez-Pérez, P. N. Landin, U. Gustavsson, T. Eriksson, and C. Fager, "Prediction of nonlinear distortion in wideband active antenna arrays," *IEEE Trans. Microw. Theory Tech.*, vol. 65, no. 11, pp. 4550–4563, 2017.

[63] F. M. Barradas, P. M. Tomé, J. M. Gomes, T. R. Cunha, P. M. Cabral, and J. C. Pedro, "Power, linearity, and efficiency prediction for mimo arrays with antenna coupling," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 12, pp. 5284–5297, 2017.

[64] J. Lu, Z. Kuai, X. Zhu, and N. Zhang, "A High-Isolation Dual-Polarization Microstrip Patch Antenna With Quasi-Cross-Shaped Coupling Slot," *IEEE Trans. Antennas Propag.*, vol. 59, pp. 2713–2717, July 2011.

[65] X. Konstantinou, J. D. Albrecht, and J. Papapolymerou, "Using active load-pull with modulated signals to optimize power and linearity," in *2020 94th ARFTG Microwave Measurement Symposium (ARFTG)*, pp. 1–4, 2020.

[66] J. Wood, "System-level design considerations for digital pre-distortion of wireless base station transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1880–1890, 2017.

[67] W. Chen, G. Lv, X. Liu, and Z. Feng, "Energy-efficient Doherty power amplifier MMIC and beamforming-oriented digital predistortion for 5G massive MIMO application," in *2017 IEEE Asia Pacific Microw. Conf. (APMC)*, pp. 391–394, 2017.

[68] N. Tervo, J. Aikio, T. Tuovinen, T. Rahkonen, and A. Parssinen, "Digital predistortion of amplitude varying phased array utilising over-the-air combining," in *2017 IEEE MTT-S Int'l Microw. Symp. (IMS)*, pp. 1165–1168, 2017.

[69] E. Ng, Y. Beltagy, P. Mitran, and S. Boumaiza, "Single-input single-output digital predistortion of power amplifier arrays in millimeter wave rf beamforming transmitters," in *2018 IEEE/MTT-S Int'l Microw. Symp. (IMS)*, pp. 481–484, 2018.

[70] X. Liu, Q. Zhang, W. Chen, H. Feng, L. Chen, F. M. Ghannouchi, and Z. Feng, "Beam-oriented digital predistortion for 5G massive MIMO hybrid beamforming transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 7, pp. 3419–3432, 2018.

[71] M. Abdelaziz, L. Anttila, A. Brihuega, F. Tufvesson, and M. Valkama, "Digital predistortion for hybrid mimo transmitters," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 3, pp. 445–454, 2018.

[72] E. Ng, Y. Beltagy, G. Scarlato, A. Ben Ayed, P. Mitran, and S. Boumaiza, "Digital predistortion of millimeter-wave rf beamforming arrays using low number of steering angle-dependent coefficient sets," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 11, pp. 4479–4492, 2019.

[73] S. A. Bassam, M. Helaoui, and F. M. Ghannouchi, "Crossover digital predistorter for the compensation of crosstalk and nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 5, pp. 1119–1128, 2009.

[74] P. Jaraut, M. Rawat, and F. M. Ghannouchi, "Composite neural network digital predistortion model for joint mitigation of crosstalk, I/Q imbalance, nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5011–5020, 2018.

[75] J. G. Lim, H. Yu, E. Traore, M. Almoneer, J. Xia, and S. Boumaiza, "Modulated-Signal-Based EM/RF/DSP Co-Simulation Framework for Predictive Analysis of Fully Digital MIMO Transmitters," in *2021 51st Eur. Microw. Conf. (EuMC)*, pp. 454–457, 2022.

APPENDICES

Appendix A

Analysis of Linearizability of the Balanced Power Amplifier in a MIMO Transmitter

Balanced PA architecture is analyzed. Balanced PA is inherently linear and load-resilient but with a limited range of tolerable output impedance variation [7]. Theoretical analysis on balanced PA is extended by incorporating matching networks as ABCD matrices. After analyzing the balanced PA, a research plan is proposed.

A.1 Application of the Co-Simulation Framework to the Analysis of Balanced PA-Based 2×2 and 4×4 MIMO Transmitter

Antenna mutual coupling and load modulation complicate system-level massive MIMO transmitter design. The complexity of DPD modeling and antenna design can be relaxed if PAs can be load-insensitive. Balanced PA is a potential candidate for use in massive MIMO transmitters due to its inherent linearity and robustness to load variation. However, it has a limitation on how much variation it can tolerate [7]. This section first discusses the background theory of balanced PA and then expands on the theory to increase the allowable load variation range and to investigate if it can improve system-level behaviour using the developed co-simulation framework in Sec. 3.

A.1.1 Background Theory

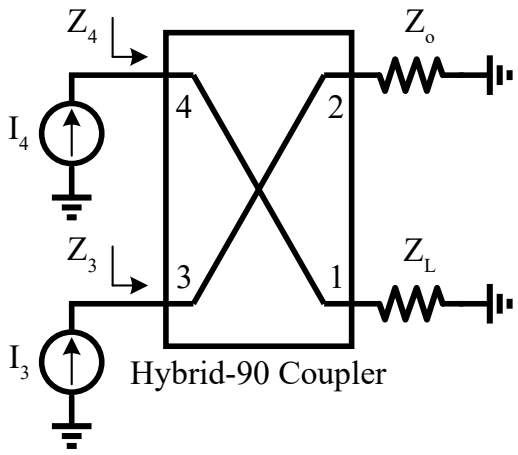


Figure A.1: **Balanced PA Schematic.**

Fig. A.1 shows a balanced PA schematic used by the authors in [7]. The current sources represent transistors and the isolation port (at 2) is matched to Z_o. The output port (at 1) is subject to load modulation with any value of Z_L. Fig. A.2 shows gain curves of a balanced PA at various load impedances. When different output impedance values are presented, the gain of each top and bottom PA varies from 9−15 dB. However, they change in the opposite direction, *balancing out* the fluctuations. Thus, the total output gain variation is only 0.5−1 dB. This balancing feature creates inherent linearity and robustness against load modulation. However, the outstanding question is how large of a variation the balanced PA can tolerate? This can be carried out by analyzing Z_3 and Z_4 in fig. A.1. With the currents at each port flowing into the hybrid-90 coupler, the coupler has the following impedance matrix characteristics:

$$\begin{bmatrix} \mathbf{V_1} \\ \mathbf{V_2} \\ \mathbf{V_3} \\ \mathbf{V_4} \end{bmatrix} = Z_o \begin{bmatrix} 0 & -j & 0 & -j\sqrt{2} \\ -j & 0 & -j\sqrt{2} & 0 \\ 0 & -j\sqrt{2} & 0 & -j \\ -j\sqrt{2} & 0 & -j & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I_1} \\ \mathbf{I_2} \\ \mathbf{I_3} \\ \mathbf{I_4} \end{bmatrix} \tag{A.1}$$

Voltages and currents at each port in terms of I_3, I_4, and Z_L can be calculated to find how load modulation affects the outputs of each transistor.

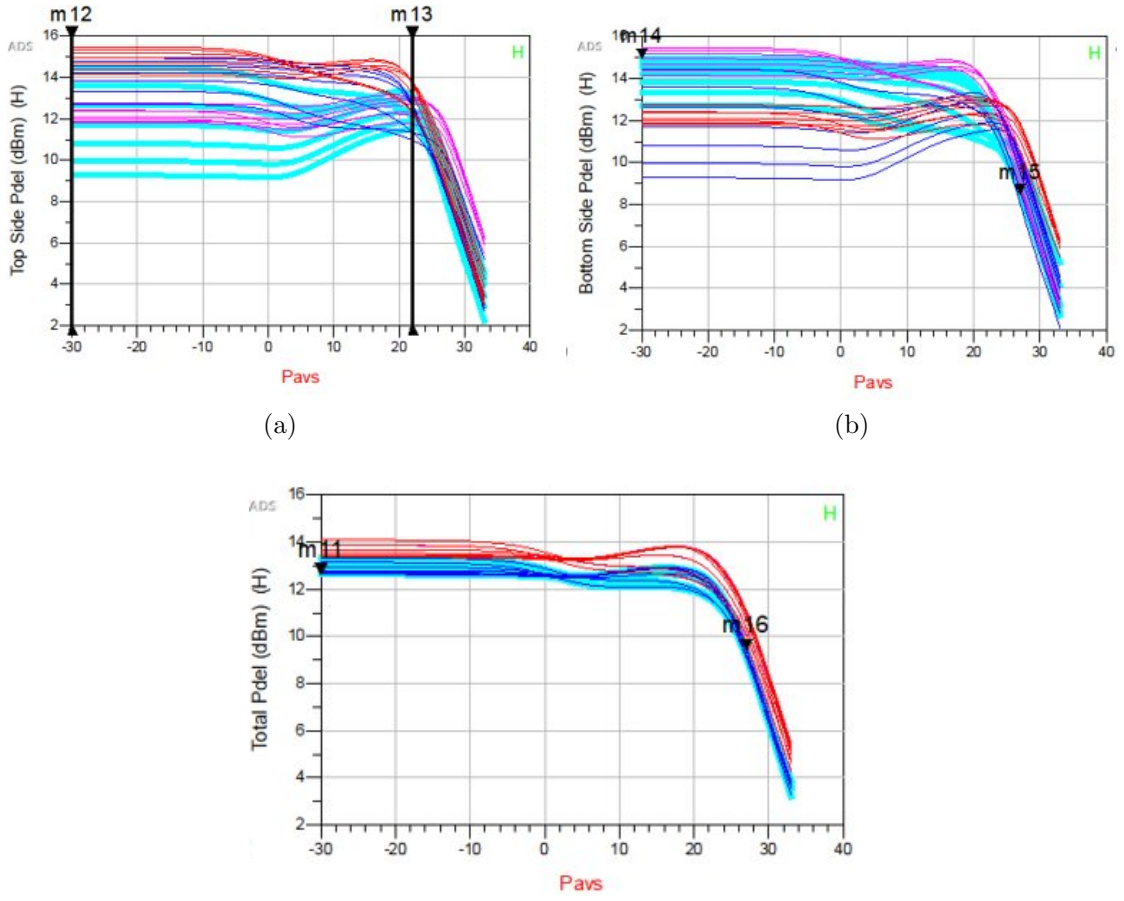


Figure A.2: **Balanced PA Gain Curves.** (a) Top PA, (b) Bottom PA, (c) Total Ouptut. Different colours represent different output load impedances.

$$V_1 = \sqrt{2}(Z_o||Z_L)(I_3 - jI_4), \quad I_1 = -\sqrt{2}\left(\frac{Z_o}{Z_o + Z_L}\right)(I_3 - jI_4) \tag{A.2}$$

$$V_1^- = \frac{V_1 + Z_o I_1}{\sqrt{Z_o}} = \sqrt{2Z_o}(I_3 - jI_4) \tag{A.3}$$

$$V_2 = \frac{Z_o}{\sqrt{2}}[-j(1+\Gamma_L)I_3 + (1-\Gamma_L)I_4], \quad I_2 = \frac{-1}{\sqrt{2}}[-j(1+\Gamma_L)I_3 + (1-\Gamma_L)I_4] \tag{A.4}$$

$$V_3 = Z_o[(1+\Gamma_L)I_3 - j\Gamma_L I_4], \quad I_3 = -jI \tag{A.5}$$

$$V_4 = Z_o[(1-\Gamma_L)I_4 - j\Gamma_L I_3], \quad I_4 = I \tag{A.6}$$

where Γ_L is the reflection coefficient of Z_L, and I_3 and I_4 are 90° out of phase due to input hybrid-90 coupler for 3-dB input power splitting. An interesting result is that the coupled signal (V_1^-), which is used for training DPD coefficients, is independent of the PA output impedance. I_3 and I_4 are proportional to the input signal, so the linearizability of the balanced PA using SISO DPD is guaranteed. For any mismatched loads at the output port, the voltages at ports 3 and 4 are affected by each other's current. The resulting modulated impedances at these ports are

$$Z_3 = \frac{V_3}{-I_3} = Z_o(1 + 2\Gamma_L), \quad \Gamma_3 = \frac{\Gamma_L}{1 + \Gamma_L} \tag{A.7}$$

$$Z_4 = \frac{V_4}{-I_4} = Z_o(1 - 2\Gamma_L), \quad \Gamma_4 = \frac{-\Gamma_L}{1 - \Gamma_L} \tag{A.8}$$

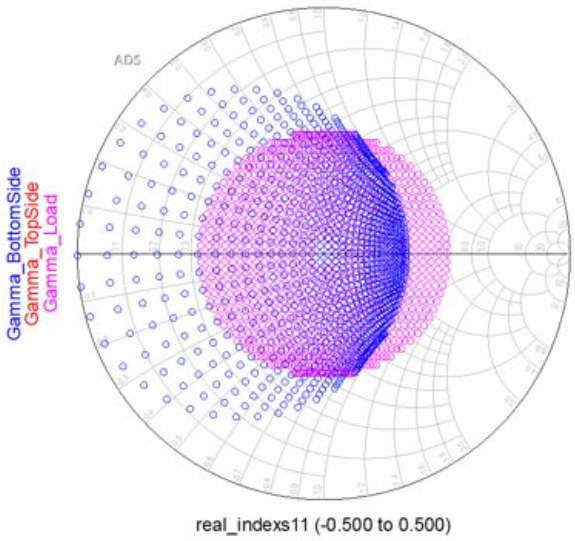


Figure A.3: **Balanced PA Impedances.**

Fig. A.3 shows Γ_3 and Γ_4 when Γ_L varies from 0 to 0.5 in its magnitude. When $\Gamma_L = \pm 0.5$, one side of the balanced PA sees a short-circuit from load modulation. For $|\Gamma_L| > 0.5$, the loads can expand outside of Smith chart and present negative impedances. Thus, when a transistor is simply modeled as a current source, the balanced PA cannot tolerate a wide range of load modulation. However, in a PA design, a matching network is present after the transistor output. The following section expands the balanced PA analysis taking the matching network into account.

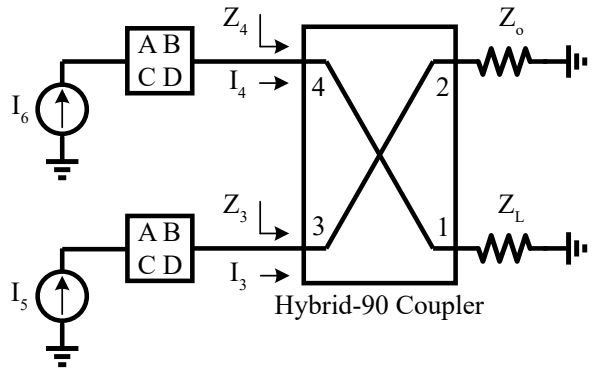


Figure A.4: **Balanced PA Schematic with Matching Networks.**

A.1.2 Matching Network Analysis

Fig. A.4 shows an updated balanced PA schematic with matching networks as ABCD matrices. Due to the addition of the ABCD matrices, I_3 and I_4 need to be re-written in terms of I_5 and I_6, respectively.

$$V_3 = Z_o\left[(1+\Gamma_L)I_3 - j\Gamma_L I_4\right], \quad I_3 = -C\left(\frac{AZ_3+B}{CZ_3+D}\right)I_5 + AI_5 = \frac{I_5}{CZ_3+D} \tag{A.9}$$

$$V_4 = Z_o\left[(1-\Gamma_L)I_4 - j\Gamma_L I_3\right], \quad I_4 = -C\left(\frac{AZ_4+B}{CZ_4+D}\right)I_6 + AI_6 = \frac{I_6}{CZ_4+D} \tag{A.10}$$

$$V_1^- = \frac{V_1 + Z_o I_1}{\sqrt{Z_o}} = \sqrt{2Z_o}\frac{(I_5 - jI_6)}{CZ_o+D} \tag{A.11}$$

$$\tag{A.12}$$

The resulting Z_3 and Z_4 impedances then change to

$$Z_3 = \frac{V_3}{-I_3} = Z_o\left(1+\Gamma_L+\Gamma_L\frac{CZ_3+D}{CZ_4+D}\right), \quad \Gamma_3 = \frac{\Gamma_L}{1+\Gamma_L\Gamma_{out}} \tag{A.13}$$

$$Z_4 = \frac{V_4}{-I_4} = Z_o\left(1-\Gamma_L-\Gamma_L\frac{CZ_4+D}{CZ_3+D}\right), \quad \Gamma_4 = \frac{-\Gamma_L}{1-\Gamma_L\Gamma_{out}} \tag{A.14}$$

where Γ_{out} is the reflection coefficient of $Z_{out} = D/C$ that is the Norton equivalent output impedance of the ABCD matrix. The analysis shows that if the output matching network

is designed to match Z_o at the output, the denominators of $\Gamma_{3,4}$ become 1, resulting in $\Gamma_{3,4} = \pm\Gamma_L$ at all times. This allows the impedances to stay within Smith chart for all passive loads at the output port.

Fig. A.5 compares current profiles and load modulation for $\Gamma_{out} = 0$ and $\Gamma_{out} = 1$ for $Z_L = 150\ \Omega$. Fig. A.5a shows that when $\Gamma_{out} = 1$, the currents of both PAs are coupled and follow each other until one PA saturates first. On the other hand, when $\Gamma_{out} = 0$, they are *decoupled* and maintain a fixed current ratio. These current profiles are reflected on the load modulation in fig. A.5b. Since the current ratio is fixed for $\Gamma_{out} = 0$, the impedances are also fixed. However, for $\Gamma_{out} = 1$, the top side impedance stays at short circuit until the bottom side begins to saturate. For the same output impedance, $\Gamma_{out} = 0$ has better robustness throughout the entire input power range. Therefore, the value of Γ_{out} can be a design parameter for a given range of Γ_L variation. For keeping all loads inside Smith chart, the following equations can be derived:

$$|\Gamma_3| \leq 1, \quad |\Gamma_4| \leq 1 \tag{A.15}$$

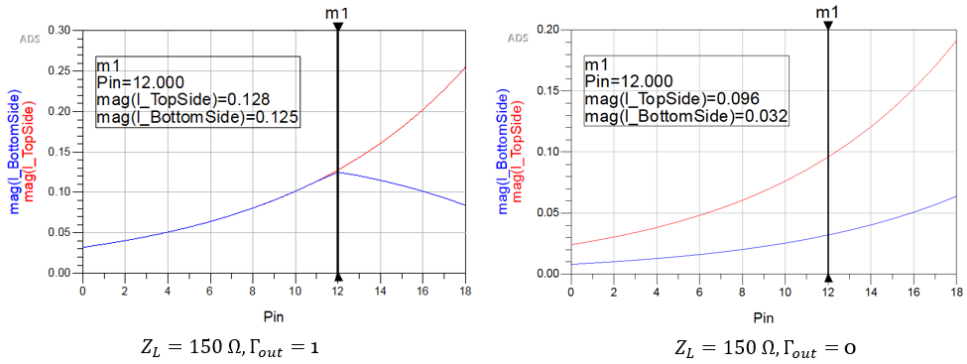
$$\left|\frac{\Gamma_L}{1+\Gamma_L\Gamma_{out}}\right| \leq 1, \quad \left|\frac{-\Gamma_L}{1-\Gamma_L\Gamma_{out}}\right| \leq 1 \tag{A.16}$$

$$|\Gamma_L| \leq |1+\Gamma_L\Gamma_{out}| \leq 1+|\Gamma_L||\Gamma_{out}| \tag{A.17}$$

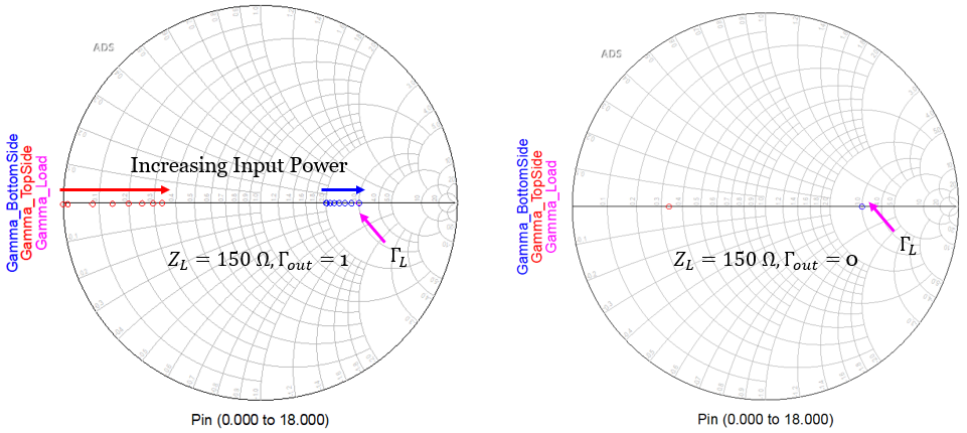
$$|-\Gamma_L| \leq |1-\Gamma_L\Gamma_{out}| \leq 1-|\Gamma_L||\Gamma_{out}| \tag{A.18}$$

$$\frac{|\Gamma_L|-1}{|\Gamma_L|} \leq 0 \leq |\Gamma_{out}| \leq \frac{1-|\Gamma_L|}{|\Gamma_L|} \tag{A.19}$$

Fig. A.6 shows a comparison of two different Γ_{out} values. For $\Gamma_{out} = 0.52e^{j26.4^\circ}$, the tolerable output impedance is $|\Gamma_L| \leq 0.658$ to keep the loads presented to individual PAs inside Smith chart. Fig. A.6a shows $|\Gamma_L| \leq 0.658$ and the loads are indeed inside Smith chart. Fig. A.6b presents a case for $\Gamma_{out} = 0.058e^{j112.4^\circ}$, in which $|\Gamma_L| \leq 0.945$. Learning from this analysis, the balanced PA should not consider transistors as current sources anymore. The presence of matching networks can greatly affect the range of load modulation it can tolerate.

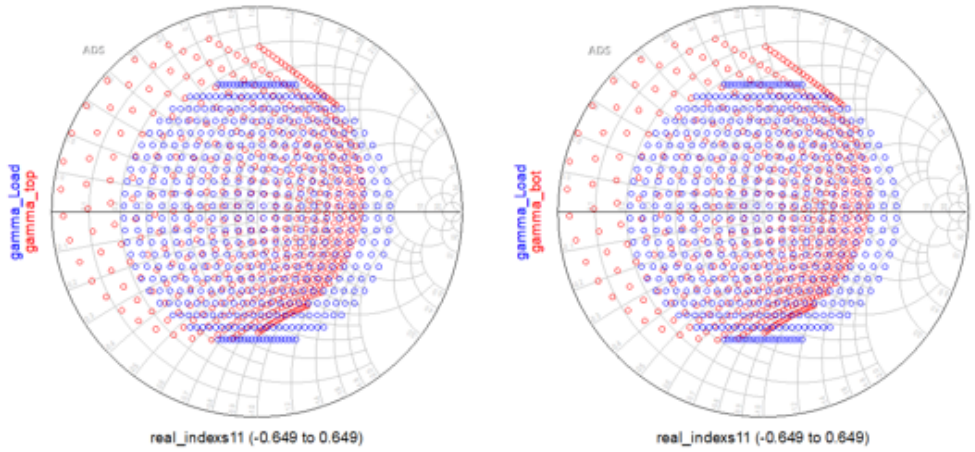


(a)

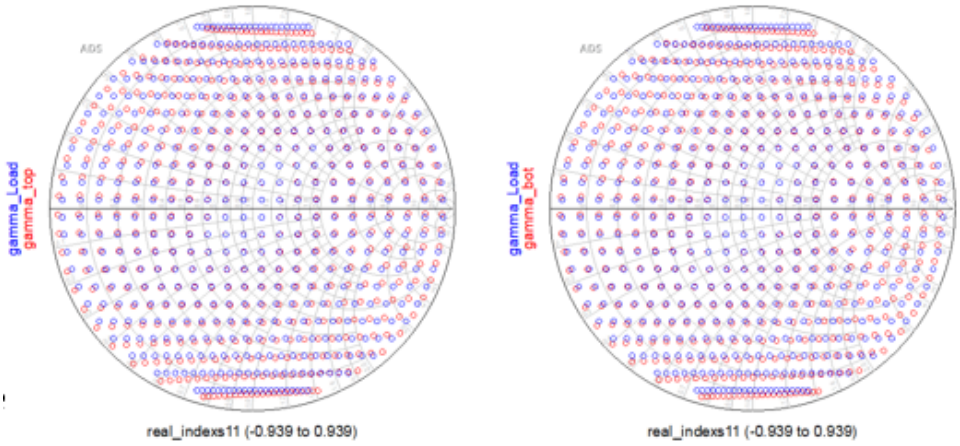


(b)

Figure A.5: **Balanced PA Γ_{out} Comparison.** (a) Current profiles, and (b) Impedances.

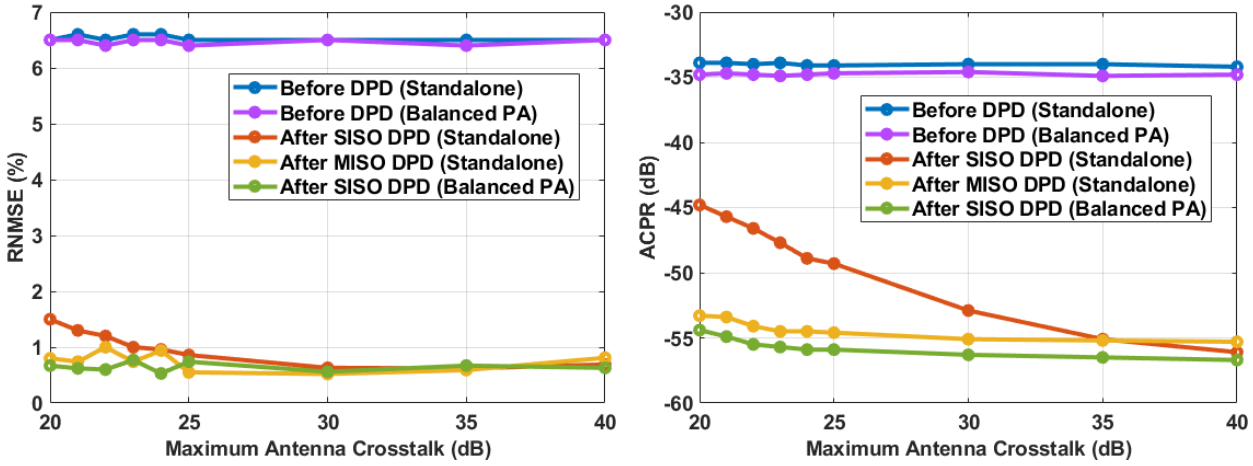


(a)



(b)

Figure A.6: **Effect of Choice of Γ_{out} for Balanced PA.** (a) $\Gamma_{out} = 0.52e^{j26.4^\circ}$, and (b) $\Gamma_{out} = 0.058e^{j112.4^\circ}$.



(a) (b)

Figure A.7: **DPD Applied on Class-AB PA and its balanced configuration in a 4×4 Fully Digital MIMO Transmitter for Various Maximum Antenna mutual coupling Magnitude.** (a) RNMSE and (b) ACPR.

A.1.3 16-Channel Fully Digital MIMO Transmitter Co-Simulation

For 16-channel fully digital MIMO transmitter co-simulation, standalone PA with $\Gamma_{out} = 0.058e^{j112.4^\circ}$ in fig. B.1 are replaced by its balanced configuration. Fig. A.7 shows the comparison of stanalone and balanced PA in 4×4 fully digital MIMO transmitter front-end before and after SISO DPD. It shows that before DPD, both standalone and balanced PA configurations behave similarly in RNMSE and ACPR. However, after SISO DPD, balanced PA configuration has a significant improvement over the standalone case, especially in ACPR. When the balanced PA is employed, SISO DPD can linearize the transmitter front-end system to the level of MISO DPD for the standalone PA case.

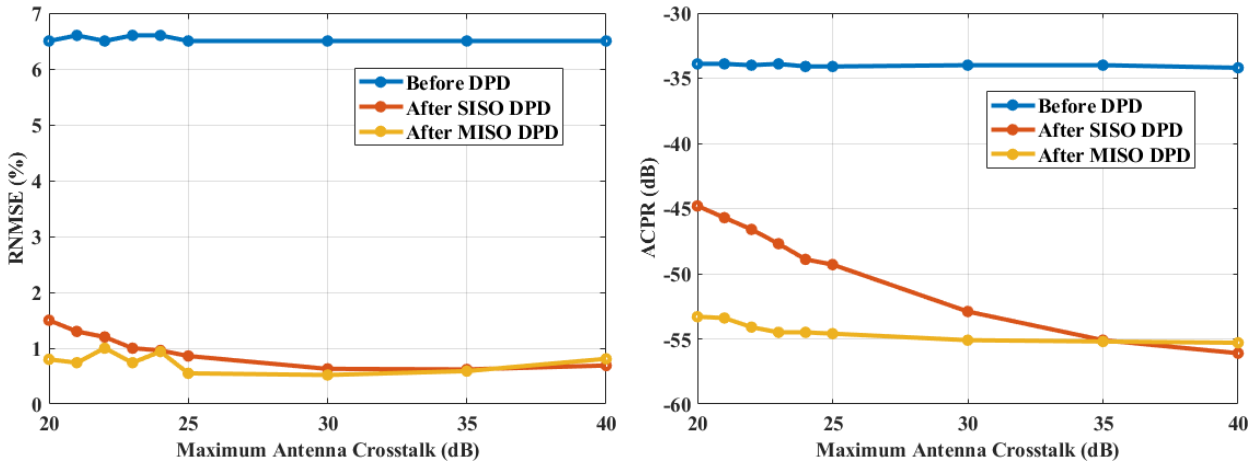
Appendix B

Massive MIMO Transmitter Front-End Analysis

B.1 Introduction

For optimal system-level massive MIMO transmitter performance, building blocks should be co-designed. DPD Modelling, PA efficiency and linearity, and PA output impedance modulation from precoders and antenna mutual coupling are three main challenging realizations of the fully digital MIMO transmitter. Reducing the complexity of such transmitter implementation requires contributions from all three components. SISO DPD has reduced complexity compared to DISO DPD, but the use of SISO DPD requires the subsequent PA to be linear and load-resilient against its output impedance modulation. Linear and load-resilient PA design also needs information about the output impedances that are the products of precoders and antenna mismatches and mutual coupling. Thus, effects of antenna mutual coupling and mismatches, and precoders on output impedance modulation are analyzed first. A method for such analysis is also described. Once the range of output impedance variation can be calculated, an appropriate PA topology should be selected or designed.

B.2 Analysis of the 4×4 MIMO Transmitter Front-End Non-Linearity as a Function of the Level of Antenna Mutual Coupling



(a) (b)

Figure B.1: **DPD Applied on Class-AB PA in a 4×4 Fully Digital MIMO Transmitter for Various Maximum Antenna Mutual Coupling Magnitude.** (a) RNMSE and (b) ACPR.

The first analysis examines how antenna mutual coupling factors affect PAs' linearizability using both SISO and MISO DPD. An ideal antenna mutual coupling module was created using ideal models in ADS with the characteristics of a real 4×4 antenna at its centre frequency. Thus, the antenna mutual coupling in its 4×4 configuration is preserved but the magnitudes of all mutual couplings can be changed by the same factor of C, which corresponds to the separation distance between antennas. Figure B.1 shows how the ACPR and RNMSE of the combined output signals of all 16 channels change with the varying mutual coupling factors before and after SISO and MISO DPDs. The maximum antenna mutual coupling value is utilized as a reference value that is swept from 20−40 dB. The results show that the antenna mutual coupling mainly affects the ACPR parameter and that the linearizability of the transmitter becomes similar for both DPDs for the antenna mutual coupling factor larger than 35 dB. For achieving ACPR below −50 dB for SISO DPD, the maximum antenna mutual coupling should be below −25 dB. For new DPD algorithms, PA and antenna designs, this study can be performed to analyze the antenna

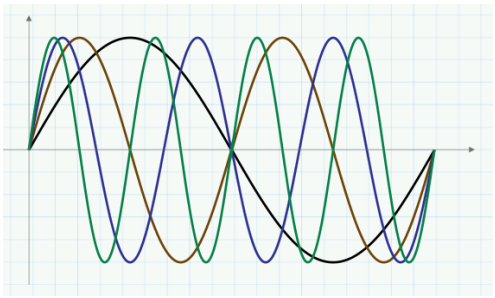


Figure B.2: **Time-Domain OFDM Symbol.** Reprinted from [12].

mutual coupling threshold to achieve a certain linearity requirements for the fully digital MIMO transmitter. This investigation is the first step towards PA-antenna co-design for the overall system-level transmitter linearity as the design goal.

B.3 Analysis of the 4×4 MIMO Transmitter Front-End Active Load Modulation for Uncorrelated Modulated Input Signals as a Function of the Level of Antenna mutual coupling

The second investigation is a study of PA output impedance modulation due to the antenna mutual coupling and mismatches in both time and frequency domain. Since the real signal source is in the time domain, real-time antenna impedance modulation was studied first from the DSP perspective. However, sampled signals in the time domain consist of multiple frequency components because the OFDM signal has many subcarriers. While DPD linearizes PAs in the time domain, the PA and antenna are designed in the frequency domain. Thus, PA output impedance modulation analysis was also performed in the frequency domain to understand how the PA-antenna co-design can be done.

A full OFDM signal consists of many OFDM symbols. In each symbol, digitally-modulated sub-carriers propagate in the CW form, as shown in fig. B.2. Since PAs are designed using CW excitations in the frequency domain, PA output impedances can be calculated from each OFDM symbol to check how much output impedance variations PAs must tolerate. This can be extended to PA-antenna co-design for engineering statistical PA output impedance modulation. Therefore, for the frequency domain analysis, $a_2(t)$ and $b_2(t)$ are sorted by OFDM symbols and then each symbol is Fourier-transformed separately.

Fig. B.3, B.4, and B.5 show the frequency domain impedances of 4×4 PAs in fully digital MIMO transmitter front-end, driven by sixteen independent 8 dB PAPR, 100 MHz bandwidth OFDM signals, when the maximum antenna mutual couplings are −20 dB, −25 dB, and −35 dB, respectively, obtained from results presented in fig. B.1. Each data point represents an impedance at each sub-carrier frequency and the effect of antenna mismatch is omitted here.

For the case with maximum −20 dB antenna mutual coupling, which is the typical scenario, most data points are confined within average VSWR circle of 1.66. The minimum and maximum VSWR values are 1.42 (top right corner; (d)) and 2.21 (top left of centre four; (f)), respectively. On average, PAs need to tolerate a range of 30−83 Ω output load modulation in the real part, while the worst case is 22.6−110.6 Ω.

For the case with maximum −25 dB antenna mutual coupling, at which SISO DPD can linearize the PAs below −50 dB ACPR, most data points are confined within average VSWR circle of 1.34. The minimum and maximum VSWR values are 1.21 (bottom left corner; (m)) and 1.61 (top edge; (b)), respectively. On average, PAs need to tolerate a range of 37.3−67 Ω output load modulation in the real part, while the worst case is 31−80.5 Ω. The worst case is similar to the average output load modulation of −20 dB case.

For the case with maximum −35 dB antenna mutual coupling, when SISO and MISO DPD perform similarly, most data points are confined within average VSWR circle of 1.07. The minimum and maximum VSWR values are 1.05 and 1.09, respectively. This is close to having constant 50 Ω loads at all times. SISO and MISO DPD can perform similarly when the output load is constant. Otherwise, SISO DPD is worse as expected.

Table. B.1 summarizes the PA output load modulation results. The results show tha if PAs can be designed to reduce the load modulation from 1.66 to 1.34 VSWR, SISO DPD can achieve ACPR below −50 dB, which gives 5 dB margin from the requirement.

Table B.1: **Comparison of the effect of antenna mutual coupling factor for 16-chain fully digital MIMO transmitter, driven with uncorrelated 100 MHz, 8 dB PAPR OFDM signals.**

Maximum mutual coupling	Average VSWR	Worst VSWR
−20 dB	1.66 (30−83 Ω)	2.21 (22.6−110.6 Ω)
−25 dB	1.34 (37.3−67 Ω)	1.61 (31−80.5 Ω)
−35 dB	1.07 (46.7−53.5 Ω)	1.09 (45.9−54.5 Ω)

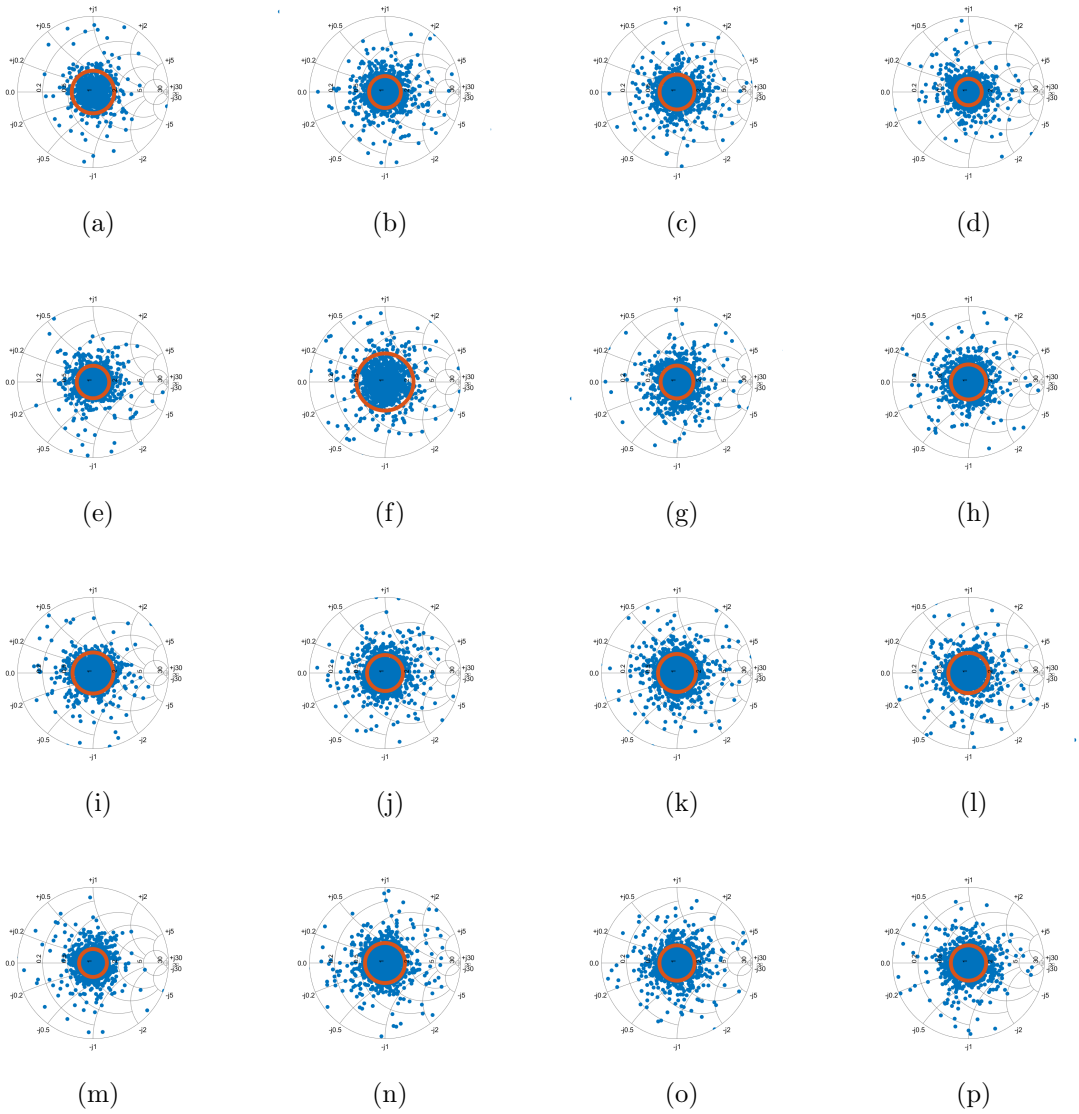
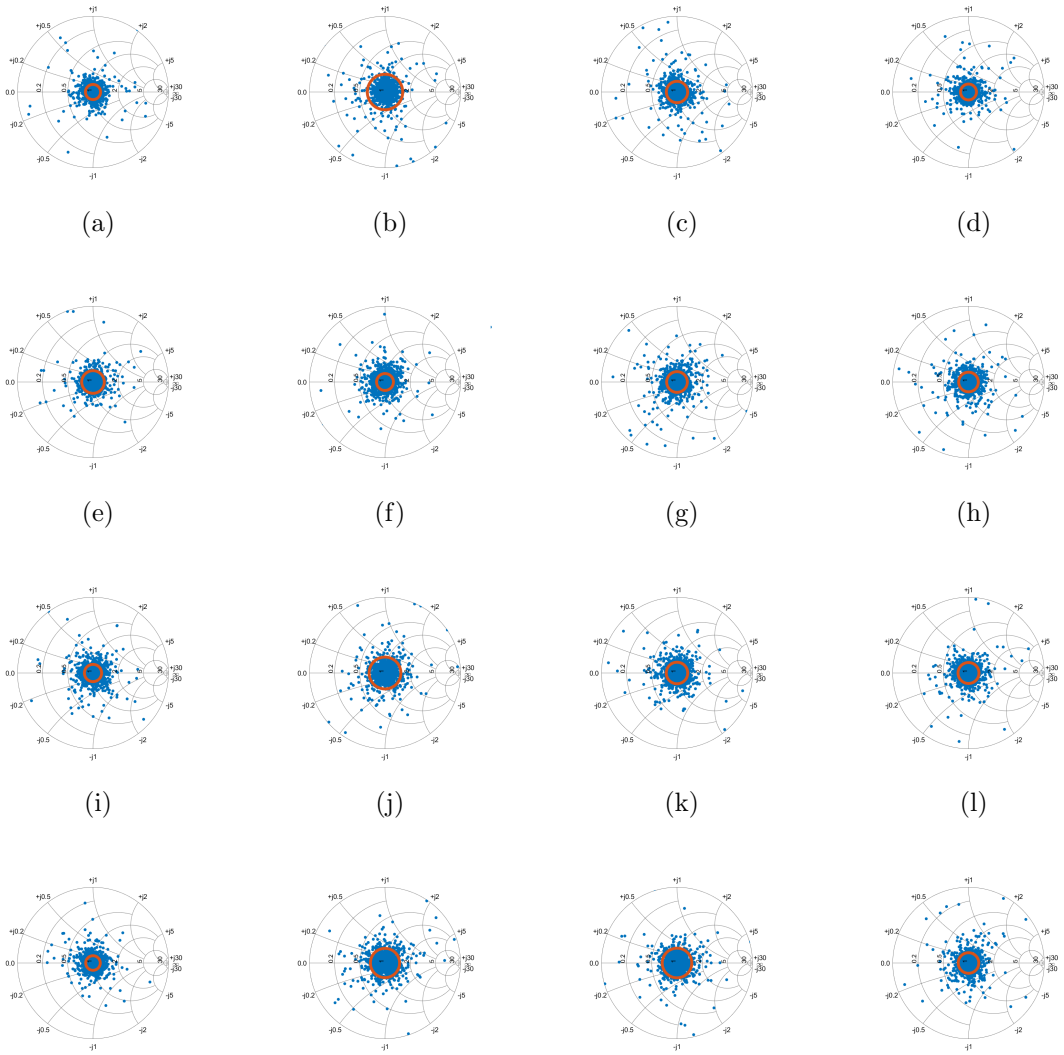


Figure B.3: **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p).



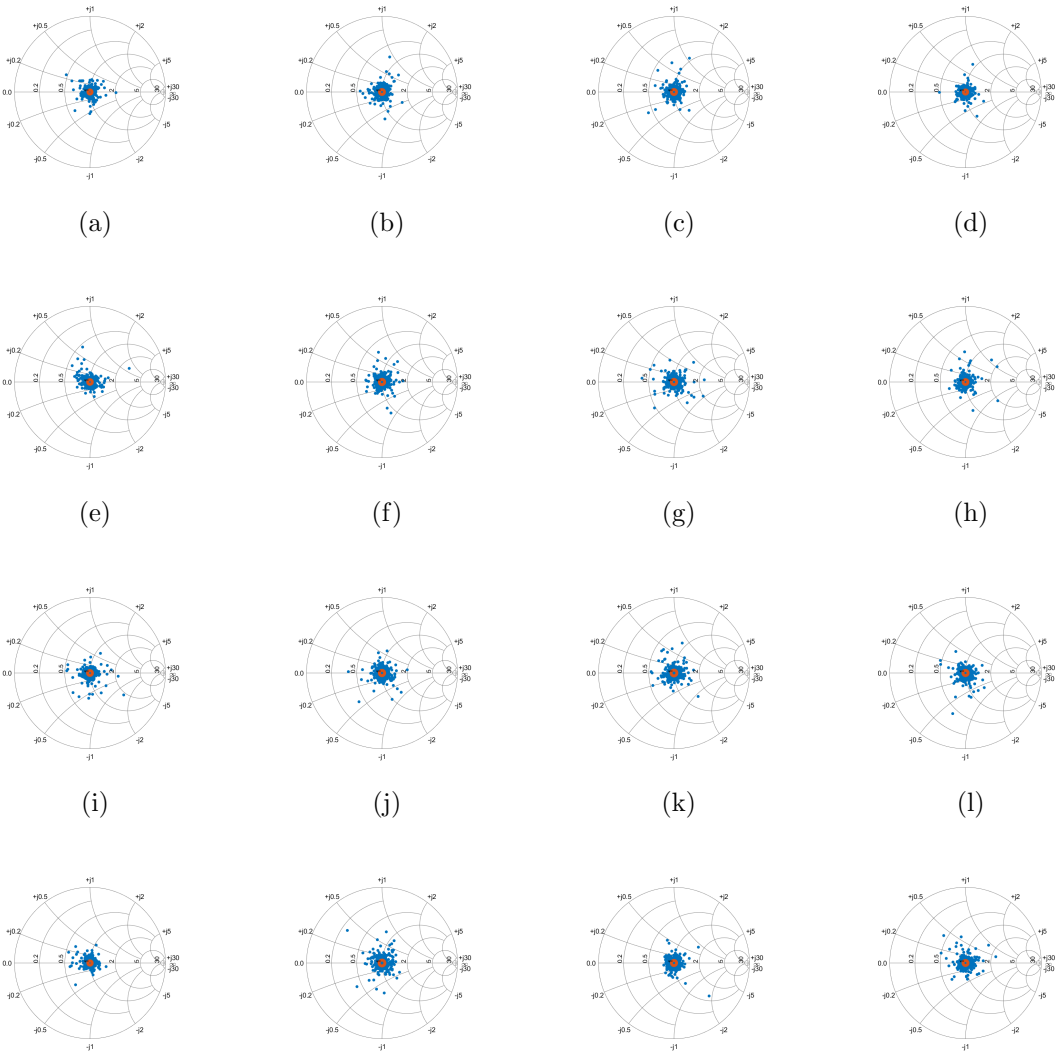
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure B.4: **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −25 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p).



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure B.5: **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −35 dB Antenna mutual coupling.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p).

Table B.2: **Comparison of the effect of precoders for 16-chain fully digital MIMO transmitter, driven with 256-QAM, 100 MHz, 8 dB PAPR OFDM signals.**

Precoder	Best VSWR	Average VSWR	Worst VSWR
Uncorrelated	1.43 (35−71.5 Ω)	1.66 (30−83 Ω)	2.21 (22.6−110.6 Ω)
Conventional Zero-Forcing	1.25 (40−62.5 Ω)	2.06 (24.3−103 Ω)	5.31 (9.4−265.5 Ω)
Equal-Power Zero-Forcing	1.25 (40−62.5 Ω)	1.76 (28.4−88 Ω)	2.65 (18.9−132.5 Ω)

B.4 Analysis of the Active Load Modulation for Precoded Modulated Input Signals

This section shows how precoders can affect PA output impedances using the developed co-simulation framework in Sec. 3. Precoders introduce correlation between signals in different RF chains, modifying the output impedances presented to each PA. How different these impedances are from those from uncorrelated signals should be analyzed. Fig. **??** shows impedances presented to four different PAs in a 2×2 fully digital MIMO transmitter when 4-QAM, 100 MHz, 8 dB PAPR uncorrelated signals are applied. Each RF signal has the same average power in this case. Most of the loads are confined in VSWR circle of 2:1.

Fig. B.6 and B.7 show output impedance modulation under 256-QAM, 8 dB PAPR, 100 MHz OFDM signal in a 4×4 transmitter with maximum −20 dB antenna mutual coupling when conventional zero-forcing precoder and equal-power zero-forcing precoder are applied, respectively. As the modulation order and the number of chains increase, the output impedance plots appear similar to those of the uncorrelated signal cases with 4-QAM, 2×2 fully digital MIMO transmitter in fig. **??** and **??**.

Table B.2 summarizes how different precoders can affect PA output impedance modulation. The range of VSWR circles across channels is wider for precoded cases. Best VSWR corresponds to a high power chain surrounded by much lower power chains so less modulation is experienced. On the other hand, the worst VSWR occurs when a low power chain is surrounded by higher power chains. It clearly shows that the conventional zero-forcing precoder is not as effective in minimizing load modulation in low power chains as the equal-power zero-forcing precoder.

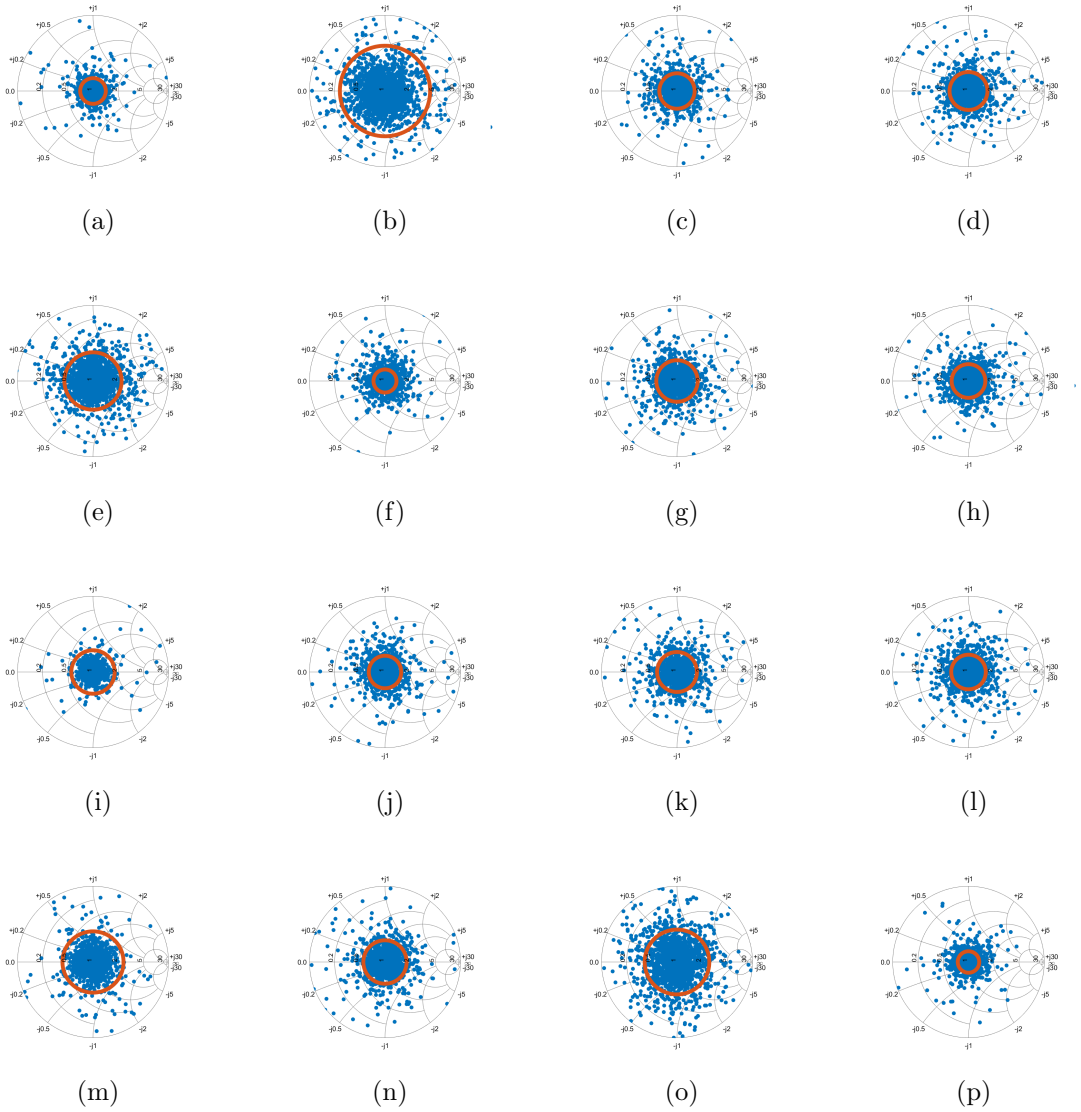


Figure B.6: **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling with Conventional Zero-Forcinsg.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). As the modulation order increases, the impedance plots appear similar to those under uncorrelated signals.

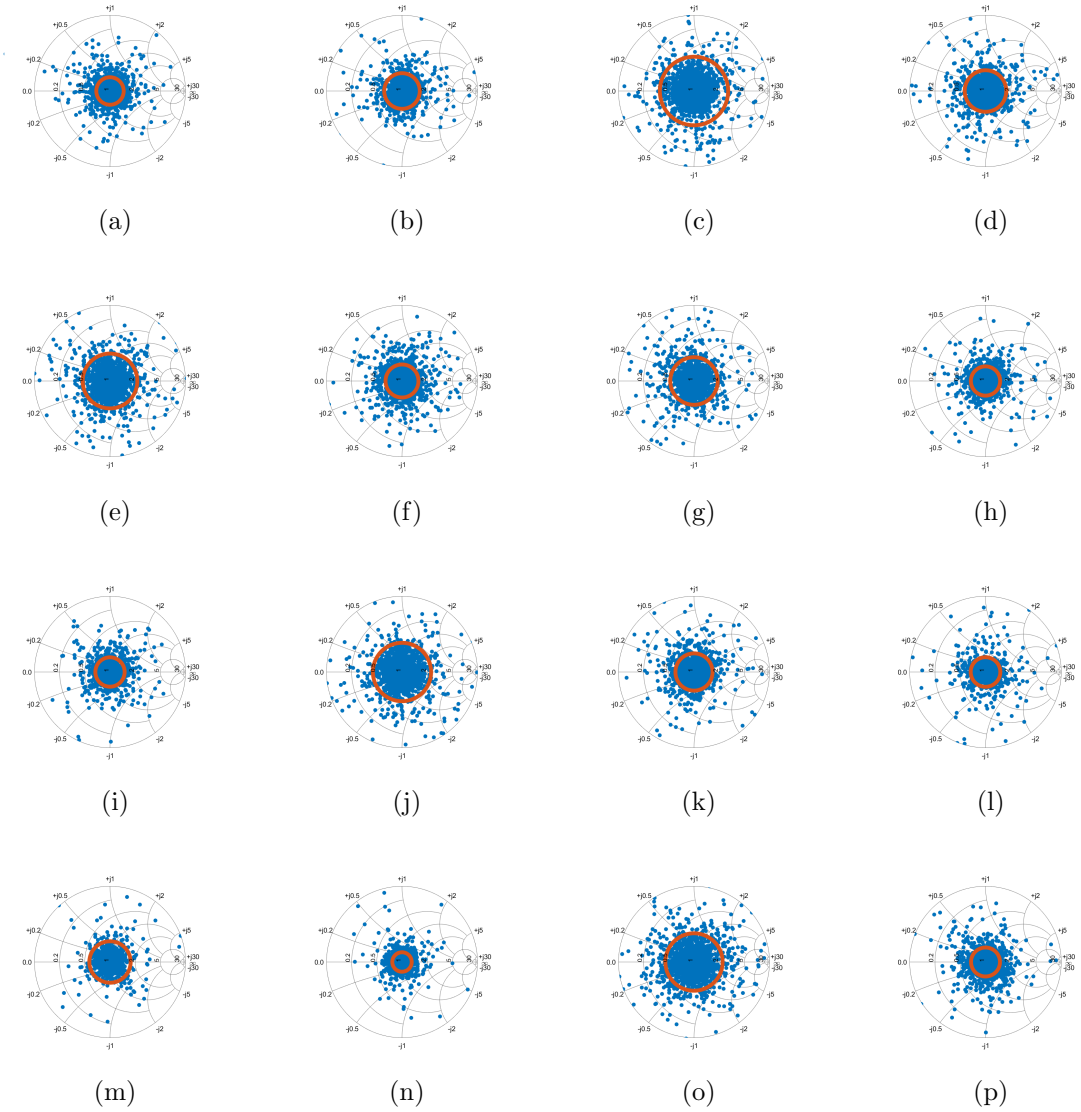


Figure B.7: **Frequency Domain Antenna Impedance Modulation Under 256-QAM, 8 dB PAPR, 100 MHz OFDM Signal in a 4×4 Transmitter with Maximum −20 dB Antenna mutual coupling with Equal-Power Zero-Forcinsg.** All impedances are normalized by the average impedance of each RF chain. Each blue data point corresponds to the impedance of each sub-carrier frequency. Orange circles represent average VSWR circles within which most of the sub-carrier impedances reside. RF chain numbering follows an ascending order from (a) to (p). As the modulation order increases, the impedance plots appear similar to those under uncorrelated signals.

B.5 Conclusion

Analyses in PA linearization and output impedance modulation showed that if the antenna mutual coupling is better than −35 dB, SISO and MISO DPDs give the same level of linearization. In addition, for antenna mutual coupling below −25 dB, SISO DPD can linearize 4×4 fully digital MIMO transmitter front-end down to −50 dB ACPR. PA output impedance modulation should be reduced from average VSWR circle of 1.66 to 1.33 for −25 dB or to 1.07 for −35 dB. A new PA design for reducing the output impedance sensitivity can be crucial in reducing the complexity of DPD modelling. Furthermore, previous research on load modulation trained PAs with uncorrelated signals and did not include the effects of precoders that reflect on channel sensitivity. However, precoders can significantly transform PA output impedances. Thus, a type of signals for DPD training should be carefully considered. The time-domain active load-pull characterization setup showed that it can accurately predict a PA behaviour in a MIMO transmitter front-end configuration. The time-domain and frequency-domain impedance analysis as well as the application of precoders will be further investigated for the development of a new linear and load-resilient PA topology.

Appendix C

Preliminary 2×2 Full Measurement-based Emulation

For the verification of the emulated MIMO transmitter configuration, simulated and measured MIMO emulation results are compared to full MIMO transmitter system measurement results. Fig. C.1 shows full 2×2 MIMO transmitter measurement setup and Fig. C.2 shows the emulated MIMO transmitter measurement setup. The vector network analyzer (VNA) in the emulation setup is calibrated for measuring $a_1[n]$, $b_1[n]$, $a_2[n]$, and $b_2[n]$. Two RF signal generators for $a_1[n]$ and $a_2[n]$ are calibrated in absolute and relative amplitude and phase, time delay, and frequency response to correctly arrive at the input and output of the PA, respectively. Four Class-AB PAs of the same design are characterized at 3.5 GHz center frequency with 100 MHz modulation bandwidth and 9 dB PAPR uncorrelated signals, and are linearized by CRV Pruned DPD algorithm with indirect learning and 71 coefficients.

Table C.1 compares full 2×2 MIMO transmitter measurements and its emulation in measurements and simulations, and Fig. C.3 shows the resulting spectra without and with DPD for each channel. The PA placed for corresponding RF chain is used for emulation for better prediction. For both emulation results in measurements and simulations, the process and fabrication variations between different PAs cannot be fully reflected. Therefore, discrepancies between full 2×2 system and emulation results exist.

Emulation results in measurements and simulations show almost identical results with and without DPD. Without DPD, RNMSE and ACPR results are within 1.5% and 1 dB differences, while with DPD, RNMSE and ACPR results are within 1.5% and 2.7 dB differences. Without DPD, the non-linearity of the PA dominates its performance over

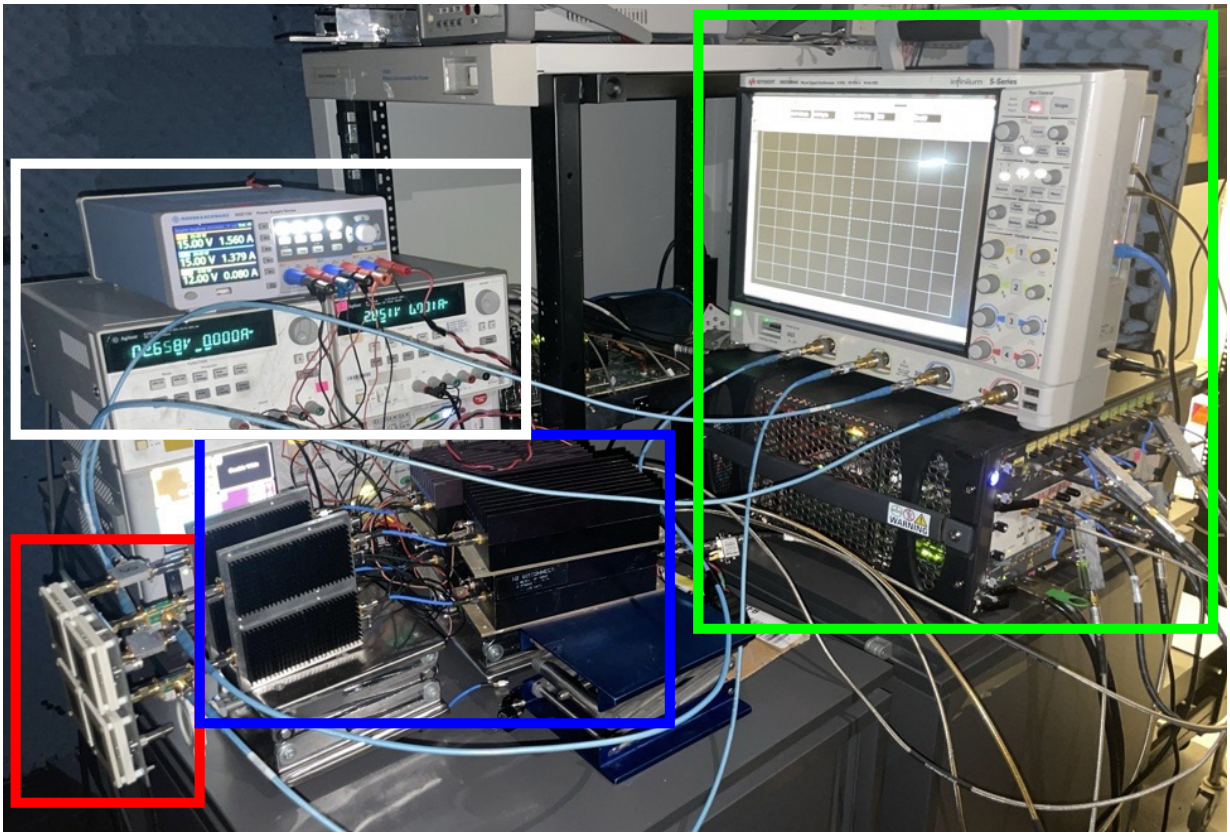


Figure C.1: **Full 2×2 MIMO Transmitter Measurement Setup.** Red box: antenna array and directional couplers as transmitter observation receiver (TOR), Blue box: class AB PA and driver, Green box: signal generator and oscilloscope, White box: DC power supplies

the antenna mutual coupling, but with DPD, the antenna mutual coupling dictates the PA linearity performance. It shows that simulation results of a PA can accurately predict its non-linearity in measurements. For with DPD results, the PA output impedances are compared in Fig. C.4. It shows that the radius of the PA output impedances in measurements is slightly larger than that in simulations. This difference between them yield to a slight difference in the simulated and measured linearization performance despite having the similar starting points.

When comparing full 2×2 measurements to both emulation results, channel #1 and #3 show asymmetry in ACPR without DPD but ACPR with DPD are the same or within 1 dB. This means that the presented PA output impedances in emulation closely rep-

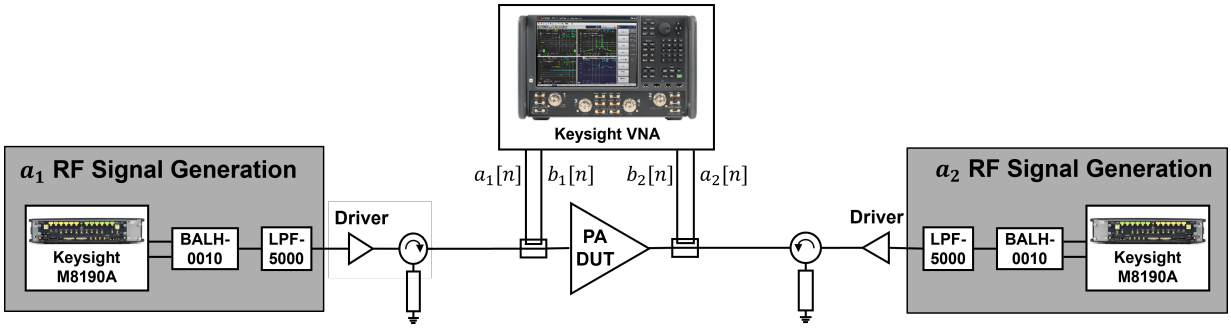


Figure C.2: **Active Load-Pull-Based Emulated MIMO Transmitter Measurement Setup.**

Table C.1: **Comparison of Full MIMO Transmitter Performance and MIMO Emulation Configuration Employing Class AB PA under 100 MHz, 8 dB PAPR OFDM, 256 QAM OFDM Signal.**

	SISO DPD	Channel #1			Channel #2			Channel #3			Channel #4		
		RNMSE (%)	ACPR (dB)	Power (dBm)	RNMSE (%)	ACPR (dB)	Power (dBm)	RNMSE (%)	ACPR (dB)	Power (dBm)	RNMSE (%)	ACPR (dB)	Power (dBm)
Full 2×2 Meas.	Without DPD	10.6	−33.5/−34.2	27.3	11.3	−34.0/−35.9	27.3	14.2	−32.5/−36.4	27.3	11.6	−33.0/−35.9	27.4
	With DPD	4.3	−47.0/−48.7	27.1	7.5	−44.9/−47.1	27.2	4.7	−46.6/−49.5	26.9	4.8	−46.4/−48.1	27.1
Emulation Meas.	Without DPD	7.5	−36.2/−35.1	27.4	7.5	−35.6/−35.1	27.4	8.3	−36.0/−35.0	27.4	8.2	−35.7/−34.7	27.3
	With DPD	2.3	−47.0/−48.7	27.0	2.4	−48.5/−49.3	27.1	2.2	−47.5/−48.6	27.1	2.5	−49.3/−50.3	27.0
Emulation Sim.	Without DPD	6.8	−36.8/−36.0	27.2	6.7	−36.3/−35.8	27.2	6.7	−36.7/−35.8	27.2	6.7	−36.5/−35.3	27.2
	With DPD	1.2	−48.2/−50.4	27.0	1.0	−51.2/−51.5	27.1	1.0	−49.1/−50.8	27	0.9	−51.7/−51.1	27.1

resented the full MIMO transmitter configuration. On the other hand, channel #2 and #4 show discrepancies in spectra. This indicates that, similar to fig. C.4, the actual antenna mutual coupling in the MIMO transmitter front-end is worse that the formulated $a_2[n]$ in emulation. However, emulation results can predict the general trends of the PA characterization and linearization in MIMO transmitter front-end.

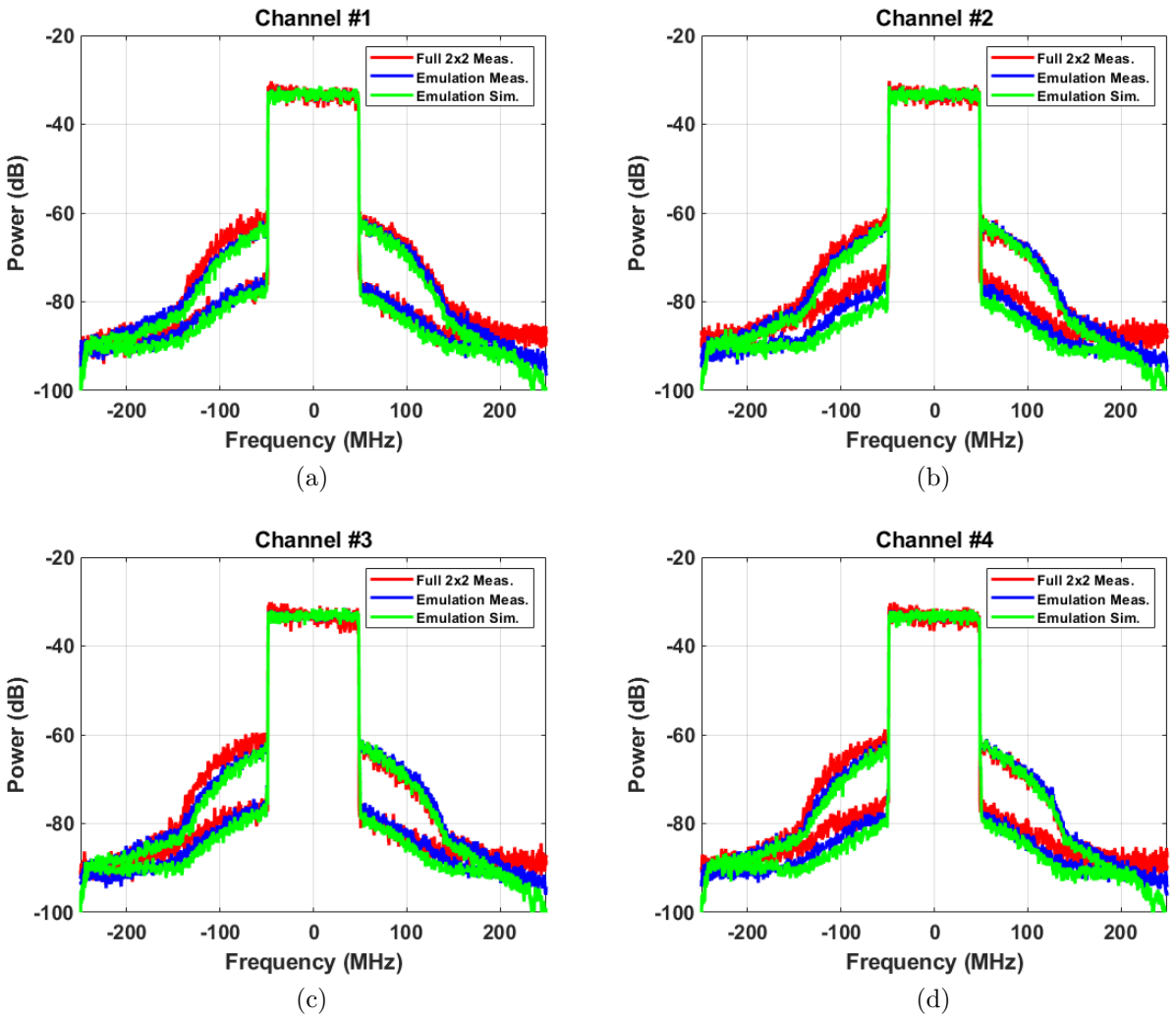
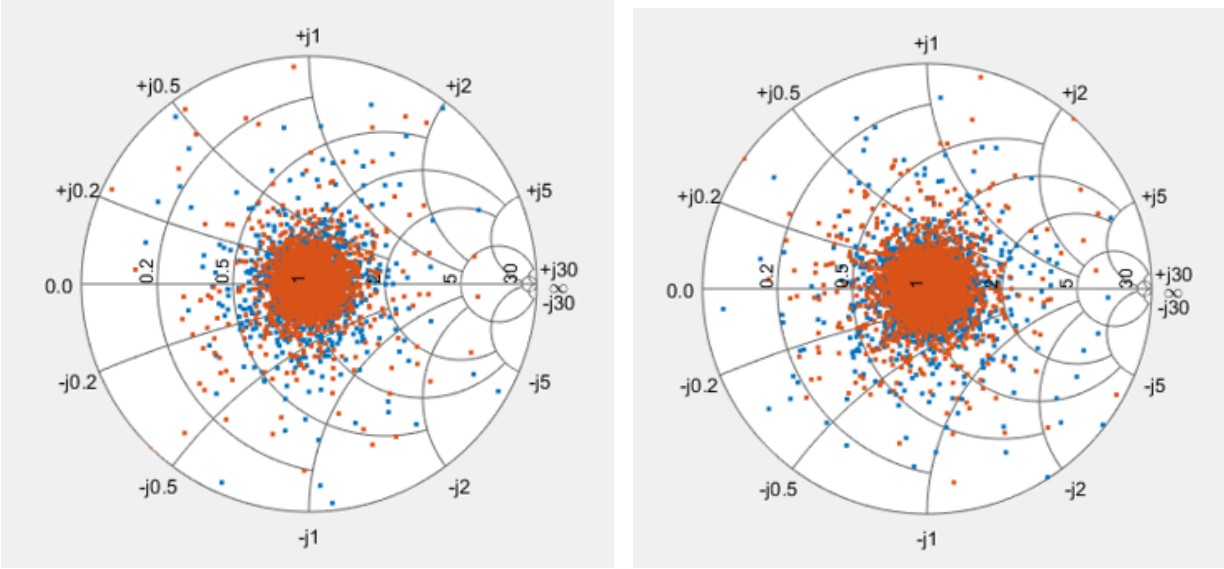


Figure C.3: **PA Output Spectra Under 100 MHz, 8 dB PAPR OFDM, 256 QAM OFDM Signal.**



(a) (b)

Figure C.4: **PA Sub-Carrier Load Impedance Variation Across 100 MHz Bandwidth.** (a) Simulation, (b) Measurement. The results are normalized to 50 Ω.

Appendix D

A Two-Tier Low-Complexity Linearization Architecture for Fully Digital mMIMO Transmitters

Jin Gyu Lim*, *Graduate Student Member, IEEE,* Ahmed Ben Ayed*, *Member, IEEE,* and Slim Boumaiza, *Senior Member, IEEE*

***Abstract*—This paper presents a two-tier, low-complexity linearization framework for fully digital multiple-input multiple-output (MIMO) transmitters that jointly minimizes total radiated power–adjacent channel power ratio (TRP-ACPR) and user error vector magnitude (EVM). The proposed architecture consists of: (i) a single-input single-output (SISO) digital predistorter (DPD), trained using a feedback signal processed through an in-band notch filter to isolate and suppress out-of-band distortion in each channel, and (ii) a low-order linear MIMO pre-equalizer placed before the precoder to compensate for in-band distortion at the user receivers. The two stages are jointly trained, enabling simultaneous minimization of TRP-ACPR and EVM without increasing precoder complexity or relying on multi-input or high-order nonlinear DPD models. Experimental results from a 16-channel fully digital transmitter operating at 3.5 GHz in a two-user line-of-sight scenario show that the pre-equalizer reduces user EVM to 0.7%, while a 13-coefficient per-antenna SISO-DPD improves TRP-ACPR from -33 dB to -43 dB and improves ACPR at users 1 and 2 from -36.5 dB to -49.6 dB and from -37 dB to -51.2 dB, respectively.**

***Index Terms*—massive MIMO transmitter, MIMO, digital predistortion, DPD, precoders, power amplifiers**

I. Introduction

Digital beamforming has become the preferred architecture for sub-6 GHz massive multiple-input multiple-output (mMIMO) systems, where each antenna is equipped with its own RF chain and digital-to-analog converter. By precoding user signals in the digital domain, the transmitter can deliver the desired waveforms to each user with high beamforming gain and effective interference suppression. To maximize power efficiency and radiated output, power amplifiers (PAs) are typically operated near saturation and consequently introduce nonlinear distortion that degrades both in-band signal quality and out-of-band spectral emissions. If unmitigated, these impairments can lead to poor total radiated power–adjacent channel power ratio (TRP-ACPR) and degraded user error vector magnitude (EVM).

Digital predistortion (DPD) is the standard solution for linearizing single-antenna transmitters [1]. Extending DPD to multi-antenna arrays, however, is considerably more challenging due to mutual coupling between antennas. Coupling produces time-varying incident waves at each PA output, leading to dynamic load modulation that alters the effective nonlinearity experienced by each chain. Several works have proposed MIMO-DPD architectures [2], [3] in which each DPD processes multiple input signals, but these approaches scale poorly with the number of RF chains.

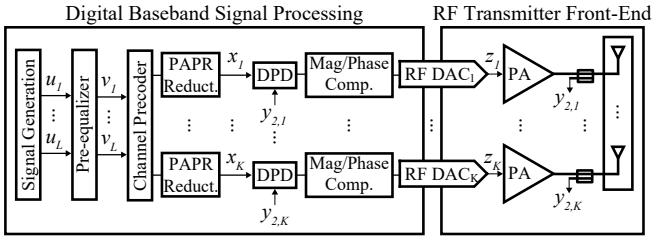


Fig. 1. Block diagram of proposed two-tier low-complexity linearization architecture for fully digital mMIMO transmitter.

To reduce complexity, dual-input single-output (DISO) DPD architectures were introduced [4], where each DPD is fed with both the intended transmit signal and an estimate of the incident wave at the PA—obtained either from reverse couplers, antenna S-parameters, or coupling models. Although DISO-DPD significantly reduces complexity compared to full MIMO-DPD, prior work often assumes uncorrelated PA inputs—an assumption that does not hold in multi-user precoding scenarios where PA inputs are highly correlated and strongly dependent on the channel.

More recently, alternative approaches [5] have proposed applying DPD before the precoding stage in mMIMO systems. While these methods reduce the number of DPD blocks to scale with the number of users, they cannot guarantee per-antenna spectral compliance and therefore cannot ensure TRP–ACPR minimization. In addition, placing the DPD upstream of the precoder introduces a strong dependence on the channel realization: as the channel changes, the effective nonlinear transformation observed at each user also changes, forcing the DPD to be retrained for every new channel state. This substantially increases training overhead and operational complexity. Furthermore, the nonlinear expansion introduced by pre-precoder DPD increases the bandwidth of the signals entering the precoder, thereby imposing additional computational load on the precoder stage.

To address these limitations, this work introduces a low-complexity linearization architecture tailored for mMIMO transmitters. The proposed solution adopts a two-tier strategy: (i) a per-antenna SISO-DPD that targets TRP-ACPR improvement and is trained using feedback processed through an in-band notch filter to isolate and suppress out-of-band distortion, and (ii) a low-order linear MIMO pre-equalizer placed before

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by the Ontario Research Fund (ORF). (Corresponding author: Jin Gyu Lim)

This article was presented at the IEEE International MTT Symposia (IMS2026), Boston, MA, USA, 7 -12 June 2026.

*Jin Gyu Lim and Ahmed Ben Ayed contributed equally to this work.

the precoder to reduce user EVM. The proposed architecture is compatible with conventional precoders, avoids the rapid complexity growth associated with scaling array size, and remains effective under realistic multi-user channel conditions. Experimental results obtained using a 16-element transmitter verify that the approach simultaneously improves user EVM and TRP-ACPR while maintaining very low DPD complexity.

II. Proposed Linearization Method Formulation

Fig 1 illustrates the proposed low-complexity linearization architecture for a fully digital K-antenna mMIMO transmitter. The system processes L user data streams, each represented by a complex baseband equivalent (CBBE) signal. A low-order linear MIMO pre-equalizer is applied directly to the user signals to mitigate in-band distortion at the receivers. The pre-equalized signals are then mapped onto the K transmit chains through a conventional precoder $\mathbf{W} \in \mathbb{C}^{K\times L}$. The corresponding downlink channel is denoted by $\mathbf{H} \in \mathbb{C}^{L\times K}$. Because the proposed approach does not modify the structure of $\mathbf{W}$, it remains fully compatible with standard multi-user MIMO processing.

For notational convenience, all signals are expressed in the discrete-time complex baseband domain and denoted by a tilde. For channel $k \in \{1,\dots,K\}$, let $\tilde{x}_k[n]$, $\tilde{z}_k[n]$, and $\tilde{y}_k[n]$ denote the precoder output, predistorted signal, and PA output, respectively. The corresponding regressor vectors of length $M+1$ are defined as $\tilde{\boldsymbol{x}}_k(n) = [\tilde{x}_k[n],\dots,\tilde{x}_k[n-M]]$, with $\tilde{\boldsymbol{z}}_k(n)$ and $\tilde{\boldsymbol{y}}_k(n)$ defined analogously. Similarly, let L denote the number of users. For each user $\ell \in \{1,\dots,L\}$, let $\tilde{u}_\ell[n]$ denote the input signals intended for that user, $\tilde{v}_\ell[n]$ the corresponding pre-equalizer output signal, and $\tilde{r}_\ell[n]$ the received signal at the user. Define $\tilde{\mathbf{u}}[n] = [\tilde{u}_1[n],\dots,\tilde{u}_L[n]]^T$, and the vectors $\tilde{\mathbf{v}}[n]$ and $\tilde{\mathbf{r}}[n]$ are constructed in the same way by stacking $\tilde{v}_\ell[n]$ and $\tilde{r}_\ell[n]$, respectively.

A. Tier I: SISO Predistortion for TRP–ACPR Minimization

The predistorter applied to channel k is a SISO nonlinear function with P basis functions $\{\psi_p(\cdot)\}$ and is expressed as,

$$\tilde{z}_k[n] = \tilde{x}_k[n] + \boldsymbol{\psi}_k(n)^T\,\boldsymbol{\alpha}_k, \tag{1}$$

where $\boldsymbol{\psi}_k(n) = [\psi_1(\tilde{\boldsymbol{x}}_k[n]),\dots,\psi_P(\tilde{\boldsymbol{x}}_k[n])]^T$, $\boldsymbol{\alpha}_k = [\alpha_{k,1},\dots,\alpha_{k,P}]^T$, and $\alpha_{k,p}$ is the coefficient associated with the pth basis function and kth channel.

The TRP–ACPR metric is governed by the out-of-band distortion generated at the PA outputs. Let $\mathcal{B}\{\cdot\}$ denote a linear notch operator that suppresses the in-band component and preserves only the out-of-band distortion. Applied to the PA output of channel k, it yields

$$\bar{y}_k[n] = \mathcal{B}\{\tilde{y}_k[n]\}. \tag{2}$$

Because TRP–ACPR is proportional to the total out-of-band power radiated across all transmit paths, the DPD coefficients $\boldsymbol{\alpha}_k$ are identified by minimizing the accumulated out-of-band energy, defined as

$$J_{\mathrm{TRP}} = \sum_{k=1}^{K}\sum_{m=0}^{M} |\bar{y}_k[n-m]|^2. \tag{3}$$

The coefficient vector $\boldsymbol{\alpha}_k$ is then updated iteratively using a quasi-Newton method for real-valued cost functions with complex parameters, leading to the update expression

$$\hat{\boldsymbol{\alpha}}_k^{(j+1)} = \hat{\boldsymbol{\alpha}}_k^{(j)} - \mu\,\bar{\boldsymbol{\Psi}}_k(n_j)^\dagger\,\bar{\boldsymbol{y}}_k(n_j), \tag{4}$$

where $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse, $\bar{\boldsymbol{\Psi}}_k(n_j) = [\bar{\boldsymbol{\psi}}_k(n_j),\dots,\bar{\boldsymbol{\psi}}_k(n_j - M)]^T$, and $\bar{\boldsymbol{\psi}}_k(n_j) = \mathcal{B}\{\boldsymbol{\psi}_k(n_j)\}$. Here, μ is the learning rate and j is the DPD iteration counter. A key feature of the proposed method is that the notch operator $\mathcal{B}\{\cdot\}$ is used only during coefficient identification in (4), but not during the predistortion operation (1). As a result, the DPD focuses exclusively on suppressing out-of-band radiation, ensuring TRP–ACPR minimization with very low complexity, while leaving in-band linear and weakly nonlinear errors at the PA outputs largely uncorrected.

B. Tier II: MIMO Pre-Equalizer for User EVM Mitigation

Residual in-band distortion that is not addressed by the predistorter is corrected by a MIMO pre-equalizer operating on the user signals before the precoder.

For each user $\ell \in \{1,\dots,L\}$, the MIMO pre-equalizer output signal, $\tilde{v}_\ell[n]$, is obtained using all user inputs as,

$$\tilde{v}_\ell[n] = \boldsymbol{\phi}(n)^T\boldsymbol{\beta}_\ell, \tag{5}$$

where

$$\boldsymbol{\phi}(n) = [\tilde{u}_1[n],\dots,\tilde{u}_1[n-M'],\dots,\tilde{u}_L[n-M']]^T, \tag{6}$$

M' is the number of filter taps, and $\boldsymbol{\beta}_\ell$ is the linear MIMO filter coefficient vector associated with user ℓ.

Under ideal linear PAs, the users' received signals can be obtained as follows,

$$\tilde{\mathbf{d}}[n] = \mathbf{H}\mathbf{W}\,\tilde{\mathbf{u}}[n]. \tag{7}$$

The pre-precoder equalizer therefore seeks the coefficient sets $\boldsymbol{\beta}_\ell$ that force the actual received signal $\tilde{\mathbf{r}}[n]$ to match $\tilde{\mathbf{d}}[n]$ in (7). This is equivalent to minimizing the following cost function,

$$J_{\mathrm{EVM}} = \sum_{\ell=1}^{L}\sum_{m=0}^{M}\left|\tilde{r}_\ell[n-m] - \tilde{d}_\ell[n-m]\right|^2. \tag{8}$$

Using Newton's method, the coefficients $\boldsymbol{\beta}_\ell$ that minimize the cost function in (8) can be obtained iteratively as follows,

$$\hat{\boldsymbol{\beta}}_\ell^{(j+1)} = \hat{\boldsymbol{\beta}}_\ell^{(j)} - \mu\,\hat{\boldsymbol{\Phi}}(n_j)^\dagger\left(\tilde{\boldsymbol{r}}_\ell(n_j) - \tilde{\boldsymbol{d}}_\ell(n_j)\right)^T, \tag{9}$$

where μ is the learning rate, j is the iteration index, and

$$\hat{\boldsymbol{\Phi}}(n_j) = [\hat{\boldsymbol{\phi}}(n_j),\dots,\hat{\boldsymbol{\phi}}(n_j - M)]^T.$$

Note, $\hat{\boldsymbol{\phi}}(n_j)$ is constructed from the ideal reference signal as

$$\hat{\boldsymbol{\phi}}(n_j) = [\tilde{d}_1[n_j],\dots,\tilde{d}_1[n_j - M'],\dots,\tilde{d}_L[n_j - M']]^T. \tag{10}$$

At each iteration, the DPD coefficients $\boldsymbol{\alpha}_k$ and the pre-equalizer coefficients $\boldsymbol{\beta}_\ell$ are updated using (4) and (9). The updated coefficients are then applied to generate the signals for iteration $j+1$ using the following,

$$\tilde{v}_\ell[n_{j+1}] = \boldsymbol{\phi}(n_{j+1})^T\boldsymbol{\beta}_\ell^{(j+1)}, \tag{11}$$

and,

$$\tilde{z}_k[n_{j+1}] = \tilde{x}_k[n_{j+1}] + \boldsymbol{\psi}_k(n_{j+1})^T\boldsymbol{\alpha}_k^{(j+1)}. \tag{12}$$

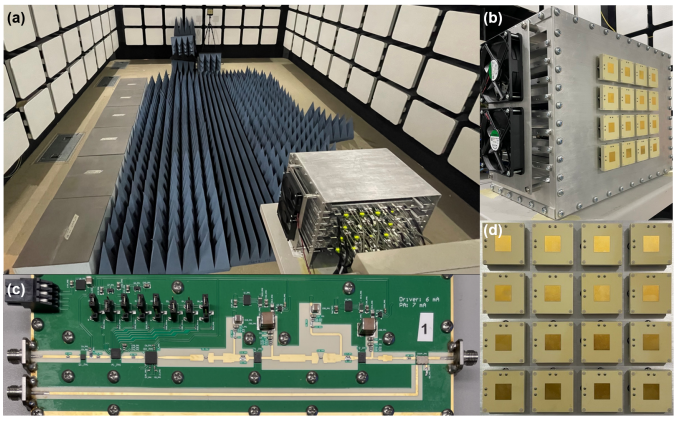


Fig. 2. (a) Full 4×4 mMIMO transmitter front-end inside a $4\times5\times5$ m^3 anechoic chamber, (b) front-view of the transmitter front-end, (c) multistage PA, (d) 4×4 stacked patch antenna array.

III. Experimental Validation

Fig. 2 shows the experimental setup. An AMD ZCU216 RFSoC is used to generate the 16 baseband waveforms feeding the MIMO array and to perform DAC conversion and RF upconversion. In this experimental validation, maximum-ratio transmission (MRT) precoding is adopted for two line-of-sight (LoS) users located at $(\theta_1, \phi_1) = (15°, 0°)$ and $(\theta_2, \phi_2) = (15°, 90°)$. To limit the PAPR at the inputs of the DPD and the power amplifiers, a clipping-and-filtering crest-factor reduction stage is applied after the precoder and before the DPD module. The resulting 16-channel waveforms are uploaded to the RFSoC for playback. Each antenna branch is magnitude- and phase-calibrated. This calibration is iteratively refined to ensure that, after predistortion and pre-equalization, all K RF chains maintain the correct relative gain and phase required for proper array operation. PA outputs are sampled through directional couplers and routed back to the RFSoC receivers for closed-loop coefficient updates.

The PAs under test are custom-designed multistage units consisting of an integrated pre-driver followed by a two-stage GaN final amplifier. The pre-driver provides approximately 25 dB gain over $2.7-3.8$ GHz. The final stage, based on two 6-W Wolfspeed CGHV1F006S devices, provides $25-28$ dB small-signal gain with $1.4-3$ dB gain compression at $P_{\text{sat}} = 38$ dBm. The array employs stacked-patch antennas with better than -20 dB return loss and worst-case mutual coupling of -24 dB at 3.5 GHz. For DPD, each transmit chain uses a SISO complexity-reduced Volterra (CRV) model [6] with 13 coefficients. The model employs nonlinearity order 5, nonlinear memory depth 3, and no linear memory. The linear pre-precoder equalizer uses 11 taps, resulting in a total of 230 coefficients for the entire array. For comparison, a full MIMO-DPD implementation using the same nonlinear order and memory depth would require approximately 1.7×10^5 coefficients to linearize the entire array.

Fig. 3 summarizes the measured root normalized mean square error (RNMSE) and ACPR performance before and after applying the proposed linearization method. The top panel reports the RNMSE at the PA outputs and at the two user locations. At the PA level, the RNMSE ranges from 30% (worst branch) to 10% (best branch) and remains essentially

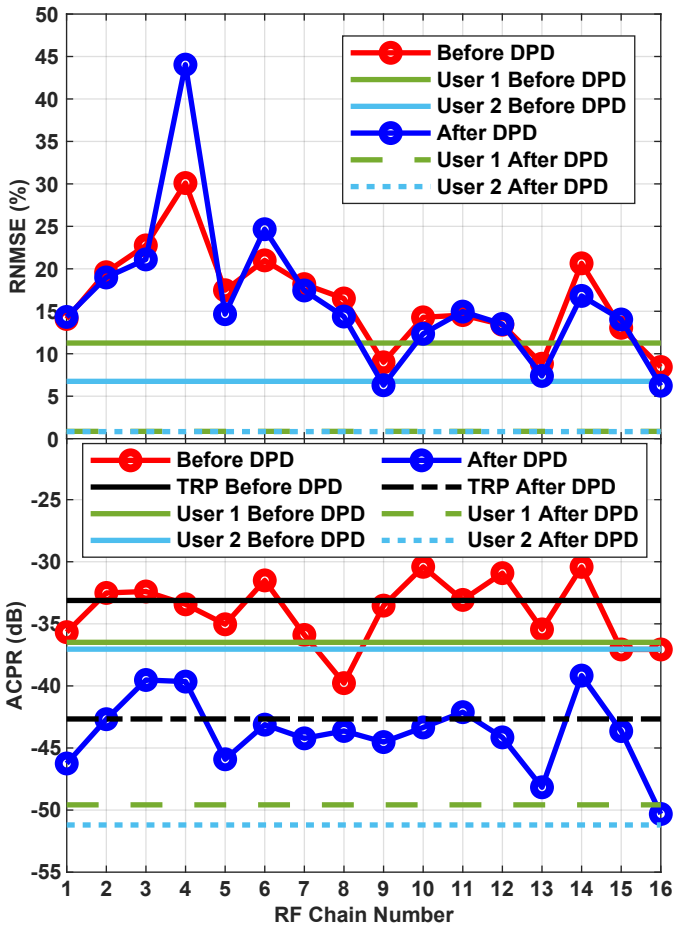


Fig. 3. Measurement results before and after 11-coefficient pre-equalizer per user and 13-coefficient SISO DPD per RF chain.

unchanged after linearization, as expected given the use of an in-band notch during DPD training. In contrast, the RNMSE observed at the user locations improves substantially, decreasing from approximately 11% and 6.8% down to 0.8% for both users—corresponding to an EVM of about 0.7%. Although no in-band correction occurs directly at the PA outputs, the combination of the pre-equalizer and the precoder restores linearity at the user side after spatial combining.

The bottom panel of Fig. 3 reports the ACPR performance measured at the PA outputs, the TRP-ACPR, and the ACPR at the user locations. At the PA level, the ACPR lies between -30 dB and -40 dB before linearization and improves by about 10 dB after applying the proposed method. The TRP-ACPR improves from -33.1 dB to -43 dB. The user ACPR exhibits the strongest enhancement: for user 1 it improves from -36.5 dB to -49.6 dB, and for user 2 from -37.0 dB to -51.2 dB. In both cases, the user ACPR remains lower than the TRP-ACPR by 3-4 dB before linearization and by 6-8 dB after linearization, reflecting the non-coherent combination of distortion across the RF chains, which leads to partial cancellation of out-of-band distortion in the user directions.

Fig. 4 shows the measured power spectral density of the user signals before and after applying the proposed two-

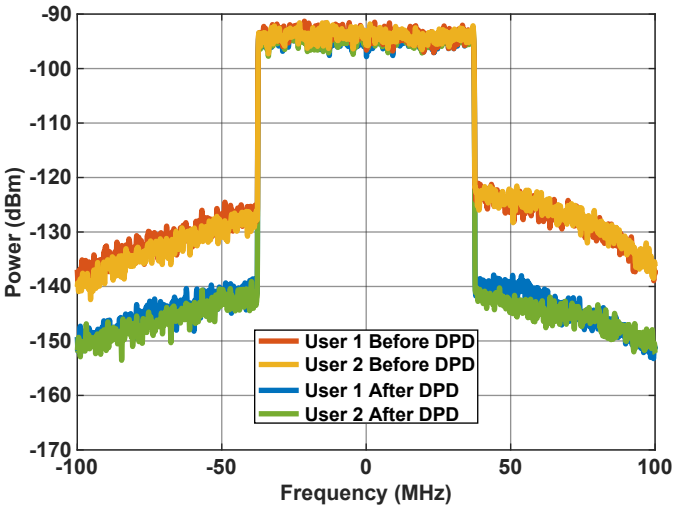


Fig. 4. Measured PSD of the user before and after the proposed two-tier linearization.

tier linearization method. The figure demonstrates that the proposed approach fully suppresses the distortion observed at the user side and confirm the capability of the proposed two-tier method to deliver clean, distortion-free user signals while simultaneously satisfying ACPR and EVM requirements.

IV. Conclusion

A two-tier, low-complexity linearization architecture for precoded MIMO transmitters was presented, targeting simultaneous TRP–ACPR compliance at the transmitter and stringent EVM performance at the users. The method combines a per-branch SISO DPD, trained using a notch-filtered feedback signal to focus exclusively on out-of-band distortion, with a low-order linear MIMO pre-equalizer placed ahead of the precoder to correct residual in-band distortion. Experimental validation on a 16-channel transmitter serving two users at 3.5 GHz demonstrated that this architecture reduces user EVM to approximately 0.7%, while a 13-coefficient SISO DPD per RF chain improves TRP–ACPR from -33.1 dB to -43 dB. In addition, the user ACPR improves from roughly -36.5 dB to -49.6 dB for user 1 and from -37.0 dB to -51.2 dB for user 2. These results confirm that excellent spectral and in-band linearity can be achieved without multi-input or high-order nonlinear DPD models and without increasing precoder complexity, making the proposed approach highly attractive for digital mMIMO transmitters.

References

[1] H. Yu, E. Traore, M. Almoneer, J. G. Lim, J. Xia, and S. Boumaiza, "Experimental study of the effects of antenna crosstalk on the linearity and efficiency of 5G sub-6 GHz wideband 2 × 2 transmitter arrays," in *2021 IEEE MTT-S Int Microw. Symp. (IMS)*, 2021, pp. 869–872.

[2] S. A. Bassam, M. Helaoui, and F. M. Ghannouchi, "Crossover digital predistorter for the compensation of crosstalk and nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 5, pp. 1119–1128, 2009.

[3] P. Jaraut, M. Rawat, and F. M. Ghannouchi, "Composite neural network digital predistortion model for joint mitigation of crosstalk, I/Q imbalance, nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5011–5020, 2018.

[4] K. Hausmair, P. N. Landin, U. Gustavsson, C. Fager, and T. Eriksson, "Digital predistortion for multi-antenna transmitters affected by antenna crosstalk," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 3, pp. 1524–1535, 2017.

[5] A. Brihuega, L. Anttila, and M. Valkama, "Beam-level frequency-domain digital predistortion for OFDM massive MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, pp. 1–16, 2022.

[6] F. Mkadem, M. C. Fares, S. Boumaiza, and J. Wood, "Complexity-reduced Volterra series model for power amplifier digital predistortion," *Analog Integr. Circuits Signal Process.*, vol. 79, no. 2, p. 331–343, 2014.

Appendix E

On the Viability of Using a Subset of Transmitter-Observation Receivers for Training a Common DPD in Fully Digital MIMO Transmitters

Jin Gyu Lim, *Graduate Student Member, IEEE,* Hoda Barkhordar-pour, *Graduate Student Member, IEEE,* Ahmed Ben Ayed, *Graduate Student Member, IEEE,* Patrick Mitran, *Senior Member, IEEE,* and Slim Boumaiza, *Senior Member, IEEE*

***Abstract*—This paper lays the foundation for reducing the required number of transmitter-observation receivers (TORs) for training DPD in fully digital massive multiple-input, multiple-output (MIMO) transmitters. Specifically, it investigates the viability of applying the same, common set of digital predistortion (DPD) coefficients to linearize all RF chains in fully digital massive MIMO transmitters. First, it is shown that if all RF chains are operated at the same output power, the common set of DPD coefficients can be found by simply averaging the coefficients obtained by training each RF chain on its own. This suggests that only a few chains may be needed for training provided the chains are a representative sample. Experimental results are then conducted where 1 and 3 chains are used for training. It is found that when training for 1 chain, significant variations in normalized mean square error (NMSE) and adjacent channel power ratio (ACPR) of up to 9 dB across the chains are realized. For training with 3 chains, the common set of DPD coefficients can reduce the variation to 1 to 2 dB. Finally, after over-the-air (OTA) combining, excellent linearization performance is found for 3 chains.**

***Index Terms*—massive MIMO transmitter, common digital predistortion, power amplifiers.**

I. Introduction

Digital pre-distortion (DPD) is a mature technique for linearizing power amplifiers (PA) in single-input, single-output (SISO) transmitters. However, the development of massive multiple-input, multiple-output (MIMO) transmitter systems poses significant new challenges including the linearization of an array of PAs. For a DPD solution to be viable for fully digital massive MIMO transmitters, key design criteria must be met. This includes the design, training method, and implementation of DPD engine (e.g., SISO and dual-input single-output (DISO), choice of basis functions, etc) which are particularly important given the possibly significant computational burden and power overhead due to the large number of PAs. In addition, limiting the number of transmitter-observation receivers (TORs) in the transmitter array is also critical as it is not practical to dedicate one for each PA.

Manuscript received 27 February 2023; revised 8 April 2023; accepted 10 April 2023. The authors would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Ontario Research Fund - Research Excellence (ORF-RE), and Ericsson Canada Inc.

Jin Gyu Lim and Hoda Barkhordar-pour contributed equally to this work. The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (email: j29lim@uwaterloo.ca).

This article was presented at the IEEE MTT-S International Microwave Symposium (IMS 2023), San Diego, CA, USA, June 11–16, 2023.

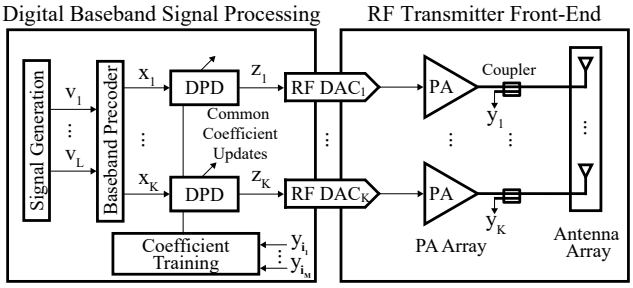


Fig. 1. Block diagram of proposed fully digital MIMO transmitter. L is the number of users, K the number of RF chains, and a subset of M ($\leq K$) TORs are used to train the DPD.

Previous research on linearizing fully digital beamforming arrays has focused on the effects of load modulation, introduced by antenna cross-coupling in the absence of isolators after the PAs [1]–[3]. Specifically, this load modulation makes linearizing the PAs with conventional SISO DPD models difficult. Recent developments in neural network-based DPD [4], [5] could be utilized to improve PA modeling under load modulation. In [6], [7], it was proposed to employ a crosstalk and mismatch module (CTMM) to estimate the reflected wave from each antenna. The estimated wave going back into each PA then served as the second input to a DISO DPD model. However, a TOR was still required for each PA to acquire the output of each PA to train the DISO DPDs.

In [8], the feasibility of reducing the number of TORs in a hybrid beamforming configuration was investigated. Specifically, only one TOR was used to receive a feedback signal from one PA and only one set of DPD coefficients was then trained from the feedback signal. These were subsequently applied to all PAs in the system. The results showed that using a single TOR does not optimally linearize all PAs, resulting in 8 to 9 dB performance variation in adjacent channel power ratio (ACPR) over the different PAs.

In this paper, we study the feasibility of using a subset of $M \leq K$ TORs (where K is the number of chains) in fully digital massive MIMO transmitters to train one common set of DPD coefficients that will be applied to all RF chains. We first develop the DPD training theory that operates with $M \leq K$ TORs to obtain one set of common DPD coefficients. The theory is then validated experimentally on $K = 8$

identical multistage PAs. Specifically, we consider the case of $M = 1$ and $M = 3$ TORs with 100 MHz, 8 dB peak-to-average power ratio (PAPR) orthogonal frequency-division multiplexing (OFDM) signals. We find that for $M = 3$ TORs, one common set of DPD coefficients can provide consistent and tolerable performances within 1 to 2 dB variations on all eight RF chains while for $M = 1$, the variation is up to 9 dB (similar to the findings in [8]). Finally, we find that after OTA combining, the performance obtained by $M = 3$ is close to the case when each PA's DPD is trained by its dedicated TOR.

II. Formulation of A Common DPD Training

Fig. 1 shows the block diagram of a fully digital K-chain MIMO transmitter. Let the sampled complex baseband equivalent (CBBE) input and output signals for the kth RF chain be $x_k[n]$ and $y_k[n]$, respectively. We assume that the DPD model has P basis functions $\psi_1(.), \psi_2(.), \ldots, \psi_P(.)$ so that for the kth chain

$$z_k[n] = \sum_{p=1}^{P} \alpha_{k,p} \psi_p(\mathbf{x}_k[n]) \tag{1}$$

where $\mathbf{x}_k[n] = (x_k[n], \ldots, x_k[n-N])^T$, N is the memory depth, and $\alpha_{k,p}$ is the pth coefficient of the kth chain. A traditional approach to finding the coefficients is by indirect learning, i.e., letting $\boldsymbol{\alpha}_k = [\alpha_{k,1}, \ldots, \alpha_{k,P}]^T$, then

$$\boldsymbol{\alpha_k} = (\Psi_k^H \Psi_k)^{-1} \Psi_k \bar{\mathbf{x}}_k, \tag{2}$$

where $\bar{\mathbf{x}}_k = (x_k[n], \ldots, x_k[n-Q])^T$,

$$\Psi_k = \begin{pmatrix} \psi_1(\mathbf{y}_k[n]) & \cdots & \psi_P(\mathbf{y}_k[n]) \\ \psi_1(\mathbf{y}_k[n-1]) & \cdots & \psi_P(\mathbf{y}_k[n-1]) \\ \vdots & \vdots & \vdots \\ \psi_1(\mathbf{y}_k[n-Q]) & \cdots & \psi_P(\mathbf{y}_k[n-Q]) \end{pmatrix}, \tag{3}$$

$\mathbf{y}_k[n] = (y_k[n], \ldots, y_k[n-N])^T$ and Q denotes the memory depth. In this case, the DPD coefficients are obtained by training the DPD of chain k based on the outputs of chain k only.

Now, consider the case that we wish to find one set of DPD model coefficients $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_P]^T$ to serve as a common set for chains $i_1, i_2, \ldots, i_M \in \{1, \ldots, K\}$. Specifically, in such a case,

$$z_{i_1}[n] = \sum\nolimits_p \alpha_p \psi_p(\mathbf{x}_{i_1}[n]) \tag{4}$$

$$\vdots$$

$$z_{i_M}[n] = \sum\nolimits_p \alpha_p \psi_p(\mathbf{x}_{i_M}[n]). \tag{5}$$

Following the indirect learning approach, the set of coefficients $\boldsymbol{\alpha}$ should be chosen as the least squares solution to

$$x_{i_m}[\ell] = \sum\nolimits_p \alpha_p \psi(\mathbf{y}_{i_m}[\ell]) \tag{6}$$

for all $\ell = n-N, \ldots, n$ and all $m = 1, \ldots, M$. In matrix form this is

$$\begin{pmatrix} \bar{\mathbf{x}}_{i_1} \\ \vdots \\ \bar{\mathbf{x}}_{i_M} \end{pmatrix} = \begin{pmatrix} \Psi_{i_1} \\ \vdots \\ \Psi_{i_M} \end{pmatrix} \boldsymbol{\alpha} \tag{7}$$

The least-squares solution is then

$$\boldsymbol{\alpha} = \left(\sum\nolimits_{m=1}^{M} \Psi_{i_m}^H \Psi_{i_m} \right)^{-1} \left(\sum\nolimits_{m'=1}^{M} \Psi_{i_{m'}}^H \bar{\mathbf{x}}_{i_{m'}} \right) \tag{8}$$

The term $(\Psi_{i_m}^H \Psi_{i_m})$ represents the correlation matrix of the i_m-th RF chain's basis matrix, and is directly dependent on the PA output power. As the DPD coefficients are therefore dependent on the PA output power, one set of DPD coefficients for arbitrary PA output power across the M RF chains will likely not perform well for all chains. Consequently, we assume that the average output power of each of the M PAs is the same.

Now, represent the output signals of the RF chains by

$$y_{i_m}[n] = y[n] + e_{i_m}[n], \qquad \text{for } 1 \le m \le M \tag{9}$$

where the nominal signal is $y[n] = \frac{1}{M} \sum_{m=1}^{M} y_{i_m}[n]$ and $e_{i_m}[n]$ is the residual error signal. It is assumed that the powers of the residual error signals are small enough to be negligible. This is experimentally validated in Section III. Thus, the correlation matrix for each of the M RF chains can be approximated by their average as follows:

$$\Psi_{i_{m'}}^H \Psi_{i_{m'}} \approx \frac{1}{M} \sum\nolimits_{m=1}^{M} \Psi_{i_m}^H \Psi_{i_m} \tag{10}$$

Taking (10) with equality and substituting into (2), we obtain

$$\boldsymbol{\alpha}_{i_{m'}} = \left(\frac{1}{M} \sum\nolimits_{m=1}^{M} \Psi_{i_m}^H \Psi_{i_m} \right)^{-1} \Psi_{i_{m'}} \bar{\mathbf{x}}_{i_{m'}} \tag{11}$$

Re-arranging (8) and then substituting (11) into the result,

$$\begin{aligned} \boldsymbol{\alpha} &= \frac{1}{M} \sum\nolimits_{m'=1}^{M} \left(\left(\frac{1}{M} \sum\nolimits_{m=1}^{M} \Psi_{i_m}^H \Psi_{i_m} \right)^{-1} \Psi_{i_{m'}}^H \bar{\mathbf{x}}_{i_{m'}} \right) \\ &= \frac{1}{M} \sum\nolimits_{m'=1}^{M} \boldsymbol{\alpha}_{i_{m'}} \end{aligned} \tag{12}$$

Eq. (12) shows that given the same average output power for the M PAs $i_1, i_2, \ldots i_M$, averaging individually trained DPD coefficients $\boldsymbol{\alpha}_{i_1}, \boldsymbol{\alpha}_{i_2}, \ldots \boldsymbol{\alpha}_{i_M}$, is equivalent to obtaining a common set of coefficients $\boldsymbol{\alpha}$ obtained by training on the output signals from all the M TORs.

It is to note that one need not find a common set of DPD coefficients $\boldsymbol{\alpha}$ by averaging over all K DPDs corresponding to the K PAs. Since the common set of coefficients $\boldsymbol{\alpha}$ is obtained by averaging the $\boldsymbol{\alpha}_k$ sets, it may be accurately estimated by averaging over a much smaller sample size M compared to K, provided this smaller sample size is representative. This observation can significantly reduce the complexity of the TORs for fully digital MIMO transmitters as the number of channels increases. In the next section, it will be seen that $M = 3$ is sufficient for an array comprised of $K = 8$ chains.

III. Experimental Validation

A. Measurement Setup

Fig. 2 shows the block diagram of the measurement setup used in the validation experiments. An arbitrary waveform generator (AWG; Keysight M8190A) is used to synthesize a 100 MHz, 8 dB PAPR OFDM signal that is centered at 3.5 GHz. The test signal is then fed to a balun (Marki

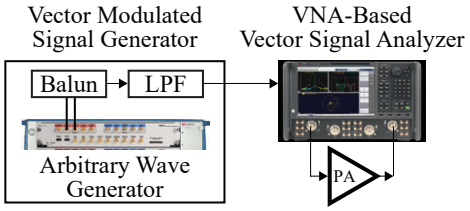


Fig. 2. Modulated-Signal-Based PA characterization setup.

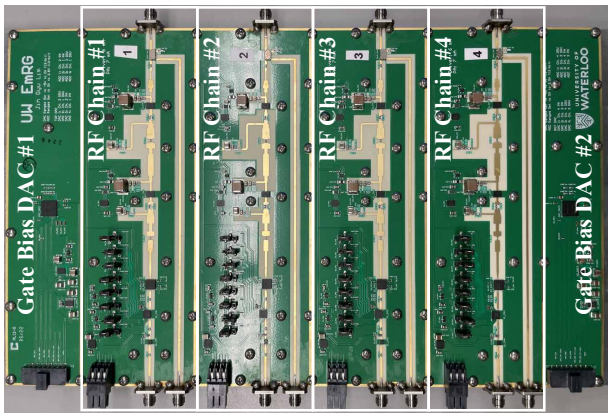


Fig. 3. Instance of one fabricated RF front-end panel comprised of four chains used for experiments.

BALH-0010), and then to a low-pass filter (LPF; Mini Circuits VLF-5500+) before feeding the RF chains. The sampled output signals of the RF chains are then fed to one of the ports of a vector network analyzer (VNA; Keysight PNA-X N5242B) that serves as a TOR. In the following experiments, to manage the complexity of the PA linearization process, the effects of antenna crosstalk were excluded. The DPD model used in these experiments is the complexity-reduced Volterra model [9].

B. RF Chain

Fig. 3 shows one instance of a 4-channel RF chain used in the experiments. Each RF chain consists of a variable gain amplifier and a two-stage PA. The variable gain amplifier has two fixed-gain pre-drivers and a digitally controlled attenuator. An attenuator (Analog Devices HMC1122) is placed between two pre-drivers (Qorvo QPA9127 & QPA9842) to provide a linear, variable gain of $23-35.5$ dB, allowing the peak input signal power to fall between -19.5 dBm and -7 dBm. The two-stage PA is designed to operate over $3.2-3.8$ GHz with two 6 W transistors (Wolfspeed CGHV1F006S). The driver and main stages of the PA are co-designed to linearize the output performance as much as possible, resulting in the small signal gain of $25-28$ dB and $1.4-3$ dB gain compression at $P_{sat} = 38$ dBm. The peak drain efficiency of the two-stage PA is $\approx 50\%$. For the measurements, the attenuators are tuned to coarsely match the output average power across all eight RF chains within ± 0.5 dB.

C. Measurement Results

Table I shows the ratio of the power of the residual error $e_i[n]$ to the power of its nominal reference signal $y[n]$ before DPD and when all eight chains have equal output power, validating the assumption in Section II that the output correlation matrices are similar.

TABLE I
POWER RATIO OF THE RESIDUAL ERROR SIGNAL $e_i[n]$ TO THE REFERENCE NOMINAL SIGNAL $y[n]$ AS MEASURED ON EIGHT IDENTICAL MULTISTAGE PAS.

	RF Chain Number							
	1	2	3	4	5	6	7	8
Error (dB)	-41.1	-41.6	-40.0	-44.9	-42.1	-41.6	-45.1	-42.7

Two sets of experiments are conducted with different numbers of TORs: one is for $M = 1$ and the other for $M = 3$. When $M = 1$, each RF chain is trained individually and its DPD coefficients are applied to all eight chains. For $M = 3$, a random subset of three RF chains is chosen and their averaged DPD coefficients are applied to all eight RF chains. Normalized mean square error (NMSE) and adjacent-channel power ratio (ACPR) measurements before and after applying DPD are shown in Fig. 4. Lower and upper whiskers represent minimum and maximum values, respectively, while the edges of the boxes represent lower and upper quartiles. The lines inside the boxes indicate the medians. The x-axis labels represent the RF chains used for training the DPD. When this number is between 1 and 8, this is the single RF chain on which the DPD was trained. When this is a triple (a, b, c), the DPD coefficients were obtained by averaging the DPD coefficients from training on chains a, b and c.

For $M = 1$, the minimum parameter values are always obtained with each RF chain's own DPD coefficients. RF chain 2 has the lowest minimum NMSE and ACPR values but 7 dB and 9 dB variations are observed in NMSE and ACPR, respectively. This is due to RF chain 2 being the most linear device so its set of DPD coefficients struggles to correct other devices' worse non-linearity. Using DPD coefficients of average-performing RF chains, such as RF chains 1 and 7, can reduce the variations to 3 to 4 dB. For $M = 1$, variations not only exist across all RF chains for a given set of coefficients but also between different sets of coefficients. Thus, $M = 1$ cannot guarantee consistent transmitter performance.

For $M = 3$, four random sets of three RF chains are selected and common DPD coefficients for each set are calculated using (12) and then applied to all eight RF chains. The worst set $(3, 7, 8)$ has a similar variation to that of average-performing RF chain indexes for $M = 1$. Sets $(2, 3, 7)$ and $(3, 4, 5)$ show excellent results with only 1 to 2 dB of variation. The results indicate that by only sampling a few RF chains, a common set of DPD coefficients can be trained to linearize all RF chains in fully digital MIMO transmitters with small performance variations.

While the performance at the transmitter output is important, the quality of the signal in the far-field matters most. The signal is a vector-weighted summation of all eight channel outputs. Fig. 5 compares power spectral density (PSD) of the far-field OTA signal when i) each chain has its individual DPD coefficients, ii) the DPD coefficients are obtained by linearizing RF chain 2, and iii) the four cases of $M = 3$ are considered. The common set of DPD coefficients found with $M = 3$ provides comparable signal quality to the case

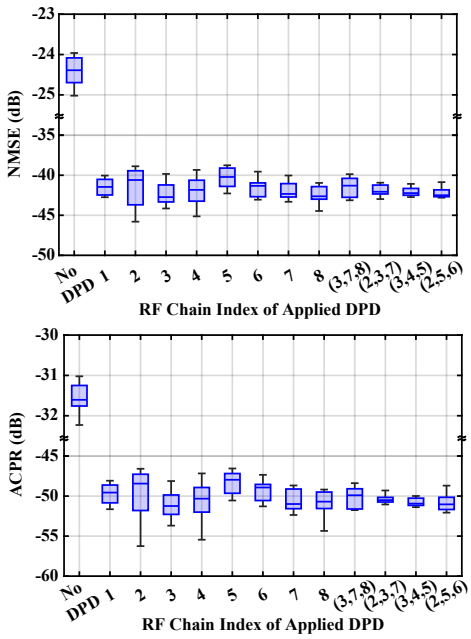


Fig. 4. NMSE with and without DPD (top), ACPR with and without DPD (bottom). The x-axis indicates which DPD coefficient set was applied to all eight chains.

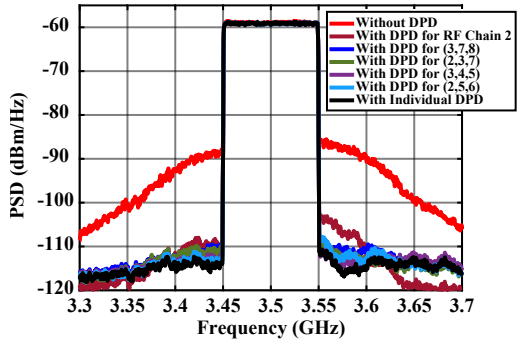


Fig. 5. PSD comparison at an estimated over-the-air boresight receiver after applying different DPD coefficients on all eight RF chains.

where individual DPD coefficients are used.

IV. Conclusion

This paper investigated a method to reduce the required number of transmitter-observation receivers (TORs) for training DPD in fully digital massive MIMO transmitters. Specifically, it investigated applying one set of common DPD coefficients, trained using $M = 1$ and $M = 3$ TORs, to eight RF chains. First, it was shown that if all RF chains are operated at the same output power, the common DPD coefficients can be calculated by simply averaging the individually trained DPD coefficients of the M selected RF chains. Experimental results found that, for $M = 1$, inconsistent performance with variations up to 9 dB for NMSE and ACPR are observed. For $M = 3$, the common DPD coefficients constrain performance variations across all eight RF chains to 1 to 2 dB. These results show significant promise in using only a small subset of RF chains for DPD training. The generality of the proposed method to antenna arrays with non-negligible crosstalk will be a subject of future research investigation.

References

[1] C. Fager, X. Bland, K. Hausmair, J. C. Cahuana, and T. Eriksson, "Prediction of smart antenna transmitter characteristics using a new behavioral modeling approach," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Tampa, FL, USA, Jun. 2014, pp. 1–4.

[2] K. Hausmair, S. Gustafsson, C. Sánchez-Pérez, P. N. Landin, U. Gustavsson, T. Eriksson, and C. Fager, "Prediction of nonlinear distortion in wideband active antenna arrays," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 11, pp. 4550–4563, Nov. 2017.

[3] F. M. Barradas, P. M. Tomé, J. M. Gomes, T. R. Cunha, P. M. Cabral, and J. C. Pedro, "Power, linearity, and efficiency prediction for MIMO arrays with antenna coupling," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 12, pp. 5284–5297, Dec. 2017.

[4] T. Kobal, Y. Li, X. Wang, and A. Zhu, "Digital predistortion of RF power amplifiers with phase-gated recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 6, pp. 3291–3299, 2022.

[5] T. Kobal and A. Zhu, "Digital predistortion of RF power amplifiers with decomposed vector rotation-based recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4900–4909, 2022.

[6] Q. Luo, X.-W. Zhu, C. Yu, and W. Hong, "Single-receiver over-the-air digital predistortion for massive MIMO transmitters with antenna crosstalk," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 1, pp. 301–315, 2020.

[7] K. Hausmair, P. N. Landin, U. Gustavsson, C. Fager, and T. Eriksson, "Digital predistortion for multi-antenna transmitters affected by antenna crosstalk," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 3, pp. 1524–1535, 2018.

[8] L. Liu, W. Chen, L. Ma, and H. Sun, "Single-PA-feedback digital predistortion for beamforming MIMO transmitter," in *Proc. IEEE Int. Conf. Microw. Millim. Wave Technol. (ICMMT)*, vol. 2, 2016, pp. 573–575.

[9] F. Mkadem, M. C. Fares, S. Boumaiza, and J. Wood, "Complexity-reduced Volterra series model for power amplifier digital predistortion," *Analog Integr. Circ. Sig. Process.*, vol. 79, no. 2, p. 331–343, 2014.

Appendix F

Digital Control Source Code

Listing F.1: Class for Analog Devices HMC1122 Attenuator

```
classdef HMC1122

    properties
       attenValue;
    end

    methods
        function obj = HMC1122_Atten(obj, attenVal) %attenVal in dB

        % HMC 1122 is a 6-bit device, 000000: 31.5 dB, 111111:0 dB, 0.5 dB step,
        % [0-63], XOR desired attenuation value with 63 using bitxor(attVal, 63)
        % P/S pin must be set to high for serial control mode
        % For power-up attenuation value, D0-D5 should be grounded such that
        % regardless of LE signal value, the start-up attenuation will be 31.5 dB

            if attenVal >= 0 && attenVal <= 31.5
                attenValSPI = bitxor(ceil(attenVal*2), 63);
            else
                disp('Out of range: Attenuation value should be between 0 and 31.5 with
                    0.5 step');
                disp('Setting it to 31.5 dB');
                attenValSPI = 0;
            end
            obj.attenValue = attenValSPI;
        end

        function currentAttenValue = getCurrentAttenValue(obj)
            currentAttenValue = obj.attenValue;
        end

    end
end
```

Listing F.2: Class for Texas Instruments AMC7834 for Transistor Gate Bias Control

```
classdef AMC7834

    properties % User Defined Properties

        PowerModeAddress       = '02';
        AMCConfig0Address      = '10';
        ADCMUXAddress          = '12';
        DACRangeAddress        = '16';
        DACClampConfigAddress  = '17'; % Bit 0 & 1, value 1: clamp on, value 0: clamp off
        DACADCTriggerAddress   = '1C';
        softwareResetAddress   = '1D';
        AMCStatusAddress       = '1F';
        ADCInternalAddress     = ['20', '21', '22', '23']; % Bipolar DAC measurements
        ADCExternalAddress     = ['24', '25', '26', '27']; % Drain current measurement
        internalTempAddress    = '2D';
        DACDataAddresses       = ['30', '31', '32', '33'];

    end
    methods
        % Constructer of this class
        function obj = AMC7834(varargin)

        end

        % Initialization step #1
        function Data = setSoftwareReset(obj)
            % NI USB-8452 accepts decimal values for digital comm.
            % This function performs software reset
            % first byte: reg. addr
            % second & third bytes: reset bytes are 00
            Data = [hex2dec(obj.softwareResetAddress) 0 12];
        end

        % Initialization step #2
        function Data = setDACRanges(obj)
            % NI USB-8452 accepts decimal values for digital comm.
            % This function sets bipolar DAC ranges
            % first byte: reg. addr
            % second byte: reserved to be 0
            % third byte: either '0101' (5) or '1010' (10) sets the ranges
            % to -5 V to 0 V
            Data = [hex2dec(obj.DACRangeAddress) 0 5];
        end

        % Initialization step #3
        function Data = setADCMUX(obj)
            % NI USB-8452 accepts decimal values for digital comm.
            % This function activates ADC channels for reading
            % first byte: reg. addr
            % second byte: activate local temp. sensor only ('0100 0000')
            % third byte: activate all external and internal ADCs
            % ('1111 1111'). External for drain currents and Internal for
            % DAC value readings
            Data = [hex2dec(obj.ADCMUXAddress) 64 255];
        end
```

```
56
57          % Initialization step #4
58          function Data = setAMCConfig0(obj)
59              % NI USB-8452 accepts decimal values for digital comm.
60              % This function configures ADC mode to direct-mode. This mode
61              % converts the signals after a trigger and stays ready.
62              % first byte: reg. addr
63              % second byte: activate direct-mode ADC and keep other fields
64              % unchanged ('0001 0011')
65              % third byte: 0 by default
66              Data = [hex2dec(obj.AMCConfig0Address) 19 0];
67          end
68
69          % Initialization step #5
70          function Data = setPowerModeON(obj)
71              % NI USB-8452 accepts decimal values for digital comm.
72              % This function activates ADC and DAC
73              % first byte: reg. addr
74              % second byte: reserved to be 0
75              % third byte: 2 ('10') for activation
76              Data = [hex2dec(obj.PowerModeAddress) 0 2];
77          end
78
79          % Release DAC clamps for gate voltage biasing
80          function Data = releaseDACClamp(obj)
81              % NI USB-8452 accepts decimal values for digital comm.
82              % This function either clamps or releases DAC outputs.
83              % first byte: reg. addr
84              % second byte: reserved to be 0
85              % third byte: Only activate DAC12 (Bit 0) & DAC34 (Bit 1) while
86              % keeping other DACs clamped. Value 1: Clamp, Value 0: Release
87              % ('0011 1100')
88              Data = [hex2dec(obj.DACClampConfigAddress) 0 hex2dec('3C')];
89          end
90
91          function Data = checkADCReady(obj)
92              % 2^7 adds Read command to the Reg. Addr byte
93              Data = [hex2dec(obj.AMCStatusAddress)+(2^7) 0 0];
94          end
95
96          function Data = setADCtrigger(obj)
97              % 0x1C ADC trigger (needs to be ADC ready)
98              Data = [hex2dec(obj.DACADCTriggerAddress) 0 1];
99          end
100
101         function Data = getDAC1(obj)
102             % 2^7 adds Read command to the Reg. Addr byte
103             address = uint8 (hex2dec(obj.DACDataAddresses(1:2))+(2^7));
104             Data = [address 0 0];
105         end
106
107         function Data = getDAC2(obj)
108             % 2^7 adds Read command to the Reg. Addr byte
109             Data = [hex2dec(obj.DACDataAddresses(3:4))+(2^7) 0 0];
110         end
111
112         function Data = getDAC3(obj)
```

```
113             % 2^7 adds Read command to the Reg. Addr byte
114             Data = [hex2dec(obj.DACDataAddresses(5:6))+(2^7) 0 0];
115         end
116
117         function Data = getDAC4(obj)
118             % 2^7 adds Read command to the Reg. Addr byte
119             Data = [hex2dec(obj.DACDataAddresses(7:8))+(2^7) 0 0];
120         end
121
122         function Data = setDAC1(obj, Value)
123             if Value >= -5 && Value <= 0
124                 DACvalue = floor((Value+5)/4.9988*4095);
125                 DACvalue = dec2bin(DACvalue,12);
126                 DACvalue = strcat('0000', DACvalue);
127
128                 Data = [hex2dec(obj.DACDataAddresses(1:2)) bin2dec(DACvalue(1:8)) bin2dec(
                        DACvalue(9:16))];
129             else
130                 disp('Out of range: DAC value should be between -5 V and -0.00122 V with
                        1.22 mV step');
131                 disp('Please set it again');
132             end
133         end
134
135         function Data = setDAC2(obj, Value)
136             if Value >= -5 && Value <= 0
137                 DACvalue = floor((Value+5)/4.9988*4095);
138                 DACvalue = dec2bin(DACvalue,12);
139                 DACvalue = strcat('0000', DACvalue);
140
141                 Data = [hex2dec(obj.DACDataAddresses(3:4)) bin2dec(DACvalue(1:8)) bin2dec(
                        DACvalue(9:16))];
142             else
143                 disp('Out of range: DAC value should be between -5 V and -0.00122 V with
                        1.22 mV step');
144                 disp('Please set it again');
145             end
146         end
147
148         function Data = setDAC3(obj, Value)
149             if Value >= -5 && Value <= 0
150                 DACvalue = floor((Value+5)/4.9988*4095);
151                 DACvalue = dec2bin(DACvalue,12);
152                 DACvalue = strcat('0000', DACvalue);
153
154                 Data = [hex2dec(obj.DACDataAddresses(5:6)) bin2dec(DACvalue(1:8)) bin2dec(
                        DACvalue(9:16))];
155             else
156                 disp('Out of range: DAC value should be between -5 V and -0.00122 V with
                        1.22 mV step');
157                 disp('Please set it again');
158             end
159         end
160
161         function Data = setDAC4(obj, Value)
162             if Value >= -5 && Value <= 0
163                 DACvalue = floor((Value+5)/4.9988*4095);
```

```
164                 DACvalue = dec2bin(DACvalue,12);
165                 DACvalue = strcat('0000', DACvalue);
166
167                 Data = [hex2dec(obj.DACDataAddresses(7:8)) bin2dec(DACvalue(1:8)) bin2dec(
                        DACvalue(9:16))];
168             else
169                 disp('Out of range: DAC value should be between -5 V and -0.00122 V with
                        1.22 mV step');
170                 disp('Please set it again');
171             end
172         end
173
174
175         function Data = getADCInt1(obj)
176             % 2^7 adds Read command to the Reg. Addr byte
177             Data = [hex2dec(obj.ADCInternalAddress(1:2))+(2^7) 0 0];
178         end
179
180         function Data = getADCInt2(obj)
181             % 2^7 adds Read command to the Reg. Addr byte
182             Data = [hex2dec(obj.ADCInternalAddress(3:4))+(2^7) 0 0];
183         end
184
185         function Data = getADCInt3(obj)
186             % 2^7 adds Read command to the Reg. Addr byte
187             Data = [hex2dec(obj.ADCInternalAddress(5:6))+(2^7) 0 0];
188         end
189
190         function Data = getADCInt4(obj)
191             % 2^7 adds Read command to the Reg. Addr byte
192             Data = [hex2dec(obj.ADCInternalAddress(7:8))+(2^7) 0 0];
193         end
194
195         function Data = getADCExt1(obj)
196             % 2^7 adds Read command to the Reg. Addr byte
197             Data = [hex2dec(obj.ADCExternalAddress(1:2))+(2^7) 0 0];
198         end
199
200         function Data = getADCExt2(obj)
201             % 2^7 adds Read command to the Reg. Addr byte
202             Data = [hex2dec(obj.ADCExternalAddress(3:4))+(2^7) 0 0];
203         end
204
205         function Data = getADCExt3(obj)
206             % 2^7 adds Read command to the Reg. Addr byte
207             Data = [hex2dec(obj.ADCExternalAddress(5:6))+(2^7) 0 0];
208         end
209
210         function Data = getADCExt4(obj)
211             % 2^7 adds Read command to the Reg. Addr byte
212             Data = [hex2dec(obj.ADCExternalAddress(7:8))+(2^7) 0 0];
213         end
214
215         function Data = getLocalTemp(obj)
216             % 2^7 adds Read command to the Reg. Addr byte
217             Data = [hex2dec(obj.internalTempAddress)+(2^7) 0 0];
218         end
```

```
219
220     end
221 end
```

Listing F.3: Class for Controlling One AMC7834 and Two HMC1122

```
1  classdef DigitalControl
2
3      properties
4          digitalInterface;
5          AMC7834;
6          HMC1122_TOP;
7          HMC1122_BOT;
8          SPI_CS;
9          I2C_GPIO_ADDR;
10         AMC7834_CS;
11         HMC1122_TOP_CS;
12         HMC1122_BOT_CS;
13     end
14
15     methods
16         function obj = DigitalControl(varargin)
17             obj.digitalInterface = varargin{1};
18             obj.AMC7834          = varargin{2};
19             obj.HMC1122_TOP      = varargin{3};
20             obj.HMC1122_BOT      = varargin{4};
21             obj.SPI_CS           = varargin{5};
22             obj.I2C_GPIO_ADDR    = varargin{6};
23             obj.AMC7834_CS       = varargin{7};
24             obj.HMC1122_TOP_CS   = varargin{8};
25             obj.HMC1122_BOT_CS   = varargin{9};
26         end
27
28         % AMC7834
29         function obj = AMC7834_init(obj)
30
31             % Software Reset
32
33             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
34             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
35             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setSoftwareReset, 3,
                   24);
36             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
37
38             % Set DAC ranges to -5 V to 0 V
39             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
40             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setDACRanges, 3, 24);
41             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
42
43             % Set ADC MUX, 4 ext, 4 int, 1 local temp
44             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
45             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setADCMUX, 3, 24);
46             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
47
48             % Set AMC Config 0 for direct-mode ADC conversion
49             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
50             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setAMCConfig0, 3, 24);
```

```
51              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
52
53              % Activate ADC and DAC
54              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
55              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setPowerModeON, 3, 24)
                    ;
56              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
57              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
58          end
59
60          function obj = releaseDACClamps(obj)
61              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
62              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
63              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.releaseDACClamp, 3,
                    24);
64              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
65              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
66          end
67
68          function [obj, ReadData]= getDAC1(obj)
69              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
70              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
71              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getDAC1, 3, 24);
72              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
73              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
74          end
75
76          function [obj, ReadData]= getDAC2(obj)
77              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
78              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
79              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getDAC2, 3, 24);
80              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
81              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
82          end
83
84          function [obj, ReadData]= getDAC3(obj)
85              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
86              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
87              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getDAC3, 3, 24);
88              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
89              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
90          end
91
92          function [obj, ReadData]= getDAC4(obj)
93              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
94              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
95              [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getDAC4, 3, 24);
96              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
97              obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
98          end
99
100         function obj = setDAC1(obj, Value)
101             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
102             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
103             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setDAC1(Value), 3, 24)
                    ;
104             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
```

```
105             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
106         end
107
108         function obj = setDAC2(obj, Value)
109             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
110             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
111             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setDAC2(Value), 3, 24)
                    ;
112             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
113             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
114         end
115
116         function obj = setDAC3(obj, Value)
117             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
118             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
119             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setDAC3(Value), 3, 24)
                    ;
120             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
121             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
122         end
123
124         function obj = setDAC4(obj, Value)
125             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
126             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
127             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setDAC4(Value), 3, 24)
                    ;
128             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
129             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
130         end
131
132         function [obj, readyFlag] = checkADCReady(obj)
133             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
134             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
135             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.checkADCReady, 3, 24);
136             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
137             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
138
139             % If readyFlag = 1, ADC ready
140             % If readyFlag = 0, ADC running
141             data = bitand(ReadData(1,2), 64);
142             if (data/64 == 1)
143                 readyFlag = 1;
144             else
145                 readyFlag = 0;
146             end
147
148         end
149
150         function obj = setADCTrigger(obj)
151             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
152             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
153             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.setADCtrigger, 3, 24);
154             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
155             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
156         end
157
158         function obj = readADC(obj)
```

```
159             ready = 0;
160             while (~ready)
161                 [obj, ready] = obj.checkADCReady;
162             end
163             % Trigger ADC read
164             obj = obj.setADCTrigger;
165             while (~ready)
166                 [obj, ready] = obj.checkADCReady;
167             end
168         end
169
170         function [obj, value] = convertSPIDataToDec(obj, data)
171             value = strcat(dec2bin(data(1,2),8), dec2bin(data(1,3),8));
172             value = bin2dec(value);
173         end
174
175         % Measures DAC1 value
176         function [obj, voltage] = getADCInt1(obj)
177             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
178             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
179             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCInt1, 3, 24);
180             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
181             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
182
183             [obj, data] = obj.convertSPIDataToDec(ReadData);
184             voltage = 2.5 + 3*(2.5*data/4096 - 2.5);
185         end
186
187         % Measures DAC2 value
188         function [obj, voltage] = getADCInt2(obj)
189             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
190             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
191             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCInt2, 3, 24);
192             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
193             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
194
195             [obj, data] = obj.convertSPIDataToDec(ReadData);
196             voltage = 2.5 + 3*(2.5*data/4096 - 2.5);
197         end
198
199         % Measures DAC3 value
200         function [obj, voltage] = getADCInt3(obj)
201             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
202             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
203             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCInt3, 3, 24);
204             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
205             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
206
207             [obj, data] = obj.convertSPIDataToDec(ReadData);
208             voltage = 2.5 + 3*(2.5*data/4096 - 2.5);
209         end
210
211         % Measures DAC4 value
212         function [obj, voltage] = getADCInt4(obj)
213             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
214             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
215             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCInt4, 3, 24);
```

```
216             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
217             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
218
219             [obj, data] = obj.convertSPIDataToDec(ReadData);
220             voltage = 2.5 + 3*(2.5*data/4096 - 2.5);
221         end
222
223         % Measures External ADC1 value for Drain current measurement
224         function [obj, current] = getDrainCurrent1(obj)
225             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
226             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
227             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCExt1, 3, 24);
228             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
229             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
230
231             [obj, data] = obj.convertSPIDataToDec(ReadData);
232             voltage = 2.5*data/4096;
233             current = voltage/5*1000; % in mA
234         end
235
236         % Measures External ADC2 value for Drain current measurement
237         function [obj, current] = getDrainCurrent2(obj)
238             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
239             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
240             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCExt2, 3, 24);
241             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
242             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
243
244             [obj, data] = obj.convertSPIDataToDec(ReadData);
245             voltage = 2.5*data/4096;
246             current = voltage/5*1000; % in mA
247         end
248
249         % Measures External ADC3 value for Drain current measurement
250         function [obj, current] = getDrainCurrent3(obj)
251             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
252             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
253             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCExt3, 3, 24);
254             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
255             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
256
257             [obj, data] = obj.convertSPIDataToDec(ReadData);
258             voltage = 2.5*data/4096;
259             current = voltage/5*1000; % in mA
260         end
261
262         % Measures External ADC4 value for Drain current measurement
263         function [obj, current] = getDrainCurrent4(obj)
264             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
265             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
266             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getADCExt4, 3, 24);
267             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
268             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
269
270             [obj, data] = obj.convertSPIDataToDec(ReadData);
271             voltage = 2.5*data/4096;
272             current = voltage/5*1000; % in mA
```

```
273         end
274
275         function [obj, temp] = getChipTemp(obj)
276             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.AMC7834_CS);
277             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.AMC7834_CS);
278             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, obj.AMC7834.getLocalTemp, 3, 24);
279             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
280             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
281
282             [obj, data] = obj.convertSPIDataToDec(ReadData);
283             if data >= 2048
284                 temp = -1*(4096-data)/4;
285             else
286                 temp = data/4;
287             end
288         end
289
290         % HMC1122 Top
291         function obj = setHMC1122Top(obj, attenVal)
292             obj.HMC1122_TOP = obj.HMC1122_TOP.HMC1122_Atten(attenVal);
293
294             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.HMC1122_TOP_CS);
295             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.HMC1122_TOP_CS);
296             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, ...
297                 obj.HMC1122_TOP.getCurrentAttenValue, 1, 6);
298             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
299             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
300         end
301
302         % HMC1122 Bottom
303         function obj = setHMC1122Bot(obj, attenVal)
304             obj.HMC1122_BOT = obj.HMC1122_BOT.HMC1122_Atten(attenVal);
305
306             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, obj.HMC1122_BOT_CS);
307             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, obj.HMC1122_BOT_CS);
308             [obj, ReadData] = obj.WriteRead(obj.SPI_CS, ...
309                 obj.HMC1122_BOT.getCurrentAttenValue, 1, 6);
310             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 1, 255);
311             obj = obj.I2CWrite(obj.I2C_GPIO_ADDR, 3, 255);
312         end
313
314
315
316
317         function [obj, ReadData] = WriteRead(obj, ChipSelect, Data, byteSize,
                NumBitsPerSample)
318
319             % Open Digital Interface
320             [obj.digitalInterface, obj.digitalInterface.DeviceHandle] ...
321                 = obj.digitalInterface.ni845xOpen;
322
323             % 3.3 V VDD
324             obj.digitalInterface = obj.digitalInterface.ni845xSetIoVoltageLevel...
325                 (obj.digitalInterface.kNi845x25Volts);
326
327             % Begin script for read and write and enable SPI
328             [obj.digitalInterface, obj.digitalInterface.ScriptHandle] = obj.
```

```
            digitalInterface.ni845xSpiScriptOpen;
329         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptEnableSPI;
330
331         % Configure Clock rate, Clock polarity, Clock edge, NumBitsPerSample
332         obj.digitalInterface = obj.digitalInterface.setNi845xSpiScript...
333             (48, obj.digitalInterface.kNi845xSpiClockPolarityIdleLow, ...
334             obj.digitalInterface.kNi845xSpiClockPhaseFirstEdge, NumBitsPerSample);
335
336         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptNumBitsPerSample(
                NumBitsPerSample);
337
338         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptCSLow(ChipSelect);
339
340         [obj.digitalInterface, ScriptReadIndex] = obj.digitalInterface.
                ni845xSpiScriptWriteRead(byteSize, Data);
341
342         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptCSHigh(ChipSelect);
343
344
345         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptDisableSPI;
346         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptRun;
347
348         [obj.digitalInterface, ReadSize] = obj.digitalInterface.
                ni845xSpiScriptExtractReadDataSize(ScriptReadIndex);
349         [obj.digitalInterface, ReadData] = obj.digitalInterface.
                ni845xSpiScriptExtractReadData(ScriptReadIndex, ReadSize);
350
351         obj.digitalInterface = obj.digitalInterface.ni845xSpiScriptClose;
352         obj.digitalInterface = obj.digitalInterface.ni845xClose;
353     end
354
355     function [obj, ReadData] = I2CRead(obj, SlaveAddr, RegAddr)
356
357         [obj.digitalInterface, obj.digitalInterface.DeviceHandle] = ...
358             obj.digitalInterface.ni845xOpen;
359
360         obj.digitalInterface = ...
361             obj.digitalInterface.ni845xSetIoVoltageLevel(obj.digitalInterface.
                    kNi845x33Volts);
362
363         [obj.digitalInterface, obj.digitalInterface.ScriptHandle2] = ...
364             obj.digitalInterface.ni845xI2cScriptOpen;
365
366         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptClockRate(100);
367         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptPullupEnable(1);
368         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptAckPollTimeout(
                uint16(250));
369         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptIssueStart;
370         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptAddressWrite(
                SlaveAddr);
371         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptWrite(1, RegAddr);
372         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptIssueStart;
373         obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptAddressRead(
                SlaveAddr);
374
375         % Byte size and ACK/NACK
376         [obj.digitalInterface, ScriptReadIndex] = ...
```

```
377                 obj.digitalInterface.ni845xI2cScriptRead(1, obj.digitalInterface.
                        kNi845xI2cNakTrue);
378
379
380             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptIssueStop;
381             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptRun;
382
383             [obj.digitalInterface, ReadSize] = ...
384                 obj.digitalInterface.ni845xI2cScriptExtractReadDataSize(ScriptReadIndex);
385             [obj.digitalInterface, ReadData] = ...
386                 obj.digitalInterface.ni845xI2cScriptExtractReadData(ScriptReadIndex,
                        ReadSize);
387
388             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptClose;
389
390             obj.digitalInterface = obj.digitalInterface.ni845xClose;
391         end
392
393         function obj = I2CWrite(obj, SlaveAddr, RegAddr, data)
394
395             [obj.digitalInterface, obj.digitalInterface.DeviceHandle] = ...
396                 obj.digitalInterface.ni845xOpen;
397
398             obj.digitalInterface = ...
399                 obj.digitalInterface.ni845xSetIoVoltageLevel(obj.digitalInterface.
                        kNi845x33Volts);
400
401             [obj.digitalInterface, obj.digitalInterface.ScriptHandle2] = ...
402                 obj.digitalInterface.ni845xI2cScriptOpen;
403
404             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptClockRate(100);
405             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptPullupEnable(1);
406             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptAckPollTimeout(
                    uint16(250));
407             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptIssueStart;
408             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptAddressWrite(
                    SlaveAddr);
409             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptWrite(1, RegAddr);
410             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptWrite(1, data);
411
412             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptIssueStop;
413             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptRun;
414
415             obj.digitalInterface = obj.digitalInterface.ni845xI2cScriptClose;
416             obj.digitalInterface = obj.digitalInterface.ni845xClose;
417         end
418
419     end
420 end
```

Listing F.4: Class for Controlling One 1x4 PA Array

```
1  classdef PA_Array
2      %UNTITLED Summary of this class goes here
3      %   Detailed explanation goes here
4
5      properties
```

```
6           TopTwoChannels;
7           BottomTwoChannels;
8       end
9
10      methods
11          function obj = PA_Array(varargin)
12              % Set Top and Bottom two channels (Ch. 1, 2, 3, 4)
13              % Varargin{1}: NI USB-8452 Control board
14              % Varargin{2}: SPI Buffer CS
15              % Varargin{3}: I2C GPIO ADDR TOP
16              % Varargin{4}: I2C GPIO ADDR BOT
17              % Varargin{5}: GPIO CS for AMC7834     (same for top and bot)
18              % Varargin{6}: GPIO CS for HMC1122 TOP (same for top and bot)
19              % Varargin{7}: GPIO CS for HMC1122 BOT (same for top and bot)
20
21              obj.TopTwoChannels = DigitalControl(varargin{1}, AMC7834, ...
22                  HMC1122, HMC1122, varargin{2}, varargin{3}, ...
23                  varargin{5}, varargin{6}, varargin{7});
24
25              obj.BottomTwoChannels = DigitalControl(varargin{1}, AMC7834, ...
26                  HMC1122, HMC1122, varargin{2}, varargin{4}, ...
27                  varargin{5}, varargin{6}, varargin{7});
28
29          end
30          %%%%%%%% AMC7834 DAC/ADC Init %%%%%%%%
31          function obj = TopAMC7834_init(obj)
32              obj.TopTwoChannels = obj.TopTwoChannels.AMC7834_init;
33          end
34
35          function obj = BotAMC7834_init(obj)
36              obj.BottomTwoChannels = obj.BottomTwoChannels.AMC7834_init;
37          end
38
39          %%%%%%%%%%%%%%%%%%%%%% HMC1122 Control %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40          function obj = setHMC1122_CH1(obj, attenVal)
41              obj.TopTwoChannels = obj.TopTwoChannels.setHMC1122Top(attenVal);
42          end
43
44          function obj = setHMC1122_CH2(obj, attenVal)
45              obj.TopTwoChannels = obj.TopTwoChannels.setHMC1122Bot(attenVal);
46          end
47
48          function obj = setHMC1122_CH3(obj, attenVal)
49              obj.BottomTwoChannels = obj.BottomTwoChannels.setHMC1122Top(attenVal);
50          end
51
52          function obj = setHMC1122_CH4(obj, attenVal)
53              obj.BottomTwoChannels = obj.BottomTwoChannels.setHMC1122Bot(attenVal);
54          end
55
56          %%%%%%%% AMC7834 DAC Release %%%%%%%%
57          function obj = releaseCH1CH2DACClamps(obj)
58              obj.TopTwoChannels = obj.TopTwoChannels.releaseDACClamps;
59          end
60
61          function obj = releaseCH3CH4DACClamps(obj)
62              obj.BottomTwoChannels = obj.BottomTwoChannels.releaseDACClamps;
```

```
63          end
64
65          %%%%%%%% Set DAC Values (Gate Voltages) %%%%%%%%
66          function obj = setCH1DrvGate(obj, Value)
67              obj.TopTwoChannels = obj.TopTwoChannels.setDAC2(Value);
68          end
69
70          function obj = setCH1PAGate(obj, Value)
71              obj.TopTwoChannels = obj.TopTwoChannels.setDAC4(Value);
72          end
73
74          function obj = setCH2DrvGate(obj, Value)
75              obj.TopTwoChannels = obj.TopTwoChannels.setDAC1(Value);
76          end
77
78          function obj = setCH2PAGate(obj, Value)
79              obj.TopTwoChannels = obj.TopTwoChannels.setDAC3(Value);
80          end
81
82          function obj = setCH3DrvGate(obj, Value)
83              obj.BottomTwoChannels = obj.BottomTwoChannels.setDAC4(Value);
84          end
85
86          function obj = setCH3PAGate(obj, Value)
87              obj.BottomTwoChannels = obj.BottomTwoChannels.setDAC2(Value);
88          end
89
90          function obj = setCH4DrvGate(obj, Value)
91              obj.BottomTwoChannels = obj.BottomTwoChannels.setDAC3(Value);
92          end
93
94          function obj = setCH4PAGate(obj, Value)
95              obj.BottomTwoChannels = obj.BottomTwoChannels.setDAC1(Value);
96          end
97
98          %%%%%%%% Get DAC Values (Gate Voltages) %%%%%%%%
99          function [obj, voltage] = getCH1DrvGate(obj)
100             obj.TopTwoChannels = obj.TopTwoChannels.readADC;
101             [obj.TopTwoChannels, voltage] = obj.TopTwoChannels.getADCInt2;
102         end
103
104         function [obj, voltage] = getCH1PAGate(obj)
105             obj.TopTwoChannels = obj.TopTwoChannels.readADC;
106             [obj.TopTwoChannels, voltage] = obj.TopTwoChannels.getADCInt4;
107         end
108
109         function [obj, voltage] = getCH2DrvGate(obj)
110             obj.TopTwoChannels = obj.TopTwoChannels.readADC;
111             [obj.TopTwoChannels, voltage] = obj.TopTwoChannels.getADCInt1;
112         end
113
114         function [obj, voltage] = getCH2PAGate(obj)
115             obj.TopTwoChannels = obj.TopTwoChannels.readADC;
116             [obj.TopTwoChannels, voltage] = obj.TopTwoChannels.getADCInt3;
117         end
118
119         function [obj, voltage] = getCH3DrvGate(obj)
```

```
120             obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
121             [obj.BottomTwoChannels, voltage] = obj.BottomTwoChannels.getADCInt4;
122         end
123
124         function [obj, voltage] = getCH3PAGate(obj)
125             obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
126             [obj.BottomTwoChannels, voltage] = obj.BottomTwoChannels.getADCInt2;
127         end
128
129         function [obj, voltage] = getCH4DrvGate(obj)
130             obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
131             [obj.BottomTwoChannels, voltage] = obj.BottomTwoChannels.getADCInt3;
132         end
133
134         function [obj, voltage] = getCH4PAGate(obj)
135             obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
136             [obj.BottomTwoChannels, voltage] = obj.BottomTwoChannels.getADCInt1;
137         end
138
139         %%%%%%%% Get ADC Values (Drain Currents) %%%%%%%%
140         function [obj, current] = getCH1DrvDrainCurrent(obj)
141             % Returns an average value over 10 measurements
142             tmp = zeros(1,10);
143             for i=1:10
144                 obj.TopTwoChannels = obj.TopTwoChannels.readADC;
145                 [obj.TopTwoChannels, tmp(1,i)] = obj.TopTwoChannels.getDrainCurrent4;
146             end
147             current = round(mean(tmp),1);
148         end
149
150         function [obj, current] = getCH1PADrainCurrent(obj)
151             % Returns an average value over 10 measurements
152             tmp = zeros(1,10);
153             for i=1:10
154                 obj.TopTwoChannels = obj.TopTwoChannels.readADC;
155                 [obj.TopTwoChannels, tmp(1,i)] = obj.TopTwoChannels.getDrainCurrent2;
156             end
157             current = round(mean(tmp),1);
158         end
159
160         function [obj, current] = getCH2DrvDrainCurrent(obj)
161             % Returns an average value over 10 measurements
162             tmp = zeros(1,10);
163             for i=1:10
164                 obj.TopTwoChannels = obj.TopTwoChannels.readADC;
165                 [obj.TopTwoChannels, tmp(1,i)] = obj.TopTwoChannels.getDrainCurrent3;
166             end
167             current = round(mean(tmp),1);
168         end
169
170         function [obj, current] = getCH2PADrainCurrent(obj)
171             % Returns an average value over 10 measurements
172             tmp = zeros(1,10);
173             for i=1:10
174                 obj.TopTwoChannels = obj.TopTwoChannels.readADC;
175                 [obj.TopTwoChannels, tmp(1,i)] = obj.TopTwoChannels.getDrainCurrent1;
176             end
```

```
177             current = round(mean(tmp),1);
178         end
179 
180         function [obj, current] = getCH3DrvDrainCurrent(obj)
181             % Returns an average value over 10 measurements
182             tmp = zeros(1,10);
183             for i=1:10
184                 obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
185                 [obj.BottomTwoChannels, tmp(1,i)] = obj.BottomTwoChannels.getDrainCurrent2
                        ;
186             end
187             current = round(mean(tmp),1);
188         end
189 
190         function [obj, current] = getCH3PADrainCurrent(obj)
191             % Returns an average value over 10 measurements
192             tmp = zeros(1,10);
193             for i=1:10
194                 obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
195                 [obj.BottomTwoChannels, tmp(1,i)] = obj.BottomTwoChannels.getDrainCurrent4
                        ;
196             end
197             current = round(mean(tmp),1);
198         end
199 
200         function [obj, current] = getCH4DrvDrainCurrent(obj)
201             % Returns an average value over 10 measurements
202             tmp = zeros(1,10);
203             for i=1:10
204                 obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
205                 [obj.BottomTwoChannels, tmp(1,i)] = obj.BottomTwoChannels.getDrainCurrent1
                        ;
206             end
207             current = round(mean(tmp),1);
208         end
209 
210         function [obj, current] = getCH4PADrainCurrent(obj)
211             % Returns an average value over 10 measurements
212             tmp = zeros(1,10);
213             for i=1:10
214                 obj.BottomTwoChannels = obj.BottomTwoChannels.readADC;
215                 [obj.BottomTwoChannels, tmp(1,i)] = obj.BottomTwoChannels.getDrainCurrent3
                        ;
216             end
217             current = round(mean(tmp),1);
218         end
219 
220         %%%%%%%% Get AMC7834 Chip Temperature %%%%%%%%
221         function [obj, temp] = getTopAMC7834Temp(obj)
222             [obj.TopTwoChannels, temp] = obj.TopTwoChannels.getChipTemp;
223         end
224 
225         function [obj, temp] = getBotAMC7834Temp(obj)
226             [obj.BottomTwoChannels, temp] = obj.BottomTwoChannels.getChipTemp;
227         end
228     end
229 end
```

Listing F.5: Class for Controlling Full 4x4 MIMO Transmitter

```
classdef TransmitterSystem


    properties
        Arrays = [];
        digitalInterface;
        digitalConfig; % 4x2 matrix
        channelConfig; % 4x4 matrix
        SPI_CS          = [0; 1; 2; 3];

        I2C_GPIO_ADDR   = [32 34; ...
                           32 34; ...
                           33 35; ...
                           33 35];

        AMC7834_CS      = [253 253; ...
                           239 239; ...
                           253 253; ...
                           239 239];

        HMC1122_TOP_CS = [251 251; ...
                          223 223; ...
                          251 251; ...
                          223 223];

        HMC1122_BOT_CS = [247 247; ...
                          191 191; ...
                          247 247; ...
                          191 191];
        numOfActiveChannels;
        activeChannels;
    end

    methods
        function obj = TransmitterSystem(varargin)
            if length(varargin) == 3
                obj.digitalInterface = ni845xI2CSpiDio(varargin{3});
            else
                obj.digitalInterface = ni845xI2CSpiDio;
            end
            tmp = size(varargin{1});
            if tmp(1,1) ~= 4 || tmp(1,2) ~= 2
                fprintf('Wrong Digital Control Board Configuration');
                return
            elseif (sum(varargin{1}(:)==0)+sum(varargin{1}(:)==1)+sum(varargin{1}(:)==2))
                ~= 8
                fprintf('Wrong Values in Digital Control Board Configuration');
                return
            elseif sum(varargin{1}(:,1) ~= varargin{1}(:,2))
                fprintf('Please connect digital control cables properly');
                return
            else
                obj.digitalConfig = varargin{1};
            end

```

```
55              tmp = size(varargin{2});
56              if tmp(1,1) ~= 4 || tmp(1,2) ~= 4
57                  fprintf('Wrong Transmitter Channel Configuration');
58                  return
59              elseif (sum(varargin{2}(:)==0)+sum(varargin{2}(:)==1)) ~= 16
60                  fprintf('Wrong Values in Transmitter Channel Configuration');
61                  return
62              else
63                  obj.channelConfig = transpose(varargin{2});
64              end
65              obj.numOfActiveChannels = length(find(obj.channelConfig));
66              obj.activeChannels      = find(obj.channelConfig);
67
68              [row, col] = size(obj.digitalConfig);
69              k=1;
70              for i=1:row
71                      if obj.digitalConfig(i,1) ~= 0
72                          Arrays(k) = PA_Array(obj.digitalInterface, ...
73                              obj.SPI_CS(i), obj.I2C_GPIO_ADDR(i,1), obj.I2C_GPIO_ADDR(i,2),
                                    ...
74                              obj.AMC7834_CS(i,1), obj.HMC1122_TOP_CS(i,1), ...
75                              obj.HMC1122_BOT_CS(i,1));
76                          k=k+1;
77                      end
78              end
79              obj.Arrays = Arrays;
80          end
81
82          function obj = AMC7834_LEFT_Init(obj,boardNumber)
83
84              if boardNumber > length(obj.Arrays)
85                  fprintf('Exceeded board number limit');
86              else
87                  if obj.digitalConfig(boardNumber,1) == 1
88                      obj.Arrays(boardNumber) = ...
89                          obj.Arrays(boardNumber).TopAMC7834_init;
90                  elseif obj.digitalConfig(boardNumber,1) == 2
91                      obj.Arrays(boardNumber) = ...
92                          obj.Arrays(boardNumber).BotAMC7834_init;
93                  end
94              end
95          end
96
97          function obj = AMC7834_RIGHT_Init(obj,boardNumber)
98
99              if boardNumber > length(obj.Arrays)
100                 fprintf('Exceeded board number limit');
101             else
102                 if obj.digitalConfig(boardNumber,1) == 1
103                     obj.Arrays(boardNumber) = ...
104                         obj.Arrays(boardNumber).BotAMC7834_init;
105                 elseif obj.digitalConfig(boardNumber,1) == 2
106                     obj.Arrays(boardNumber) = ...
107                         obj.Arrays(boardNumber).TopAMC7834_init;
108                 end
109             end
110         end
```

```
111
112         function obj = releaseDAC_LEFT(obj,boardNumber)
113
114             if boardNumber > length(obj.Arrays)
115                 fprintf('Exceeded board number limit');
116             else
117                 if obj.digitalConfig(boardNumber,1) == 1
118                     obj.Arrays(boardNumber) = ...
119                         obj.Arrays(boardNumber).releaseCH1CH2DACClamps;
120                 elseif obj.digitalConfig(boardNumber,1) == 2
121                     obj.Arrays(boardNumber) = ...
122                         obj.Arrays(boardNumber).releaseCH3CH4DACClamps;
123                 end
124             end
125         end
126
127         function obj = releaseDAC_RIGHT(obj,boardNumber)
128
129             if boardNumber > length(obj.Arrays)
130                 fprintf('Exceeded board number limit');
131             else
132                 if obj.digitalConfig(boardNumber,1) == 1
133                     obj.Arrays(boardNumber) = ...
134                         obj.Arrays(boardNumber).releaseCH3CH4DACClamps;
135                 elseif obj.digitalConfig(boardNumber,1) == 2
136                     obj.Arrays(boardNumber) = ...
137                         obj.Arrays(boardNumber).releaseCH1CH2DACClamps;
138                 end
139             end
140         end
141
142         function obj = setDrvGate(obj,channelNumber, value)
143
144             if channelNumber > obj.numOfActiveChannels
145                 fprintf('Exceeded channel number limit');
146             else
147                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
148                 if obj.digitalConfig(boardNumber,1) == 1
149                     switch mod(obj.activeChannels(channelNumber),4)
150                         case 0
151                             obj.Arrays(boardNumber) = ...
152                                 obj.Arrays(boardNumber).setCH4DrvGate(value);
153                         case 1
154                             obj.Arrays(boardNumber) = ...
155                                 obj.Arrays(boardNumber).setCH1DrvGate(value);
156                         case 2
157                             obj.Arrays(boardNumber) = ...
158                                 obj.Arrays(boardNumber).setCH2DrvGate(value);
159                         case 3
160                             obj.Arrays(boardNumber) = ...
161                                 obj.Arrays(boardNumber).setCH3DrvGate(value);
162                     end
163                 elseif obj.digitalConfig(boardNumber,1) == 2
164                     switch mod(obj.activeChannels(channelNumber),4)
165                         case 0
166                             obj.Arrays(boardNumber) = ...
167                                 obj.Arrays(boardNumber).setCH1DrvGate(value);
```

```
                    case 1
                        obj.Arrays(boardNumber) = ...
                            obj.Arrays(boardNumber).setCH4DrvGate(value);
                    case 2
                        obj.Arrays(boardNumber) = ...
                            obj.Arrays(boardNumber).setCH3DrvGate(value);
                    case 3
                        obj.Arrays(boardNumber) = ...
                            obj.Arrays(boardNumber).setCH2DrvGate(value);
                end
            end
        end
    end

    function [obj, voltage] = getDrvGate(obj,channelNumber)

        if channelNumber > obj.numOfActiveChannels
            fprintf('Exceeded channel number limit');
        else
            boardNumber = ceil(obj.activeChannels(channelNumber)/4);
            if obj.digitalConfig(boardNumber,1) == 1
                switch mod(obj.activeChannels(channelNumber),4)
                    case 0
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH4DrvGate;
                    case 1
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH1DrvGate;
                    case 2
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH2DrvGate;
                    case 3
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH3DrvGate;
                end
            elseif obj.digitalConfig(boardNumber,1) == 2
                switch mod(obj.activeChannels(channelNumber),4)
                    case 0
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH1DrvGate;
                    case 1
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH4DrvGate;
                    case 2
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH3DrvGate;
                    case 3
                        [obj.Arrays(boardNumber), voltage] = ...
                            obj.Arrays(boardNumber).getCH2DrvGate;
                end
            end
        end
    end

    function obj = setPAGate(obj,channelNumber, value)
        if channelNumber > obj.numOfActiveChannels
            fprintf('Exceeded channel number limit');
```

```
225             else
226                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
227                 if obj.digitalConfig(boardNumber,1) == 1
228                     switch mod(obj.activeChannels(channelNumber),4)
229                         case 0
230                             obj.Arrays(boardNumber) = ...
231                                 obj.Arrays(boardNumber).setCH4PAGate(value);
232                         case 1
233                             obj.Arrays(boardNumber) = ...
234                                 obj.Arrays(boardNumber).setCH1PAGate(value);
235                         case 2
236                             obj.Arrays(boardNumber) = ...
237                                 obj.Arrays(boardNumber).setCH2PAGate(value);
238                         case 3
239                             obj.Arrays(boardNumber) = ...
240                                 obj.Arrays(boardNumber).setCH3PAGate(value);
241                     end
242                 elseif obj.digitalConfig(boardNumber,1) == 2
243                     switch mod(obj.activeChannels(channelNumber),4)
244                         case 0
245                             obj.Arrays(boardNumber) = ...
246                                 obj.Arrays(boardNumber).setCH1PAGate(value);
247                         case 1
248                             obj.Arrays(boardNumber) = ...
249                                 obj.Arrays(boardNumber).setCH4PAGate(value);
250                         case 2
251                             obj.Arrays(boardNumber) = ...
252                                 obj.Arrays(boardNumber).setCH3PAGate(value);
253                         case 3
254                             obj.Arrays(boardNumber) = ...
255                                 obj.Arrays(boardNumber).setCH2PAGate(value);
256                     end
257                 end
258             end
259         end
260
261         function [obj, voltage] = getPAGate(obj,channelNumber)
262
263             if channelNumber > obj.numOfActiveChannels
264                 fprintf('Exceeded channel number limit');
265             else
266                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
267                 if obj.digitalConfig(boardNumber,1) == 1
268                     switch mod(obj.activeChannels(channelNumber),4)
269                         case 0
270                             [obj.Arrays(boardNumber), voltage] = ...
271                                 obj.Arrays(boardNumber).getCH4PAGate;
272                         case 1
273                             [obj.Arrays(boardNumber), voltage] = ...
274                                 obj.Arrays(boardNumber).getCH1PAGate;
275                         case 2
276                             [obj.Arrays(boardNumber), voltage] = ...
277                                 obj.Arrays(boardNumber).getCH2PAGate;
278                         case 3
279                             [obj.Arrays(boardNumber), voltage] = ...
280                                 obj.Arrays(boardNumber).getCH3PAGate;
281                     end
```

```
282                 elseif obj.digitalConfig(boardNumber,1) == 2
283                     switch mod(obj.activeChannels(channelNumber),4)
284                         case 0
285                             [obj.Arrays(boardNumber), voltage] = ...
286                                 obj.Arrays(boardNumber).getCH1PAGate;
287                         case 1
288                             [obj.Arrays(boardNumber), voltage] = ...
289                                 obj.Arrays(boardNumber).getCH4PAGate;
290                         case 2
291                             [obj.Arrays(boardNumber), voltage] = ...
292                                 obj.Arrays(boardNumber).getCH3PAGate;
293                         case 3
294                             [obj.Arrays(boardNumber), voltage] = ...
295                                 obj.Arrays(boardNumber).getCH2PAGate;
296                     end
297                 end
298             end
299         end
300
301         function [obj, current] = getDrvCurrent(obj,channelNumber)
302
303             if channelNumber > obj.numOfActiveChannels
304                 fprintf('Exceeded channel number limit');
305             else
306                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
307                 if obj.digitalConfig(boardNumber,1) == 1
308                     switch mod(obj.activeChannels(channelNumber),4)
309                         case 0
310                             [obj.Arrays(boardNumber), current] = ...
311                                 obj.Arrays(boardNumber).getCH4DrvDrainCurrent;
312                         case 1
313                             [obj.Arrays(boardNumber), current] = ...
314                                 obj.Arrays(boardNumber).getCH1DrvDrainCurrent;
315                         case 2
316                             [obj.Arrays(boardNumber), current] = ...
317                                 obj.Arrays(boardNumber).getCH2DrvDrainCurrent;
318                         case 3
319                             [obj.Arrays(boardNumber), current] = ...
320                                 obj.Arrays(boardNumber).getCH3DrvDrainCurrent;
321                     end
322                 elseif obj.digitalConfig(boardNumber,1) == 2
323                     switch mod(obj.activeChannels(channelNumber),4)
324                         case 0
325                             [obj.Arrays(boardNumber), current] = ...
326                                 obj.Arrays(boardNumber).getCH1DrvDrainCurrent;
327                         case 1
328                             [obj.Arrays(boardNumber), current] = ...
329                                 obj.Arrays(boardNumber).getCH4DrvDrainCurrent;
330                         case 2
331                             [obj.Arrays(boardNumber), current] = ...
332                                 obj.Arrays(boardNumber).getCH3DrvDrainCurrent;
333                         case 3
334                             [obj.Arrays(boardNumber), current] = ...
335                                 obj.Arrays(boardNumber).getCH2DrvDrainCurrent;
336                     end
337                 end
338             end
```

```
339         end
340
341         function [obj, current] = getPACurrent(obj,channelNumber)
342
343             if channelNumber > obj.numOfActiveChannels
344                 fprintf('Exceeded channel number limit');
345             else
346                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
347                 if obj.digitalConfig(boardNumber,1) == 1
348                     switch mod(obj.activeChannels(channelNumber),4)
349                         case 0
350                             [obj.Arrays(boardNumber), current] = ...
351                                 obj.Arrays(boardNumber).getCH4PADrainCurrent;
352                         case 1
353                             [obj.Arrays(boardNumber), current] = ...
354                                 obj.Arrays(boardNumber).getCH1PADrainCurrent;
355                         case 2
356                             [obj.Arrays(boardNumber), current] = ...
357                                 obj.Arrays(boardNumber).getCH2PADrainCurrent;
358                         case 3
359                             [obj.Arrays(boardNumber), current] = ...
360                                 obj.Arrays(boardNumber).getCH3PADrainCurrent;
361                     end
362                 elseif obj.digitalConfig(boardNumber,1) == 2
363                     switch mod(obj.activeChannels(channelNumber),4)
364                         case 0
365                             [obj.Arrays(boardNumber), current] = ...
366                                 obj.Arrays(boardNumber).getCH1PADrainCurrent;
367                         case 1
368                             [obj.Arrays(boardNumber), current] = ...
369                                 obj.Arrays(boardNumber).getCH4PADrainCurrent;
370                         case 2
371                             [obj.Arrays(boardNumber), current] = ...
372                                 obj.Arrays(boardNumber).getCH3PADrainCurrent;
373                         case 3
374                             [obj.Arrays(boardNumber), current] = ...
375                                 obj.Arrays(boardNumber).getCH2PADrainCurrent;
376                     end
377                 end
378             end
379         end
380
381
382         function obj = setAtten(obj,channelNumber, value)
383             if channelNumber > obj.numOfActiveChannels
384                 fprintf('Exceeded channel number limit');
385             else
386                 boardNumber = ceil(obj.activeChannels(channelNumber)/4);
387                 if obj.digitalConfig(boardNumber,1) == 1
388                     switch mod(obj.activeChannels(channelNumber),4)
389                         case 0
390                             obj.Arrays(boardNumber) = ...
391                                 obj.Arrays(boardNumber).setHMC1122_CH4(value);
392                         case 1
393                             obj.Arrays(boardNumber) = ...
394                                 obj.Arrays(boardNumber).setHMC1122_CH1(value);
395                         case 2
```

```
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH2(value);
                            case 3
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH3(value);
                        end
                    elseif obj.digitalConfig(boardNumber,1) == 2
                        switch mod(obj.activeChannels(channelNumber),4)
                            case 0
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH1(value);
                            case 1
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH4(value);
                            case 2
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH3(value);
                            case 3
                                obj.Arrays(boardNumber) = ...
                                    obj.Arrays(boardNumber).setHMC1122_CH2(value);
                        end
                    end
                end
            end

    end
end
```

Listing F.6: Main File

```
% mex -setup if compiler throws an error
% Use MinGW64 Compiler (C)
% loadlibrary('ni845x', 'ni845x.h');

%{
    0: No connection
    1: PCB facing up
    2: PCB facing down
%}
digital_board_config = [1 1;
                        1 1;
                        1 1;
                        1 1];

%{
    Channels are from left to right, top to bottom
    facing the connection view

    0: Inactive channel
    1: Active channel

    1 2 3 4
    5 6 7 8
    9 10 11 12
    13 14 15 16
%}
channel_config = [1 1 1 1;
```

```
                    1 1 1 1;
                    1 1 1 1;
                    1 1 1 1];


tt = TransmitterSystem(digital_board_config, channel_config, '1D4300A');

tt = tt.AMC7834_LEFT_Init(1);  % Board number from top to bottom
tt = tt.AMC7834_RIGHT_Init(1); % Board number from top to bottom
tt = tt.AMC7834_LEFT_Init(2);  % Board number from top to bottom
tt = tt.AMC7834_RIGHT_Init(2); % Board number from top to bottom
tt = tt.AMC7834_LEFT_Init(3);  % Board number from top to bottom
tt = tt.AMC7834_RIGHT_Init(3); % Board number from top to bottom
tt = tt.AMC7834_LEFT_Init(4);  % Board number from top to bottom
tt = tt.AMC7834_RIGHT_Init(4); % Board number from top to bottom

tt = tt.releaseDAC_LEFT(1);    % Board number from top to bottom
tt = tt.releaseDAC_RIGHT(1);   % Board number from top to bottom
tt = tt.releaseDAC_LEFT(2);    % Board number from top to bottom
tt = tt.releaseDAC_RIGHT(2);   % Board number from top to bottom
tt = tt.releaseDAC_LEFT(3);    % Board number from top to bottom
tt = tt.releaseDAC_RIGHT(3);   % Board number from top to bottom
tt = tt.releaseDAC_LEFT(4);    % Board number from top to bottom
tt = tt.releaseDAC_RIGHT(4);   % Board number from top to bottom

tt = tt.setDrvGate(1, -2.84);
tt = tt.setDrvGate(2, -2.86);
tt = tt.setDrvGate(3, -2.87);
tt = tt.setDrvGate(4, -2.93);
tt = tt.setDrvGate(5, -2.88);
tt = tt.setDrvGate(6, -2.85);
tt = tt.setDrvGate(7, -2.83);
tt = tt.setDrvGate(8, -2.83);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(9, -2.88);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(10, -2.87);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(11, -2.89);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(12, -2.885);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(13, -2.84);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(14, -2.89);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(15, -2.88);  % Channel number from top to bottom, Gate Volt
tt = tt.setDrvGate(16, -3.04);  % Channel number from top to bottom, Gate Volt

% tt = tt.setDrvGate(1, -5);
% tt = tt.setDrvGate(2, -5);
% tt = tt.setDrvGate(3, -5);
% tt = tt.setDrvGate(4, -5);
% tt = tt.setDrvGate(5, -5);
% tt = tt.setDrvGate(6, -5);
% tt = tt.setDrvGate(7, -5);
% tt = tt.setDrvGate(8, -5);
% tt = tt.setDrvGate(9, -5);
% tt = tt.setDrvGate(10, -5);
% tt = tt.setDrvGate(11, -5);
% tt = tt.setDrvGate(12, -5);
% tt = tt.setDrvGate(13, -5);
% tt = tt.setDrvGate(14, -5);
% tt = tt.setDrvGate(15, -5);
```

```
% tt = tt.setDrvGate(16, -5);
%
%
% [tt, Drv_Voltage1] = tt.getDrvGate(1);   % Channel number from top to bottom
% [tt, Drv_Current1] = tt.getDrvCurrent(1);% Channel number from top to bottom
%
% [tt, Drv_Voltage2] = tt.getDrvGate(2);   % Channel number from top to bottom
% [tt, Drv_Current2] = tt.getDrvCurrent(2);% Channel number from top to bottom
%
% [tt, Drv_Voltage3] = tt.getDrvGate(3);   % Channel number from top to bottom
% [tt, Drv_Current3] = tt.getDrvCurrent(3);% Channel number from top to bottom
%
% [tt, Drv_Voltage4] = tt.getDrvGate(4);   % Channe4 number from top to bottom
% [tt, Drv_Current4] = tt.getDrvCurrent(4);% Channe4 number from top to bottom
%
% [tt, Drv_Voltage5] = tt.getDrvGate(5);   % Channel number from top to bottom
% [tt, Drv_Current5] = tt.getDrvCurrent(5);% Channel number from top to bottom
%
% [tt, Drv_Voltage6] = tt.getDrvGate(6);   % Channel number from top to bottom
% [tt, Drv_Current6] = tt.getDrvCurrent(6);% Channel number from top to bottom
%
% [tt, Drv_Voltage7] = tt.getDrvGate(7);   % Channel number from top to bottom
% [tt, Drv_Current7] = tt.getDrvCurrent(7);% Channel number from top to bottom
%
% [tt, Drv_Voltage8] = tt.getDrvGate(8);   % Channe4 number from top to bottom
% [tt, Drv_Current8] = tt.getDrvCurrent(8);% Channe4 number from top to bottom
%
% [tt, Drv_Voltage9] = tt.getDrvGate(9);   % Channel number from top to bottom
% [tt, Drv_Current9] = tt.getDrvCurrent(9);% Channel number from top to bottom
%
% [tt, Drv_Voltage10] = tt.getDrvGate(10);   % Channel number from top to bottom
% [tt, Drv_Current10] = tt.getDrvCurrent(10);% Channel number from top to bottom
%
% [tt, Drv_Voltage11] = tt.getDrvGate(11);   % Channel number from top to bottom
% [tt, Drv_Current11] = tt.getDrvCurrent(11);% Channel number from top to bottom
%
% [tt, Drv_Voltage12] = tt.getDrvGate(12);   % Channe4 number from top to bottom
% [tt, Drv_Current12] = tt.getDrvCurrent(12);% Channe4 number from top to bottom
%
% [tt, Drv_Voltage13] = tt.getDrvGate(13);   % Channel number from top to bottom
% [tt, Drv_Current13] = tt.getDrvCurrent(13);% Channel number from top to bottom
%
% [tt, Drv_Voltage14] = tt.getDrvGate(14);   % Channel number from top to bottom
% [tt, Drv_Current14] = tt.getDrvCurrent(14);% Channel number from top to bottom
%
% [tt, Drv_Voltage15] = tt.getDrvGate(15);   % Channel number from top to bottom
% [tt, Drv_Current15] = tt.getDrvCurrent(15);% Channel number from top to bottom
%
% [tt, Drv_Voltage16] = tt.getDrvGate(16);   % Channe4 number from top to bottom
% [tt, Drv_Current16] = tt.getDrvCurrent(16);% Channe4 number from top to bottom
%

tt = tt.setPAGate(1, -2.8);
tt = tt.setPAGate(2, -2.80);
tt = tt.setPAGate(3, -2.98);
tt = tt.setPAGate(4, -2.78);
tt = tt.setPAGate(5, -2.815);
```

```
tt = tt.setPAGate(6, -2.76);
tt = tt.setPAGate(7, -2.81);
tt = tt.setPAGate(8, -2.82);
tt = tt.setPAGate(9, -2.88);
tt = tt.setPAGate(10, -2.85);
tt = tt.setPAGate(11, -2.84);
tt = tt.setPAGate(12, -2.81);
tt = tt.setPAGate(13, -2.94);
tt = tt.setPAGate(14, -2.81);
tt = tt.setPAGate(15, -2.84);
tt = tt.setPAGate(16, -2.97);

% tt = tt.setPAGate(1, -5);
% tt = tt.setPAGate(2, -5);
% tt = tt.setPAGate(3, -5);
% tt = tt.setPAGate(4, -5);
% tt = tt.setPAGate(5, -5);
% tt = tt.setPAGate(6, -5);
% tt = tt.setPAGate(7, -5);
% tt = tt.setPAGate(8, -5);
% tt = tt.setPAGate(9, -5);
% tt = tt.setPAGate(10, -5);
% tt = tt.setPAGate(11, -5);
% tt = tt.setPAGate(12, -5);
% tt = tt.setPAGate(13, -5);
% tt = tt.setPAGate(14, -5);
% tt = tt.setPAGate(15, -5);
% tt = tt.setPAGate(16, -5);
%
% [tt, PA_Voltage1] = tt.getPAGate(1);   % Channel number from top to bottom
% [tt, PA_Current1] = tt.getPACurrent(1);% Channel number from top to bottom
%
% [tt, PA_Voltage2] = tt.getPAGate(2);   % Channel number from top to bottom
% [tt, PA_Current2] = tt.getPACurrent(2);% Channel number from top to bottom
%
% [tt, PA_Voltage3] = tt.getPAGate(3);   % Channel number from top to bottom
% [tt, PA_Current3] = tt.getPACurrent(3);% Channel number from top to bottom
%
% [tt, PA_Voltage4] = tt.getPAGate(4);   % Channel number from top to bottom
% [tt, PA_Current4] = tt.getPACurrent(4);% Channel number from top to bottom
%
% [tt, PA_Voltage5] = tt.getPAGate(5);   % Channel number from top to bottom
% [tt, PA_Current5] = tt.getPACurrent(5);% Channel number from top to bottom
%
% [tt, PA_Voltage6] = tt.getPAGate(6);   % Channel number from top to bottom
% [tt, PA_Current6] = tt.getPACurrent(6);% Channel number from top to bottom
%
% [tt, PA_Voltage7] = tt.getPAGate(7);   % Channel number from top to bottom
% [tt, PA_Current7] = tt.getPACurrent(7);% Channel number from top to bottom
%
% [tt, PA_Voltage8] = tt.getPAGate(8);   % Channel number from top to bottom
% [tt, PA_Current8] = tt.getPACurrent(8);% Channe
%
% [tt, PA_Voltage9] = tt.getPAGate(9);   % Channel number from top to bottom
% [tt, PA_Current9] = tt.getPACurrent(9);% Channel number from top to bottom
%
% [tt, PA_Voltage10] = tt.getPAGate(10);   % Channel number from top to bottom
```

```
% [tt, PA_Current10] = tt.getPACurrent(10);% Channel number from top to bottom
%
% [tt, PA_Voltage11] = tt.getPAGate(11);    % Channel number from top to bottom
% [tt, PA_Current11] = tt.getPACurrent(11);% Channel number from top to bottom
%
% [tt, PA_Voltage12] = tt.getPAGate(12);    % Channel number from top to bottom
% [tt, PA_Current12] = tt.getPACurrent(12);% Channel number from top to bottom
%
% [tt, PA_Voltage13] = tt.getPAGate(13);    % Channel number from top to bottom
% [tt, PA_Current13] = tt.getPACurrent(13);% Channe
%
% [tt, PA_Voltage14] = tt.getPAGate(14);    % Channel number from top to bottom
% [tt, PA_Current14] = tt.getPACurrent(14);% Channel number from top to bottom
%
% [tt, PA_Voltage15] = tt.getPAGate(15);
% [tt, PA_Current15] = tt.getPACurrent(15);
%
% [tt, PA_Voltage16] = tt.getPAGate(16);
% [tt, PA_Current16] = tt.getPACurrent(16);
```