thesis for the degree of Doctor of Philosophy

Integrating Deep Learning for Hardware Impairments Mitigation in Communication Systems

Yibo Wu

CHALMERS
UNIVERSITY OF TECHNOLOGY

Communication Systems Group
Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2024

Integrating Deep Learning for Hardware Impairments Mitigation in Communication Systems

YIBO WU

ISBN: 978-91-8103-114-0
Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie 5572
ISSN 0346-718X

This thesis has been prepared using LaTeX.

Communication Systems Group
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

Printed by Chalmers Reproservice
Gothenburg, Sweden, October, 2024.

To my wife and my parents.

Abstract

The rapid growth of wireless communication, especially with the deployment of massive multiple-input multiple-output (MIMO) systems, has driven up power consumption, requiring more cost-efficient hardware. However, lower-cost components often introduce impairments such as power amplifier (PA) nonlinearity, phase noise (PN), in-phase and quadrature (IQ) imbalance, and mutual coupling, leading to degraded system performance. Classical model-based methods face challenges in effectively mitigating these impairments while maintaining performance and low computational complexity. In contrast, deep learning techniques offer a promising alternative by providing more adaptable and efficient solutions. This thesis explores deep learning-based methods for mitigating hardware impairments in communication systems, aiming to enhance both performance and power efficiency.

We first focus on single-input single-output (SISO) systems. In Paper A, we address the joint mitigation of IQ imbalance and PA nonlinearity using digital predistortion (DPD). Classical model-based methods often underperform in joint mitigation, and existing neural network (NN)-based methods are computationally demanding. To resolve these issues, we propose a novel NN-based DPD model combined with an NN pruning technique. This approach provides a more power-efficient solution for mitigating the combined impairments of the PA and IQ modulator compared to existing models. In Paper B, we further explore training DPD with low-sampling rate data. Supervised learning methods rely on high sampling-rate feedback paths, which are costly in wideband and multi-antenna scenarios. To overcome this, we introduce a reinforcement learning (RL)-based DPD learning algorithm that reduces the reliance on such high-sampling rate feedback paths while maintaining effective learning, making the DPD optimization more power-efficient.

We next shift our focus to mitigation techniques in massive MIMO systems. In Paper C, we tackle the joint mitigation of PA linearization and antenna crosstalk in massive multi-user (MU) MIMO orthogonal frequency division multiplexing (OFDM) networks. In such systems, the large number of antennas and corresponding PAs significantly increase the computational complexity of conventional DPD. While deploying DPD per user equipment (UE) in the frequency-domain (FD) instead of per PA in the time-domain (TD) can reduce complexity, the literature lacks proper FD DPD models. To address this, we propose a low-complexity FD convolutional neural network (CNN)-based DPD model. This model is effective in a line-of-sight (LOS) channel with fewer UEs.

Finally, Paper D explores joint channel and PN estimation in cell-free massive MIMO OFDM systems. Several previous studies assume single-carrier PN models to OFDM systems, leading to mismatches and overly optimistic performance predictions. We consider two setups: shared and separate local oscillators (LOs) between distributed access points (APs), which introduce uncorrelated and correlated PN,

respectively. We propose novel distributed and centralized joint PN and channel estimators, including a deep learning-based channel estimator, which demonstrates improved performance in both PN and channel estimation.

To summarize, this thesis explores the implementation of various deep learning-based techniques to effectively mitigate hardware impairments across different communication systems.

Keywords: 5G, 6G, hardware impairments mitigation, deep learning, power amplifier (PA), digital predistortion (DPD), local oscillator (LO), phase noise (PN), in-phase and quadrature (IQ) imbalance, antenna crosstalk, mutual coupling, massive MIMO, channel estimation, achievable rate.

List of Publications

This thesis is based on the following publications:

[A] **Y. Wu**, U. Gustavsson, A. Graell i Amat and H. Wymeersch, "Low complexity joint impairment mitigation of I/Q modulator and PA using neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 54-64, Jan. 2022.

[B] **Y. Wu**, J. Song, C. Häger, U. Gustavsson, A. Graell i Amat and H. Wymeersch, "Symbol-based over-the-air digital predistortion using reinforcement learning," in *IEEE International Conference on Communications*, Seoul, South Korea, pp. 2615-2620, May 2022.

[C] **Y. Wu**, U. Gustavsson, M. Valkama, A. Graell i Amat, and H. Wymeersch, "Time vs. frequency domain DPD for massive MIMO: methods and performance analysis," *IEEE Transaction on Wireless Communications*, major revision, Jul. 2024.

[D] **Y. Wu**, L. Sanguinetti, M. F. Keskin, U. Gustavsson, A. Graell i Amat, and H. Wymeersch, "Uplink cell-free massive MIMO OFDM with phase noise-aware channel estimation: separate and shared LOs," submitted to *IEEE Transaction on Wireless Communications*, Sep. 2024.

Publications by the author not included in this thesis:

[E] **Y. Wu**, U. Gustavsson, A. Graell i Amat and H. Wymeersch, "Residual neural networks for digital predistortion," in *IEEE Global Communications Conference*, Taipei, Taiwan, pp. 1-6, Dec. 2020.

[F] **Y. Wu**, U. Gustavsson, M. Valkama, A. Graell i Amat and H. Wymeersch, "Frequency-domain digital predistortion for massive MU-MIMO-OFDM downlink," in *IEEE Global Communications Conference*, Rio de Janeiro, Brazil, pp. 579-584, Dec. 2022.

[G] **Y. Wu**, L. Sanguinetti, U. Gustavsson, A. Graell i Amat and H. Wymeersch, "Impact of phase noise on uplink cell-free massive MIMO OFDM," in *IEEE Global Communications Conference*, Kuala Lumpur, Malaysia, pp. 5829-5834, Dec. 2023.

[H] **Y. Wu**, F. Keskin, U. Gustavsson, G. Seco-Granados, E. G. Larsson, and H. Wymeersch, "Location and map-assisted wideband phase and time calibration between distributed antennas," in *IEEE Global Communications Conference Workshop*, Cape Town, South Africa, Dec. 2024, to appear.

[I] Y. Ge, **Y. Wu**, F. Jiang, O. Kaltiokallio, J. Talvitie, M. Valkama, L. Svensson, and H. Wymeersch, "Iterated posterior linearization PMB filter for 5G SLAM," in *IEEE International Conference on Communications*, Seoul, Republic of Korea, pp. 877-882, May 2022.

[J] J. Song, V. Lauinger, **Y. Wu**, C. Häger, J. Schröder,A. Graell i Amat, L. Schmalen, and H. Wymeersch, "Blind channel equalization using vector-quantized variational autoencoders," submitted to *IEEE Transaction on Communications*, 2024.

[K] S. Rivetti, J. M. Mateos-Ramos, **Y. Wu**, J. Song, M. F. Keskin, V. Yajnanarayana, C. Häger, and H. Wymeersch, "Spatial signal design for positioning via end-to-end learning," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 525-529, Mar. 2023.

[L] J. M. Mateos-Ramos, J. Song, **Y. Wu**, C. Häger, M. F. Keskin, V. Yajnanarayana, and H. Wymeersch, "End-to-end learning for integrated sensing and communication," in *IEEE International Conference on Communications*, Seoul, Republic of Korea, pp. 1942-1947, May 2022.

[M] H. Chen, S. R. Aghdam, M. F. Keskin, **Y. Wu**, S. Lindberg, A. Wolfgang, U. Gustavsson, T. Eriksson, and H. Wymeersch, "MCRB-based performance analysis of 6G localization under hardware impairments," in *IEEE International Conference on Communications Workshops*, Seoul, Republic of Korea, pp. 115-120, May 2022.

[N] H. Guo, H. Wymeersch, B. Makki, H. Chen, **Y. Wu**, G. Durisi, M. F. Keskin, M. H. Moghaddam, C. Madapatha, H. Yu, P. Hammarberg, H. Kim, and T. Svensson, "Integrated communication, localization, and sensing in 6G D-MIMO networks," *IEEE Wireless Communications Magazine*, minor revision, Oct. 2024.

Acknowledgments

As I stand at the end of this long and winding PhD journey, my heart is filled with immense gratitude. Reflecting on this path, I realize that it has been shaped not just by the work itself but by the incredible people who have walked beside me, offering their support, wisdom, and kindness. It is hard to imagine coming this far without the support and kindness I have received. I simply could not have come this far without you all—you have not only helped shape my work but have made this journey a deeply meaningful and unforgettable chapter of my life.

First and foremost, a heartfelt thank you to my main supervisor and examiner, Prof. Henk Wymeersch. Henk, your endless support, both academic and personal, has been the compass that has guided me through this PhD journey. Whenever I found myself lost in a complex problem or in need of direction, you were always there to help me steer the PhD ship back on course. Your efficiency and forward-thinking approach have taught me lessons I will carry with me long after this PhD is over. I also want to sincerely thank my co-supervisors, Prof. Alexandre Graell i Amat and Dr. Ulf Gustavsson. Alex, your "massive" and "critical" feedback always push me to improve my work, and they really shape how I write (and "complain"!). Ulf, thank you for the "longest" support. You have been with me since my Master thesis, and I have cherished every moment of our partnership, from research breakthroughs to shared jokes. I look forward to continuing our journey together.

Special thanks to Prof. Luca Sanguinetti for the invaluable discussions during my time in Pisa on cell-free massive MIMO, and for continuing to guide me with your insight and expertise. My gratitude also extend to Prof. Mikko Valkama for your collaboration and insightful discussions on hardware impairment mitigation. Thanks to Prof. Alexios Balatsoukas-Stimming for your constructive feedback during my Midway seminar and for the deep discussions on deep learning topics. I am also grateful to Prof. Thomas Eriksson for generously sharing your deep knowledge of PA nonlinearity. The "preamble" of this Journey starts from your guidance as my Master thesis examiner. I would also like to thank Prof. Gonzalo Seco-Granados and Prof. Erik G. Larsson for the fruitful discussions and guidance on phase calibration that have broadened my perspective.

To my colleagues at ComSys, you made the office feel like home. Thank you for the many moments of laughter we have shared in the lunchroom. A big thanks to Prof. Fredrik Brännström and Prof. Erik Ström for leading the group with such excellence. Fredrik, it was an honor to be your TA, and Erik, thanks for your care during our talks. To my office mates—Yu, Mohammad, Shen, Hao, Hui, and Hyowon—you have made every day lighter with your company. A special thanks to Jinxiang, José, Kamran, Sharief, and the entire ComSys group, you have all been a source of support

and joy along the way.

My deepest thanks to my colleagues at Ericsson. Milad, thank you for being such a great friend and always sharing a laugh. Xinlin, Jingya, Adrián, Mingquan, and Colette, your encouragement since day one has meant so much. Lise, it has been a joy to share this industrial PhD journey with you, and I wish you all the best with your work ahead. Anders and Henrik, Thank you for your steady leadership and and unwavering support through this journey. A big thank you to my colleagues in the Antenna Systems group, and to Mikael, Mohamed, Simone, Markus, Niklas, and so many more in Ericsson for all your support.

I would also like to express my gratitude to my friends, Adriana, Bile P., Di P., Chi Z., Chuanxu Z., Chenliang L., Guo W., Haoge L., Huang Z., Hang Y., Junjie L., Jiaxi Y., Jian Z., Jun L., Liqin D., Lin X., Le X., Ke R., Kunyuan H., Pei W., Ruiqi R., little Qiuyu C., Rui X., Siqi W., Sining A., Wenxuan L., Xinjian G., Xinyan L., Xixi L., Xin W., Xin J., Xue B., Xunbo Y., Xueyin G., Xiang L., Yaxiong L., Ye C., Yifei Q., Yifei Z., Yue W., Yuchen Z., Zhihang W., Zhiyuan W., Zhejian C., Zongyu Z., Ziyu Z., Ze Z., and many more. Your friendship has been a source of joy throughout these past years.

To my parents and family, your constant support has been a source of strength throughout this journey. Every step of this journey has been strengthened by your love, and I cannot thank you enough.

And finally, to my dearest wife, Yidong, no words can truly express how grateful I am for you. From the very beginning, you have been my rock, giving me endless love, support, encouragement, and patience. You have made countless sacrifices, standing by me through every good and bad moment, always believing in me when I struggled to believe in myself. I could not have done this without you by my side. I love you, more than words can say.

YIBO WU
Göteborg, October, 2024.

Financial Support

This work was supported in part by the Swedish Foundation for Strategic Research (SSF), grant no. ID19-0021.

Acronyms

2D	two-dimensional
5G	fifth-generation mobile network
6G	sixth-generation mobile network
ACLR	adjacent channel leakage ratio
ACPR	adjacent channel power ratio
ADC	analog-to-digital converter
AM/AM	amplitude-to-amplitude
AM/PM	amplitude-to-phase
AP	access point
ASIC	application-specific integrated circuit
AWGN	additive white Gaussian noise
BER	bit error rate
BPF	bandpass filter
BS	base-station
CFO	carrier frequency offset
CNN	convolutional neural network
CPU	central processing unit
CP	cyclic prefix
CPE	common phase error
CSI	channel state information
DAC	digital-to-analog converter
DFT	discrete Fourier transform
DOA	direction-of-arrival
DPD	digital predistortion

EVM	error vector magnitude
FD	frequency-domain
FIR	finite impulse response
FLOP	floating point operations
FPGA	field-programmable gate array
GPT	generative transformer
GPU	graphics processing unit
GMP	generalized memory polynomial
I	in-phase
ICI	inter-carrier interference
IF	intermediate frequency
ISI	intersymbol interference
IDFT	inverse discrete Fourier transform
ILA	indirect learning architecture
IQ	in-phase and quadrature
IUI	inter-user interference
LDPC	low-density parity-check
LNA	low noise amplifier
LOS	line-of-sight
LMMSE	linear minimum mean square error
LPF	lowpass filter
LO	local oscillator
LuT	lookup table
MLP	multilayer perceptron
MIMO	multiple-input multiple-output
MMSE	minimum mean square error

mm-Wave	millimeter-wave
MP	memory polynomial
MSE	mean square error
MU	multi-user
NMSE	normalized mean squared error
NN	neural network
OFDM	orthogonal frequency division multiplexing
OOB	out-of-band
OTA	over-the-air
PA	power amplifier
PDF	probability density function
PH	parallel Hammerstein
PLL	phase-locked loop
PN	phase noise
PSD	power spectral density
PSK	phase shift keying
PTRS	phase tracking reference signals
Q	quadrature
QAM	quadrature amplitude modulation
ReLU	rectified linear unit
RF	radio frequency
RL	reinforcement learning
RVTDNN	real-valued time-delay neural network
SE	spectral efficiency
SER	symbol error rate
SGD	stochastic gradient descent

SISO	single-input single-output
SNR	signal-to-noise ratio
TD	time-domain
TDD	time division duplexing
UE	user equipment
ZF	zero forcing

Contents

Abstract i

List of Papers iv

Acknowledgements vii

Acronyms x

I Overview 1

1 Introduction 3
- 1.1 Background . . . 3
- 1.2 Thesis Outline . . . 5
- 1.3 Notation . . . 5

2 Hardware Impairments and Mitigation in Communication Systems 7
- 2.1 Communication With Ideal Hardware . . . 7
  - 2.1.1 Transmitter . . . 8
  - 2.1.2 Channel . . . 11
  - 2.1.3 Receiver . . . 12
- 2.2 Hardware Impairments . . . 14
  - 2.2.1 Power Amplifier Nonlinearity . . . 15
  - 2.2.2 IQ Imbalance . . . 17
  - 2.2.3 Phase Noise . . . 19
  - 2.2.4 Antenna Crosstalk . . . 22
  - 2.2.5 Other Hardware Impairments . . . 24

2.3 Classical Mitigation Techniques . . . 25
2.4 Performance and Complexity Metrics . . . 31
2.4.1 Performance Metrics . . . 31
2.4.2 Complexity Metrics . . . 32

3 Deep Learning for Communication Systems 35
3.1 Deep Learning Introduction . . . 35
3.1.1 Types of Deep Learning Techniques . . . 36
3.1.2 Implementation of Deep Learning . . . 40
3.2 Deep Learning for the Physical Layer in Communication Systems . . . 44

4 Hardware Impairments Mitigation Using Deep Learning 47
4.1 Motivations, Challenges, and Applications . . . 47
4.2 Deep Learning for DPD . . . 49
4.2.1 Optimization deep learning-based DPD . . . 49
4.2.2 Neural Network-based DPD in the Time-Domain and Frequency-Domain . . . 53
4.2.3 Joint Mitigation of Hardware Impairments Using DPD . . . 54

5 Contributions and Future Work 57
5.1 Paper A . . . 57
5.2 Paper B . . . 58
5.3 Paper C . . . 58
5.4 Paper D . . . 59
5.5 Future work . . . 60

References 61

II Papers 81

A Low Complexity Joint Impairment Mitigation of I/Q Modulator and PA Using Neural Networks A1
1 Introduction . . . A3
2 System Model . . . A5
2.1 IQ Imbalance . . . A6
2.2 PA Nonlinearity . . . A7
2.3 Digital Predistortion . . . A7
3 Preliminaries . . . A8
3.1 DPD Parameter Identification by ILA . . . A8
3.2 Shortcut Connections . . . A9

4 Shortcut Neural Networks for DPD . . . A10
4.1 Shortcut Connections for IQ-PA System . . . A10
4.2 SVDEN Architecture . . . A11
4.3 Neural Network Pruning . . . A13
4.4 Computational Complexity . . . A14
5 Experimental results . . . A15
5.1 Setup . . . A15
5.2 Results . . . A17
6 Conclusion . . . A23
References . . . A24

B Symbol-Based Over-the-Air Digital Predistortion Using Reinforcement Learning B1
1 Introduction . . . B3
2 System model . . . B5
2.1 System Model . . . B5
2.2 Indirect Learning Architecture . . . B6
3 Proposed Symbol-Based DPD Optimization Algorithm . . . B7
3.1 Symbol-Based Criterion . . . B7
3.2 Supervised Learning . . . B7
3.3 Reinforcement Learning . . . B7
4 Numerical Results . . . B11
4.1 Setup . . . B11
4.2 Simulation results . . . B12
5 Conclusion . . . B15
References . . . B15

C Time vs. Frequency Domain DPD for Massive MIMO: Methods and Performance Analysis C1
1 Introduction . . . C3
2 System Model . . . C6
2.1 Linear MU-MIMO-OFDM System Model . . . C6
2.2 Nonlinear Power Amplifier in MU-MIMO-OFDM . . . C9
2.3 Linear and Nonlinear Crosstalk . . . C10
3 Relaxation of OOB Linearization Requirements in Massive MIMO Systems . . . C11
3.1 OOB Linearization Relaxation in a Line-of-Sight Channel . . . C12
3.2 OOB Linearization Relaxation in Isotropic Fading Channels . . . C14
4 Frequency Domain Convolutional Neural Network DPD . . . C15
4.1 State-of-the art DPD . . . C15
4.2 Proposed FD-CNN: Architecture . . . C16
4.3 Proposed FD-CNN: Parameter Learning . . . C18

4.4 Proposed FD-CNN: Complexity Analysis and Implementation Discussion . . . C20
4.5 Qualitative Comparison of DPD Strategies . . . C22
5 Numerical Results . . . C24
5.1 Figures of Merit for Linearity in Massive MIMO . . . C24
5.2 Simulation Setup . . . C25
5.3 Simulation results . . . C26
6 Conclusion . . . C34
References . . . C36

D Uplink cell-free massive MIMO OFDM with phase noise-aware channel estimation: separate and shared LOs D1
1 Introduction . . . D3
1.1 Motivation . . . D4
1.2 Contributions . . . D5
1.3 Paper Outline and Notation . . . D6
2 Preliminary . . . D6
2.1 Block Fading Channel Model . . . D6
2.2 Uplink Pilot Assignment . . . D7
3 Phase Noise-Impaired Cell-free MIMO OFDM . . . D8
3.1 Phase Noise Model . . . D8
3.2 Signal Model with Phase Noise . . . D10
3.3 Mismatched Signal Model with Phase Noise . . . D13
4 PN-aware Distributed Channel Estimation . . . D14
4.1 Distributed Joint Channel and CPE Estimation . . . D14
4.2 Distributed Deep Learning Channel Estimation . . . D15
5 PN-aware Centralized Channel Estimation . . . D16
5.1 Centralized LMMSE CPE Estimation given Channels . . . D17
5.2 Distributed LMMSE Channel Estimation Given CPEs . . . D18
6 Uplink Data Transmission and Achievable SE . . . D18
6.1 Scalable Uplink Data Transmission . . . D19
6.2 Uplink Spectral Efficiency . . . D20
7 Numerical Results . . . D20
7.1 Metrics and Baselines . . . D21
7.2 Scenario 1: Separate LOs across APs . . . D21
7.3 Scenario 2: Common LO shared by APs . . . D26
8 Conclusion . . . D29
References . . . D33

Part I

Overview

CHAPTER 1

Introduction

1.1 Background

Wireless communication has become an essential part of modern life, connecting billions of people worldwide. Since the early 1970s, the mobile wireless industry has evolved through several generations, leading to the current fifth generation network (5G) and the future 6G [1], [2]. According to the Ericsson mobility report [3], by the end of 2023, there are over 8.5 billion mobile subscriptions globally, with 1.7 billion being 5G subscriptions. This number is expected to rise to nearly 5.6 billion by 2029, making up 60 percent of a total of 9.3 billion mobile subscriptions. The increasing demand for mobile data calls for faster speeds, lower delays, and more reliable connections. Achieving these goals requires advanced technologies such as massive multiple-input multiple-output (MIMO), which uses a large number of antennas to serve multiple users at the same time, greatly increasing network capacity and making more efficient use of available radio frequencies [4], [5].

However, these technological advancements lead to a significant increase in energy consumption and cost. For instance, massive MIMO systems require numerous radio frequency (RF) chains, each including components such as power amplifiers (PAs), digital-to-analog converters (DACs), and local oscillators (LOs). These components are not only expensive but also consume a lot of energy. The radio access network (RAN), which includes all the equipment connecting mobile devices to the network, accounts for about 73% of the total energy consumption by mobile operators as reported in [6]. As mobile data usage continues to grow and technologies such as massive MIMO become more common, the energy demands and operational costs of

these networks will increase further. This raises a critical question: *How can we meet the performance requirements of wireless communication systems while maintaining or even reducing their energy consumption?*

One effective way to address these issues is to use less expensive hardware components in both cellular and cell-free massive MIMO systems [7]–[10]. While this approach reduces initial costs and energy use, it introduces hardware imperfections that can degrade the quality of communication signals. For example, imperfect PAs can cause nonlinear distortions [11], and LOs can introduce phase noise (PN)[12]. To use these affordable components while still meeting key communication requirements, such as spectral efficiency (SE), and bit error rate (BER) in 5G and 6G systems [13], it is essential to implement effective techniques to reduce the impact of these impairments. However, designing these mitigation techniques presents its own challenges. For example, these techniques consume additional power and must be carefully balanced to mitigate impairments without significantly increasing energy consumption. Also, managing computational complexity while maintaining strong performance is a major challenge [11]. Each hardware impairment uniquely affects the system, requiring tailored models for effective mitigation. For instance, linearizing nonlinear distortions from PAs requires different strategies compared to handling PN from oscillators. The critical research question is: *How can we develop methods that mitigate hardware impairments more efficiently?*

Traditional model-based mitigation methods rely on predefined mathematical models to address hardware impairments in wireless systems. While these methods can be effective, they often struggle to balance performance between complexity [14]–[17]. Accurately modeling complex hardware behaviors can make these approaches overly complicated, leading to higher computational costs and reducing their suitability for real-time applications. Additionally, different types of hardware impairments typically co-existing because hardware components are by nature cascaded in the transistors, making it difficult to develop a single optimal mathematical model that can effectively handle all combined impairments.

Deep learning has proven highly effective in various applications, including natural language processing [18], object detection [19], and autonomous driving [20]. It also holds great potential in communication systems, offering improvements in both performance and efficiency in many scenaios [21]–[27]. Recent research by IBM demonstrates that deploying deep learning methods on a designed analog chip can reduce power consumption by up to 14 times compared to conventional systems for natural language tasks [28]. This breakthrough significantly motivates the practicality and cost-effectiveness of deep learning in many other applications. In the context of hardware impairment mitigation, deep learning presents a promising alternative for addressing these challenges [15], [29]–[31]. Unlike traditional model-based approaches, deep learning models learn the system characteristics directly from data, enabling them to adapt to various types of impairments without requiring detailed,

handcrafted models—particularly in cases where simple models do not exist.

Deep learning-based mitigation methods come with their own challenges. One major issue is the difficulty in obtaining labeled data, as measurements for specific hardware impairments are often inaccessible, yet labeled data is crucial for enabling supervised learning. Another challenge is balancing high performance with low computational complexity, which demands careful design of both the model structure and the training algorithm to address various hardware impairments [32]. Striking this balance is critical for ensuring that deep learning-based mitigation is both effective and practical in real-world scenarios [14], [15]. Ultimately, the choice between model-based and deep learning-based mitigation approaches depends on the specific requirements and constraints of each situation.

In this thesis, deep learning-based techniques for mitigating hardware impairments are explored. Various types of imperfect hardware are included, such as PAs, DACs, lowpass filters (LPFs), and LOs, and antenna chains. The investigation spans different communication scenarios, including single-carrier, orthogonal frequency division multiplexing (OFDM), single-input single-output (SISO), cellular, and cell-free massive MIMO systems. By developing models that effectively correct hardware impairments without imposing excessive computational burdens, we aim to contribute to the advancement of efficient and practical wireless communication technologies.

1.2 Thesis Outline

This thesis is structured as a collection of papers and is divided into two parts. Part I provides an introduction, while Part II contains the included publications. The remainder of Part I is organized as follows: Chapter 2 presents an overview of generic communication systems, various hardware impairment models, classical model-based mitigation techniques, and performance and complexity metrics used for evaluation. Chapter 3 covers the fundamentals of deep learning, focusing on relevant models, implementations, and their applications in communication systems. Chapter 4 explores the use of deep learning for mitigating hardware impairments, with digital predistortion (DPD) as an example of pre-compensating distortion from hardware impairments. Finally, Chapter 5 summarizes the appended publications and suggests potential future research directions.

1.3 Notation

Through Part I of the thesis, the following notation is used. Lowercase and uppercase boldface letters denote column vectors and matrices, such as $\boldsymbol{x}$ and $\boldsymbol{X}$. $\boldsymbol{X}^{\mathsf{T}}$ and $\boldsymbol{X}^{\mathsf{H}}$ denote the transpose and Hermitian transpose of matrix $\boldsymbol{X}$. x_n and x_{n_1,n_2} denotes the n-th element of a vector $\boldsymbol{x}$ and the (n_1, n_2) element of a matrix. $\boldsymbol{x}_{n:n-M}$ denotes a vector consisting of the $(n-M)$-th to the n-th elements of $\boldsymbol{x}$. $x(t)$ denotes a

continuous-time signal at time t. $\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively. $\mathbf{0}_{B\times U}$ and $\boldsymbol{I}_U$ denote the $B \times U$ all-zeros matrix and the $U \times U$ identity matrix, respectively. $\mathbb{E}_x\{\cdot\}$ denotes the expectation with respect to x. $f_{\boldsymbol{\alpha}}(x(t))$ denotes the function $f_{\boldsymbol{\alpha}}$ with continuous-time input $x(t)$ and a parameters vector $\boldsymbol{\alpha}$ (if any). $f_{\boldsymbol{\alpha}}(x_{n_1}, \cdots, x_{n_M}) : \mathbb{C}^M \to \mathbb{R}^N$ denotes a function with a discrete-time complex-valued input vector $[x_{n_1}, \cdots, x_{n_M}]^\mathsf{T}$, a real-valued output vector of length N, and a parameters vector $\boldsymbol{\alpha}$ (if any). $f(\{x_n\}_{n=1}^N)$ denotes a function with input of a set of samples $\{x_n\}_{n=1}^N$. The real and imaginary parts of a complex symbol x_n are denoted by $\Re\{x_n\}$ and $\Im\{x_n\}$. $\mathcal{N}(\mathbf{0}_{U\times 1}, \sigma^2\boldsymbol{I}_U)$ denotes the real-valued zero-mean Gaussian distribution with covariance matrix $\sigma^2\boldsymbol{I}_U$.

There are few inconsistencies in the notation between the introduction part of the thesis and the appended papers. In such cases, we follow the notation used within each individual paper.

CHAPTER 2

Hardware Impairments and Mitigation in Communication Systems

This chapter begins with introducing a generic communication system with its digital processing and analog components in the transmitter and receiver. Details of various hardware impairments are then introduced, including PA nonlinearities, LO PN, in-phase and quadrature (IQ) imbalance, and antenna crosstalk. Subsequently, the traditional model-based mitigation techniques used to mitigate each of these impairments are discussed.

2.1 Communication With Ideal Hardware

Coherent communication is essential in wireless systems, ensuring that the transmitted and received signals maintain consistent time and phase alignment. This coherence is crucial for accurately decoding the transmitted information and achieving high data rates with low error probabilities [33], [34]. However, hardware imperfections at the transmitter and receiver can cause signal distortion, breaking the coherence, and degrading system performance. This section introduces the basics of a generic communication system, including digital processing and hardware components, under the assumption of ideal hardware without impairments.

Figure 2.1 illustrates a generic wireless communication system comprising three main components: the transmitter, the channel, and the receiver. The *transmitter* converts source data into modulated signals optimized for transmission across the *channel*, which can consist of various media, such as air, cables, or fiber optics. As

Figure 2.1: A generic communication system.

the signal propagates through the channel, it experiences degradation such as noise and attenuation, degrading its quality. The *receiver* demodulates the received signal and converts it to the received data, compensating for the distortions introduced by the channel. This basic architecture forms the backbone of various communication networks, including cellular and cell-free massive MIMO systems, which are within the scope of this thesis.

such as

2.1.1 Transmitter

The transmitter is responsible for preparing and sending information bits from the source to the receiver through the communication channel. It comprises several key components that usually work in sequence to ensure accurate and efficient transmission of data.

Encoder

Before the encoding process in a communication system, information such as text, images, and audio is first converted into bits at the application layer. Techniques such as pulse code modulation (PCM) for audio or JPEG compression for images [35], [36] are commonly used to efficiently represent this data by reducing redundancy and minimizing its size. Once converted into bits, the channel encoder performs error correction coding by adding controlled redundancy to protect the data from errors during transmission. Common coding techniques include block codes such as Hamming codes [37], convolutional codes [38], turbo codes [39], and low-density parity-check (LDPC) codes [40]. These codes enable the receiver to detect and correct transmission errors, offering a balance between error correction capability and data rate efficiency.

Modulator

Once the data is encoded, the modulator converts the bits into a signal suitable for transmission over the wireless channel. Modulation involves varying certain properties of a carrier wave, such as amplitude, frequency, or phase, to encode the information. Common modulation schemes include quadrature amplitude modulation (QAM), phase shift keying (PSK), and frequency division multiplexing (FDM) [33],

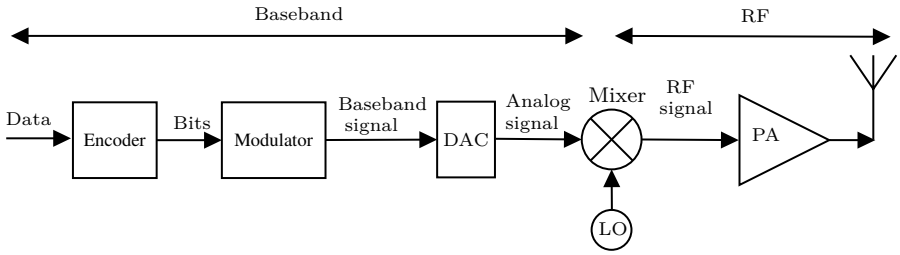


Figure 2.2: A simplified direct RF transmitter in communication systems. The baseband signal is modulated and upconverted directly to the target RF frequency.

[41]. Symbols are represented as points in a constellation diagram, with different points corresponding to different bit patterns. The higher the order of QAM, the more bits can be represented by each symbol, which increases the data rate without requiring additional bandwidth. For instance, 16-QAM can represent 4 bits per symbol, whereas 64-QAM can represent 6 bits per symbol. However, higher-order QAM schemes are more susceptible to noise, which can increase the BER. This trade-off between data rate and reliability is a critical consideration in the design of communication systems, where maintaining a low BER is essential for effective data transmission.

Following modulation, pulse shaping is typically employed to control the bandwidth of the transmitted signal and reduce intersymbol interference (ISI) [33]. Consider a N length sequence of transmission symbols $s_0, \ldots, s_N$ that modulates a pulse $v(t)$ and superposes linearly to form the pulse-shaped signal,

$$x(t) = \sum_{n=0}^{N} s_n v(t - nT_s), \tag{2.1}$$

where T_s is the symbol time. The proper design of $v(t)$ is crucial for achieving the desired performance in communication systems. Effective pulse shaping ensures that the transmitted symbols occupy the minimum necessary bandwidth while maintaining spectral compliance with regulatory requirements. Common pulse shaping filters include the raised cosine filter and the root raised cosine filter [42].

Digital-to-Analog Converter

The DAC converts the discrete-time digital signal into a continuous-time analog signal suitable for further analog processing. It consists of a transcoder followed by a reconstruction stage. The transcoder outputs an analog signal with an amplitude approximation of the digital input, which is then filtered to reconstruct a smooth analog waveform in the reconstruction stage [43], [44]. The analog signal output by

the DAC can be expressed as

$$x^{\mathrm{DAC}}(t) = \sum_{n} \mathcal{Q}(x(nT_{\mathrm{samp}}))f^{\mathrm{DAC}}(t - nT_{\mathrm{samp}}), \tag{2.2}$$

where T_{samp} is the sampling time, $\mathcal{Q}$ is the quantization function based on the Q^{DAC}-bit resolution of the DAC, and $x(nT_{\mathrm{samp}})$ represents the n-th digital sample. The function $f^{\mathrm{DAC}}(t)$ denotes the DAC reconstruction filter, which is typically designed as a LPF. This filter is crucial for smoothing the staircase output and effectively reconstructing the desired analog signal. In the ideal case of an infinite resolution perfect DAC, $x^{\mathrm{DAC}}(t) = x(t)$.

Up-conversion

The mixer shifts the frequency of the baseband or intermediate frequency signal to the desired RF band for transmission, a process known as up-conversion. The mixer mixes the modulated basedband signal, which is the output of the DAC, with a carrier signal generated by the LO. This up-conversion process can be expressed as [45]

$$x^{\mathrm{RF}}(t) = A_x(t)\cos(2\pi f_c t + \phi_x(t)), \tag{2.3}$$

where $x^{\mathrm{RF}}(t)$ is RF signal, $A_x(t)$ and $\phi_x(t)$ are the amplitude and phase of the baseband signal $x^{\mathrm{DAC}}(t)$, respectively, and f_c is the frequency of the LO, known as the carrier frequency.

In quadrature modulation schemes, such as QAM, up-conversion is typically implemented by mixing the in-phase and quadrature components with cosine and sine carriers, respectively. Equation (2.3) can be expanded as [45]

$$\begin{aligned} x_{\mathrm{RF}}(t) &= x_{\mathrm{I}}(t)\cos(2\pi f_c t) + x_{\mathrm{Q}}(t)\sin(2\pi f_c t) && (2.4)\\ &= \Re\{x(t)e^{-j2\pi f_c t}\}, && (2.5) \end{aligned}$$

where $x_{\mathrm{I}}(t)$ and $x_{\mathrm{Q}}(t)$ are the in-phase and quadrature components of the baseband signal $x^{\mathrm{DAC}}(t)$, and $\Re$ is the real operation.

Power Amplifier

The PA amplifies the modulated RF signal to a power level suitable for transmission over the air. It is a crucial component in communication systems, as it ensures that the transmitted signal has sufficient power to reach the intended receiver over the communication channel. The PA must provide sufficient gain while maintaining linearity to prevent distortion, which can significantly degrade signal quality. Nonlinearities in the PA can cause spectral regrowth and adjacent channel interference, which not only degrade the performance of the system in use but also lead to inter-

cell and inter-system interference, negatively impacting neighboring systems [11], [46], [47]. In an ideal scenario, the PA has a linear transfer function, represented as,

$$x^{\mathrm{PA}}(t) = Gx^{\mathrm{RF}}(t), \tag{2.6}$$

where G is the gain of the PA, $x^{\mathrm{RF}}(t)$ is the input RF signal, and $x^{\mathrm{PA}}(t)$ is the output signal. The gain G is typically expressed in dB as:

$$G_{\mathrm{dB}} = 10\log_{10}\left(\frac{P_{\mathrm{out}}}{P_{\mathrm{in}}}\right), \tag{2.7}$$

where $P_{\mathrm{out}} = \mathbb{E}_x\{|x^{\mathrm{PA}}(t)|^2\}$ and $P_{\mathrm{in}} = \mathbb{E}_x\{|x^{\mathrm{RF}}(t)|^2\}$ are the output and input power, respectively.

Antenna

The antenna converts the electrical RF signal into electromagnetic waves for wireless propagation. Its performance depends on gain, radiation pattern, and polarization. Gain indicates how well energy is directed, while the radiation pattern shows energy distribution. Polarization, whether linear or circular, impacts signal reception and overall communication performance [48].

2.1.2 Channel

The communication channel introduces various losses that can degrade the quality of the received signal. These impairments include noise, path loss, fading, interference, and multipath propagation. Path loss represents the reduction in power of the signal as it propagates through space. Multipath fading is caused by the signal may reach the receiver via multiple paths due to reflection, diffraction, and scattering, which can be characterized statistically using models such as Rayleigh and Rician Fading [49]. The noise from the channel is commonly modeled as additive white Gaussian noise (AWGN) thanks to the central limit theorem, where various random processes (such as thermal noise from resistors, shot noise in semiconductor devices, and flicker noise) combine to produce a resultant noise that approximates a Gaussian distribution. Passing the transmitted amplified signal $x^{\mathrm{PA}}(t)$ though an AWGN channel, the received signals $y(t)$ is

$$y(t) = x^{\mathrm{PA}}(t) + w(t), \tag{2.8}$$

where $w(t)$ is a white Gaussian noise process with a power spectral density (PSD) N_0.

2.1.3 Receiver

The receiver captures the transmitted signal and processes it to recover the original information while correcting any errors introduced during transmission. It performs the inverse operations of the transmitter, such as demodulation and decoding, to reconstruct the transmitted data as accurately as possible.

Down-conversion

The received RF signal is typically filtered by a wide bandpass filter (BPF) to remove out-of-band noise and interference. After filtering, the signal is amplified by an low noise amplifier (LNA), which boosts the weak incoming signal with minimal additional noise. The LNA is also an amplifier and can introduce nonlinear distortion. In the down-conversion process, the received RF signal $y(t)$ is converted to an intermediate frequency (IF) or baseband using a mixer that combines the received signal with a signal from the LO. For QAM modulation, a quadrature receiver is often employed, where the signal is down-converted by multiplying it with cosine and sine carriers, $\cos(2\pi f_c')$ and $\sin(2\pi f_c')$, via two mixers. The carrier frequency f_c' of the LO is typically set to match the transmitted frequency, f_c. This quadrature down-conversion separates the signal into its in-phase and quadrature components, represented mathematically by [45]

$$\hat{x}_{\mathrm{I}}(t) = y(t)\cos(2\pi f_c' t), \tag{2.9}$$

and

$$\hat{x}_{\mathrm{Q}}(t) = y(t)\sin(2\pi f_c' t), \tag{2.10}$$

where the outputs $\hat{x}_{\mathrm{I}}(t)$ and $\hat{x}_{\mathrm{Q}}(t)$ are filtered by two LPF filters, $f_{\mathrm{I}}^{\mathrm{LPF}}$ and $f_{\mathrm{Q}}^{\mathrm{LPF}}$, to obtained the in-phase and quadrature components of the received baseband signal,

$$\hat{x}(t) = f_{\mathrm{I}}^{\mathrm{LPF}}\left(\hat{x}_{\mathrm{I}}(t)\right) + 1j \times f_{\mathrm{Q}}^{\mathrm{LPF}}\left(\hat{x}_{\mathrm{Q}}(t)\right). \tag{2.11}$$

In the ideal case of perfect mixer, LO, and LPFs, the down-conversion process can be re-written as

$$\hat{x}(t) = f^{\mathrm{LPF}}\left(y(t)e^{j2\pi f_c t}\right). \tag{2.12}$$

Analog-to-Digital Converter

After down-conversion, the analog-to-digital converter (ADC) converts the continuous-time analog signal into discrete-time digital samples for further digital signal processing. In the ADC, a LPF is used to the continuous-time analog signal to avoid aliasing

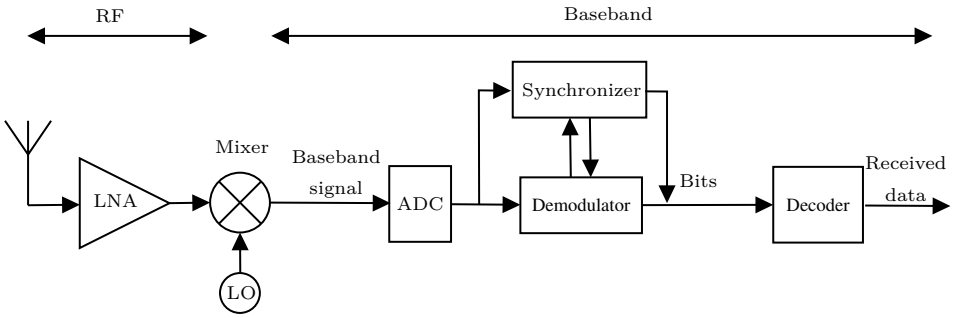


Figure 2.3: A simplified receiver. Not all parts are present in all systems.

effects before it is sampled by the sampling circuit. After which, the discrete-time output of the sampling circuit is fed to a Q^{ADC}-bit quantizer, which outputs quantized digital output. High-resolution ADC leads to high power consumption and may also be non-linear in its behavior, causing further distortion. The power consumption of an ADC increases exponentially with the number of bits and increases linearly with the sampling rate, thus the power consumption of ADCs is especially problematic in systems with large bandwidths and many RF chains, such as millimeter-wave (mm-Wave) systems and massive MIMO systems [7], [9].

Demodulator and Synchronizer

The demodulator recovers the transmitted symbols $\{\hat{s}_n\}_{n=1}^{N}$ from the received baseband signal $\hat{x}(t)$, and before or during this process, synchronization is usually needed to ensure that the timing and frequency of the baseband signal are aligned with the references.

The synchronizer can be implemented by analog circuits or digital techniques. It usually involves the following parts:

- Time synchronization: Both symbol and frame detection are crucial components of time synchronization in communication systems, ensuring the receiver can accurately sample and decode the transmitted signal. Frame detection is a high-level process that identifies the start of a data frame, which contains multiple symbols, ensuring the receiver aligns with the overall data structure. Symbol detection, a lower-level process, focuses on identifying the start of each symbol within the frame to ensure accurate sampling and avoid inter-symbol interference. Both detections are commonly achieved using preamble sequences or pilot symbols, where the receiver correlates the incoming signal with known reference signals to synchronize timing [50], [51].

- Carrier phase and frequency synchronization: Aligning phase and frequency offsets between the received and transmitted signal due to any frequency and

phase mismatches between the received signal and the receiver's LO signal. It is common to use methods such as the phase-locked loop (PLL), which is a control system that iteratively adapts its signal whose phase is locked to the phase of the received signal, both digital and analog PLL systems are used [52]. Generally, the pilot symbols used in the time synchronization can provide coarse synchronization for PLL to fine-tune the frequency and phase offsets [51].

After the synchronization, it is common to use a matched filter to maximize the signal-to-noise ratio (SNR) for optimal symbol detection. The matched filter is designed to "match" the shape of the transmitted signal by correlating the received signal with a time-reversed and conjugated version of the expected transmitted signal. The received baseband signal, $\hat{x}(t)$, pass through the matched filter to obtained the output $\tilde{x}(t)$ by

$$\tilde{x}(t) = \int_{-\infty}^{\infty} \hat{x}(\tau) v'(t-\tau) d\tau. \tag{2.13}$$

Once the baseband signal passes through the matched filter, it is sampled at the correct time instants, typically at the peak of the filter output, corresponding to the symbol intervals to obtain the received symbols $\{\hat{s}_n\}_{n=1}^{N}$. These sampled points are then de-mapped to bits based on a predefined constellation scheme such as QAM. A receiver with matched filters is known as the matched filter receiver, which is an optimal receiver to maximize the SNR in the presence of AWGN [45].

Decoder

The decoder reverses the operations performed by the encoder. In the error correction stage, it removes the redundancy introduced during channel encoding and corrects any transmission errors. Algorithms such as the Viterbi Algorithm are used for decoding convolutional codes [53], belief propagation for LDPC codes, and turbo decoding for turbo codes. The decoder aims to minimize the BER. Finally, the source decoder reconstructs the original information from the corrected binary data.

2.2 Hardware Impairments

Coherent transmission in communication systems relies on the accuracy and performance of hardware components. However, imperfections in these components can introduce impairments that degrade the quality of the transmitted signal. This section discusses several hardware impairments considered in this thesis, including PA nonlinearities, PN, IQ imbalance, and antenna crosstalk (also known as mutual coupling). Each of these impairments affects coherent transmission in different ways, and specific mitigation techniques are needed to reduce their impact, which is covered in Chapter 2.3.

2.2.1 Power Amplifier Nonlinearity

In practice, PAs exhibit nonlinear behavior, especially when operating near their saturation point to maximize efficiency. This nonlinearity causes distortion, leading to spectral regrowth and in-band distortion, which can interfere with adjacent channels and degrade signal quality.

As shown in the transmitter in Figure 2.4, the RF signal $x^{\mathrm{PA}}(t)$ is amplified by the PA, which acts as a nonlinear system with memory effects. The memory effects mean that the PA output at any given time depends not only on the current input, $x^{\mathrm{RF}}(t)$, but also on the inputs in the previous time slots. A generic PA behavior in continuous-time can be represented by the nonlinear function f^{PA}

$$x^{\mathrm{PA}}(t) = f^{\mathrm{PA}}\left(\left\{x^{\mathrm{RF}}(\tau) : \tau \in \left[t - T^{\mathrm{PA}}, t\right]\right\}\right), \tag{2.14}$$

where the input RF signal is with a memory length in time T^{PA}, and $x^{\mathrm{PA}}(t)$ is the PA output signal at time t. Memory effects are mainly due to the frequency-dependent behavior of the PA and thus are more considerable for wideband signals. From the amplitude and phase perspective, this nonlinearity introduces both amplitude modulation to amplitude modulation (AM/AM) and amplitude modulation to phase modulation (AM/PM) distortions. From a spectrum perspective, this nonlinearity leads to both in-band and out-of-band (OOB) errors. The former needs to be minimized to ensure low information transmission errors, while the latter needs to kept low to avoid interference with other systems using adjacent frequencies.

There are many models used to represent PA behavior, such as Volterra series [54]–[57], Wiener [58], Hammerstein [59], memory polynomial (MP) [55], generalized memory polynomial (GMP) [57], neural network (NN)-based [60], [61], where Wiener, Hammerstein, MP, and GMP models are subsets of the full Volterra series with different focus. A widely used PA model is the GMP model, which is a simplified version of the Volterra series that removes certain polynomial cross-terms. Specifically, the GMP model describes the input-output of the PA in discrete-time as

$$\begin{aligned} x_n^{\mathrm{PA}} = & \sum_{q=0}^{K^{\mathrm{PA}}-1} \sum_{m=0}^{M^{\mathrm{PA}}} a_{q,m} x_{n-m}^{\mathrm{RF}} \left|x_{n-m}^{\mathrm{RF}}\right|^q \\ & + \sum_{q=1}^{K^{\mathrm{PA}}-1} \sum_{m=0}^{M^{\mathrm{PA}}} \sum_{g=1}^{G^{\mathrm{PA}}} \left(\acute{a}_{q,m,g} x_{n-m}^{\mathrm{RF}} \left|x_{n-m-g}^{\mathrm{RF}}\right|^q + \tilde{a}_{q,m,g} x_{n-m}^{\mathrm{RF}} \left|x_{n-m+g}^{\mathrm{RF}}\right|^q \right) \end{aligned} \tag{2.15}$$

where x_n^{RF} and x_n^{PA} are the time samples of $x^{\mathrm{RF}}(t)$ and $y^{\mathrm{PA}}(t)$ at time instance nT_{samp}, K^{PA} represents the nonlinear order, M^{PA} is the memory length, and G^{PA} is the cross-term length. The coefficients $a_{q,m}$, $\acute{a}_{q,m,g}$, and $\tilde{a}_{q,m,g}$ are complex-valued parameters. The GMP model (2.15) consists of polynomial terms with memory components, divided into three parts. The first part, known as the MP model [55],

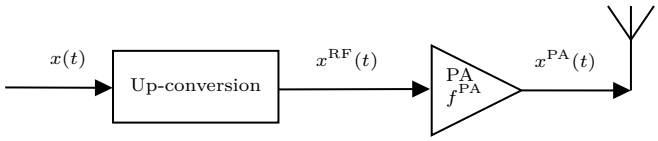
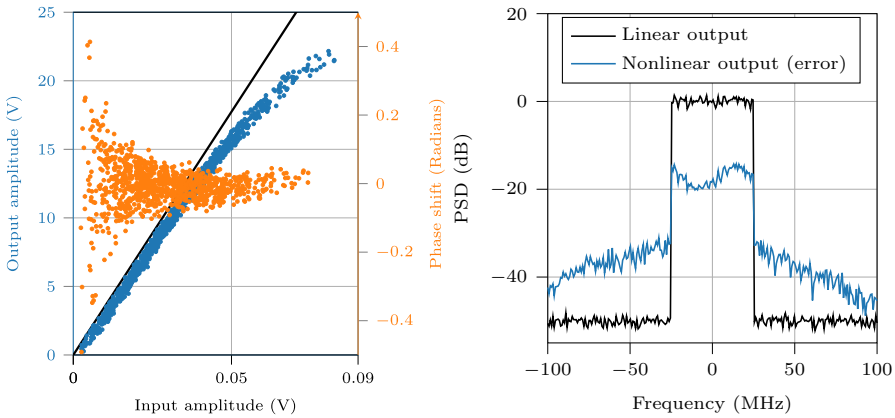


Figure 2.4: Block diagram of the PA in a transmitter. The up-conversion represents the process of converting the baseband signal $x(t)$ to the RF signal $x^{\mathrm{RF}}(t)$, such as an IQ modulator.

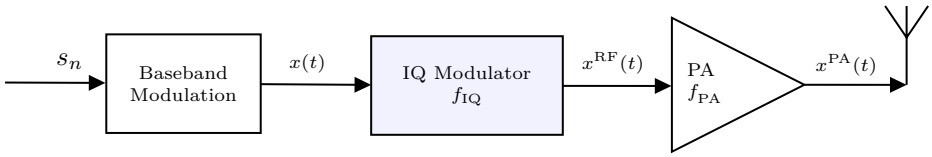


(a) AM/AM and AM/PM characteristics. **(b)** PSDs.

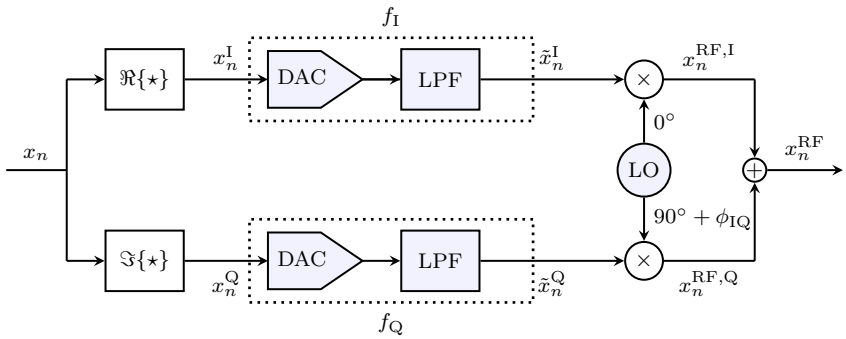
Figure 2.5: AM/AM, AM/PM, and PSDs of a GMP-based PA. Signal bandwidth 50 MHz.

includes $K^{\mathrm{PA}}M^{\mathrm{PA}}$ coefficients, $\{a_{q,m}\}$, capturing interactions of the input signal itself. The second part contains lagging cross-terms, with $K^{\mathrm{PA}}M^{\mathrm{PA}}G^{\mathrm{PA}}$ coefficients, $\{\acute{a}_{q,m}\}$, capturing interactions between current and previous inputs. The final part consists of leading cross-terms, also with $K^{\mathrm{PA}}M^{\mathrm{PA}}G^{\mathrm{PA}}$ coefficients, $\{\tilde{a}_{q,m}\}$, capturing interactions between the current and future inputs.

Figure 2.5a illustrates the nonlinear behavior of the PA, showing the AM/AM and AM/PM characteristics using a GMP model with $K^{\mathrm{PA}} = 7$, $M^{\mathrm{PA}} = 5$, and $G^{\mathrm{PA}} = 3$. The input signal is an OFDM signal with a bandwidth of 50 MHz. The AM/AM and AM/PM characteristics describe how the output amplitude and phase of the PA change in response to variations in input amplitude. Ideally, the PA should exhibit a linear relationship, where the output power increases proportionally with the input power, as depicted by the black solid line. However, due to nonlinearities, the AM/AM and AM/PM plots in Figure 2.5a demonstrate saturation in the output amplitude at high input levels, with corresponding phase distortion in the same region. Furthermore, Figure 2.5b presents the PSD of the PA output, comparing the results with and without nonlinearities. This figure highlights the introduction of both in-band and OOB distortions due to the nonlinear behavior of the PA.



(a) The IQ modulator up-converts baseband signal $x(t)$ to RF signal $x^{\mathrm{RF}}(t)$, which is then amplified by the PA. The baseband modulation represents the process of converting symbols $\{s_n\}$ to the baseband signal $x(t)$ (2.1).



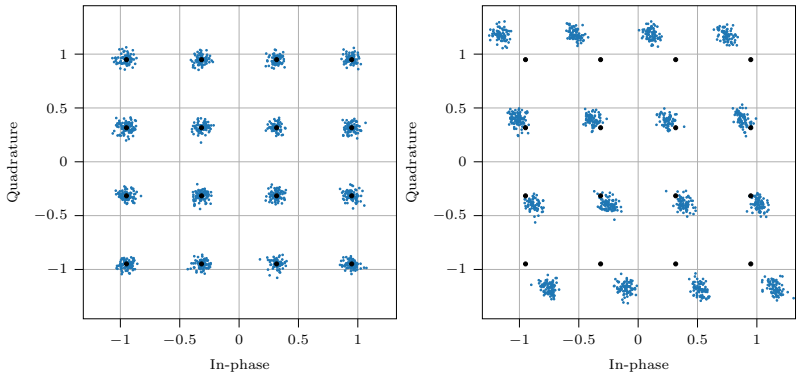
(b) Components of an IQ modulator in the direct conversion transmitter. Imperfect DACs, LPFs, mixers, and LO cause IQ imbalance. Discrete-time modeling is considered.

Figure 2.6: IQ modulator.

2.2.2 IQ Imbalance

IQ imbalance is an important impairment in wireless systems, caused by mismatches in gain and phase between the in-phase and quadrature signal paths of the IQ (de)modulator, also known as quadrature (de)modulator. These imbalances can significantly degrade the performance of wireless communication systems, affecting both transmission and reception [62]–[69]. Figure 2.6a shows the block diagram of an IQ modulator at the transmitter, and Figure 2.6b shows detailed components of the IQ modulator. Gain mismatches typically arise from differences in the DACs or LPFs within the in-phase and quadrature branches, while phase mismatches result from imperfections in the LO during up- and down-conversion processes [67]. Different from the PA that introduces nonlinear distortion, IQ imbalance is primarily modeled as linear distortion. It is generally frequency-dependent, impacting the signal across different frequencies [69].

Figure 2.6b shows a detailed process of the IQ modulator. Although IQ imbalance arises from continuous-time operations on analog signals, it is common and simpler to use a discrete-time model as long as the sampling frequency meets the Nquirist rate. Also, modeling in discrete-time allows for easy mitigation of IQ imbalance in the digital domain. Therefore, discrete-time signals are considered here. For example,



(a) Without IQ imbalance **(b)** With IQ imbalance

Figure 2.7: 16-QAM symbols with and without frequency-selective IQ imbalance in a single-carrier system. Gain mismatch $g_0^{\mathrm{I}}/g_0^{\mathrm{Q}} = 1$ dB, phase mismatch $\phi_{\mathrm{IQ}} = 8^\circ$, and SNR = 20 dB. The function of I and Q branches, f_{I} and f_{Q}, are realized by two FIR filters from [69].

$x_n = x(nT_{\mathrm{samp}})$ and $x_n^{\mathrm{RF}} = x^{\mathrm{RF}}(nT_{\mathrm{samp}})$ are the n-th sample of baseband signal $x(t)$ and RF signal $x^{\mathrm{RF}}(t)$ at sample n, respectively. The discrete-time baseband signal x_n is up-converted by the IQ modulator to passband, where its real and imaginal parts, $x_n^{\mathrm{I}} = \Re x_n$ and $x_n^{\mathrm{Q}} = \Im x_n$ are sent to the I and Q branches of the modulator, respectively. The combination effect of imperfect DAC and LPF is represented by the nonlinear function $f_{\mathrm{I}} : \mathbb{R}^{M^{\mathrm{I}}+1} \to \mathbb{R}$ and $f_{\mathrm{Q}} : \mathbb{R}^{M^{\mathrm{Q}}+1} \to \mathbb{R}$ for the I and Q branches, respectively, where M^{I} and M^{Q} represent the memory lengths of any memory effects in LPFs and DACs. In practice, f_{I} and f_{Q} are commonly modeled by two finite impulse response (FIR) filters, which model the linear frequency-selective IQ imbalance. The input-output relations of the DAC-LPF in the I and Q branches can be expressed in discrete-time as [69]

$$\tilde{x}_n^{\mathrm{I}} = f_{\mathrm{I}}(\boldsymbol{x}_n^{\mathrm{I}}) = \sum_{m=0}^{M^{\mathrm{I}}} g_m^{\mathrm{I}} x_{n-m}^{\mathrm{I}}, \tag{2.16}$$

$$\tilde{x}_n^{\mathrm{Q}} = f_{\mathrm{Q}}(\boldsymbol{x}_n^{\mathrm{Q}}) = \sum_{m=0}^{M^{\mathrm{Q}}} g_m^{\mathrm{Q}} x_{n-m}^{\mathrm{Q}}, \tag{2.17}$$

where $\boldsymbol{x}_n^{\mathrm{I}} \triangleq [x_n^{\mathrm{I}}, \cdots, x_{n-M^{\mathrm{I}}}^{\mathrm{I}}]^{\mathsf{T}}$ and $\boldsymbol{x}_n^{\mathrm{Q}} \triangleq [x_n^{\mathrm{Q}}, \cdots, x_{n-M^{\mathrm{Q}}}^{\mathrm{Q}}]^{\mathsf{T}}$ are the input signal at time sample n with memory, $\tilde{x}_n^{\mathrm{I}}$ and $\tilde{x}_n^{\mathrm{Q}}$ are the corresponding output at time sample n of the DAC-LPF for the I and Q branches, respectively. g_m^{I} and g_m^{Q} are the coefficients of the in-phase and quadrature FIR filters, respectively, M^{I} and M^{Q} are the number of filter taps.

The DAC-LPF outputs, $\tilde{x}_n^{\mathrm{I}}$ and $\tilde{x}_n^{\mathrm{I}}$, are up-converted by two separate mixers using

the carrier signals from the LO. Ideally, the carrier signals from LO should be exactly 90 degrees out of phase, however, imperfections in the LO, mixer, and other analog components can cause a phase imbalance ϕ_{IQ}. The IQ modulator output at time sample n, x_n^{RF}, is given by

$$x_n^{\mathrm{RF}} = x_n^{\mathrm{RF,I}} + 1j x_n^{\mathrm{RF,Q}}, \tag{2.18}$$

where $x_n^{\mathrm{RF,I}} = \tilde{x}_n^{\mathrm{I}} - \sin(\phi_{\mathrm{IQ}})\tilde{x}_n^{\mathrm{Q}}$ and $\tilde{x}_n^{\mathrm{Q}} = \cos(\phi_{\mathrm{IQ}})\tilde{x}_n^{\mathrm{Q}}$. (2.18) can be rewritten as

$$\begin{aligned} x_n^{\mathrm{PA}} &= \tilde{x}_n^{\mathrm{I}} - \sin(\phi_{\mathrm{IQ}})\tilde{x}_n^{\mathrm{Q}} + 1j\cos(\phi_{\mathrm{IQ}})\tilde{x}_n^{\mathrm{Q}} \\ &= \tilde{x}_n^{\mathrm{I}} + 1j e^{1j\phi_{\mathrm{IQ}}}\tilde{x}_n^{\mathrm{Q}} \\ &= \sum_{m=0}^{M^{\mathrm{I}}} g_m^{\mathrm{I}} x_{n-m}^{\mathrm{I}} + 1j e^{1j\phi_{\mathrm{IQ}}} \sum_{m=0}^{M^{\mathrm{Q}}} g_m^{\mathrm{Q}} x_{n-m}^{\mathrm{Q}}. \end{aligned} \tag{2.19}$$

(2.19) can be rewritten in a compact form using the nonlinear function $f_{\mathrm{IQ}} : \mathbb{C}^{M^{\mathrm{IQ}}+1} \to \mathbb{C}$ with memory length $M^{\mathrm{IQ}} \triangleq \max\left(M^{\mathrm{I}}, M^{\mathrm{Q}}\right)$ to represent the IQ modulator system,

$$x_n^{\mathrm{RF}} = f_{\mathrm{IQ}}\left(x_n, \ldots, x_{n-M^{\mathrm{IQ}}}\right) = f_{\mathrm{IQ}}\left(\boldsymbol{x}_{n:n-M^{\mathrm{IQ}}}\right), \tag{2.20}$$

where $\boldsymbol{x}_{n:n-M^{\mathrm{IQ}}} = [x_n, \ldots, x_{n-M^{\mathrm{IQ}}}]^{\mathsf{T}}$. In the case of a perfect IQ modulator, $\phi_{\mathrm{IQ}} = 0$, $M^{\mathrm{I}} = M^{\mathrm{Q}} = 0$, $g_0^{\mathrm{I}} = g_0^{\mathrm{Q}} = 1$.

Figure 2.7 presents an example of a constellation diagram of 16-QAM with and without linear, frequency-dependent IQ imbalance. The imbalance includes a gain mismatch of $g_0^{\mathrm{I}}/g_0^{\mathrm{Q}} = 1$ dB, a phase mismatch of $\phi_{\mathrm{IQ}} = 8°$, and an SNR of 20 dB. The results demonstrate how IQ imbalance distorts the ideal grid structure of the 16-QAM constellation, affecting both the phase and amplitude of the symbols. IQ imbalance also degrades channel estimation accuracy, impairing beamforming, reducing beam directivity, increasing, for instance, OOB distortion [63], [67], [70]–[72].

2.2.3 Phase Noise

Maintaining phase coherence between the transmitter and receiver is crucial for communication blocks such as demodulation and beamforming. For example, in massive MIMO systems, phase coherence is necessary for accurate beamforming to enable signals from multiple antennas to combine constructively, achieving array gain and ensuring high spectral efficiency. However, imperfections in LOs introduce PN at both the transmitter and receiver in the process of up-conversion and down-conversion, which disrupts this phase coherence and degrades system performance. For instance, in OFDM systems, PN causes inter-carrier interference (ICI), leading to signal overlap between neighboring subcarriers and reducing spectral efficiency [73], [74]. Similarly, in multi-user (MU) massive MIMO, PN results in inter-user interference (ICI), mak-

ing it harder to separate users and ultimately reducing data rates [74], [75]. The effects of PN become more severe at higher frequencies, such as mmWave bands, due to the manufacturing properties of LOs. Further research has explored the impact of PN in both cellular and cell-free massive MIMO systems [76]–[82], as well as in OFDM [73], [83]–[86] and single-carrier systems [87], [88]. Advanced compensation techniques are necessary to mitigate PN and maintain system performance.

Depending on the specific implementation of the LO, different PN models may be applied. Typically, two practical LO configurations are used: a free-running oscillator or a PLL synthesizer. The PN from a free-running oscillator at a UE can be described by a discrete-time Wiener process as[1],

$$\phi_t = \phi_{t-1} + \delta_{\phi_t}, \tag{2.21}$$

where the incremental PN δ_{ϕ_t} at time sample t follows a Gaussian distribution $\mathcal{N}(0, \sigma_\phi^2)$. When the UE up-converts the baseband signal $x(t)$ to RF following (2.5), due to the PN, the corresponding discrete-time RF signal with PN at time sample t is given by

$$x_t^{\mathrm{RF}} = \Re\{x_t e^{-j2\pi f_c t T_{\mathrm{samp}}} e^{j\phi_t}\}. \tag{2.22}$$

Similarly, the PN from a base-station (BS) or access point (AP) at time sample t can be modeled as,

$$\varphi_t = \varphi_{t-1} + \delta_{\varphi_t}, \tag{2.23}$$

where the incremental PN δ_{φ_t} also follows a Gaussian distribution $\mathcal{N}(0, \sigma_\varphi^2)$. When the BS down-converts the received RF signal $y(t)$ to baseband following (2.12), due to the PN, the corresponding discrete-time baseband signal with PN at time sample t is given by

$$\hat{x}_t = f_{\mathrm{LPF}}\big(y_t e^{j2\pi f_c t T_{\mathrm{samp}}} e^{j\varphi_t}\big). \tag{2.24}$$

The PN-impaired signal from the UE is received at the BS, where the PN at time sample t from the UE and BS are combined as [2]

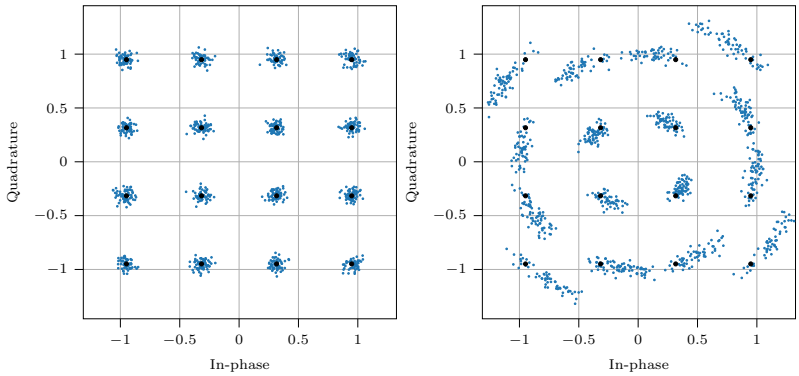
$$\theta_t = \phi_t + \varphi_t, \tag{2.25}$$

where θ_t is the received PN at the BS.

The variance of the PN increments is related to the quality of the LO, carrier

[1] In this subsection, a different notation is used to simplify the model of PN in OFDM systems: t represents the time sample index in the time-domain (TD), while n is used to denote the subcarrier index in the frequency-domain (FD).

[2] Assume perfect time synchronization, frequency synchronization, and ideal hardware components, with only PN from LOs.



(a) Without PN. **(b)** With PN.

Figure 2.8: 16-QAM symbols with and without PN in a single-carrier system. PN variance $\sigma^2 = 1$, SNR = 20 dB.

frequency, and sampling time, which can be modeled as [73]

$$\sigma_i^2 = 4\pi^2 f_c^2 \gamma_i T_s, \quad \text{for } i \in \{\phi, \varphi\}, \tag{2.26}$$

where γ_ϕ and γ_φ are constants related to the LO quality of the UE and BS. An example value is 10^{-17} [76]. It is important to note that in cellular massive MIMO networks, multiple antennas might share a common LO, while in distributed antenna deployments, such as in cell-free massive MIMO networks, each antenna may use an individual LO. Consequently, the PN associated with different antennas may be the same or independent. Figure 2.8 shows the constellation diagrams of received 16-QAM symbols with and without PN, i.e., demodulated from (2.12) and (2.24). PN variance $\sigma^2 = 1$, and SNR = 20 dB. The constellation affected by PN exhibits a noticeable rotation of the symbols. In the case of severe PN, the symbol points may also spread out more, complicating detection and leading to a rise in symbol error rate (SER).

Considering an OFDM system with N subcarriers, the PN realizations for an arbitrary OFDM symbol forms a PN vector, $\boldsymbol{\theta} = [\theta_0, \cdots, \theta_t, \cdots, \theta_{N-1}]^\mathsf{T}$. The corresponding PN vector that affects the carrier signal is

$$e^{j\boldsymbol{\theta}} \triangleq [e^{j\theta_0}, \cdots, e^{j\theta_{N-1}}]^\mathsf{T}, \tag{2.27}$$

and its corresponding FD phase-drift vector is expressed as $\boldsymbol{J} \in \mathbb{C}^N$. Its i-th entry J_i, for $i = -N/2, \cdots, N/2 - 1$, is given by [73, Eq. (6)]

$$J_i = \frac{1}{N} \sum_{t=0}^{N-1} e^{j\theta_t} e^{-j2\pi ti/N}, \tag{2.28}$$

where the first phase-drift element, J_0, is also known as the common phase error (CPE).

Consider an MU case with U user equipments (UEs). In the absence of PN, the received uplink symbols from UE u to BS over subcarrier n, y_n, is given by

$$y_n = \sum_{u=1}^{K} h_{u,n} s_{u,n}, \tag{2.29}$$

where $h_{u,n}$ is the channel between UE u and the BS, and $s_{u,n}$ is the transmitted symbol from UE u at subcarrier n. In the presence of PN, (2.29) can be written as

$$y_n = \sum_{u=1}^{U} \left(J_{u,0} h_{u,n} s_{u,n} + \zeta_{u,n} \right), \tag{2.30}$$

where $J_{u,0}$ denotes the CPE of UE u, and the ICI component over subcarrier n, $\zeta_{u,n}$, is given by

$$\zeta_{u,n} = \sum_{n'=0, n' \neq n}^{N-1} J_{u,n-n'} h_{u,n'} s_{u,n'}. \tag{2.31}$$

Comparing (2.30) with (2.29), one notes that the presence of PN introduces the channel aging effect, meaning the effective channel begins to deviate from the true channel h, with this deviation increasing over time. PN disrupts the OFDM orthogonality and the pilot orthogonality between UEs, leading to ICI, which negatively impacts channel estimation as well as the corresponding uplink combining and downlink precoding.

2.2.4 Antenna Crosstalk

In MIMO systems, antenna crosstalk, also referred to as *mutual coupling*, occurs due to the interaction between closely spaced antennas and signal paths within an array [89], [90]. This crosstalk can cause both nonlinear and linear distortions. Antenna crosstalk presents a time-varying load-impedance to each PA through mutual coupling, which in terms creates nonlinear distortion. This particular source of distortion is not covered in most conventional PA modeling, and requires specific attention, both in terms of modeling and compensation [91]. Linear crosstalk, on the other hand, typically arises after the PA [89]. Antenna crosstalk can negatively impact system capacity [92], carrier frequency offset estimation [93], channel estimation [94], beamforming accuracy [95], [96], and PA linearization [91]. Specifically, within the scope of this thesis, antenna crosstalk affects beamforming by causing beams to become less focused or misaligned, and nonlinear crosstalk worsens the PA's nonlinear characteristics. This results in spectral regrowth, increased in-band errors, and unwanted

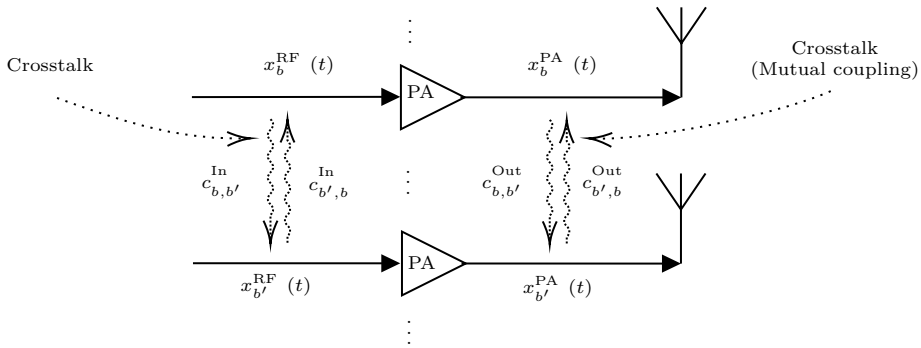


Figure 2.9: Antenna crosstalk between RF chains happens before and after the PAs. Linear discrete-time antenna crosstalk models (2.32) and (2.35) are used.

emissions in the OOB.

The effects of antenna crosstalk are challenging to model accurately because it originates from multiple sources and interacts with other hardware impairments such as nonlinear PAs [90], [91]. Nonlinear crosstalk arises from nonlinear elements in the RF chain, such as the PA, which produce harmonics and intermodulation products that couple into nearby antenna elements. Figure 2.9 illustrates antenna crosstalk between RF chains before and after the PAs. Although antenna crosstalk affects analog signals in continuous-time, it is common to model this effect in discrete-time for further mitigation in the digital domain. In an array with B RF chains and antennas, the impact of crosstalk on the PA input signal $x_{b,n}^{\mathrm{RF}}$ at time sample n and RF chain b can be described by a discrete-time nonlinear function $f_b^{\text{Cross-In}} : \mathbb{C}^B \to \mathbb{C}$,

$$\tilde{x}_{b,n}^{\mathrm{RF}} = f_b^{\text{Cross-In}}(\boldsymbol{x}_n^{\mathrm{RF}}) \tag{2.32}$$

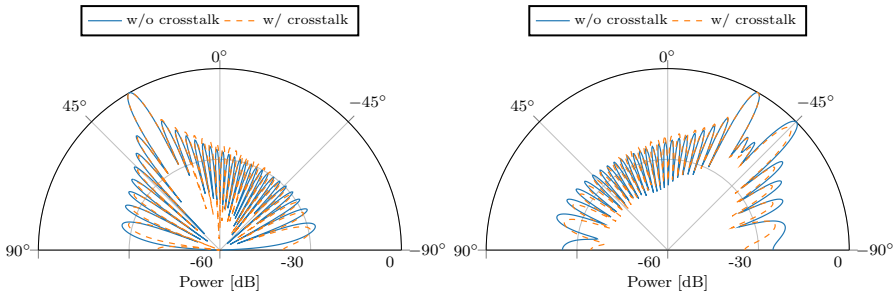
where $\boldsymbol{x}_n^{\mathrm{RF}} = [x_{1,n}^{\mathrm{RF}}, \cdots, x_{B,n}^{\mathrm{RF}}]^{\mathsf{T}} \in \mathbb{C}^B$, and $\tilde{x}_{b,n}^{\mathrm{RF}}$ is the input RF signal of PA b at time sample n with crosstalk, which introduces nonlinear crosstalk when the output $\tilde{x}_{b,n}^{\mathrm{RF}}$ is passing through a nonlinear PA such as (2.15).

The antenna crosstalk between PA output signals, i.e., mutual coupling, can be modeled by a discrete-time nonlinear function $f_b^{\text{Cross-Out}} : \mathbb{C}^B \to \mathbb{C}$,

$$\tilde{x}_{b,n}^{\mathrm{PA}} = f_b^{\text{Cross-Out}}(\boldsymbol{x}_n^{\mathrm{PA}}), \tag{2.33}$$

where $\boldsymbol{x}_n^{\mathrm{PA}} = [x_{1,t}^{\mathrm{PA}}, \cdots, x_{B,t}^{\mathrm{PA}}]^{\mathsf{T}} \in \mathbb{C}^B$, and $\tilde{x}_{b,t}^{\mathrm{PA}}$ is the output signal of PA b at time sample n with crosstalk.

In practice, crosstalk between RF chains is commonly modeled as linear relations. A simplified yet accurate and very common linear crosstalk model is adopted [89]. In this model, only linear crosstalk between RF chains is modeled, which reduces



(a) 1 UE. **(b)** 2 UEs

Figure 2.10: Radiation patterns from a uniform linear array consisting of 32 antennas with zero-forcing precoding in a LOS channel, comparing scenarios with and without linear antenna crosstalk.

$f_b^{\text{Cross-In}}$ to a linear function, and (2.32) can be rewritten as [89]

$$\tilde{x}_{b,t}^{\text{RF}} = x_{b,t}^{\text{RF}} + \sum_{\substack{b'=0 \\ b' \neq b}}^{B-1} c_{b,b'}^{\text{In}} x_{b',t}^{\text{RF}}. \tag{2.34}$$

Here $c_{b,b'}^{\text{In}} \in \mathbb{C}$ is the crosstalk parameter between the inputs of PA b and b'. In practice, this crosstalk parameter's amplitude represents the crosstalk strength, which is proportional to the square of the physical distance between two RF chains. The phase of the crosstalk parameter depends on the effective physical distance relative to the wavelength [89]. A linear crosstalk between antennas model reduces $f_b^{\text{Cross-Out}}$ to a linear function, and (2.33) can be rewritten as [89]

$$\tilde{x}_b^{\text{PA}} = x_b^{\text{PA}} + \sum_{\substack{b'=0 \\ b' \neq b}}^{B-1} c_{b,b'}^{\text{Out}} x_{b'}^{\text{PA}}, \tag{2.35}$$

where $c_{b,b'}^{\text{Out}} \in \mathbb{C}$ is the crosstalk parameter between the outputs of PA b and b'. Figure 2.10 shows the radiation beampatterns from a uniform linear array consisting of $B = 32$ antennas with zero-forcing precoding in an LOS channel, comparing the case with and without antenna crosstalk. Only linear antenna crosstalk is considered with ideal PAs. It shows that the antenna crosstalk distorts the beamforming accuracy, especially for the sidelobes and the case of 2 UEs.

2.2.5 Other Hardware Impairments

Other hardware impairments, such as those from ADCs and DACs, are commonly modeled as AWGN. This modeling approach is especially prevalent in OFDM sys-

tems, even for some deterministic hardware impairments. The reasoning behind this approximation is that although hardware impairments originate in the TD, they spread across the FD when transformed into the FD. This spread causes the impairments at any single subcarrier or frequency to reflect contributions from various time-domain effects. As a result, using the central limit theorem, these impairments can be reasonably modeled as Gaussian distributed noise in the FD, aligning with the assumption of AWGN. This approach is widely employed in the literature [64], [79], [97]–[101] thanks to its simplicity and reasonable accuracy. It has proven effective in assessing the impact of such impairments on massive MIMO systems compared to more deterministic hardware impairment models [102]. Although this model is effective for evaluating the impact of hardware impairments, it is less practical for designing mitigation strategies.

2.3 Classical Mitigation Techniques

In this section, classical model-based mitigation techniques to mitigate the PA nonlinearity, IQ imbalance, PN, and antenna crosstalk are discussed.

Digital Predistorion

To address the nonlinear behavior of PAs, DPD is widely used. It works by introducing pre-distortion in the digital domain to counteract the distortion caused by PA nonlinearity, thereby linearizing the overall system response. Implementing DPD reduces spectral regrowth, improves in-band signal quality, and minimizes adjacent channel interference. These improvements are critical for maintaining low error rates, high data rates, and efficient spectrum use [11], [103]. Figure 2.11 shows the block diagram of the cascaded DPD and PA system, assuming ideal hardware between the DPD and the PA. Additionally, Figure 2.11 provides a simplified illustration of the inverse behavior of the DPD and the PA without considering memory effects.

The DPD behavior in continuous-time can be represented by the nonlinear function f^{DPD}

$$x^{\mathrm{DPD}}(t) = f^{\mathrm{DPD}}\Big(\big\{x(\tau) : \tau \in \big[t - T^{\mathrm{DPD}}, t\big]\ \big\}\Big), \tag{2.36}$$

where the baseband input signal of DPD is with a memory length in time T^{DPD}, and $x^{\mathrm{DPD}}(t)$ is the DPD output signal at time t. The output $x^{\mathrm{DPD}}(t)$ is sent to the PA. There are various DPD models to represent the DPD function f_{DPD}, including the lookup table (LuT)[104], Volterra series-based model [105], MP model [55], [106], GMP [57], [107], Wiener model [108], and Hammerstein model [109]. Among these, the GMP model [57] is recognized for offering the best tradeoff between complexity and performance, making it widely adopted. In practice, these DPD models usu-

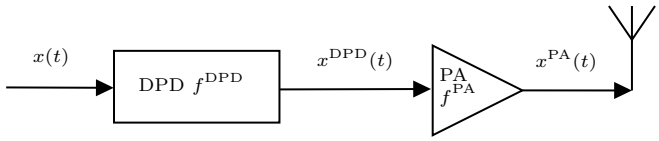


Figure 2.11: Block diagram of the DPD-PA system. Note that DPD is in the digital domain, and PA is in the analog domain. Components between DPD and PA such as DAC and mixers are omitted here.

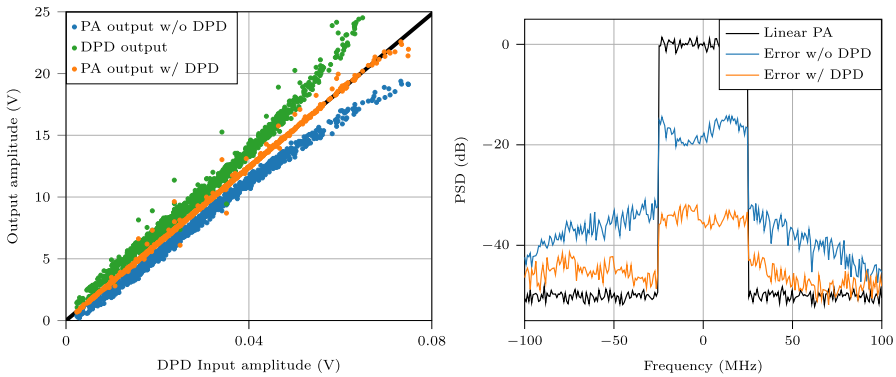
ally represent the continuous-time DPD function f^{DPD} in discrete-time to simplify analysis and implementation.

Specifically, for an arbitrary PA, the input-output relation of the DPD using the GMP model in discrete-time at time sample n can be expressed as

$$\begin{aligned} x_n^{\mathrm{DPD}} = & \sum_{k=0}^{K^{\mathrm{DPD}}-1} \sum_{m=0}^{M^{\mathrm{DPD}}} d_{q,m} x_{n-m} \left|x_{n-m}\right|^q \\ & + \sum_{q=1}^{K^{\mathrm{DPD}}-1} \sum_{m=0}^{M^{\mathrm{DPD}}} \sum_{g=1}^{G^{\mathrm{DPD}}} \left(\acute{d}_{q,m,g} x_{n-m} \left|x_{n-m-g}\right|^q + \tilde{d}_{q,m,g} x_{n-m} \left|x_{n-m+g}\right|^q \right), \end{aligned} \tag{2.37}$$

where x_n and x_n^{DPD} are the discrete-time DPD input and output signal at time sample n. K^{DPD} is the nonlinear order parameter, M^{DPD} is memory length, G^{DPD} is the cross-term length, $d_{q,m}$, $\acute{d}_{q,m,g}$, and $\tilde{d}_{q,m,g}$ are complex-valued coefficients. Figure 2.12 gives an example of using a GMP-based DPD to linearize a GMP-based PA used in Figure 2.5. Specifically, the AM/AM results in Figure 2.12a illustrate that DPD tends to increase the output amplitude to compensate for the reduction in the nonlinear PA, and the PSD results in Figure 2.12b show that DPD effectively reduces both the in-band and OOB distortion.

The implementation of DPD brings significant computational challenges, which grow with the number of PAs and the signal bandwidth. This issue is especially noticeable in large antenna arrays, where up to hundreds of PAs need to be linearized. While each PA in such arrays typically uses less power compared to traditional SISO systems, the processing demands for DPD remain the same, creating an imbalance where DPD consumes a disproportionate amount of power [11]. Using separate DPD units for each PA increases the overall computational load. To reduce this complexity, shared DPD methods, where one DPD is used across multiple PAs in a subarray, have been explored, but this often requires hybrid or analog beamforming, which can reduce performance compared to digital precoding. Additionally, DPD often requires oversampling at rates up to five times the signal bandwidth to control nonlinear PA effects such as spectral regrowth, which further drives up the power consumption.



(a) AM/AM. **(b)** PSDs.

Figure 2.12: AM/AM and PSD results with and without DPD. Both PA and DPD are GMP models. Signal bandwidth 50 MHz.

These factors highlight the growing need for simplified DPD solutions that maintain performance while reducing complexity.

IQ Imbalance Compensation

IQ imbalance can severely degrade the performance of communication systems. Compensation for IQ imbalance can be implemented either at the receiver [110], [111] or at the transmitter before transmission [67], [69]. An example of IQ imbalance compensation at the transmitter is adopted using the approach from [67], which is shown in Figure 2.13. IQ pre-compensator compensates the IQ imbalance in the digital domain using models such as FIR filters or Volterra series. Before it is applied, its parameters are estimated by minimizing the mean square error (MSE) between the baseband signal $x(t)$ and the IQ imbalance impaired baseband signal $y(t)$ collected via a feedback path

$$\hat{\boldsymbol{\alpha}}^{\mathrm{IQ}} = \arg\min_{\boldsymbol{\alpha}^{\mathrm{IQ}}} \mathbb{E}_x \left\{ |x(t) - y(t)|^2 \right\}, \tag{2.38}$$

where $\boldsymbol{\alpha}^{\mathrm{IQ}}$ denotes the parameters of any IQ pre-compensator. The estimation can be performed using algorithms such as least squares to derive an inverse model of the IQ imbalance, f_{IQ}. A straightforward method to estimate this inverse model is to input the received signal $y(t)$ into the IQ pre-compensator and use the original baseband signal $x(t)$ as the reference output during the estimation process [67]. Figure 2.13 shows the constellation diagram of transmitted 16 QAM symbols with and without an IQ pre-compensator. A frequency-selective IQ imbalance is used, with a gain mismatch $g_0^{\mathrm{I}}/g_0^{\mathrm{Q}} = 1$ dB, phase mismatch $\phi_{\mathrm{IQ}} = 8°$, and SNR = 20 dB. The function of I and Q branches, f_{I} and f_{Q}, are realized by two FIR filters from [69]. A frequency-flat

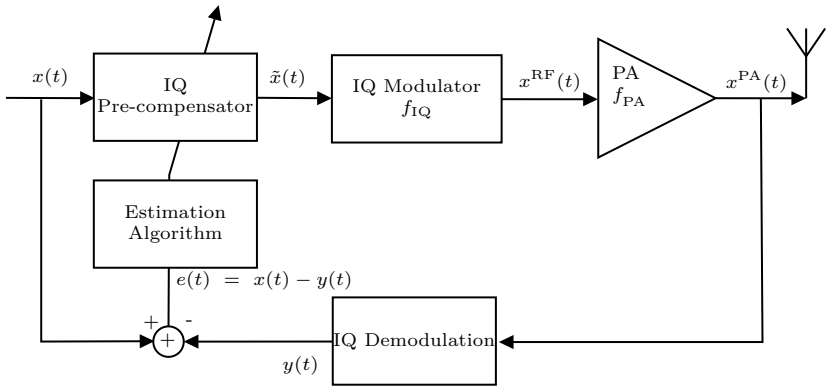


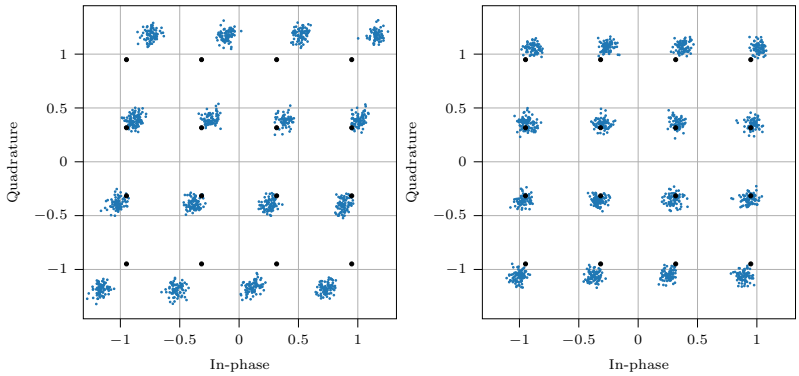
Figure 2.13: IQ imbalance pre-compensator at the transmitter. The pre-compensator is in the digital domain, whose parameters are estimated by minimizing the MSE between the baseband signal $x(t)$ and the IQ imbalance impaired baseband signal $y(t)$ collected via a feedback path.

linear IQ pre-compensator model is estimated using the least squares algorithm. It shows that residual IQ imbalance remains after the IQ pre-compensator, mainly due to the fact that the IQ imbalance is frequency-dependent while the pre-compensator is frequency-flat.

Phase Noise Mitigation

Mitigating PN is crucial, especially in high-frequency systems such as mmWave bands since the variance of the PN grows with the square of the carrier frequency. PN mitigation is generally carried out at the receiver using techniques such as pilot-based estimation, which relies on pilots to estimate PN. As discussed in Chapter 2.2.3, PN leads to both CPE and ICI in OFDM systems. The PN mitigation can be done by only mitigating the CPE [112], [113] or more advanced mitigation including ICI mitigation [73], [86]. A more complex iterative approach uses decision-directed methods [73], [114], [115], which not only uses pilots but also refines PN estimates by iteratively incorporating decisions on the transmitted data symbols to enhance estimation accuracy. The choice of PN mitigation techniques depends on system requirements and cost considerations. For example, in the 3GPP standardization [116], pilot-based estimation is employed, where the phase tracking reference signals (PTRS) are strategically placed with specified spacing in both time (between every 2 or 4 resource blocks) and frequency (between 1, 2, or 4 OFDM symbol times). These configurations can be selected based on factors such as channel conditions, including mobility, and hardware impairments such as LO quality.

The problem of estimating the phase shifts J_i up to a certain order can be formulated as a linear estimation problem. For a l-th order estimation, the goal is to



(a) Without IQ imbalance compensator **(b)** With IQ imbalance compensator

Figure 2.14: Constellation diagram of transmitted 16-QAM symbols with frequency-selective IQ imbalance, under the case with and without frequency-flat IQ imbalance compensator. Gain mismatch $g_0^{\mathrm{I}}/g_0^{\mathrm{Q}} = 1$ dB, phase mismatch $\phi_{\mathrm{IQ}} = 8^\circ$, SNR = 20 dB. Residual IQ imbalance remains due to residual frequency-selective IQ imbalance.

estimate the vector $\boldsymbol{J}_l = [J_0, J_1, J_{-1}, \cdots, J_l, J_{-l}]^{\mathsf{T}}$, which contains $(2l+1)$ elements of the phase shift vector $\boldsymbol{J}$. For example, the 0-th order phase shift represents the CPE. This estimation can be solved using the least squares algorithm as detailed in [73], and Figure 2.15 demonstrates the reconstructed PN using different orders of estimated phase shifts $\hat{\boldsymbol{J}}_l$. The figure illustrates that even using the second-order phase drift coefficients yields a close approximation of the true PN realization. This is due to the lowpass nature of PN, where only a few frequency components are necessary to accurately capture its frequency response.

Antenna Crosstalk Mitigation

Increasing the physical distance between antennas is an effective way to reduce crosstalk, making careful antenna placement crucial based on the operating frequency and environmental factors. However, at high frequencies such as mmWave, the small spacing between antennas can lead to significant coupling effects [90], [117]. Another approach involves using decoupling networks, which employ components such as inductors, capacitors, and resistors to mitigate coupling and improve antenna isolation [118]. These networks use techniques such as phase shifts, impedance adjustments, and electrical isolation to separate antennas and reduce crosstalk. However, incorporating these components can increase design costs, which is a challenge for large-scale systems such as massive MIMO that prioritize integrated designs and aim to avoid expensive and bulky components, such as isolators between PAs and antennas, to keep system complexity and cost low [119].

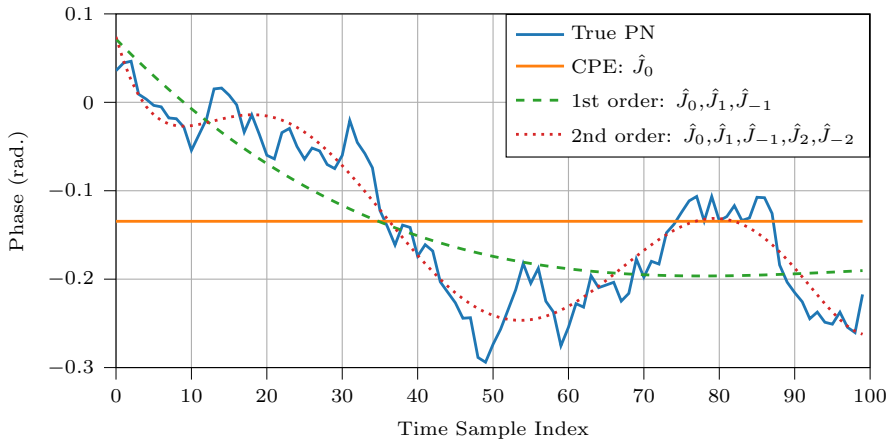


Figure 2.15: Example of reconstructed PN in the TD using different orders of estimated phase shifts $\{\hat{J}_i\}$ in the FD. The phase shifts are estimated in the FD, which is then used to reconstruct the PN in the TD. OFDM with 100 subcarriers.

Except for these decoupling techniques in the analog domain, similar to IQ precompensator and DPD, antenna crosstalk can also be mitigated in the digital domain [89], [91], [120]. The mutual coupling between antennas creates a non-constant load impedance for the PAs, which in turn alters their non-linear behavior. For instance, linear antenna crosstalk between RF chains before the PA leads to nonlinear distortion because of passing through the nonlinear PA. Therefore one can jointly mitigate the antenna crosstalk and the PA nonlinearity using DPD [89], [91], which can effectively mitigate both linear and nonlinear crosstalk. Here, the antenna crosstalk parameters can be represented in a matrix form

$$\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_B] \in \mathbb{C}^{B\times B}, \tag{2.39}$$

where the crosstalk parameters of the other $B-1$ antenna to the b-th antenna is

$$\boldsymbol{c}_b = [c_{b,1}^{\text{Out}}, \cdots, c_{b,b-1}^{\text{Out}}, 1, c_{b,b+1}^{\text{Out}}, \cdots, c_{b,B}^{\text{Out}}]^{\mathsf{T}} \in \mathbb{C}^{B}. \tag{2.40}$$

The crosstalk parameter between the PA output b and b', $c_{b,b'}^{\text{Out}}$, is defined in (2.35). This antenna crosstalk matrix can be estimated using measurements at the array side in an anechoic chamber or at the receiver side [121], [122]. The former is more accurate and more robust to noise but is time-consuming and costly. The latter is less costly and more adaptive to different antenna configurations and channels. With an estimated mutual coupling matrix, $\hat{\boldsymbol{C}}$, one can mitigate the antenna crosstalk. In practice, other hardware impairments such as nonlinear PAs affect the estimation, providing potential for nonlinear estimation using deep learning method [123].

2.4 Performance and Complexity Metrics

2.4.1 Performance Metrics

The performance of a hardware impairment mitigation method can be measured by normalized mean squared error (NMSE), error vector magnitude (EVM), and adjacent channel power ratio (ACPR). The NMSE is defined as [124]

$$\mathrm{NMSE} = 10\log_{10}\frac{\mathbb{E}_y\left\{|y_n-\hat{y}_n|^2\right\}}{\mathbb{E}_y\left\{|y_n|^2\right\}}, \tag{2.41}$$

where y_n and $\hat{y}_n$ are the original and measured signal at time sample n, respectively. $\mathbb{E}_y$ denotes the expectation over all a set of samples $\{y_n\}$. NMSE is usually used to measure the distortion level arising from hardware impairments such as PA nonlinearity, and IQ imbalance. The EVM is defined as [125, Eq. (2.4)]

$$\mathrm{EVM} = \sqrt{\frac{\mathbb{E}_y\left\{|s_n-\hat{s}_n|^2\right\}}{\mathbb{E}_y\{|s_n|^2\}}}, \tag{2.42}$$

where s_n and $\hat{s}_n$ are the n-th ideal and measured symbols. EVM results are commonly expressed as a percentage. EVM is commonly used to measure the distortion between the transmitted and received symbols due to many hardware impairments such as the PN, IQ imbalance, PA nonlinearity, and antenna crosstalk. Table 2.1 shows the transmitter EVM requirements for different QAM in 5G new radio (NR) from 3GPP standardization [126]. The ACPR is defined as [124]

$$\mathrm{ACPR} = 10\log_{10}\frac{\int_{\text{adjacent}}|Y(f)|^2\mathrm{d}f}{\int_{\text{main.}}|Y(f)|^2\mathrm{d}f}, \tag{2.43}$$

where $Y(f)$ denotes the Fourier transform of the signal $y(t)$. ACPR measures the ratio of power in the adjacent band to the power in the main band. It is therefore used to measure the OOB emission levels due to nonlinear hardware impairments, such as PA nonlinearity, nonlinear IQ imbalance, and nonlinear antenna crosstalk. In (2.43), the integration in the numerator is performed over the adjacent channel (the one with the larger power between the lower and upper adjacent channels), while the denominator integrates over the main channel. Table 2.2 outlines the ACPR requirements for different frequency bands in 5G NR, as specified by 3GPP standardization [126].

Besides these metrics, other commonly used metrics include SER, which measures the fraction of symbols that are received incorrectly. SNR quantifies how much stronger the signal is compared to the noise, and signal-to-interference-and-noise ratio (SINR) extends this by also considering interference from sources such as interference between UEs and hardware impairments. Both SNR and SINR can be used to

Table 2.1: Transmitted EVM requirements for different QAM in 5G NR from 3GPP standardization [127].

	QPSK	16-QAM	64-QAM	256 QAM
EVM (%)	17.5	12.5	8	3.5

Table 2.2: ACPR requirements for different frequency bands in 5G NR from 3GPP standardization [127].

	sub-6GHz	28GHz	39GHz
ACPR (dBc)	-45	-28	-26

determine the channel capacity, where SINR is more commonly used in scenarios with interference, such as MU MIMO systems.

2.4.2 Complexity Metrics

When comparing algorithm complexity, common methods include big-O notation, parameter count, and floating point operationss (FLOPs). Each offers insights from different perspectives, with varying suitability depending on the algorithm and application.

Measure by Bachmann-Landau measure

Bachmann-Landau measure, commonly known as big-O notation, $\mathcal{O}(\cdot)$, is often used to describe the asymptotic upper bound of an algorithm's growth with respect to its input size [128]. It is commonly used to measure algorithm complexity in communication systems [129]–[131]. While it provides valuable insight into how an algorithm scales in terms of runtime or space requirements, it focuses mainly on theoretical growth. It does not offer precise information about the actual runtime or computational cost in practical applications. Moreover, big-O notation does not consider important factors such as memory usage or specific hardware architectures, making it less suitable for evaluating power consumption or performance in real-world scenarios, such as hardware mitigation algorithms in communication systems.

Measure by Number of Parameters

Another common method for assessing algorithm complexity is by analyzing the number of parameters, which is widely used in both NN-based and conventional model-based algorithms, including hardware impairment mitigation studies [132]–[135]. This measure is particularly relevant when implementing algorithms on field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), or Graphics processing units (GPUs), as it impacts memory usage and indirectly

relates to computational complexity and power consumption. A higher number of parameters requires more memory and logic resources, which can affect performance and power efficiency. When deploying such as FPGAs and GPUs, limited on-chip memory and logic resources may be strained by large parameter sets, while on ASICs, this could result in increased circuit area and higher power consumptions.

While the number of parameters can be useful for comparing algorithms implemented on the same platform, it is still an approximation of computational complexity. It does not fully account for the exact number or types of operations involved, such as additions or multiplications, which influence the true computational load. Additionally, a model with many parameters could still be efficient if operations are highly parallelized or the parameters are sparsely utilized. Conversely, a model with fewer parameters might be computationally intensive due to complex, non-linear operations. Therefore, the number of parameters is more suitable for estimating memory requirements than for comparing the computational complexity or power consumption of different algorithms, particularly in tasks involving complex operations, such as hardware impairment mitigation techniques.

Measure by Number of FLOPs

The number of FLOPs is a more precise metric for evaluating computational complexity compared to methods such as big-O notation or counting the number of parameters. Unlike these measures, which either provide a theoretical upper bound or focus on memory usage, FLOPs directly count fundamental operations such as addition, subtraction, and multiplication, offering a more detailed view of actual computational demands. Different variants of this metric are widely applied to measure algorithm complexity in communication systems such as uplink precoding [136], channel estimation [100], [137], downlink combining [138], and PA behavior modeling, such as Volterra series-based models [135], [139], [140], and deep learning-based methods [141]–[148]. It can also capture the operations needed during both training and inference for deep learning-based methods [141]. Aside from *training complexity*, where deep learning-based methods are resource-intensive due to large datasets and iterative updates, FLOPs serve as a useful metric for comparing *inference complexity* with conventional methods. This is particularly relevant for applications such as DPD, where real-time processing contributes significantly to total power consumption [135]. The strength of the FLOPs metric also lies in its hardware relevance such as GPUs and FPGAs. It is highly related to processing time and power consumption on different hardware platforms. However, the metric is less useful for algorithms involving significant non-arithmetic operations, such as memory access or control logic, which are not accounted for in FLOPs. Despite these limitations, FLOPs remain a widely accepted metric for assessing the computational complexity of signal processing algorithms and deep learning models.

CHAPTER 3

Deep Learning for Communication Systems

This chapter starts by introducing the fundamentals of deep learning, covering three types of deep learning techniques. It then explains different NN architectures and provides details on how to effectively train them. Finally, the chapter explores the application of deep learning in the physical layer of communication systems, summarizing and discussing three key motivations for using deep learning in this field, along with the corresponding studies driven by these motivations.

3.1 Deep Learning Introduction

Deep learning is a subset of machine learning focused on multi-layered NNs. The term "deep" refers to the numerous hidden layers (depth of an NN), which enable the network to learn complex patterns and intricate relationships within the data [149]. Although the history of deep learning is extensive, its impact has significantly increased in recent years due to the vast amounts of data generated by Internet-scale applications and advances in computing infrastructure, particularly through GPUs. These improvements have allowed deep learning models to grow in size and complexity, enabling them to tackle increasingly challenging applications with greater accuracy over time [149], [150]. Deep learning has become highly effective in solving various applications, including natural language processing [18], object detection [19], autonomous driving [20], and image semantic segmentation [151]. In the following sections, the basics of deep learning is introduced, covering learning methods, NN architectures, implementation, and training.

3.1.1 Types of Deep Learning Techniques

Training of deep learning models can be broadly categorized as supervised, unsupervised, and reinforcement learning algorithms [149].

Supervised Learning

Supervised learning involves training a model on a dataset that contains pairs of inputs and their corresponding correct outputs, or labels. The term "supervised" reflects the fact that the designer, as the "supervisor", provide the labeled data during the training phase [150]. During training, the model learns the mapping between inputs and outputs by adjusting its internal parameters based on the error between its predictions and the true labels, typically using an optimization algorithm such as gradient descent. The goal is for the model to generalize this learned mapping to make accurate predictions on new, unseen inputs. Figure 3.1 illustrates this process. Supervised learning is widely applied to both classification (e.g., identifying handwritten digits) and regression tasks (e.g., predicting housing prices based on features such as size and location). In classification, the model assigns inputs to one of several predefined categories. Regression, on the other hand, predicts a continuous value.

Specifically, consider a supervised learning regression task with input $\boldsymbol{x}$, labeled output $\boldsymbol{y}$, and model $f_{\boldsymbol{\alpha}}$ with parameters $\boldsymbol{\alpha}$. The prediction step in supervised learning is,

$$\hat{\boldsymbol{y}} = f_{\boldsymbol{\alpha}}(\boldsymbol{x}), \tag{3.1}$$

and the training step involves minimizing the error between the model output $\hat{\boldsymbol{y}}$ and labeled output $\boldsymbol{y}$. Supervised learning comes with both benefits and drawbacks. On the positive side, when there is a sufficient amount of labeled data, it achieves high accuracy and can be applied to a wide range of tasks, from classification to regression. This versatility has made supervised learning the dominant approach in deep learning, particularly in areas with abundant labeled data. It is also effective

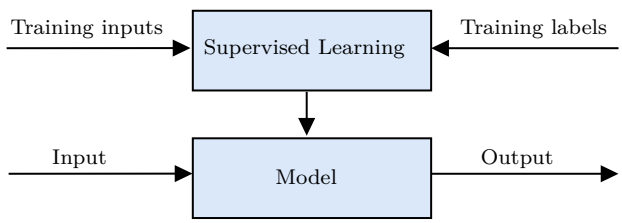


Figure 3.1: Supervised learning diagram. Training inputs and corresponding training labels are used to update a model's parameters, enabling it to learn the relationship between inputs and outputs. Once trained, the model can predict outputs for new, unseen data.

for fine-tuning advanced models for specific tasks, such as adapting a pre-trained Chatgenerative transformer (GPT) model for particular applications such as question answering or text classification [152]. However, the need for a large volume of labeled data can be a significant limitation, both in terms of cost and time. In summary, supervised learning is highly effective for tasks with large datasets and well-defined objectives, offering excellent accuracy when sufficient labeled data is available.

Unsupervised Learning

Unlike supervised learning, which relies on labeled input-output pairs provided by a "supervisor" to learn a mapping from inputs to outputs, unsupervised learning trains a model to uncover patterns or structures in data without labeled outputs. Common applications of unsupervised learning include clustering and dimensionality reduction [153]–[155]. Traditional clustering tasks, such as image compression, customer segmentation, and anomaly detection, are key examples of unsupervised learning applications and play a crucial role in preprocessing and understanding large datasets. One classic example of unsupervised learning is K-means clustering, where a dataset is divided into clusters such that data points within the same cluster are more similar to each other than to those in different clusters [156]. Another advanced application of unsupervised learning can be seen in the pre-training phase of large language models such as the ChatGPT [152], where the model processes vast amounts of unlabeled text data (e.g., from the internet, books, and other sources) to learn to predict the next word in a sequence, a task known as language modeling. This pre-training enables the model to build a foundational understanding of language structure before being fine-tuned with supervised learning. Overall, unsupervised learning is valuable for exploring and interpreting large unlabeled datasets, though it typically yields less precise predictions compared to supervised methods and its evaluation can be more challenging [155].

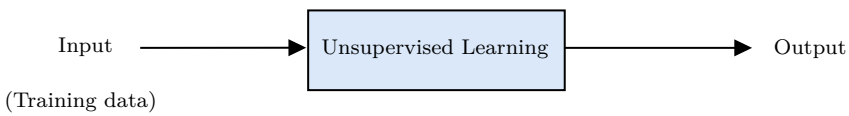


Figure 3.2: Unsupervised learning discovers hidden patterns or structures in data without relying on labeled outputs.

Reinforcement Learning

Reinforcement learning (RL) involves an autonomous *agent* learning to take *actions* by interacting with an *environment* through trial-and-error and feedback mechanisms [149], [157]. Unlike supervised learning, where models are trained on predefined labeled datasets, RL is based on learning from experiences and adapting through

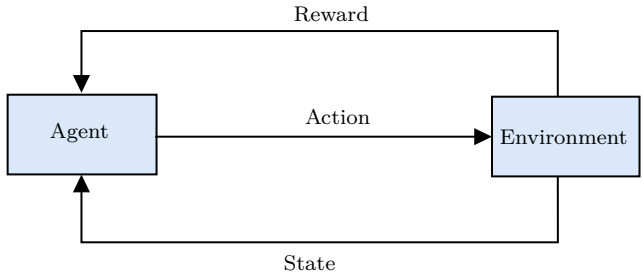


Figure 3.3: Reinforcement learning diagram: The agent takes actions based on the current state following a certain policy, interacts with the environment, receives rewards, and observes new states. Over time, the agent adjusts its policy based on the accumulated experiences to learn optimal behavior.

feedback in the form of *rewards*. At each step, the agent observes the current *state* of the environment, selects an action and receives a *reward* and a new *state* from the environment. The reward, which reflects how beneficial the chosen action was, can either be naturally derived from the environment or be custom-designed by the system's developer to encourage specific behaviors. The ultimate goal of the agent is to learn an optimal strategy, or *policy*, that maximizes the cumulative reward over time by mapping states to actions effectively.

A key benefit of RL is its ability to enable an agent to autonomously make decisions in complex and dynamic environments where it is impractical to define explicit instructions for every scenario. By learning from experience, RL excels in tasks that require a sequence of actions to achieve long-term objectives. Its versatility has led to widespread applications across various fields [158]–[161]. For example, in game playing, RL has achieved superhuman performance in games like Chess, Go, and video games, as demonstrated by AlphaGo [158]. In robotic control, RL is used to train robots for tasks such as manipulation, locomotion, and navigation, allowing them to develop advanced motor skills [159]. Additionally, in autonomous vehicles, RL plays a key role in learning driving policies through interaction with simulated environments, advancing self-driving technology [160].

A *policy* in RL is a strategy or rule that defines how an agent selects actions. It effectively maps states, or observations of states, to actions, acting as the agent's "brain". The policy can be either deterministic, where the agent always takes the same action for a given state, or stochastic, where the agent selects actions according to a probability distribution over possible actions for each state. Some examples of RL policy techniques are Q-learning [162], which focuses on learning a value function to implicitly derive a policy, policy gradient methods [163], which directly optimizes the policy, and deep RL [164], which leverages deep NNs to handle complex environments. In continuous action spaces, policies are often represented as Gaussian distributions to model stochastic action selection. In this thesis, a Gaussian policy

is introduced [165], [166], which can be applied in communication systems. In this context, consider a memoryless RL problem, meaning there is no action-dependent transition to the future states, which simplifies the model. In Gaussian policy, the policy maps from state observations $\boldsymbol{s} \in \mathbb{R}^N$ to mean actions

$$\boldsymbol{\mu} = \pi_{\boldsymbol{\alpha}}(\boldsymbol{s}), \tag{3.2}$$

where $\pi_{\boldsymbol{\alpha}}(\boldsymbol{s})$ denote the policy parameterized by parameters $\boldsymbol{\alpha}$, which could be realized by a NN. Given the mean action $\boldsymbol{\mu}$ and a vector of perturbations from a Gaussian distribution $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}_{N\times 1}, \sigma_\pi^2 \boldsymbol{I}_N)$ with diagonal variance σ_π^2, action samples $\boldsymbol{a}$ can be computed with[1]

$$\boldsymbol{a} = \sqrt{1-\sigma_\pi^2}\boldsymbol{\mu} + \boldsymbol{w}. \tag{3.3}$$

The log-likelihood of an arbitrary action sample a_n, using policy $\pi_{\boldsymbol{\alpha}}(a_n|s_n)$, is given by

$$\log \pi(a_n|s_n) \propto -\frac{\left|a_n - \sqrt{1-\sigma_\pi^2}[\pi_{\boldsymbol{\alpha}}(\boldsymbol{s})]_n\right|^2}{\sigma_\pi^2}. \tag{3.4}$$

This Gaussian policy introduces variations to the agent's actions, enabling the agent to explore the environment in a manner consistent with Gaussian statistics.

After taking an action $\boldsymbol{a}$, the environment transitions to a new state. A reward is then computed by a reward function R, which provides feedback on how good or bad the action was. Often, R is simplified to depend only on the current state and is usually custom-designed to encourage specific behaviors $R(\boldsymbol{s})$. The rewards are used to compute the return r, which is commonly the cumulative sum of rewards over time $r = \sum_t R(\boldsymbol{s}_t)$, where $\boldsymbol{s}_t$ represents the state at time t. The RL goal is to maximize the expected return $J(\pi_{\boldsymbol{\alpha}})$,

$$J(\pi_{\boldsymbol{\alpha}}) = \mathbb{E}_{\pi_{\boldsymbol{\alpha}}}\{r\}. \tag{3.5}$$

This RL optimization problem can be formulated as

$$\hat{\pi}_{\boldsymbol{\alpha}} = \arg\max_{\pi_{\boldsymbol{\alpha}}} J(\pi_{\boldsymbol{\alpha}}). \tag{3.6}$$

The optimization in (3.6) can be achieved by calculating the gradients of the expected return with respect to the parameters and updating the parameters using the gradient descent method, as explained in Chapter 3.1.2.

RL has several benefits compared to other learning techniques. It enables agents to learn and adapt autonomously without needing explicit programming for every situ-

[1] A simplified non-correlated Gaussian policy is considered.

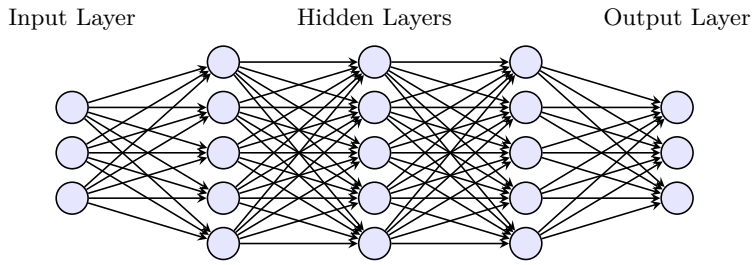


Figure 3.4: An example of a fully-connected NN with 3 hidden layers.

ation, and it excels in optimizing long-term goals by focusing on cumulative rewards. However, RL has its limitations. It typically requires a large number of interactions with the environment, which can be computationally expensive and time-consuming. Additionally, designing effective RL algorithms involves careful tuning of reward functions, exploration strategies, and hyperparameters, which can be challenging and complex [167].

3.1.2 Implementation of Deep Learning

In this section, the implementation of deep learning is presented, including NN architectures, and key components such as loss functions, activation functions, residual connections, and gradient updates.

Fully-connected Neural Networks

At the core of deep learning is the use of NNs with multiple layers to learn data representations. One of the most fundamental types of neural networks is the multilayer perceptron (MLP), a specific instance of a fully connected NN. Figure 3.4 shows an example structure of an MLP, where multiple layers of neurons are arranged in a feedforward architecture including input and output layers, and three hidden layers. These layers allow the network to capture intricate patterns in data by progressively learning more abstract features. This process can be mathematically represented as,

$$\boldsymbol{s}_l = f_\sigma(\boldsymbol{W}_l \boldsymbol{s}_{l-1} + \boldsymbol{e}_l), \tag{3.7}$$

where $\boldsymbol{s}_{l-1}$ and $\boldsymbol{s}_l$ are the input and output of layer l. Here, $\boldsymbol{W}_l$ is the weight matrix of layer l, and $\boldsymbol{e}_l$ is the bias vector of layer l, and $f_\sigma(\cdot)$ is the activation function, which is typically element-wise function on arbitrary input x.

Convolutional Neural Networks

Fully-connected layers are not efficient for processing input data with multiple dimensions, such as images, because fully connecting each input leads to an exponential

increase in the number of parameters as the input size grows. This causes significant memory and computational costs. Convolutional neural network (CNN), first proposed by LeCun et al. in 1998, is designed to address this inefficiency by using local receptive fields, shared weights, and pooling layers to process high-dimensional data such as images more effectively [168]. CNNs have since been widely adopted for tasks such as image classification, as demonstrated in AlexNet's success in the ImageNet competition in 2012, object detection, and segmentation tasks [151], [169]. These applications leverage CNN's ability to automatically capture spatial hierarchies of features, significantly reducing computational complexity while maintaining performance in high-dimensional tasks. Figure. 3.5 shows an example of the convolutional operation in CNN. The convolutional process can be written as

$$\tilde{\boldsymbol{S}}_l^k = \sum_{c=1}^{C} \boldsymbol{W}_l^k \circledast \boldsymbol{S}_{l-1}^c, \tag{3.8}$$

where $\tilde{\boldsymbol{S}}_l^k$ is the output at the k-th feature map in the l-th layer, and $\circledast$ denotes the *cross-correlation* operation [149, Eq. 9.6],

$$\tilde{S}_l^k[i,j] = \sum_m \sum_n S_{l-1}^c[i+m, j+n] W_l^k[m,n], \tag{3.9}$$

After this, the output $\tilde{\boldsymbol{S}}_l^k$ at the k-th feature map, and also the input to the next layer, is obtained by applying bias $\boldsymbol{E}_l^k$ and activation function as

$$\boldsymbol{S}_l^k = f_\sigma\left(\tilde{\boldsymbol{S}}_l^k + \boldsymbol{E}_l^k\right). \tag{3.10}$$

CNN-based models typically consist of both convolutional and fully-connected layers. To connect these layers, a flattened layer is commonly used, which converts the multi-dimensional output of the convolutional layers into a one-dimensional vector.

Activation Functions

Activation function $f_\sigma(\cdot)$ introduces non-linearity into NNs, and its widespread use allows NNs to model complex data patterns [149]. Without activation functions, the NN would behave like a linear regression model, regardless of the number of layers. There are various types of activation functions used in deep learning. Three common activation functions are introduced. The rectified linear unit (ReLU) activation function is given by

$$f_\sigma^{\mathrm{ReLU}}(x) = \max(0, x). \tag{3.11}$$

ReLU and its variants (like Leaky ReLU) are widely used for hidden layers due to their simplicity and effectiveness in mitigating the vanishing gradient problem [170].

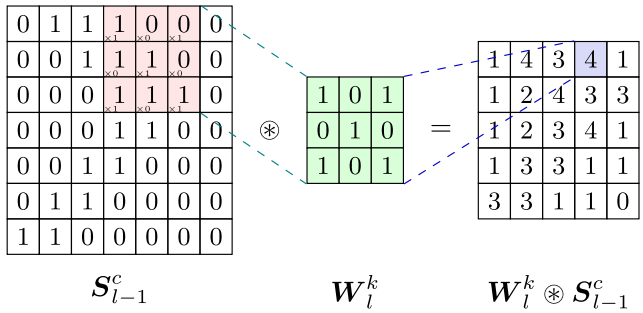


Figure 3.5: An example of the convolutional operation in CNN. A convolutional filter $\boldsymbol{W}_l^k$ cross-correlates the input $\boldsymbol{S}_{l-1}^c$ at layer $l-1$ and channel c to obtain one of the output at layer l and channel k.

Softmax activation is given by

$$f_\sigma^{\text{Softmax}}(x) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}, \tag{3.12}$$

which outputs a probability distribution over multiple classes. It is commonly used in the output layer for multi-class classification problems such as in and constellation de-mapper. Sigmoid activation is given by

$$f_\sigma^{\text{Sigmoid}}(x) = \frac{1}{1+e^{-x}}, \tag{3.13}$$

which outputs a value between -1 and 1, which is commonly used in the output layer for binary classification problems.

Residual Connections

Residual connections, also known as skip connections, are a structural feature in NNs designed to address vanishing and exploding gradient issues. By making it easier for the network to learn identity mappings, these connections facilitate the training of very deep networks and improve convergence speed by stabilizing gradient flow. ResNet [171] played a key role in popularizing residual connections, which are now widely used in many applications, including GPT [152]. These connections create shortcuts by directly passing the input of one or more layers to their output, as illustrated in Figure 3.6. A modified version of the residual connections, described in [171] can be written as

$$\boldsymbol{s}_{l+l_{\text{R}}} = f_{\text{R}}\left(\{\boldsymbol{s}_{l'}\}_{l'=l}^{l+l_{\text{R}}}\right) + \boldsymbol{W}_{\text{proj}}\boldsymbol{s}_l, \tag{3.14}$$

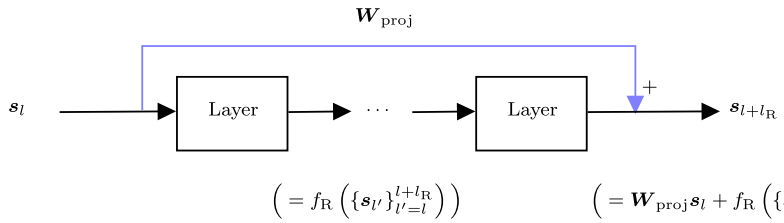


Figure 3.6: An example of a residual connection that bypasses several layers (l to $l + l_{\text{R}}$) by a residual connection layer.

where l_{R} denotes the number of bypassed layers, $f_{\text{R}}(\{\boldsymbol{s}_{l'}\}_{l'=l}^{l+l_{\text{R}}})$ denotes a function representing the layers from l to $l + l_{\text{R}}$, and $\boldsymbol{W}_{\text{proj}}$ is the projection matrix. The projection matrix can also address the dimensional mismatch between the input and the output. For identity residual connections, $\boldsymbol{W}_{\text{proj}}$ simplifies to 1, resulting in the input being directly added to the output.

Loss Functions

A loss function, also known as a cost function, measures how closely a machine learning model's predictions align with the expected outcomes. It evaluates the model's performance by calculating the difference between the predicted values and the actual target values. During training, the model iteratively updates its parameters to minimize this loss, thereby improving its accuracy. The loss functions used in this thesis are introduced.

A general *loss function* quantifies the discrepancy between the prediction $\hat{\boldsymbol{y}}$ and the actual labels $\boldsymbol{y}$ as $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$. Common loss functions include MSE and cross-entropy, with the MSE loss function defined as

$$\mathcal{L}^{\text{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{n=1}^{N} \|y_n - \hat{y}_n\|^2. \tag{3.15}$$

The MSE loss function is commonly used in regression tasks [149]. Another widely used loss function is the cross-entropy loss, given by

$$\mathcal{L}^{\text{CE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{N}\sum_{n=1}^{N} y_n \log(\hat{y}_n), \tag{3.16}$$

where $\{y_n\}_{n=1}^{N}$ and $\{\hat{y}_n\}_{n=1}^{N}$ are the true and predicted probability distributions, respectively. It is widely used in binary and multi-class classification tasks [172].

Stochastic Gradient Descent

The parameters $\boldsymbol{\alpha}$ in deep learning models, such as NNs, can be updated through back-propagation via a stochastic gradient descent (SGD) algorithm as

$$\boldsymbol{\alpha}_{j+1} = \boldsymbol{\alpha}_j - \eta \nabla_{\boldsymbol{\alpha}} \mathcal{L}_{\boldsymbol{\alpha}_j}, \tag{3.17}$$

where $\eta > 0$ denotes the learning rate and $\nabla_{\boldsymbol{\alpha}} \mathcal{L}_{\boldsymbol{\alpha}_j}$ is the derivatives of the loss function $\mathcal{L}_{\boldsymbol{\alpha}_j}$ with respect to $\boldsymbol{\alpha}_j$ at step j [173].

3.2 Deep Learning for the Physical Layer in Communication Systems

Deep learning has demonstrated promising potential in communication systems, particularly in the physical layer, with various studies exploring its applications across different scenarios. These works are primarily motivated by three key factors: enabling joint optimization of multiple communication blocks, balancing complexity with performance, and addressing model mismatches. Each of these motivations is discussed in the following sections.

Optimizing Multiple Blocks Jointly with Deep Learning

One motivation is that deep learning models can be designed to optimize multiple conventional methods in the physical layer simultaneously using one model. Conventional systems are designed using cascaded blocks—each responsible for tasks such as modulation, channel estimation, and hardware impairment compensation. These blocks are optimized separately, which can result in suboptimal system-wide performance. Deep learning methods can use one single model, e.g., an NN, to optimize multiple blocks rather than optimizing individual blocks such as modulation, coding, or channel estimation. Such examples are joint MIMO detection and channel decoding [174], joint beamforming and channel prediction [175], joint receiver design [176], [177], joint pilot design and channel estimation [178], [179], joint channel estimation and feedback [180], [181], joint scheduling and beamforming [182], joint power control and pilot assignment [183], power control and channel allocation [26], joint channel estimation and symbol detection [184], joint channel state information (CSI) feedback and precoding [185], end-to-end communication antoencoder [27], [186]–[190], joint channel feedback and estimation [191], joint constellation and pulse shaping [192], [193], and joint hardware impairments mitigation, including IQ imbalance, PA nonlinearities, and PN [15], [17], [31], [194]. These studies demonstrate the potential of deep learning to simultaneously optimize multiple conventional methods in the physical layer using a single model, rather than the traditional approach of optimizing individual blocks separately, leading to more efficient and cohesive system-wide performance.

Balancing Performance and Complexity with Deep Learning

Another motivation for using deep learning in communication systems is the need to balance performance and computational complexity. Unlike deep learning tasks that use large, deep NNs, physical layer applications typically require less layers. This is because communication systems, especially in massive MIMO setups, are real-time and resource-constrained. In these systems, the power consumed by the RF chain is high, leaving less power for the digital processing of high-bandwidth signals, making computational efficiency critical. As a result, low-complexity considerations are essential when designing deep learning models for the physical layer. The complexity aspect of using deep learning has been explored in various areas, including MIMO detection [21], [174], [195]–[198], channel estimation [22], [178], [199], [200], CSI feedback [23], [201], [202], MIMO precoding [203]–[207], communication autoencoders [24], [186], [193], [208], [209], OFDM receiver [210], direction-of-arrival (DOA) estimation [211], and power allocation [25], [212]–[214], which not only design low-complexity deep learning models but also use techniques to reduce computational complexity such as NN pruning techniques [141]–[143]. These studies demonstrate the potential of deep learning to achieve a better balance between computational complexity and performance trade-offs in the physical layer.

Addressing Model and Method Deficits with Deep Learning

Deep learning methods can address both the model and method deficits in the physical layer. The mismatches arise from the limitations of conventional, theoretically derived models or methods, which often rely on simplified or ideal assumptions, such as system linearity, that do not fully reflect real-world non-ideal conditions. Real-world scenarios include complexities such as non-linearities, noise, interference, and hardware impairments that are difficult to model accurately. In contrast, deep learning models, especially NNs, do not require such explicit assumptions. They learn directly from large datasets, capturing the complexities of the physical layer and reducing mismatches found in traditional models. For instance, in tasks such as channel estimation or MIMO detection, deep learning-based models have demonstrated improvements over conventional methods, particularly in handling non-ideal hardware and channel conditions [30], [215]–[218]. In full-duplex systems, RL can solve the ICI problem [193]. Similarly, in MIMO power allocation, deep learning methods have been shown to improve system performance while reducing computational complexity [25], [212]. These studies highlight the potential of deep learning to more effectively handle both model and method mismatches in the physical layer, particularly in addressing the complexities of real-world, non-ideal conditions that are difficult for conventional methods to model.

In the next Chapter, the application of deep learning for hardware impairment mitigation is introduced.

CHAPTER 4

Hardware Impairments Mitigation Using Deep Learning

Mitigating multiple hardware impairments in communication systems involves balancing performance with complexity, a challenge where traditional methods may face limitations. Deep learning shows the potential to offer more adaptable and efficient solutions for addressing a wider range of impairments. This chapter explores the application of deep learning in mitigating multiple hardware impairments, beginning with the motivations, challenges, and highlights in the literature. It then uses DPD as a case study to demonstrate how deep learning, through both supervised and reinforcement learning, can pre-compensate for multiple impairments, showcasing examples of joint mitigation.

4.1 Motivations, Challenges, and Applications

Deep learning offers promising solutions for addressing hardware impairments in communication systems. As discussed in Chapter 3.2, the key motivations and challenges for applying deep learning in the physical layer include addressing model and method mismatches, balancing performance and complexity, and optimizing multiple communication tasks jointly. This section explores how these motivations drive advancements in hardware impairment mitigation and the challenges involved in fully realizing the benefits of deep learning.

Joint Mitigation of Multiple Hardware Impairments

One of the strengths of deep learning is its ability to jointly address multiple hardware impairments, which is often difficult for traditional methods. In practical communication systems, impairments like IQ imbalance, PA nonlinearity, mutual coupling, and PN frequently occur together and interact with each other. This interaction complicates the mitigation process, as focusing on one impairment may cause suboptimal performance for others. For instance, PN from the LO can interfere with IQ imbalance compensation at the receiver, leading to degraded performance for both impairments. Additionally, using separate mitigation techniques for each impairment increases computational load and reduces power efficiency, especially in large-scale systems like massive MIMO, where impairments scale with the number of antennas.

While model-based mitigation methods can address multiple hardware impairments simultaneously [68], [219]–[222], they often rely on simplified assumptions that may not fully capture the complexities of real-world scenarios, leading to suboptimal performance. For instance, the parallel Hammerstein (PH) model [68] accounts for frequency-dependent IQ imbalance using a Hammerstein model, but its performance is limited when applied to systems with both frequency-dependent IQ imbalance and real PAs [15]. Similarly, [220] assumes static IQ imbalance, while [221] is based on a MP model, which may not provide enough linearization performance. Another example is in [91] which assumes linear crosstalk or specific PA behavior, which limits the general applicability of the used Volterra-based model. Moreover, linear estimation methods like least squares and linear minimum mean square error (LMMSE) are challenged by the interaction between the PN and nonlinearities [30], [74].

Deep learning-based approaches, particularly those using NNs, offer a more flexible solution. These models can learn directly from data, capturing the complex dependencies between impairments and providing more accurate compensation. This ability to jointly mitigate multiple impairments can simplify system design and reduce overall complexity. For instance, deep learning has been successfully applied to mitigate IQ imbalance, PA nonlinearity, PN, carrier frequency offset (CFO), and antenna crosstalk [15], [17], [31], [193], [223]–[226], and to mitigate PA nonlinearity jointly with constellation shaping and demodulation [193]. These aspects are also covered in Paper A, C, and D. Specifically, in Papers A and C, deep learning-based methods are shown to improve performance and reduce complexity when mitigating impairments like IQ imbalance, PA nonlinearity, and antenna crosstalk., and in Paper D, deep learning-based channel estimator is shown to improve channel and PN estimation performance.

Balance Complexity and Performance of Mitigation

While deep learning shows potential for mitigating hardware impairments, managing computational complexity is crucial for practical deployment. Massive MIMO

systems, for example, face strict computational and energy constraints, making it important to design low-complexity deep learning models. Selecting the right model architecture, such as choosing between fully connected layers and more efficient architectures like CNNs, is key to balancing complexity and performance. CNNs, for instance, handle large input dimensions more efficiently, as demonstrated in various DPD applications [17], [29], [227]. In addition, techniques like deep unfolding, which incorporate model-based knowledge into NNs, have shown improvements in both performance and complexity by turning iterative algorithms into NN layers [228]. This approach can be particularly effective in scenarios where traditional mathematical models are available but need further optimization.

When no clear mathematical model is available, deep learning offers alternatives. For example, traditional TD GMP-based models struggle to operate effectively in the frequency domain, limiting their ability to reduce power consumption. Deep learning-based DPD methods using NNs present a promising alternative, particularly in massive MIMO systems [16], as discussed in Paper C. Furthermore, deep learning offers improved performance-complexity trade-offs in tasks such as channel estimation under nonlinearities or PN, where traditional estimators like minimum mean square error (MMSE) incur significant computational costs [30].

In the next section, DPD is used as an example to show how deep learning is implemented to mitigate multiple hardware impairments.

4.2 Deep Learning for DPD

We use DPD as an example of using deep learning for hardware impairment mitigation because it is well-suited for optimization through supervised learning and can leverage deep learning's ability to compensate for multiple impairments using a single DPD block. As discussed in Chapter 4.1, supervised learning with labeled input-output data is highly effective for optimizing deep learning-based techniques. DPD is particularly easy to train, as it operates in the digital domain, where the undistorted input (baseband signal) is readily accessible and the distorted output can be obtained through feedback paths or over-the-air measurements. We first focus on how to obtain labeled data for supervised learning using the indirect learning architecture (ILA). When obtaining large amounts of labeled data is challenging, RL is employed to train deep learning-based DPD models. Additionally, low-complexity NN structures for DPD to achieve low complexity and power efficiency, covering both TD and FD DPD implementations designed to optimize performance while minimizing computational overhead.

4.2.1 Optimization deep learning-based DPD

In practice, obtaining labeled input-output data for DPD directly from real hardware is challenging, as the desired DPD output is often unknown. Additionally, separating distorted and undistorted signals is difficult when multiple impairments, such as PAs

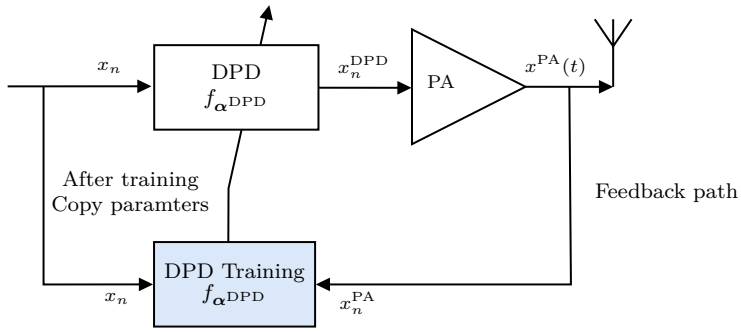


Figure 4.1: Block diagram of the ILA enabling supervised learning to train deep learning-based DPD, acquiring analog PA output $x^{\mathrm{PA}(t)}$ via feedback path to digital samples x_n^{PA}.

or IQ modulators, are involved. Furthermore, acquiring PA measurements typically requires a dedicated feedback path, which becomes costly for wideband signals. These challenges make it difficult to effectively apply supervised learning in such scenarios. In cases where labeled DPD output is not available, one approach is to use simulated data from model-based hardware models. However, this approach not only adds complexity but is also limited in performance, as simulations may not fully capture real hardware behavior, leading to potentially suboptimal results when deployed. To address this, RL offers a promising alternative, as it eliminates the need for labeled data and allows models to learn effectively through direct interaction with the environment.

DPD Learning by Supervised Learning

To enable supervised learning for DPD training, it is common to use the ILA to obtain the labeled input and output data, as shown in Figure 4.1. A deep learning-based DPD model can be trained by feeding the measured PA output signal, $\{x_n^{\mathrm{PA}}\}_{n=1}^N$, and the undistorted baseband signal $\{x_n\}_{n=1}^N$ as the labeled input and output data. In a such way, a post-inverse of the PA can be learned by the DPD. The ILA makes the training of DPD done by supervised learning. Specifically, consider a deep learning-based DPD, e.g., a NN, represented by $f_{\boldsymbol{\alpha}^{\mathrm{DPD}}}(\cdot)$ with DPD parameters $\boldsymbol{\alpha}^{\mathrm{DPD}}$, the parameters can be learned by minimizing the loss between the DPD output $x_n^{\mathrm{DPD}} = f_{\boldsymbol{\alpha}^{\mathrm{DPD}}}(\{x_n^{\mathrm{PA}}\})$ and the baseband signal x_n,

$$\hat{\boldsymbol{\alpha}}^{\mathrm{DPD}} = \arg\min_{\boldsymbol{\alpha}^{\mathrm{DPD}}} \mathbb{E}_x\big\{\mathcal{L}\left(f_{\boldsymbol{\alpha}^{\mathrm{DPD}}}(\{x_n^{\mathrm{PA}}\}), x_n\right)\big\}, \tag{4.1}$$

where the expectation is taken over the training dataset $\{x_n^{\mathrm{PA}}, x_n, \}$. In practice, the optimization of the DPD parameters (4.1) is typically implemented using the SGD

algorithm (3.17) with a batch-size N using the MSE loss function (3.15),

$$\mathcal{L}_{\text{MSE}}\left(x_n^{\text{DPD}}, x_n\right) = \frac{1}{N}\sum_{n=1}^{N}\left|x_n - x_n^{\text{DPD}}\right|^2. \tag{4.2}$$

In summary, the ILA provides a practical approach to address the challenge of obtaining an unknown or difficult-to-measure DPD output. It enables the use of straightforward supervised learning to train a deep learning-based post-distorter. However, it is important to note that the ultimate goal is to develop a pre-distorter rather than a post-distorter. Any noise or distortion in the PA output measurements can introduce bias, which may degrade DPD performance [229]. For instance, studies have shown that the ILA requires an SNR of at least 22 dB to improve linearization compared to the case without DPD [230]. Fortunately, in practice, since the ILA operates on the transmitter side, the measured data typically has a high enough SNR to minimize the impact of such bias [229].

DPD Learning by Reinforcement Learning

Learning DPD parameters through the ILA is a straightforward approach but requires a dedicated data acquisition path from the PA. This feedback path can be costly, especially for large bandwidth signals, as high sampling rates ADCs are needed, typically requiring a sampling rate exceeding five times the signal bandwidth to capture the full-band behavior of the PA. In systems with multiple RF chains, such as massive MIMO, the cost of these feedback paths can increase significantly, presenting a major challenge. To mitigate these costs, recent studies have explored the use of low sampling rate ADCs [231]–[234], which attempt to recover the full-rate PA output from under-sampled ADC data for DPD optimization. To further reduce costs, over-the-air (OTA) methods have emerged as an alternative [119], [235], [236]. These methods employ an observation receiver to capture the PA output signal over the wireless channel, enabling DPD optimization without the need for expensive high-rate feedback paths. However, these approaches are primarily effective in line-of-sight (LOS) scenarios.

The mentioned studies [119], [231]–[236] employ a sample-based approach to optimize DPD models by minimizing sampling errors between over-sampled signals. Implementing this approach is straightforward, as DPD typically operates at a similar oversampling rate as the PA to reduce in-band and OOB distortion from the PA. However, in a massive MIMO environment, the requirement for OOB linearization is significantly relaxed. This is because the in-band signal gains substantial power through downlink beamforming, while the OOB components tend to interfere destructively, resulting in lower power gain [17], [237]. The reduced need for OOB linearization encourages the use of low-complexity DPD and low-rate OTA DPD optimization methods. Paper B explores the latter approach using RL, where RL

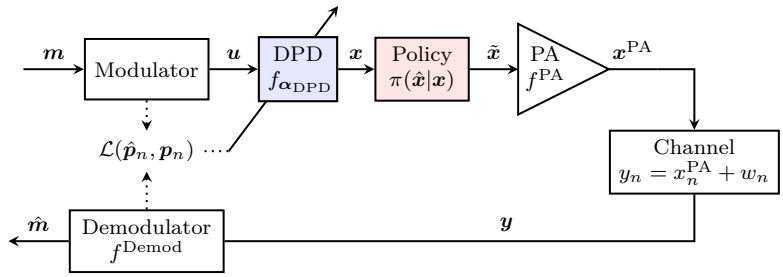


Figure 4.2: Learning DPD parameters by RL. The DPD parameters $\boldsymbol{\alpha}^{\text{DPD}}$ are optimized using OTA observations at the receiver side by minimizing a CE loss between the probabilities of the transmitted and received messages, $\{\boldsymbol{p}_n\}$ and $\{\hat{\boldsymbol{p}}_n\}$.

is employed to eliminate the need for a known channel model in conventional OTA learning algorithms. The block diagram of the RL-based DPD optimization is shown in Figure 4.2. This OTA DPD optimization algorithm is described as follows.

In the context of RL, the agent is now a deep learning-based DPD model $f_{\boldsymbol{\alpha}^{\text{DPD}}}$ with parameters $\boldsymbol{\alpha}^{\text{DPD}}$, such as a NN. The state and action are the baseband signal $\boldsymbol{u}$ and the DPD output $\boldsymbol{x}$, respectively, as defined by:

$$\boldsymbol{x} = f_{\boldsymbol{\alpha}^{\text{DPD}}}(\boldsymbol{u}). \tag{4.3}$$

The Gaussian policy $\pi(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ (3.3) is then applied to the DPD output, which adds perturbations to the DPD output to explore the environment, i.e., the nonlinear PA and channel.

The RL environment consists of the unknown nonlinear PA and channel, and other hardware components such as the ADC, DAC, mixer, oscillators, and low-noise PA at the receiver are assumed to be ideal, which can also be considered as non-ideal in specific scenarios. The function f^{Demod} represents the demodulation process that includes components such as down-conversion, matched filtering, and constellation de-mapping. For an arbitrary demodulated symbol $\hat{s}_n$ at time sample n, the de-mapper gives a prediction of the transmitted message represented by a probability vector $\hat{\boldsymbol{p}}_n \in \mathbb{R}_+^M$ across M possible messages,

$$\hat{\boldsymbol{p}}_n = f^{\text{Demod}}\left([f^{\text{PA}}(\tilde{\boldsymbol{x}})]_n + w_n\right), \tag{4.4}$$

where $w_n \in \mathcal{N}(0, \sigma^2)$ is the AWGN. This de-mapper can be realized by the maximum likelihood demapper or an NN-based de-mapper as in [27], [186], [238], which can be pre-trained to have similar decoding performance. The RL reward is calculated using the cross-entropy loss function (3.16) as

$$r = -\mathcal{L}_{\text{CE}}(\hat{\boldsymbol{p}}_n, \boldsymbol{p}_n), \tag{4.5}$$

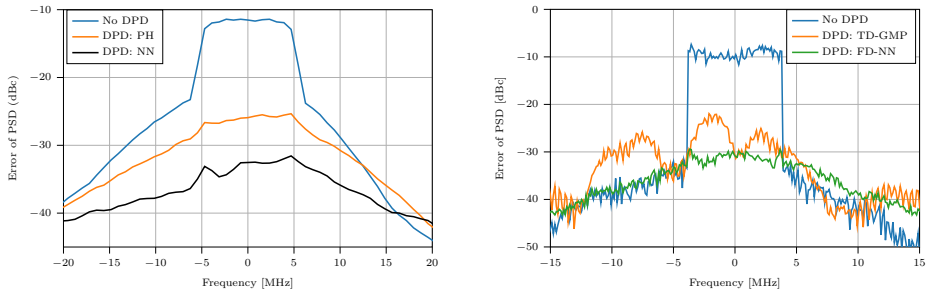
where $\boldsymbol{p}_n = [0, \cdots, 1, 0, \cdots, 0]^{\mathsf{T}} \in \mathbb{R}^M$ is the labeled message m_n in a one-hot vector form. The optimization goal is to maximize the expected return $J_{\boldsymbol{\alpha}^{\text{DPD}}} = \mathbb{E}_{\boldsymbol{m}}\{r\}$, which is realized using gradient descent methods (3.17) to update the DPD parameters $\boldsymbol{\alpha}^{\text{DPD}}$.

In cases where supervised learning is impractical due to the lack of labeled DPD input-output data, RL presents an alternative for optimizing OTA DPD parameters. This approach eliminates the need for complex feedback paths or precise models of non-ideal hardware and channels. Paper B explores the use of RL in single-carrier SISO systems addressing only PA nonlinearity, but the method can be extended to account for multiple hardware impairments. Additionally, the RL-based method has been validated by real-world measurements in a similar DPD optimization context for optical communication systems [239]. While promising, further research is needed to apply RL to multi-carrier and massive MIMO systems, where challenges like increased hardware impairments must be addressed. Nevertheless, the large antenna arrays in massive MIMO can help ease constraints such as OOB distortion, making RL an attractive solution for these scenarios. follow-up research will be crucial in fully leveraging its potential.

4.2.2 Neural Network-based DPD in the Time-Domain and Frequency-Domain

As discussed in Chapter 4.1, designing a low-complexity NN structure is one of the challenges to enabling the practical implementation of hardware impairment mitigation techniques. The structure design is substaintially exlpored for NNs-based DPD [15]–[17], [60], [61], [240]–[247], and can generally be categorized into two groups: NN-based TD-DPD [15], [60], [61], [241]–[246] and NN-based FD-DPD [16], [17], [247]. Both types operate in the baseband and digital domain but differ in where they are applied in the signal chain. The structure design for NN-based TD-DPD and FD-DPD follows different principles, but the overall goal remains the same: to minimize complexity while maximizing the linearization performance of DPD, including Paper A for designing a TD-DPD NN and Paper C for designing an FD-DPD NN.

The choice between TD-DPD and FD-DPD is far from obvious in massive MIMO systems, TD-DPD becomes computationally expensive as the number of PAs increases, with its complexity scaling directly with the number of PAs. FD-DPD, applied before the inverse discrete Fourier transform (IDFT) and digital precoding, offers a more efficient solution by scaling with the number of UEs rather than PAs, making it a more power-efficient choice for massive MIMO systems [16], [17]. FD-DPD also offers power savings since it can mitigate OOB distortion with lower oversampling rates, reducing power consumption for wideband signals in 5G and future networks. While shared DPD approaches attempt to reduce complexity by managing multiple PAs across arrays or subarrays [248]–[251], they are often con-



(a) IQ imbalance and nonlinear PA. Details of the scenario are in Paper A.

(b) Mutual coupling and nonlinear PA. Details of the scenario are in Paper C.

Figure 4.3: Comparison of deep learning-based DPD with traditional methods for hardware impairment mitigation. (a) In a SISO system, NN-based DPD effectively compensates for both PA nonlinearity and IQ imbalance with the same computational complexity as the model-based PH approach, while significantly reducing residual distortion. (b) In a MIMO system, FD NN-based DPD outperforms conventional TD GMP-based DPD by offering superior mitigation of PA nonlinearity and mutual coupling effects with significantly lower computational complexity.

strained by analog or hybrid architectures. In contrast, massive MIMO systems benefit from reduced OOB linearization requirements, where in-band signals are enhanced by beamforming and OOB components are suppressed through destructive interference [237], [252]. Despite the potential, NN-based FD-DPD is still a developing field that requires further validation [253]. Traditional DPD models like GMP, which are designed for TD mapping, are difficult to adapt to the frequency domain, as demonstrated in [16], where the solution still relies on IDFT and discrete Fourier transform (DFT) operations. Paper C further explores this problem, exploring how CNN can be used in FD-DPD to effectively handle large input sizes from OFDM subcarriers and UEs while managing complexity. This is a relatively new area, and further applications of deep learning are essential to fully explore its potential and address existing challenges.

4.2.3 Joint Mitigation of Hardware Impairments Using DPD

Numerous studies have investigated DPD for mitigating multiple hardware impairments, including both model-based [68], [219], [220] and NN-based approaches [15], [17], [222], [223], [244], [245], [254], including Paper A and Paper C. DPD is effective at mitigating multiple impairments, such as IQ imbalance, PA nonlinearity, and mutual coupling, for several reasons. First, as a pre-compensation technique, DPD compensates for these impairments directly at the transmitter. Second, NN-based DPD leverages the modeling capabilities of NNs, enabling it to outperform traditional models designed for either joint or separate compensation. Third, by using

the ILA approach, the combined effects of multiple impairments are captured at the PA output, allowing the effective training of DPD by supervised learning.

For example, Figure 4.3a shows how an NN-based DPD improves the mitigation of both IQ imbalance and PA nonlinearity with the same computational complexity as the model-based PH approach [68], unlike the NN-based method, the PH model leaves significant residual distortion, as discussed in Paper A. Similarly, Figure 4.3b shows how FD NN-based DPD can jointly pre-compensate for PA nonlinearity and mutual coupling in a massive MIMO system with significantly lower complexity. In contrast, the conventional TD-GMP-based DPD not only fails but worsens the OOB distortion, as explored in Paper C.

CHAPTER 5

Contributions and Future Work

This chapter summarizes the contributions of each included publication and suggests some potential directions for future research.

5.1 Paper A

Low Complexity Joint Impairment Mitigation of I/Q Modulator and PA Using Neural Networks

In paper A, we investigate the joint hardware impairments mitigation problem. Specifically, we address the joint mitigation of nonlinear frequency-dependent I/Q imbalance and PA in direct conversion transmitters in SISO systems. While DPD can jointly mitigate these impairments, model-based methods often underperform, and existing NN-based methods suffer from high computational complexity, making them impractical for real-world applications.

We propose a novel NN-based architecture, called the shortcut real-valued time-delay neural network (SVDEN), which incorporates shortcut connections to effectively mitigate both IQ imbalance and PA nonlinearity with lower complexity. We utilize a NN pruning algorithm to further reduce the computational complexity of the SVDEN by selectively removing connections with minimal impact on performance. Experimental results from a GaN PA demonstrate that the proposed SVDEN with pruning achieves superior linearization performance in both in-band and OOB compared to existing models, while also reducing computational complexity. The proposed SVDEN architecture, along with the pruning algorithm, offers a practical and

effective solution for joint hardware mitigation of the PA and IQ modulator in direct conversion transmitters, balancing both performance and complexity.

5.2 Paper B

Symbol-Based Over-the-Air Digital Predistortion Using Reinforcement Learning

In Paper B, we address the challenge of optimizing DPD for PA linearization in a single-carrier SISO system, focusing on low-sampling rate optimization. Traditional DPD optimization approaches that based on the ILA often require high sampling rate feedback paths, which are costly and complex for wideband signals and multi-antenna systems. Optimization using OTA observations at an observation receiver can alleviate the use of a high-sampling rate feedback path, but existing works still require over-sampling rate optimization, which is not optimal and not appropriate when the target is to minimize the SER.

By utilizing the Gaussian policy RL, we introduce a novel DPD optimization algorithm using OTA measurements. This algorithm minimizes the cross-entropy loss between the transmitted and received symbols, instead of the conventional sample-based approach between over-sampled signals. This RL-driven DPD optimization method operates without the need for hardware and channel models. Results on a GaN PA show that the RL-driven DPD optimization algorithm results in SER improvements while preserving acceptable OOB distortion levels, outperforming conventional ILA, even when using full-rate ADCs in the feedback path. This RL-driven DPD optimization algorithm offers a practical and low-complexity solution for PA linearization, reducing costs in wideband systems and potentially helpful in multi-antenna systems and other hardware impairments mitigation.

5.3 Paper C

Time vs. Frequency Domain DPD for Massive MIMO: Methods and Performance Analysis

In Paper C, we extend the hardware impairments mitigation problem from SISO systems in Paper A and Paper B to massive MU-MIMO systems. Specifically, we address the complexity challenge of PA linearization in massive MU-MIMO-OFDM systems. The deployment of numerous antennas and their corresponding PAs creates a complexity issue for DPD to achieve the required PA linearization performance. While the complexity of conventional TD DPD, whose dimension is related to the number of PAs, scales with the number of PAs, the complexity of FD DPD, whose dimension is related to the number of UEs, scales with the number of UEs.

The main motivation is that current FD-DPD models fail to fully address the DPD complexity issue with antennas because of their reliance on IDFTs and DFTs. We propose a low-complexity FD CNN DPD which reduces complexity by replacing IDFTs and DFTs with convolutional layers. We develop a new learning algorithm for FD-DPDs with differentiable structures, enabling simple supervised learning and resulting in better linearization performance of NN-based DPDs in scenarios with antenna crosstalk. We provide a comprehensive analysis of different state-of-the-art TD and FD-DPD schemes in terms of complexity and linearization performance in rich scattering and LOS channels, and also scenarios with antenna crosstalk. The analysis shows that FD-DPDs, particularly the proposed FD-CNN DPD, are preferable in LOS scenarios with few UEs. On the other hand, in scenarios with more UEs or isotropic scattering channels, significant intermodulation distortions among UEs degrade FD-DPD performance, making TD-DPD more suitable.

5.4 Paper D

Uplink Cell-Free massive MIMO OFDM with Phase Noise-Aware Channel Estimation: Separate and Shared LOs

In Paper D, we study the hardware impairments mitigation problem in cell-free massive MIMO. Specifically, we investigate the impact of PN on the uplink performance of cell-free massive MIMO OFDM systems. Hardware impairments originating from LOs in both APs and UEs pose considerable challenges in achieving phase-coherent transmission. Traditional single-carrier PN models underestimate the PN impact in OFDM systems, resulting in inaccurate channel estimation, uplink combining, and downlink precoding, and also give overly optimistic SE predictions due to underestimating the impact of PN. Moreover, using a shared common LO between APs can save costs but introduce correlated PN interference, impacting system performance without leveraging PN correlation, while a solution to address this in cell-free massive MIMO is missing in the literature. In addition to deriving novel channel estimators under the OFDM PN model, this paper also addresses two research questions. To what extent does applying the single-carrier PN model degrade the system performance in OFDM systems? What are the performance gains of using accurate channel estimation based on the OFDM PN model in OFDM systems? We consider two scenarios: uncorrelated PN with separate LOs for each AP, and correlated PN with a shared common LO for each AP, covering both conventional cell-free and radio stripes scenarios.

Specifically, we develop a novel uplink OFDM signal model for cell-free massive MIMO networks, including for both correlated and uncorrelated PN scenarios. For scenarios with separate LOs, we propose a distributed PN-aware joint LMMSE channel and CPE estimator, showing improved SE over single-carrier estimators. For

shared LO scenarios, we propose a centralized joint channel and CPE estimator that alternates between distributed channel estimation and centralized CPE estimation, leveraging PN correlation to enhance performance. Furthermore, we propose a deep learning-based distributed channel estimator to initialize the centralized estimator, improving accuracy and reducing iteration times.

5.5 Future work

In this thesis, various hardware impairment mitigation scenarios using deep learning techniques have been explored, focusing on both performance and complexity aspects, which are crucial for practical systems.

Here are some potential directions for future work:

- In Paper A, the joint mitigation of imperfect IQ modulators and PA is only explored for SISO systems. A potential direction for future work is to extend this approach to massive MIMO systems. While similar hardware impairment models from Paper C can be applied, the main challenge remains in finding a low-complexity mitigation solution for these hardware impairments in massive MIMO scenarios.

- In Paper B, RL-based DPD optimization is studied only in single-carrier SISO systems. Future work could explore its application in OFDM and MIMO systems. For OFDM systems, incorporating components like IDFT, DFT, and cyclic prefix should not directly affect the Gaussian policy RL used for DPD optimization, though the performance transition from single-carrier to OFDM systems still needs to be evaluated. Extending this method to MIMO systems presents a challenge due to the differing dimensions between the reward (related to the number of UEs) and DPD (related to the number of antennas), requiring significant modifications to the RL algorithm.

- In Paper C, although the CNN-based FD-DPD achieves a better complexity-performance trade-off, its size and complexity increase linearly with the number of data subcarriers, making it unsuitable for large bandwidth applications and increasing training costs. Additionally, it requires re-training when the bandwidth changes. Future research can focus on improving deep learning-based FD-DPD designs to address these limitations.

- In Paper D, the PN impact and PN-aware channel and PN estimators are only investigated for the uplink scenario. A future research direction could extend this investigation to the downlink, exploring the PN impact, the limitations of single-carrier PN-aware estimators, and the potential performance gains of accurate PN-aware estimators for downlink transmission.

References

[1] W. Chen, X. Lin, J. Lee, *et al.*, "5G-advanced toward 6G: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 6, pp. 1592–1619, Jun. 2023.

[2] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," *Proc. IEEE*, vol. 109, no. 7, pp. 1166–1199, Jul. 2021.

[3] Ericsson, "Ericsson mobility report," Ericsson AB, Tech. Rep., Jun. 2024.

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[5] H. Q. Ngo, *Massive MIMO: Fundamentals and system designs*. Linköping University Electronic Press, 2015, vol. 1642.

[6] E. Kolta, T. Hatt, and S. Moore, "Going green: Benchmarking the energy efficiency of mobile," GSMA Intelligence, Tech. Rep., Jun. 2021.

[7] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commu.*, vol. 16, no. 6, pp. 4038–4051, 2017.

[8] X. Yang, M. Matthaiou, J. Yang, C.-K. Wen, F. Gao, and S. Jin, "Hardware-constrained millimeter-wave systems for 5G: Challenges, opportunities, and solutions," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 44–50, Jan. 2019.

[9] X. Hu, C. Zhong, X. Chen, W. Xu, H. Lin, and Z. Zhang, "Cell-free massive MIMO systems with low resolution ADCs," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6844–6857, 2019.

[10] H. Guo, H. Wymeersch, B. Makki, *et al.*, "Integrated communication, localization, and sensing in 6G D-MIMO networks," 2024, [Oneline]. Available: `https://arxiv.org/pdf/2403.19785`.

[11] C. Fager, T. Eriksson, F. Barradas, K. Hausmair, T. Cunha, and J. C. Pedro, "Linearity and efficiency in 5G transmitters: New techniques for analyzing efficiency, linearity, and linearization in a 5G active antenna transmitter context," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 35–49, May 2019.

[12] T. H. Lee and A. Hajimiri, "Oscillator phase noise: A tutorial," *IEEE J. Solid-sJ. Solid-state Circuitstate Circuits*, vol. 35, no. 3, pp. 326–336, Mar. 2000.

[13] H. Wymeersch, H. Chen, H. Guo, *et al.*, "6G positioning and sensing through the lens of sustainability, inclusiveness, and trustworthiness," 2023, [Oneline]. Available: `https://arxiv.org/pdf/2309.13602`.

[14] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.

[15] Y. Wu, U. Gustavsson, A. Graell i Amat, and H. Wymeersch, "Low complexity joint impairment mitigation of I/Q modulator and PA using neural networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 54–64, Nov. 2021.

[16] C. Tarver, A. Balalsoukas-Slimining, C. Studer, and J. R. Cavallaro, "Virtual DPD neural network predistortion for OFDM-based MU-massive MIMO," in *Asilomar Conf. Signals, Syst. Comput.*, Oct. 2021, pp. 376–380.

[17] Y. Wu, U. Gustavsson, M. Valkama, A. Graell i Amat, H. Wymeersch, *et al.*, "Time vs. frequency domain DPD for massive MIMO: Methods and performance analysis," 2024, [Oneline]. Available: `https://arxiv.org/pdf/2402.16577`.

[18] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A deep learning system for twitter sentiment classification," in *Proc. Int. workshop on semantic evaluation (SemEval)*, 2014, pp. 208–212.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, [Oneline]. Available: `https://arxiv.org/pdf/1409.1556`.

[20] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. field robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[21] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *18th Int. Workshop Signal Process. Advances in Wireless Commun. (SPAWC)*, IEEE, 2017, pp. 1–5.

[22] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, 2018.

[23] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, 2018.

[24] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," 2017, [Oneline]. Available: `https://arxiv.org/pdf/1707.07980`.

[25] L. Sanguinetti, A. Zappone, and M. Debbah, "Deep learning power allocation in massive MIMO," in *IEEE Asilomar Conf. signals, Syst. and computers*, IEEE, 2018, pp. 1257–1261.

[26] G. Zhao, Y. Li, C. Xu, Z. Han, Y. Xing, and S. Yu, "Joint power control and channel allocation for interference mitigation based on reinforcement learning," *IEEE Access*, vol. 7, pp. 177 254–177 265, 2019.

[27] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Aug. 2019.

[28] S. Ambrogio, P. Narayanan, A. Okazaki, *et al.*, "An analog-AI chip for energy-efficient speech recognition and transcription," *Nature*, vol. 620, no. 7975, pp. 768–775, Aug. 2023.

[29] T. Liu, Y. Ye, S. Yin, *et al.*, "Digital predistortion linearization with deep neural networks for 5G power amplifiers," in *Eur. Microw Conf. Central Europ (EuMCE)*, IEEE, May 2019, pp. 216–219.

[30] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2398–2412, Dec. 2020.

[31] S. Liu, T. Wang, and S. Wang, "Joint compensation of CFO and IQ imbalance in OFDM receiver: A deep learning based approach," in *IEEE Int. Conf. Commun. in China (ICCC)*, IEEE, Jul. 2021, pp. 793–798.

[32] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications," *Development and Analysis of Deep Learning Architectures*, pp. 223–266, Nov. 2019.

[33] J. G. Proakis and M. Salehi, *Digital communications*. McGraw-hill, 2008.

[34] E. A. Lee and D. G. Messerschmitt, *Digital communication*. Springer Science & Business Media, 2012.

[35] B. Waggener and W. N. Waggener, *Pulse code modulation techniques*. Springer Science & Business Media, 1995.

[36] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. consumer electron.*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[37] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[38] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[39] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. IEEE Int. Conf. Commun.*, IEEE, vol. 2, May 1993, pp. 1064–1070.

[40] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. theory*, vol. 8, no. 1, pp. 21–28, 1962.

[41] T. S. Rappaport, *Wireless communications: principles and practice*. Cambridge University Press, 2024.

[42] N. S. Alagha and P. Kabal, "Generalized raised-cosine filters," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 989–997, 1999.

[43] D. A. Johns and K. Martin, *Analog integrated circuit design*. John Wiley & Sons, 2008.

[44] M. J. Pelgrom and M. J. Pelgrom, *Analog-to-digital conversion*. Springer, 2013.

[45] J. B. Anderson, *Digital transmission engineering*. John Wiley & Sons, 2006.

[46] M. K. Kazimierczuk, *RF power amplifiers*. John Wiley & Sons, 2014.

[47] S. C. Cripps, *Advanced techniques in RF power amplifier design*. Artech House, 2002.

[48] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.

[49] P. M. Shankar, *Fading and shadowing in wireless systems*. Springer, 2017.

[50] G. Kolumbán, M. P. Kennedy, and L. O. Chua, "The role of synchronization in digital communications using chaos. I. fundamentals of digital communications," *IEEE Trans. circuits Syst. I: Fundam. Theory and Appl.*, vol. 44, no. 10, pp. 927–936, 1997.

[51] U. Mengali, *Synchronization techniques for digital receivers*. Springer Science & Business Media, 2013.

[52] F. M. Gardner, *Phaselock techniques*. John Wiley & Sons, 2005.

[53] G. D. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[54] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems*. Krieger Publishing Co., Inc., 2006.

[55] L. Ding, G. T. Zhou, D. R. Morgan, *et al.*, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[56] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based Volterra behavioral modeling of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4323–4332, Dec. 2006.

[57] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[58] P. Celka, N. J. Bershad, and J.-M. Vesin, "Stochastic gradient identification of polynomial Wiener systems: Analysis and application," *IEEE Trans. Signal Process.*, vol. 49, no. 2, pp. 301–313, Feb. 2001.

[59] A. E. Nordsjo and L.-H. Zetterberg, "Identification of certain time-varying nonlinear Wiener and Hammerstein systems," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 577–592, Mar. 2001.

[60] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[61] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[62] M. Valkama, M. Renfors, and V. Koivunen, "Advanced methods for I/Q imbalance compensation in communication receivers," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2335–2344, Oct. 2001.

[63] A. Tarighat, R. Bagheri, and A. H. Sayed, "Compensation schemes and performance analysis of IQ imbalances in OFDM receivers," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3257–3268, Aug. 2005.

[64] T. Schenk, *RF imperfections in high-rate wireless systems: impact and digital compensation*. Springer Science & Business Media, 2008.

[65] L. Anttila, M. Valkama, and M. Renfors, "Frequency-selective I/Q mismatch calibration of wideband direct-conversion transmitters," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 55, no. 4, pp. 359–363, Apr. 2008.

[66] L. Ding, Z. Ma, D. R. Morgan, M. Zierdt, and G. T. Zhou, "Compensation of frequency-dependent gain/phase imbalance in predistortion linearization systems," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 55, no. 1, pp. 390–397, Feb. 2008.

[67] H. Cao, A. S. Tehrani, C. Fager, T. Eriksson, and H. Zirath, "I/Q imbalance compensation using a nonlinear modeling approach," *IEEE Trans. Microw. Theory Tech.*, vol. 57, no. 3, pp. 513–518, Mar. 2009.

[68] L. Anttila, P. Händel, and M. Valkama, "Joint mitigation of power amplifier and I/Q modulator impairments in broadband direct-conversion transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 4, pp. 730–739, Mar. 2010.

[69] Z. Zhu, X. Huang, and H. Leung, "Joint I/Q mismatch and distortion compensation in direct conversion transmitters," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2941–2951, May 2013.

[70] T. C. Schenk, E. Fledderus, and P. F. Smulders, "Performance analysis of zero-IF MIMO OFDM transceivers with IQ imbalance," *J. Commun.*, vol. 2, no. 7, pp. 9–19, 2007.

[71] B. Maham, O. Tirkkonen, and A. Hjorungnes, "Impact of transceiver I/Q imbalance on transmit diversity of beamforming OFDM systems," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 643–648, Jan. 2012.

[72] S. Zarei, W. Gerstacker, and R. Schober, "I/Q imbalance aware widely-linear precoding for downlink massive MIMO systems," in *IEEE Global Commun. Conf. Workshops*, IEEE, 2014, pp. 301–307.

[73] D. Petrovic, W. Rave, and G. Fettweis, "Effects of phase noise on OFDM systems with and without PLL: Characterization and compensation," *IEEE Trans. Commun.*, vol. 55, no. 8, pp. 1607–1616, Aug. 2007.

[74] Y. Wu, L. Sanguinetti, U. Gustavsson, A. Graell i Amat, and H. Wymeersch, "Impact of phase noise on uplink cell-free massive MIMO OFDM," in *IEEE Global Commun. Conf.*, Dec. 2023, pp. 5829–5834.

[75] A. Pitarokoilis, S. K. Mohammed, and E. G. Larsson, "Effect of oscillator phase noise on uplink performance of large MU-MIMO systems," in *2012 50th Annual Allerton Conf. Communication, Control, and Comput. (Allerton)*, IEEE, 2012, pp. 1190–1197.

[76] E. Björnson, M. Matthaiou, and M. Debbah, "Massive MIMO with non-ideal arbitrary arrays: Hardware scaling laws and circuit-aware design," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4353–4368, Apr. 2015.

[77] A. Pitarokoilis, E. Björnson, and E. G. Larsson, "Performance of the massive MIMO uplink with OFDM and phase noise," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1595–1598, Jun. 2016.

[78] A. Pitarokoilis, S. K. Mohammed, and E. G. Larsson, "Uplink performance of time-reversal MRC in massive MIMO systems subject to phase noise," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 711–723, Sep. 2014.

[79] A. Papazafeiropoulos, E. Björnson, P. Kourtessis, S. Chatzinotas, and J. M. Senior, "Scalable cell-free massive MIMO systems: Impact of hardware impairments," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9701–9715, Sep. 2021.

[80] J. Zheng, J. Zhang, J. Cheng, V. C. Leung, D. W. K. Ng, and B. Ai, "Asynchronous Cell-Free massive MIMO with rate-splitting," *IEEE J. Sel. Areas Commun.*, Jan, 2023.

[81] S.-N. Jin, D.-W. Yue, and H. H. Nguyen, "Spectral efficiency of a frequency-selective cell-free massive MIMO system with phase noise," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 483–487, Nov. 2020.

[82] Ö. Özdogan, E. Björnson, and J. Zhang, "Performance of cell-free massive MIMO with Rician fading and phase shifts," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5299–5315, Aug. 2019.

[83] L. Piazzo and P. Mandarini, "Analysis of phase noise effects in OFDM modems," *IEEE Trans. Commun.*, vol. 50, no. 10, pp. 1696–1705, Oct. 2002.

[84] S. Wu and Y. Bar-Ness, "OFDM systems in the presence of phase noise: Consequences and solutions," *IEEE Trans. Commun.*, vol. 52, no. 11, pp. 1988–1996, Nov. 2004.

[85] S. Wu, P. Liu, and Y. Bar-Ness, "Phase noise estimation and mitigation for OFDM systems," *IEEE Trans. Wireless Commu.*, vol. 5, no. 12, pp. 3616–3625, 2006.

[86] Q. Zou, A. Tarighat, and A. H. Sayed, "Compensation of phase noise in OFDM wireless systems," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5407–5424, Nov. 2007.

[87] T. Pollet, M. Moeneclaey, I. Jeanclaude, and H. Sari, "Effect of carrier phase jitter on single-carrier and multi-carrier qam systems," in *Proc. IEEE Int. Conf. Commun.*, IEEE, vol. 2, 1995, pp. 1046–1050.

[88] Y. Gao, A. P. T. Lau, S. Yan, and C. Lu, "Low-complexity and phase noise tolerant carrier phase estimation for dual-polarization 16-QAM systems," *Optics Express*, vol. 19, no. 22, pp. 21 717–21 729, Oct. 2011.

[89] S. Amin, P. N. Landin, P. Händel, and D. Rönnow, "Behavioral modeling and linearization of crosstalk and memory effects in RF MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 4, pp. 810–823, Mar. 2014.

[90] X. Chen, S. Zhang, and Q. Li, "A review of mutual coupling in MIMO systems," *IEEE Access*, vol. 6, pp. 24 706–24 719, Apr. 2018.

[91] 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, Sep. 2017.

[92] K.-H. Chen and J.-F. Kiang, "Effect of mutual coupling on the channel capacity of MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 398–403, Jan. 2015.

[93] Y. Wu, J. Bergmans, and S. Attallah, "Effects of antenna correlation and mutual coupling on the carrier frequency offset estimation in MIMO systems," in *IEEE Int. Conf. Wireless Commun. Netw. and Mobile Comput. (WiCOM)*, IEEE, Sep. 2010, pp. 1–4.

[94] X. Liu and M. E. Bialkowski, "Effect of antenna mutual coupling on MIMO channel estimation and capacity," *Int. J. Antennas and Propag.*, vol. 2010, no. 1, p. 306 173, 2010.

[95] B. Liao and S.-C. Chan, "Adaptive beamforming for uniform linear arrays with unknown mutual coupling," *IEEE Antennas and Wireless Propag. Lett.*, vol. 11, pp. 464–467, 2012.

[96] A. Ramos, T. Varum, and J. N. Matos, "A review on mutual coupling reduction techniques in mmwaves structures and massive MIMO arrays," *IEEE Access*, 2023.

[97] C. Studer, M. Wenk, and A. Burg, "MIMO transmission with residual transmit-RF impairments," in *IEEE Int. ITG Workshop on smart antennas (WSA)*, IEEE, 2010, pp. 189–196.

[98] P. Zetterberg, "Experimental investigation of tdd reciprocity-based zero-forcing transmit precoding," *EURASIP J. Advances in Signal Process.*, vol. 2011, pp. 1–10, 2011.

[99] E. Björnson, P. Zetterberg, and M. Bengtsson, "Optimal coordinated beamforming in the multicell downlink with transceiver impairments," in *IEEE Global Commun. Conf. (GLOBECOM)*, IEEE, Dec. 2012, pp. 4775–4780.

[100] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Trans. Inf. theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[101] H. Chen, S. R. Aghdam, M. F. Keskin, *et al.*, "MCRB-based performance analysis of 6G localization under hardware impairments," in *IEEE Int. Conf. Commun. Workshops*, IEEE, 2022, pp. 115–120.

[102] U. Gustavsson, C. Sanchéz-Perez, T. Eriksson, *et al.*, "On the impact of hardware impairments on massive MIMO," in *IEEE Global Commun. Conf. Workshops*, IEEE, Dec. 2014, pp. 294–300.

[103] F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," *IEEE Microw. Mag.*, vol. 10, no. 7, pp. 52–64, 2009.

[104] O. Hammi, F. M. Ghannouchi, S. Boumaiza, and B. Vassilakis, "A data-based nested LUT model for RF power amplifiers exhibiting memory effects," *IEEE Micro. Wireless Compon. Lett.*, vol. 17, no. 10, pp. 712–714, 2007.

[105] C. Eun and E. J. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.

[106] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[107] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1468–1479, 2004.

[108] P. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *2005 Asia-Pacific Microw. Conf. Proceed.*, IEEE, vol. 2, 2005, 4–pp.

[109] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Augmented Hammerstein predistorter for linearization of broad-band wireless transmitters," *IEEE Trans. Microw. Theory and Technol.*, vol. 54, no. 4, pp. 1340–1349, 2006.

[110] B. Narasimhan, D. Wang, S. Narayanan, H. Minn, and N. Al-Dhahir, "Digital compensation of frequency-dependent joint Tx/Rx I/Q imbalance in OFDM systems under high mobility," *IEEE J. Sel. Topics Sig. Process.*, vol. 3, no. 3, pp. 405–417, 2009.

[111] Y. Tsai, C.-P. Yen, and X. Wang, "Blind frequency-dependent I/Q imbalance compensation for direct-conversion receivers," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 1976–1986, 2010.

[112] P. Robertson and S. Kaiser, "Analysis of the effects of phase-noise in orthogonal frequency division multiplex (OFDM) systems," in *Proc. IEEE Int. Conf. Commun.*, IEEE, vol. 3, 1995, pp. 1652–1657.

[113] S. Wu and Y. Bar-Ness, "A phase noise suppression algorithm for OFDM-based wlans," *IEEE Commun. Lett.*, vol. 6, no. 12, pp. 535–537, 2002.

[114] S. Bittner, W. Rave, and G. Fettweis, "Joint iterative transmitter and receiver phase noise correction using soft information," in *IEEE Int. Conf. Commun.*, IEEE, 2007, pp. 2847–2852.

[115] S. Bittner, E. Zimmermann, and G. Fettweis, "Exploiting phase noise properties in the design of MIMO-OFDM receivers," in *IEEE Wireless Commun. Netw. Conf.*, IEEE, 2008, pp. 940–945.

[116] 3GPP, "NR; Physical layer procedures for data," 3GPP, TS 38.214, 2018, Version 16.4.0.

[117] S. Zhu, H. Liu, Z. Chen, and P. Wen, "A compact gain-enhanced Vivaldi antenna array with suppressed mutual coupling for 5G mmWave application," *IEEE Antennas Wireless Propag. Let.*, vol. 17, no. 5, pp. 776–779, 2018.

[118] M. Li, L. Jiang, and K. L. Yeung, "Novel and efficient parasitic decoupling network for closely coupled antennas," *IEEE Trans. Antennas and Propag.*, vol. 67, no. 6, pp. 3574–3585, 2019.

[119] K. Hausmair, U. Gustavsson, C. Fager, and T. Eriksson, "Modeling and linearization of multi-antenna transmitters using over-the-air measurements," in *Proc. IEEE ISCAS*, IEEE, May 2018, pp. 1–4.

[120] E. Zenteno, S. Amin, M. Isaksson, D. Rönnow, and P. Händel, "Combating the dimensionality of nonlinear MIMO amplifier predistortion by basis pursuit," in *IEEE Eur. Microw. Conf.*, IEEE, 2014, pp. 833–836.

[121] Y. Yu, H.-S. Lui, C. H. Niow, and H. T. Hui, "Improved doa estimations using the receiving mutual impedances for mutual coupling compensation: An experimental study," *IEEE Trans. Wireless Commu.*, vol. 10, no. 7, pp. 2228–2233, 2011.

[122] R. M. Lebrón, P.-S. Tsai, J. M. Emmett, C. Fulton, and J. L. Salazar-Cerreno, "Validation and testing of initial and in-situ mutual coupling-based calibration of a dual-polarized active phased array antenna," *IEEE Access*, vol. 8, pp. 78 315–78 329, 2020.

[123] Y. Xiao and Y. Wang, "Deep learning-based mutual coupling modeling and baseband decoupling algorithm for MIMO systems," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1986–1990, 2020.

[124] X. Zhang, L. E. Larson, and P. Asbeck, *Design of linear RF outphasing power amplifiers*. Artech House, 2003.

[125] R. A. Shafik, M. S. Rahman, A. R. Islam, and N. S. Ashraf, "On the error vector magnitude as a performance metric and comparative analysis," in *Int. Conf. Emerg. Technol.*, Nov. 2006, pp. 27–31.

[126] 3GPP, "NR; Base station (BS) radio transmission and reception (Release 15)," 3GPP, Tech. Rep. 38.104, 2020.

[127] 3GPP, "NR; Base station (BS) radio transmission and reception (Release 15)," 3GPP, Tech. Rep. 38.104, 2020.

[128] E. Landau, *Handbuch der Lehre von der Verteilung der Primazahlen*. BG Teubner, 1909, vol. 1.

[129] A. Chockalingam, "Low-complexity algorithms for large-MIMO detection," in *IEEE Int. Symp. Commun. Control Signal Process. (ISCCSP)*, IEEE, 2010, pp. 1–6.

[130] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surv. & Tut.*, vol. 17, no. 4, pp. 1941–1988, 2015.

[131] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Commun. Surv. & Tut.*, vol. 21, no. 4, pp. 3109–3132, 2019.

[132] S. He, Z. Li, Y. Tang, Z. Liao, F. Li, and S.-J. Lim, "Parameters compressing in deep learning," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 321–336, 2020.

[133] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, 2018.

[134] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, [Oneline]. Available: `https://arxiv.org/pdf/1710.09282`.

[135] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.

[136] S. Zarei, W. Gerstacker, R. R. Müller, and R. Schober, "Low-complexity linear precoding for downlink large-scale MIMO systems," in *IEEE Annual Int. Symp. Personal, Indoor, and Mobile Radio Commun. (PIMRC)*, IEEE, 2013, pp. 1119–1124.

[137] G. Auer, S. Sand, and A. Dammann, "Comparison of low complexity OFDM channel estimation techniques," in *Proc. of Int. OFDM Workshop*, 2003.

[138] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Apr. 2020.

[139] J. Moon and B. Kim, "Enhanced Hammerstein behavioral model for broadband wireless transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 59, no. 4, pp. 924–933, Apr. 2011.

[140] S. Afsardoost, T. Eriksson, and C. Fager, "Digital predistortion using a vector-switched model," *IEEE Trans. Microw. Theory Tech.*, vol. 60, no. 4, pp. 1166–1174, Apr. 2012.

[141] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *ICLR*, May 2016.

[142] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient DNNs," in *Adv. Neural Inform. Process. Syst. (NIPS)*, Barcelona, Spain: Curran Associates Inc., 2016, pp. 1379–1387, ISBN: 9781510838819.

[143] M. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," *ICLR*, May 2018.

[144] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," 2019, [Oneline]. Available: `https://arxiv.org/pdf/1902.09574`.

[145] S. Han, X. Liu, H. Mao, *et al.*, "EIE: efficient inference engine on compressed deep neural network," in *ACM/IEEE ISCA*, Jun. 2016, pp. 243–254.

[146] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. solid-state circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[147] S. Zhang, Z. Du, L. Zhang, *et al.*, "Cambricon-X: An accelerator for sparse neural networks," in *IEEE/ACM MICRO*, Oct. 2016, 20:1–20:12.

[148] S. Dey, K.-W. Huang, P. A. Beerel, and K. M. Chugg, "Pre-defined sparse neural networks with hardware acceleration," *IEEE J. Emerg. Sel. Topics Circuits and Syst.*, vol. 9, no. 2, pp. 332–345, Mar. 2019.

[149] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, `http://www.deeplearningbook.org`.

[150] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into deep learning*. Cambridge University Press, 2023.

[151] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural Inf. Process. systems*, vol. 25, 2012.

[152] J. Achiam, S. Adler, S. Agarwal, *et al.*, "GPT-4 technical report," 2023, [Oneline]. Available: `https://arxiv.org/pdf/2303.08774`.

[153] H. B. Barlow, "Unsupervised learning," *Neural computation*, vol. 1, no. 3, pp. 295–311, 1989.

[154] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: A brief survey," *A review of machine learning techniques for Process. multimedia content*, vol. 1, no. 2004, pp. 9–16, 2004.

[155] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*. Springer, 2016, vol. 9.

[156] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE access*, vol. 8, pp. 80 716–80 727, 2020.

[157] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[158] F.-Y. Wang, J. J. Zhang, X. Zheng, *et al.*, "Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 2, pp. 113–120, 2016.

[159] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation (Cat. No. 02CH37292)*, IEEE, vol. 4, 2002, pp. 3404–3410.

[160] B. R. Kiran, I. Sobh, V. Talpaert, *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[161] M. Yin, A. K. Veldanda, A. Trivedi, *et al.*, "Millimeter wave wireless assisted robot navigation with link state classification," *IEEE Open J. Commun. Society*, vol. 3, pp. 493–507, 2022.

[162] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. artificial intelligence*, vol. 30, 2016.

[163] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural Inf. Process. Syst.*, vol. 12, 1999.

[164] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[165] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[166] Y. Wu, J. Song, C. Häger, U. Gustavsson, A. G. i Amat, and H. Wymeersch, "Symbol-based over-the-air digital predistortion using reinforcement learning," in *IEEE Int. Conf. Commun.*, IEEE, may. 2022, pp. 2615–2620.

[167] C. Szepesvári, *Algorithms for reinforcement learning*. Springer nature, 2022.

[168] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[169] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[170] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. fourteenth Int. Conf. artificial intelligence and statistics*, JMLR Workshop and Conf. Proceed., 2011, pp. 315–323.

[171] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

[172] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural Inf. Process. Syst.*, vol. 31, 2018.

[173] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade: Second Edition*, Springer, 2012, pp. 421–436.

[174] T. Wang, L. Zhang, and S. C. Liew, "Deep learning for joint MIMO detection and channel decoding," in *IEEE 30th Annual Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*, IEEE, 2019, pp. 1–7.

[175] J. M. J. Huttunen, D. Korpi, and M. Honkala, "DeepTx: Deep learning beamforming with channel prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 3, pp. 1855–1867, Mar. 2023.

[176] M. Honkala, D. Korpi, and J. M. J. Huttunen, "DeepRx: Fully convolutional deep learning receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3925–3940, Jun. 2021.

[177] D. Korpi, M. Honkala, J. M. Huttunen, and V. Starck, "DeepRx MIMO: Convolutional MIMO detection with learned multiplicative transformations," in *IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–7.

[178] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, 2019.

[179] X.-F. Kang, Z.-H. Liu, and M. Yao, "Deep learning for joint pilot design and channel estimation in MIMO-OFDM systems," *Sensors*, vol. 22, no. 11, p. 4188, 2022.

[180] J. Jang, H. Lee, I.-M. Kim, and I. Lee, "Deep learning for multi-user MIMO systems: Joint design of pilot, limited feedback, and precoding," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7279–7293, 2022.

[181] J. Guo, T. Chen, S. Jin, G. Y. Li, X. Wang, and X. Hou, "Deep learning for joint channel estimation and feedback in massive MIMO systems," *Digital Commun. and Networks*, vol. 10, no. 1, pp. 83–93, 2024.

[182] X. Li, X. Yu, T. Sun, J. Guo, and J. Zhang, "Joint scheduling and deep learning-based beamforming for fd-MIMO systems over correlated rician fading," *IEEE Access*, vol. 7, pp. 118 297–118 309, 2019.

[183] M. U. Khan, E. Testi, M. Chiani, and E. Paolini, "Joint power control and pilot assignment in cell-free massive MIMO using deep learning," *IEEE Open J. Commun. Society*, 2024.

[184] A. K. Nair and V. Menon, "Joint channel estimation and symbol detection in MIMO-OFDM systems: A deep learning approach using bi-lstm," in *IEEE Int. Conf. communication systems & networks (COMSNETS)*, IEEE, 2022, pp. 406–411.

[185] Q. Sun, H. Zhao, J. Wang, and W. Chen, "Deep learning-based joint CSI feedback and hybrid precoding in FDD mmWave massive MIMO systems," *Entropy*, vol. 24, no. 4, p. 441, 2022.

[186] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[187] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Asilomar Conf. Signals, Syst. and Computers*, IEEE, 2018, pp. 298–303.

[188] T. Van Luong, Y. Ko, N. A. Vien, M. Matthaiou, and H. Q. Ngo, "Deep energy autoencoder for noncoherent multicarrier mu-simo systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3952–3962, Mar. 2020.

[189] J. Song, C. Häger, J. Schröder, A. G. i Amat, and H. Wymeersch, "End-to-end autoencoder for superchannel transceivers with hardware impairment," in *Optical Fiber Communication Conf.*, Optica Publishing Group, 2021, F4D–6.

[190] J. Song, V. Lauinger, Y. Wu, *et al.*, "Blind channel equalization using vector-quantized variational autoencoders," 2023, [Oneline]. Available: `https://arxiv.org/pdf/2302.11687`.

[191] Y. Liu and O. Simeone, "HyperRNN: Deep learning-aided downlink CSI acquisition via partial channel reciprocity for FDD massive MIMO," in *IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, 2021, pp. 31–35.

[192] F. A. Aoudia and J. Hoydis, "End-to-end waveform learning through joint optimization of pulse and constellation shaping," in *IEEE Global Commun. Conf. Workshops*, IEEE, Dec. 2021, pp. 1–6.

[193] D. Korpi, M. Honkala, J. M. Huttunen, F. A. Aoudia, and J. Hoydis, "Waveform learning for reduced out-of-band emissions under a nonlinear power amplifier," 2022, [Oneline]. Available: `https://arxiv.org/pdf/2201.05524`.

[194] A. Mohammadian and C. Tellambura, "Joint channel and phase noise estimation and data detection for GFDM," *IEEE Open J. Commun. Society*, vol. 2, pp. 915–933, 2021.

[195] X. Tan, Z. Zhong, Z. Zhang, X. You, and C. Zhang, "Low-complexity message passing MIMO detection algorithm with deep neural network," in *IEEE Global Conf. Signal and Inf. Process. (GlobalSIP)*, IEEE, 2018, pp. 559–563.

[196] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *IEEE Global Conf. Signal and Inf. Process. (GlobalSIP)*, IEEE, 2018, pp. 584–588.

[197] M. A. Albreem, A. H. Alhabbash, S. Shahabuddin, and M. Juntti, "Deep learning for massive MIMO uplink detectors," *IEEE Commun. Surv. & Tut.*, vol. 24, no. 1, pp. 741–766, 2021.

[198] L. V. Nguyen, N. T. Nguyen, N. H. Tran, M. Juntti, A. L. Swindlehurst, and D. H. Nguyen, "Leveraging deep neural networks for massive MIMO data detection," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 174–180, 2022.

[199] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10 325–10 329, 2019.

[200] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6315–6328, 2021.

[201] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *IEEE Int. Conf. Commun.*, IEEE, 2020, pp. 1–6.

[202] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based CSI feedback in massive MIMO systems," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 8017–8045, 2022.

[203] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, 2019.

[204] M. Zhang, J. Gao, and C. Zhong, "A deep learning-based framework for low complexity multiuser MIMO precoding design," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 11 193–11 206, 2022.

[205] J. Shi, W. Wang, X. Yi, X. Gao, and G. Y. Li, "Deep learning-based robust precoding for massive MIMO," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7429–7443, 2021.

[206] A. M. Elbir and A. K. Papazafeiropoulos, "Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 552–563, 2019.

[207] J. Yuan, H. Q. Ngo, and M. Matthaiou, "Machine learning-based channel prediction in massive MIMO with channel aging," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 2960–2973, 2020.

[208] J. M. Mateos-Ramos, J. Song, Y. Wu, *et al.*, "End-to-end learning for integrated sensing and communication," in *IEEE Int. Conf. Commun.*, IEEE, 2022, pp. 1942–1947.

[209] S. Rivetti, J. M. Mateos-Ramos, Y. Wu, *et al.*, "Spatial signal design for positioning via end-to-end learning," *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 525–529, 2023.

[210] J. Pihlajasalo, D. Korpi, M. Honkala, *et al.*, "Deep learning OFDM receivers for improved power efficiency and coverage," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5518–5535, 2023.

[211] D. Hu, Y. Zhang, L. He, and J. Wu, "Low-complexity deep-learning-based doa estimation for hybrid massive MIMO systems with uniform circular arrays," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 83–86, 2019.

[212] Y. Zhao, I. G. Niemegeers, and S. H. De Groot, "Power allocation in cell-free massive MIMO: A deep learning method," *IEEE Access*, vol. 8, pp. 87 185–87 200, 2020.

[213] T. Van Chien, T. N. Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5732–5748, 2020.

[214] M. Bashar, A. Akbari, K. Cumanan, *et al.*, "Exploiting deep learning in limited-fronthaul cell-free massive MIMO uplink," *IEEE J. Sele. Areas Commun.*, vol. 38, no. 8, pp. 1678–1697, Aug. 2020.

[215] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, 2019.

[216] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, 2019.

[217] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[218] A. Singh and S. Saha, "Machine/deep learning based estimation and detection in OFDM communication systems with various channel imperfections," *Wireless Networks*, vol. 28, no. 6, pp. 2637–2650, 2022.

[219] F. M. Ghannouchi, M. Younes, and M. Rawat, "Distortion and impairments mitigation and compensation of single-and multi-band wireless transmitters," *IET Microw. Antennas & Propag.*, vol. 7, no. 7, pp. 518–534, May 2013.

[220] Z. A. Khan, E. Zenteno, P. Händel, and M. Isaksson, "Digital predistortion for joint mitigation of I/Q imbalance and MIMO power amplifier distortion," *IEEE Trans. Microw. Theory and Technol.*, vol. 65, no. 1, pp. 322–333, Jan. 2017.

[221] M. Grimm, M. Allén, J. Marttila, M. Valkama, and R. Thomä, "Joint mitigation of nonlinear RF and baseband distortions in wideband direct-conversion receivers," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 1, pp. 166–182, 2014.

[222] Z. Yao, Y. Peng, Y. Wang, *et al.*, "A novel radio frequency fingerprint concealment method based on IQ imbalance compensation and digital pre-distortion," *IEEE Trans. Inf. Forensics and Secur.*, Jul. 2024.

[223] M. Rawat, K. Rawat, M. Younes, and F. Ghannouchi, "Joint mitigation of nonlinearity and modulator imperfections in dual-band concurrent transmitter using neural networks," *Electronics Lett.*, vol. 49, no. 4, pp. 253–255, Feb. 2013.

[224] S. Kumari, K. K. Srinivas, and P. Kumar, "Channel and carrier frequency offset equalization for OFDM based uav communications using deep learning," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 850–853, 2020.

[225] A. Mohammadian, C. Tellambura, and G. Y. Li, "Deep learning-based phase noise compensation in multicarrier systems," *Wireless Commun. Lett.*, vol. 10, no. 10, pp. 2110–2114, Jun. 2021.

[226] R. Liu, X. Xu, and X. Qin, "Joint estimation of IQ imbalance and PA nonlinearity: An iterative scheme," *Digital Signal Process.*, vol. 155, p. 104 730, 2024.

[227] X. Hu, Z. Liu, X. Yu, *et al.*, "Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers," *IEEE Trans. on Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3923–3937, Aug. 2021.

[228] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *IEEE Int. Workshop Signal Process. Syst. (SiPS)*, IEEE, 2019, pp. 266–271.

[229] H. Paaso and A. Mammela, "Comparison of direct learning and indirect learning predistortion architectures," in *IEEE Int. Symp. Wireless Commun. Syst.*, IEEE, Oct. 2008, pp. 309–313.

[230] S. Amin, E. Zenteno, P. N. Landin, D. Rönnow, M. Isaksson, and P. Händel, "Noise impact on the identification of digital predistorter parameters in the indirect learning architecture," in *Swedish Commun. Technol. Workshop (Swe-CTW)*, 2012, pp. 36–39.

[231] Y. Liu, J. J. Yan, H.-T. Dabag, and P. M. Asbeck, "Novel technique for wideband digital predistortion of power amplifiers with an under-sampling ADC," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 11, pp. 2604–2617, Oct. 2014.

[232] Z. Wang, W. Chen, G. Su, F. M. Ghannouchi, Z. Feng, and Y. Liu, "Low feedback sampling rate digital predistortion for wideband wireless transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 11, pp. 3528–3539, Sep. 2016.

[233] N. Guan, N. Wu, and H. Wang, "Digital predistortion of wideband power amplifier with single undersampling ADC," *IEEE Microw. Wireless Compon. Lett.*, vol. 27, no. 11, pp. 1016–1018, Sep. 2017.

[234] Y. Beltagy, P. Mitran, and S. Boumaiza, "Direct learning algorithm for digital predistortion training using sub-Nyquist intermediate frequency feedback signal," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 1, pp. 267–277, Nov. 2018.

[235] X. Liu, W. Chen, L. Chen, F. M. Ghannouchi, and Z. Feng, "Linearization for hybrid beamforming array utilizing embedded over-the-air diversity feedbacks," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 12, pp. 5235–5248, Oct. 2019.

[236] X. Wang, C. Yu, Y. Li, W. Hong, and A. Zhu, "Real-time single channel over-the-air data acquisition for digital predistortion of 5G massive MIMO wireless transmitters," in *Proc. MTT-S IWS, IEEE*, May, 2019, pp. 1–3.

[237] C. Mollén, U. Gustavsson, T. Eriksson, and E. G. Larsson, "Spatial characteristics of distortion radiated from antenna arrays with transceiver nonlinearities," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6663–6679, Aug. 2018.

[238] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Proc. Asilomar Conf. Signals, Syst., Comput*, Oct. 2018, pp. 298–303.

[239] J. Song, Z. He, C. Häger, *et al.*, "Over-the-fiber digital predistortion using reinforcement learning," in *Eur. Conf. Opt.l Commun. (ECOC)*, IEEE, Sep. 2021, pp. 1–4.

[240] M. Isaksson, D. Wisell, and D. Ronnow, "Nonlinear behavioral modeling of power amplifiers using radial-basis function neural networks," in *IEEE MTT-S Int. Microw. Symp. Digest*, IEEE, Jun. 2005, pp. 1967–1970.

[241] M. Rawat, K. Rawat, and F. M. Ghannouchi, "Adaptive digital predistortion of wireless power amplifiers/transmitters using dynamic real-valued focused time-delay line neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 1, pp. 95–104, Jan. 2010.

[242] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Tech.*, vol. 59, no. 4, pp. 913–923, Apr. 2011.

[243] M. Rawat and F. M. Ghannouchi, "A mutual distortion and impairment compensator for wideband direct-conversion transmitters using neural networks," *IEEE Trans. Broadcast.*, vol. 58, no. 2, pp. 168–177, Apr. 2012.

[244] D. Wang, M. Aziz, M. Helaoui, and F. M. Ghannouchi, "Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 30, no. 1, pp. 242–254, Jun. 2018.

[245] 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 and Technol.*, vol. 66, no. 11, pp. 5011–5020, 2018.

[246] Y. Wu, U. Gustavsson, A. Graell i Amat, and H. Wymeersch, "Residual neural networks for digital predistortion," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[247] Y. Wu, U. Gustavsson, M. Valkama, A. Graell i Amat, and H. Wymeersch, "Frequency-domain digital predistortion for massive MU-MIMO-OFDM downlink," in *IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2022, pp. 579–584.

[248] C. Yu, J. Jing, H. Shao, *et al.*, "Full-angle digital predistortion of 5G millimeter-wave massive MIMO transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 7, pp. 2847–2860, Jul. 2019.

[249] X. Liu, Q. Zhang, W. Chen, *et al.*, "Beam-oriented digital predistortion for 5G massive MIMO hybrid beamforming transmitters," *IEEE Trans. Microw. Theory and Techn.*, vol. 66, no. 7, pp. 3419–3432, May 2018.

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

[251] A. Brihuega, L. Anttila, and M. Valkama, "Frequency-domain digital predistortion for OFDM," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 6, pp. 816–818, Mar. 2021.

[252] C. Mollén, E. G. Larsson, U. Gustavsson, T. Eriksson, and R. W. Heath, "Out-of-band radiation from large antenna arrays," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 196–203, Apr. 2018.

[253] A. Brihuega, L. Anttila, and M. Valkama, "Beam-level frequency-domain digital predistortion for OFDM massive MIMO transmitters," *IEEE Trans. Micro. Theory Techn.*, vol. 71, no. 4, pp. 1412–1427, Nov. 2022.

[254] M. Rawat and F. M. Ghannouchi, "A mutual distortion and impairment compensator for wideband direct-conversion transmitters using neural networks," *IEEE Trans. Broadcasting*, vol. 58, no. 2, pp. 168–177, Apr. 2012.