UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

MASTER'S THESIS

TRANSFORMER NN-BASED BEHAVIORAL MODELING AND PREDISTORTION FOR WIDEBAND PAS

Author	Lesthuruge Silva
Supervisor	Prof. Nandana Rajatheva
Second Examiner	Pekka Pirinen
Technical Advisor	Hossein Rezaei

April 2023

Silva L. (2023) Transformer NN-Based Behavioral Modeling and Predistortion for Wideband PAs. University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Wireless Communications Engineering, 51 p.

ABSTRACT

This work investigates the suitability of transformer neural networks (NNs) for behavioral modeling and the predistortion of wideband power amplifiers. We propose an augmented real-valued time delay transformer NN (ARVTDTNN) model based on a transformer encoder that utilizes the multi-head attention mechanism. The inherent parallelized computation nature of transformers enables faster training and inference in the hardware implementation phase. Additionally, transformers have the potential to learn complex nonlinearities and long-term memory effects that will appear in future high-bandwidth power amplifiers. The experimental results based on 100 **MHz LDMOS Doherty PA show that the ARVTDTNN model exhibits superior or comparable performance to the state-of-the-art models in terms of normalized mean square error (NMSE) and adjacent channel power ratio (ACPR). It improves the NMSE and ACPR up to** -37.6 **dB and** -41.8 **dB, respectively. Moreover, this approach can be considered as a generic framework to solve sequence-to-one regression problems with the transformer architecture.**

Keywords: **Digital predistortion, DPD, in-phase and quadrature (I/Q) components, multi-head attention, transformer-encoder, augmented real-valued time delay transformer NN, ARVTDTNN**

TABLE OF CONTENTS

FOREWORD

This thesis explores the feasibility of using transformer neural networks for behavioral modeling and predistortion of wideband power amplifiers. The research has been supported by the Academy of Finland's 6G Flagship program under Grant 346208. I am deeply grateful to my thesis supervisor, Prof. Nandana Rajatheva, and my technical advisor, Hossein Rezaei, as well as my peer researcher, Anusha Rathnayake, for their immense support throughout my master's thesis. They gave me the freedom to explore any direction that could lead to a better solution for the research problem, and their encouragement and guidance helped me achieve this milestone. Discovering a new research area and making significant progress in this research boosted my confidence. I would also like to express my gratitude to all the professors at the University of Oulu for their valuable support. Last but not least, I would like to thank my family members and colleagues for their support and encouragement during my research period and throughout my master's degree. Without them, these achievements would not have been possible. Thank you.

Oulu, 28th April, 2023

Lesthuruge Silva

LIST OF ABBREVIATIONS AND SYMBOLS

Acronyms

1D	One Dimensional
2D	Two Dimensional
5G	Fifth Generation
6G	Sixth Generation
ACI	Adjacent Channel Interference
ACLR	Adjacent Channel Leakage Ratio
ACPR	Adjacent Channel Power Ratio
ADC	Analog to Digital Conversion
AM	Amplitude Modulation
AM/AM	Amplitude Modulation to Amplitude Modulation
AM/PM	Amplitude Modulation to Phase Modulation
ARBFNN	Augmented Radial Basis Function Neural Network
ARVTDNN	Augmented Real-Valued Time Delay Neural Network
ARVTDTNN	Augmented Real-Valued Time Delay Transformer Neural Network
BER	Bit Error Rate
BILSTM	Bidirectional Long-Short Term Memory
CNN	Convolutional Neural Network
D/A	Digital to Analog
DC	Direct Current
DNN	Deep Neural Network
DPD	Digital Predistortion
EVM	Error Vector Magnitude
FC	Fully Connected
FFN	Feed Forward Network
FFNN	Feed Forward Neural Network
FPGA	Field Programmable Gate Array
Gbps	Gigabits per second
GMP	Generalized Memory Polynomial
GPU	Graphical Processing Unit
I	In-phase
IF	Intermediate Frequency
IIP3	Third-order Input Intercept Point
IMD	Inter Modulation Distortion
IP	Intellectual Property
IQ	In-phase and Quadrature
ISI	Inter Symbol Interference
LINC	Linear Amplification with Nonlinear Components
LSTM	Long-Short Term Memory
LUT	Look-Up Table
MHz	Mega Hertz
MSE	Mean Square Error
NLP	Natural Language Processing
NMSE	Normalized Mean Square Error

NN	Neural Network
OFDM	Orthogonal Frequency Division Multiplexing
OPBO	Output Power Back-Off
PA	Power Amplifier
PAE	Power Added Efficiency
PAPR	Peak to Average Power Ratio
PM/AM	Phase Modulation to Amplitude Modulation
PM/PM	Phase Modulation to Phase Modulation
PSD	Power Spectral Density
PTI-DPD	Power Temperature Inclusive Digital Predistortion
Q	Quadrature
QAM	Quadrature Amplitude Modulation
ReLU	Rectified Linear Unit
RF	Radio Frequency
RMS	Root Mean Square
RNN	Recurrent Neural Network
RVTDCNN	Real-Valued Time Delay Convolutional Neural Network
RVTDNN	Real-Valued Time Delay Neural Network
SSPA	Solid State Power Amplifier
TDL	Tapped Delay Line
THD	Total Harmonic Distortion
TWTA	Traveling Wave Tube Amplifier

Symbols

P_{in}	Input power of the PA
P_{out}	Output power of the PA
dB	decibel
x_{in}	Baseband complex input signal of the PA
x_{out}	Baseband complex output signal of the PA
G	Complex gain of the PA
η_{PA}	Efficiency of the PA
P_{source}	Power drawn from the power supply by PA
P	Nonlinearity order
ω	Angular frequency
$S(f)$	Power spectral density
P_{max}	Maximum power level
P_{avg}	Average power level
$P_{out,sat}$	Saturation output power of the PA
y_{actual}	Actual/measured PA output
y_{model}	Modeled PA output
$G_A(A)$	AM/AM characteristics of the PA
$\phi_G(A)$	AM/PM characteristics of PA
M	Memory depth
I_{in}	Input in-phase component
Q_{in}	Input quadrature component
I_{out}	Output in-phase component
Q_{out}	Output quadrature component

X_n	Reshaped input to the model
T	Sequence length
Q	Query matrix
K	Key matrix
V	Value matrix
d_k	Hidden dimension of queries/keys
d_v	Hidden dimension of values
$< sos >$	Start of the sequence
$< eos >$	End of the sequence
d_{model}	Dimension inside the transformer
d_{in}	Number of input features
X_{emb}	Embedded output
X_{ffn}	Feed-forward network input
X_{pos}	Positional encoded output
X_{enc}	Transformer encoder output

Operators

$\lvert \cdot \rvert$	Absolute value
$\sum$	Summation operation
$\angle$	Angle

1 INTRODUCTION

This chapter serves as an introduction to the research problem, which centers around addressing RF transmitter impairments and the requirement of linearization. The chapter provides a brief explanation of the DPD technique, along with other methods for linearization. It also describes the existing problem and proposed solution. Furthermore, the chapter outlines the contribution of the thesis and provides a brief overview of the remaining sections.

1.1 RF Transmitter

Modern communication systems are expected to meet a wide range of requirements and demands, increasing the complexity of the elements in the communication chain [1]. The radio frequency (RF) transmitter is designed to manage performance metrics such as linearity, bandwidth, and power efficiency. Linearity is essential for avoiding distortions that can arise from the analog circuitry. Bandwidth is crucial for achieving higher data rates, while power efficiency affects infrastructure costs and environmental impact. The simplified block diagram of the RF transmitter is shown in Figure 1.1, consisting of data source, baseband signal processing, digital to analog converter (DAC), in-phase and quadrature (IQ) modulator, mixer, power amplifier (PA), and antenna. The transmitter converts the baseband signal containing data into a form that can be transmitted through the channel [2]. Initially, the data undergoes baseband signal processing performed in the digital domain. The processed data then undergo digital to analog (D/A) conversion. Next IQ modulator up-converts the complex signal into an intermediate frequency (IF) or directly to RF. Mixer converts the IF signal to the RF signal. However, a mixer is not required in zero-IF architectures as the baseband signal is directly converted to the RF signal. Finally, the PA amplifies the RF signal and is transmitted to the channel via the antenna.

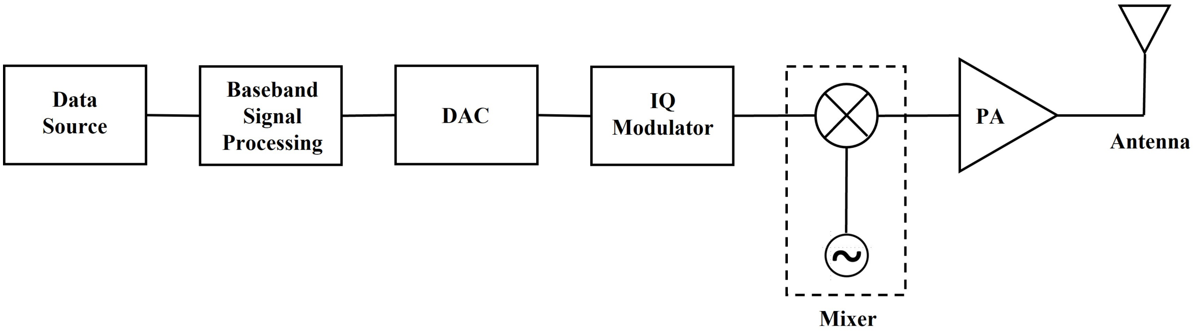


Figure 1.1. Simplified block diagram of wireless transmitter.

Each hardware component in the transmitter design introduces some imperfection to the signal, such as frequency response distortions, amplitude distortions, phase distortions, group delay distortions, direct current (DC) offset, gain, and phase imbalance [1]. Although various sources contribute to transmitter distortions, significant distortions are due to the nonlinearity present in the RF PA. Thus, compensation of PA nonlinearity is an essential aspect of transmitter design.

1.2 Power Amplifier Nonlinearity and Requirement of Linearizers

The PA is a vital component in the RF transmitter as it amplifies the IQ-modulated radio signal to a higher power level, enabling it to be transmitted through the antenna [3]. There are several classes of amplifier operation, each with different levels of nonlinearities. Figure 1.2 depicts the standard PA input-output characteristics curve, illustrating the linear and saturation regions. Usually, the PA is operated close to its saturation point to achieve maximum efficiency for high peak-to-average power ratio (PAPR) signals. However, operating near the saturation point can result in significant nonlinearities in the PA's output, leading to amplitude and phase distortion. Hence, there's an inverse relationship between PA efficiency and linearity, in which more nonlinear PAs can achieve higher efficiency. However, nonlinearity can cause spectral regrowth, leading to adjacent channel interference (ACI) and in-band distortions, increasing the bit error rate (BER) degradation. Spectral regrowth is a significant concern for telecommunication operators since the allowed frequency ranges are strictly regulated, and interference with other operators' frequency bands is not permitted [3].

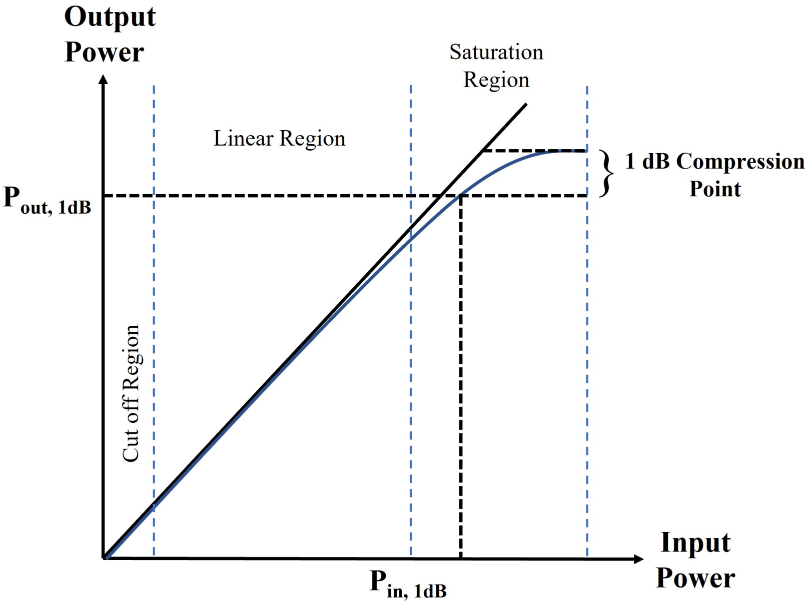


Figure 1.2. PA input-output power characteristics curve.

The rapid thermal time constants of the active devices, non-flat frequency response of the matching networks, and variations in the biasing circuit cause memory effects in the PA output [4]. That effect becomes more dominant as the signal bandwidth increases. The upcoming 6^{th} generation (6G) standard demands a significant boost in data transmission rates up to several gigabits per second (Gbps). To support that, signal bandwidths of several hundred megahertz (MHz) should be employed, which will severely increase the memory effect impairment of the PA [5]. The memory effect and nonlinearity of the PA can provoke spectral regrowth, reducing the adjacent channel power ratio (ACPR) performance. Moreover, modern communication systems utilize orthogonal frequency division multiplexing (OFDM) due to its tolerance to inter-symbol interference (ISI) and spectral efficiency. OFDM performs well in multi-path environments with frequency selective fading and has simple implementation. However, the high PAPR of OFDM makes it vulnerable to the nonlinearities of the PA [3]. Thus, linearization of PA is essential in transmitter design to avoid spectral regrowth and in-band distortions.

1.3 Digital Predistortion (DPD) and Other Linearization Techniques

The most straightforward approach to achieve linearization is to back off the input power level, ensuring that the PA operates entirely within its linear region [4]. However, this method is associated with a larger size and high cost. Another approach would be the feedforward technique, which subtracts the distortion from the output. This method is also highly costly due to requiring an extra RF amplifier and analog combining networks. Another technique called Linear amplification with non-linear components (LINC) utilizes two phase-altered nonlinear PAs for linearization. This also has the disadvantage of requiring a high-power analog combining network [4].

In recent years, DPD has become a popular alternative solution for linearization due to its simple implementation capabilities [4]. DPD helps to reduce in-band and out-band distortions while maintaining the operation close to the maximum rated power, and it also reduces the size and cost significantly.

1.4 Problem Description and Proposed Solution

PAs play a critical role in modern communication systems. However, the existing behavioral and linearization models for PAs have limitations in addressing the complex and nonlinear relationships that future high-bandwidth PAs may exhibit. This is especially problematic given the increasing demand for high data rate requirements in modern communication systems, which can significantly increase the memory effect of PAs, leading to more distortions that existing models may not be able to resolve.

Furthermore, the training and inference of recurrent neural network (RNN) and long short-term memory (LSTM) models, which have been commonly used for PA behavioral modeling and linearization, present hardware acceleration challenges due to their sequential nature. As a result, this study proposes a transformer-based deep learning solution for behavioral modeling and linearization of PAs.

The transformer model structure, introduced in 2017 in the paper "Attention is All You Need" [6] is designed with an encoder-decoder-based architecture that enables high computation parallelism in both training and inference phases. The transformer model can process the entire data sequence in parallel through its self-attention mechanism, identifying long-term sequence dependencies, and can handle complex time series dependencies that are challenging for existing sequence models. By utilizing the full extent of available hardware resources, the transformer-based model potentially overcomes the limitations of existing models, improving the design of high-bandwidth PAs for future communication systems.

1.5 Thesis Contribution

- A transformer-based PA behavioral model is introduced to utilize parallel computation in transformers during the training and inference phases.
- A novel approach is presented to handle sequence-to-one time-series regression problems by modifying the existing transformer architecture specifically designed for natural language processing (NLP) tasks.
- A transformer-based DPD architecture is introduced to linearize the PA nonlinear effects, outperforming most existing state-of-the-art solutions.

1.6 Thesis Outline

The remainder of the thesis is structured as follows:

- **Chapter 2:** All the necessary theory parts are highlighted, including PA characteristics, importance of PA linearization and existing linearization techniques.
- **Chapter 3:** This chapter provides an overview of state-of-the-art behavioral modeling and predistortion techniques, with a focus on neural network-based models.
- **Chapter 4:** This chapter discusses the use of transformer-based techniques for behavioral modeling and predistortion, exploring their underlying requirements and how they effectively linearize complex, nonlinear PAs.
- **Chapter 5:** The results of the study are presented in this chapter, addressing the problem using simulation tools and providing comparisons with existing models.
- **Chapter 6:** In this chapter, we summarise the contribution of our study and suggest potential areas for further investigation.

2 BACKGROUND ON POWER AMPLIFIER AND LINEARIZATION TECHNIQUES

This chapter offers a comprehensive overview of the all the necessary theoretical aspects related to the topic. It delves into the characteristics of the PA and emphasizes the significance of linearization. Additionally, the chapter discusses the various linearization techniques that are currently in use.

2.1 Input-Output Power Characteristics

As discussed in Chapter 1, the PA is inherently a nonlinear device, meaning that its input-output characteristics are nonlinear, as depicted in Figure 1.2 [1]. The region where the output of the PA behaves linearly is known as the linear region, which is observed at low input power values. However, beyond a certain input power level, gain compression starts to appear until it gets saturated, and it is known as the saturation region. Gain is the slope of the P_{in} vs. P_{out} graph when both powers are denoted in Watts. The 1 dB compression point measures the PA's linearity. It refers to the point at which the output power deviates from its linear value by 1 dB. In other words, the 1 dB compression point is the input power level at which the amplifier's output power starts to saturate and no longer increases linearly with the input power. This characteristic is important because it affects the signal quality being amplified. PA with a high compression point can handle high-power signals without distorting them [1].

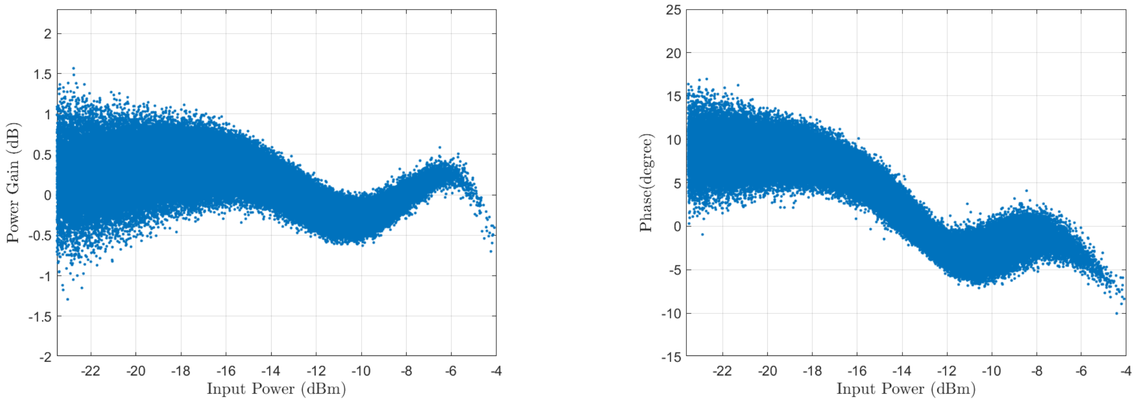
2.2 AM/AM and AM/PM Characteristics

Input-output power characteristic curve does not provide a comprehensive illustration of the PA behavior. Thus, a much better meaningful representation is provided by amplitude modulation to amplitude modulation (AM/AM) and amplitude modulation to phase modulation (AM/PM) characteristics. Generally, the nonlinear transmitters can be described by AM/AM, AM/PM, PM/AM, and PM/PM characteristics. However, PA distortion only depends on amplitude-modulated (AM) signals, and thus AM/AM and AM/PM characteristics are sufficient to realize the PA behavior. AM/AM and AM/PM distortions are generated by the nonlinearity of the PA, while PM/AM and PM/PM distortions occur by gain and phase imbalances in the frequency up-conversion [1].

Let x_{in} and x_{out} be the baseband complex input and output signals of the PA. Then the magnitude and phase of the instantaneous complex gain of the PA can be expressed as

$$\begin{gathered} |G| = \frac{|x_{out}|^2}{|x_{in}|^2}, \\ \angle G = \angle x_{out} - \angle x_{in} \end{gathered} \tag{1}$$

under the assumption that PA does not exhibit PM/AM and PM/PM distortions. As shown in Figure 2.1a, AM/AM characteristic curve is then obtained by plotting the magnitude of instantaneous gain against input power in dB. Similarly, AM/PM characteristic curve is obtained by plotting the phase of instantaneous gain along with input power, as shown in Figure 2.1b [1].



(a) AM/AM characteristics.

(b) AM/PM characteristics.

Figure 2.1. Sample AM/AM and AM/PM characteristics of a PA.

2.3 PA Performance Metrics

Various performance metrics are employed to evaluate the performance of the PA [7]. Efficiency (η_{PA}) is one of the crucial metrics of the PA, which measures the amount of power dissipated during the amplification process. High efficiency indicates that the PA is effective in minimizing power dissipation, which results in longer battery life. Ideally, unity efficiency is the best expectation, but it is not feasible in practical implementations. Efficiency can be calculated as

$$\eta_{PA} = \frac{P_{out}}{P_{source}} \tag{2}$$

where P_{out} refers to the power delivered to the load and P_{source} denotes the power that PA draws from the power supply. Efficiency may provide misleading results in certain cases; thus, power-added efficiency (PAE) is defined, and it can be obtained as

$$PAE = \frac{P_{out} - P_{in}}{P_{source}} * 100\% \tag{3}$$

where P_{in} is the power at PA input. PAE is always smaller than η_{PA}, and higher amplifier gain indicates a higher PAE [2].

The other main performance metric of PA is nonlinearity, which can be evaluated using several methods, including the third-order input intercept point (IIP3), total harmonic distortion (THD), ACPR, and error vector magnitude (EVM). ACPR and EVM are commonly used metrics to evaluate nonlinearity, with ACPR measuring out-of-band distortion and EVM measuring in-band distortion [2].

2.4 Inter-Modulation Distortion (IMD) and Spectral Regrowth

The nonlinearity of the PA causes the generation of unwanted frequency components at the output, known as inter-modulation distortion (IMD) [1]. IMD can be mathematically illustrated with a two-tone signal, and the same principle applies to wideband signals with a continuous spectrum. Let x_{in} and x_{out} be the input and output of the PA, respectively, and the nonlinearity order (P) as two. Then, x_{out} can be modeled with a second-order polynomial as

$$x_{out} = ax_{in} + bx_{in}^2 \tag{4}$$

where a and b are the model coefficients. Let the input signal be a two-tone signal having amplitudes of A_1, A_2 and angular frequencies ω_1 and ω_2 where $\omega_2 > \omega_1$. Then x_{in} can be written as

$$x_{in} = A_1 cos(\omega_1 t) + A_2 cos(\omega_2 t). \tag{5}$$

By substituting x_{in} to (4) and simplifying, x_{out} can be expressed as

$$\begin{aligned} x_{out} = [aA_1cos(\omega_1 t) + aA_2cos(\omega_2 t)] + \frac{b(A_1^2 + A_2^2)}{2} + bA_1A_2cos(\omega_2 - \omega_1)t \\ + \frac{bA_1^2}{2}cos(2\omega_1 t) + bA_1A_2cos(\omega_1 + \omega_2)t + \frac{bA_2^2}{2}cos(2\omega_2 t). \end{aligned} \tag{6}$$

In (6), first two terms with ω_1 and ω_2 frequencies show the intended amplified versions of the input signal. All other terms are unintended and generated due to the nonlinearity of the PA. Some unintended frequencies are close to the useful signal, while others are far away. The frequencies that are far away can be removed through filtering. Figure 2.2 shows the frequency domain interpretation of the transmitter output signal, highlighting the undesired frequency components. This mathematical interpretation can be extended to the N^{th}-order nonlinear model, which would result in the presence of N^{th}-order harmonics and N^{th}-order mixing terms in the output signal. Since the practical input signals are continuous, IMD is observed as a spectrum regrowth around the channel. Hence, the nonlinear effect of the PA results in a considerable amount of spectrum regrowth, which leads to generating interference in adjacent channels [1].

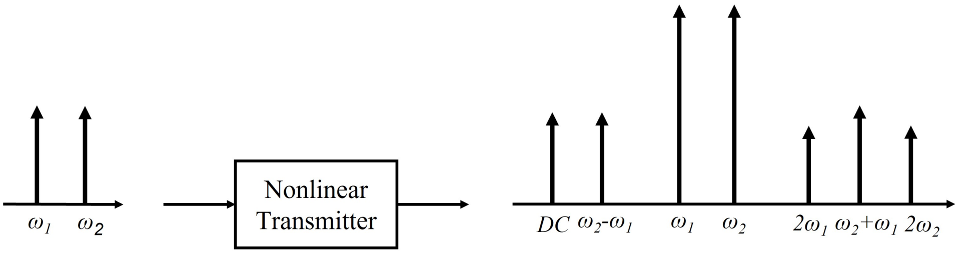


Figure 2.2. Frequency domain of a nonlinear transmitter driven by a two-tone signal.

2.5 Distortion Impact on Variable Amplitude Signals

Since the nonlinearity of the PA compresses the signals, carrying information on amplitude may cause problems [2]. The nonlinear impact can cause significant deviations in the constellation points from their ideal positions and even enter a different decision region. This increases the in-band distortion with a high BER. The Figure 2.3 illustrates the compression impact of the nonlinear PA for a 16QAM modulated signal. Corner constellation points show greater deviation as they carry high power, and can be misinterpreted in a wrong decision region.

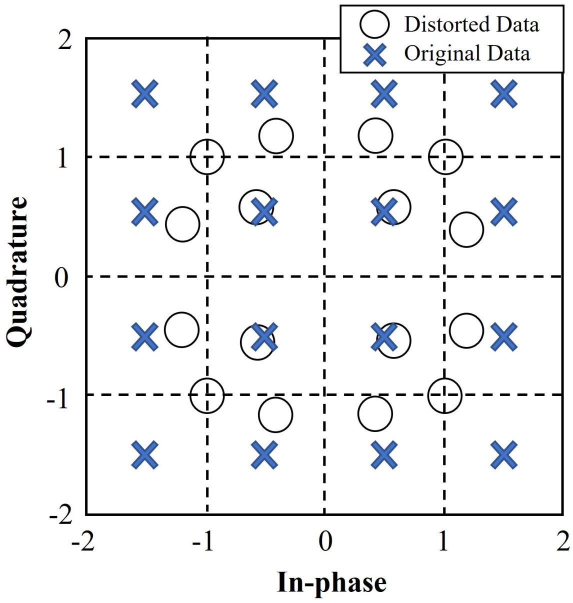


Figure 2.3. Compression impact of the nonlinear PA for a 16QAM signal.

2.6 Adjacent Channel Power Ratio (ACPR)

ACPR measures the nonlinearity of PA in the frequency domain, also known as the adjacent channel leakage ratio (ACLR) [8]. It is defined as the ratio between the mean power of the main channel and the filtered mean power in adjacent channels. ACPR can be expressed as

$$ACPR = 10log\left|\frac{\int_{main} S(f)df}{\int_{adj\_l} S(f)df + \int_{adj\_r} S(f)df}\right| \tag{7}$$

where $S(f)$ denotes the power spectral density (PSD) of the output signal. The numerator represents the mean power of the main channel, while the denominator represents the filtered mean power in both the left and right adjacent channels. ACPR is a crucial metric for measuring ACI that cannot be removed through filtering. Hence, it is important to minimize the power leakage to adjacent channels, and each communication standard includes an ACPR threshold to control that, commonly referred to as the spectrum mask [1].

Figure 2.4 illustrates the main channel across the signal bandwidth centered around 0 frequency. The lower and upper adjacent channels are defined based on an offset frequency between the center of the main channel and that of the considered adjacent channel. Typically, the first and second adjacent channels are defined for both the left and right sides. The transition bands are introduced to reduce ISI [1].

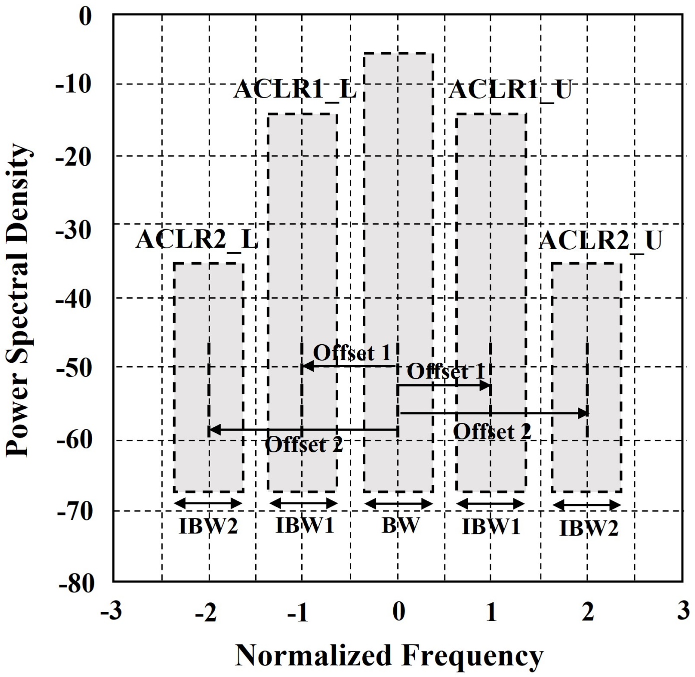


Figure 2.4. Graphical illustration of adjacent channels.

2.7 Error Vector Magnitude (EVM)

EVM is another metric used to measure the nonlinear in-band distortions caused by the PA, calculated based on the deviation in the constellation domain [1]. Figure 2.5 shows the reference constellation point obtained with no distortions and the actual constellation point. The phase error is defined as the angle between the actual signal vector and the reference signal vector, while the magnitude error is the difference in magnitude between the two. In general, transmitters could introduce either phase error, magnitude error, or a combination of both. Each communication standard specifies threshold values for EVM to maintain quality, and EVM is typically expressed as a percentage. EVM can be written as a root mean square (RMS) value as follows:

$$EVM(\%) = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}|S_{ref,i} - S_{act,i}|^2}{\frac{1}{N}\sum_{i=1}^{N}|S_{ref,i}|^2}} \tag{8}$$

where N is the total number of constellation points and $S_{ref,i}$, $S_{act,i}$ are the reference and actual constellation points for the i^{th} symbol, respectively.

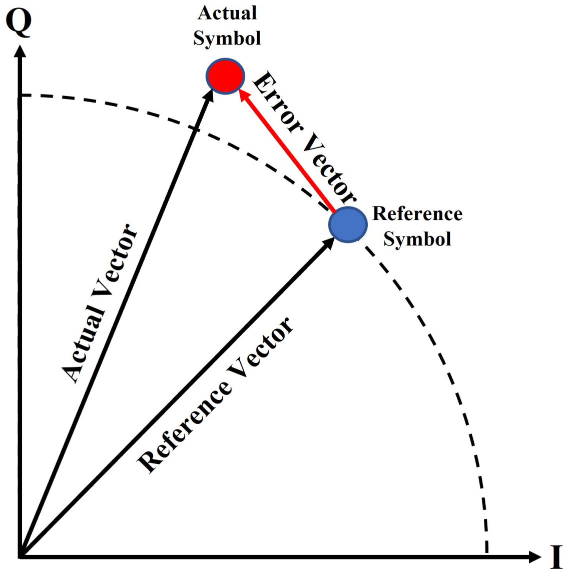


Figure 2.5. Graphical illustration of error vector.

2.8 Memory Effects of PA

Memory effects refer to systems in which the current output is not only influenced by the current input but also by one or more previous inputs [1]. Memory depth is defined as the number of input samples affecting the output. Energy-storing memory systems inherently exhibit this effect. PA systems can have multiple energy-storing circuits or other elements contributing to the system's memory effect. These elements may include capacitive/inductive elements, matching network elements, and transistor junctions. Memory effects can be classified into two main categories based on their correlation with the nonlinearity of the PA: linear memory effects and nonlinear memory effects. Linear memory effects are uncorrelated with the PA nonlinearity and can be mathematically interpreted as a linear combination of time-shifted input signals as

$$y(t) = \sum_i h_i x(t - \tau_i). \tag{9}$$

Nonlinear memory effects include the nonlinear behavior of the PA. They can be mathematically represented by including a nonlinear term that accounts for the nonlinear memory effects of PA and the linear memory effects as

$$y(t) = \sum_i h_i x(t - \tau_i) + f[x(t - \tau_1), ..., x(t - \tau_N)] \tag{10}$$

where f is the nonlinear function. Memory effects can be classified based on their origins into two categories: electrothermal memory effects and electrical memory effects. Electrothermal memory effects arise from temperature variations in active devices and have long-term memory impacts. Electrical memory effects are caused by capacitive elements and impedance-matching networks present in the PA and have short-term impacts. While memory effects may not cause significant degradation of linearity, they can still significantly impact the performance of the linearizer. Thus, proper analysis of memory effects is required for behavioral modeling and linearization of PA [1].

2.9 PA Linearization Techniques

Linearization techniques aim to mitigate the nonlinear effects of PAs by modifying either the input or output waveform and can be categorized into two types: circuit-level techniques and system-level techniques [9]. Circuit-level techniques involve modifications made at the device level and are typically more applicable to the user equipment, while system-level techniques utilize both digital and analog methods and are better suited for base station transmitters. Harmonic termination, harmonic injection, transconductance gain compensation, and thermal compensation methods are some of the main circuit-level techniques. On the other hand, power back-off, feedback, LINC, feed-forward, and predistortion are among the primary system-level techniques. Some techniques are briefly discussed below.

2.9.1 Power Back-off

The PA exhibits linearity for input signals with small power, and this phenomenon is utilized by the back-off technique [9]. The operation point of the PA is backed off from the saturation point to enable the output signal to swing fully while maintaining linearity. The back-off level should be higher than the PAPR of the waveform to preserve the linearity. $PAPR_{dB}$ of the waveform is defined as

$$PAPR_{dB} = 10 \times log_{10} \frac{P_{max,W}}{P_{avg,W}} \tag{11}$$

where P_{max} and P_{avg} are the maximum and average power levels of the waveform in Watts. Modern communication systems have PAPR ranging from $10-13$ dB [1]. Typically, back-off is achieved by reducing the input power to the PA. The back-off technique enables the PA to avoid operating in the nonlinear region. However, increasing the back-off level leads to a decrease in efficiency. In digital modulation, the back-off level is usually around $6-8$ dB below the 1 dB compression point. PA output power back-off ($OPBO_{dB}$) can be expressed as

$$OPBO_{dB} = 10 log_{10} \frac{P_{out}}{P_{out,sat}} \tag{12}$$

where P_{out} and $P_{out,sat}$ are the operating output power and saturation output power of the PA, respectively. However, the back-off technique does not attempt to overcome the trade-off between efficiency and linearity [1].

2.9.2 Feedback Linearization

The general idea of feedback linearization is to use the output of a nonlinear system and apply a transformation to it, such that the resulting signal behaves more linearly [9].

This can typically involve adding a scaled and phase-shifted version of the output signal to itself, to cancel out nonlinear effects and produce a more linear response. This concept can be incorporated in RF, IF, and baseband frequencies. The main challenge associated with feedback linearization is the stability problems due to the delay between input and output. Hence, this technique is less common in wideband communication systems. Multiple techniques are available for linearization using feedback, including RF feedback, polar loop feedback, cartesian loop feedback, envelope elimination and restoration, and LINC linearization.

RF feedback is achieved by subtracting the output RF from the input RF directly to linearize a specific section of the transmitter. Polar loop feedback is a technique where the AM/AM and AM/PM transfer functions of the PA are corrected using separate loops, which are typically implemented in the IF stage. However, it is also possible to do so in the RF stage. Cartesian loop feedback is implemented in RF, where the PA output is demodulated and generates two I/Q samples to feed into the modulator for linearization. Another linearization method is envelope elimination and restoration [9].

2.9.3 Envelope Elimination and Restoration

The main concept behind the envelope elimination and restoration linearization technique is to separate the amplitude and phase information of the signal. First, the amplitude information is removed from the signal with the help of a limiter, which allows only the phase information to pass through the PA. Then, the envelope detector extracts the amplitude information from the signal simultaneously and which is then supplied to the power supply. The bias of the PA is then adjusted to restore the envelope to the carrier.

However, the delay difference between the two paths is a major concern. Therefore, it is crucial to synchronize the amplitude and phase signals before imposing them on the RF carrier. This is typically achieved by delaying the phase information to compensate for the delay between the two paths [9].

2.9.4 Feedforward Linearization

The primary difference between feedback and feedforward methods is the location at which the error signal is compared [10]. In feedforward, the error signal is compared at the output of the system, while in feedback, the error signal is compared at the input. The undistorted input is delayed and compared with the attenuated output of the PA. This allows for measuring the introduced amplitude and phase distortion by the PA. Then the compared signal is passed through an error amplifier and compared with a delayed version of distorted PA output. By doing so, the distortion can be removed. To ensure the effective removal of distortion, it's essential to design the delay lines accurately. This involves matching the group delays of the PA and the error amplifier.

The feedforward architecture requires another PA, called the error amplifier. Unlike the main PA, the error amplifier doesn't compensate for tracking or gain errors. Consequently, it's important that this PA be linear. In addition, the error amplifier must be sufficiently robust to handle coupling at the output combiner of the overall PA.

These criteria typically result in the error amplifier being sized similarly to the main PA, which raises cost and efficiency concerns that must be addressed [10].

2.9.5 Predistortion

Predistortion is a widely used technique in modern communication systems [9]. The main idea behind this technique is to introduce distortion to the input of the PA so that the output of the PA becomes linearized. The main challenge of the predistortion technique is determining the appropriate distortion for the input signal to achieve the desired linearization of the PA output. The predistorter is placed immediately before the PA.

The predistorter is designed to have the inverse nonlinear characteristics of the PA, with the aim of achieving an overall linear output [10]. Figure 2.6 provides a graphical representation of the predistortion principle. It shows the AM/AM characteristics of the PA and predistorter, exhibiting an inverse relationship between each other. In modern broadband systems, higher bandwidths are utilized, leading to a significant memory effect in PAs. Therefore, the use of adaptive predistortion is necessary.

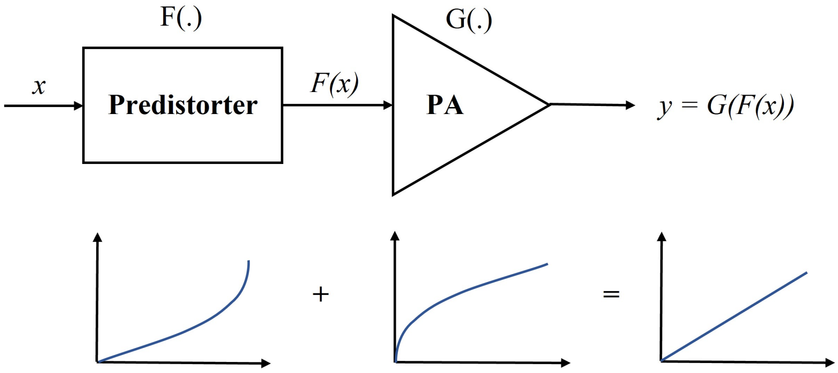


Figure 2.6. Graphical illustration of predistortion principle.

The predistortion concept can be illustrated mathematically with a few equations [10]. If the PA has a nonlinear gain function of $G(x)$, then output $y(t)$ can be expressed as

$$y(t) = G(x(t)) = K'.x(t) + nonlinear\_terms \tag{13}$$

where the input signal to the PA is represented as $x(t)$, and the linear gain of the PA is denoted by K'. The objective is to eliminate the nonlinear terms of the PA, which can be achieved through the use of a predistorter having a nonlinear gain function represented by $F(x)$ as

$$y(t) = G(F(x(t))) = K'.x(t). \tag{14}$$

For the overall system to have a linear relationship, the nonlinear gain function represented by F must be the inverse of the nonlinear gain function represented by G.

The predistortion function can be implemented in either the analog or digital domains as either fixed or adaptive functions [10]. Analog predistortion is a technique that involves using circuits composed of diodes and transistors. This technique is used in various applications, including space-borne communications and cellphone headsets. The nonlinearity of junction diodes is commonly utilized in analog predistortion techniques to modify the input signal, which is especially popular in RF domain analog predistortion. On the other hand, DPD systems are implemented using digital components, which makes them more flexible and versatile than analog ones. This is widely used in modern communication systems due to their simplicity and robustness in parameter changes. The Cartesian feedback architecture is the foundation for DPD. In technique, the output signal from the PA is first down-converted and demodulated into I/Q streams, which are then digitized using a high-resolution Analog-to-Digital Converter (ADC). The digitized output signal is then compared with the input I/Q data, and the predistorter is adjusted accordingly to minimize any distortions [10].

To assess the performance of DPD applications, various metrics are employed, including ACPR and EVM. These metrics can be computed for the linearized system output. While AM/AM and AM/PM characteristics can also be used to evaluate performance, they tend to yield qualitative results rather than numerical values. They offer a visual representation of the degree of linearization achieved, allowing for comparing different curves produced by various predistortion techniques to assess their performance [1].

3 STATE-OF-THE-ART BEHAVIORAL MODELING AND PREDISTORTION TECHNIQUES OF PA

This chapter provides a thorough overview of the current state-of-the-art in behavioral modeling and predistortion techniques, with a primary focus on models that utilize neural networks. Additionally, the chapter highlights the progression of predistortion techniques from memory-less models to more sophisticated models that incorporate adaptive memory-based approaches.

3.1 Behavioral Model

The behavioral model of an RF PA is a nonlinear dynamic model designed to emulate the PA behavior based on the knowledge of input and output signals [1]. It strives to capture the input-output relationship of the RF PA, and it can be expressed mathematically or through other means. The process of deriving a behavioral model is similar to predistortion, and most model structures can be used for both applications. Nevertheless, the performance evaluation approach differs between behavioral modeling and predistortion. As shown in Figure 3.1, in behavioral modeling, performance evaluation is typically based on comparing the output of the model to the output of the actual PA when the same input is applied to both. The signals $y_{actual}(t)$ and $y_{model}(t)$ are compared and assessed the performance based on their similarity either in the time domain or frequency domain. Then the parameters are adjusted through a feedback path to the model [1].

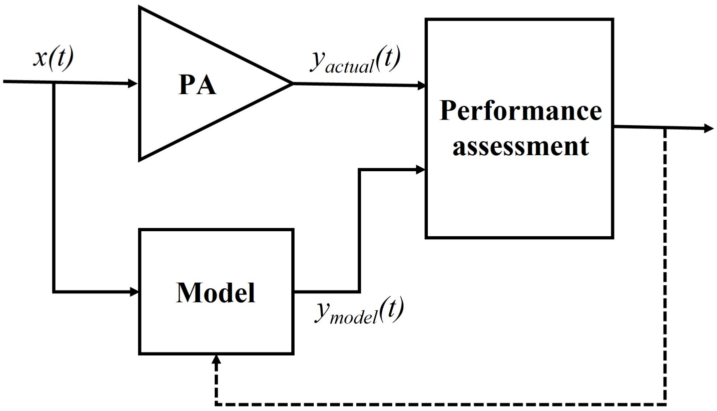


Figure 3.1. Behavioral model performance evaluation.

RF PA systems typically have highly complex internal circuitry, making it difficult to interpret each phenomenon and develop a model mathematically [10]. However, one of the primary advantages of behavioral modeling is that it requires very little or no knowledge of the internal circuitry to construct the model. In addition, behavioral modeling is essentially a mapping between inputs and outputs, which means that it can help protect the intellectual property (IP) of the device. The main considerations of behavioral modeling are the simulation speed and the level of accuracy required for the given application. Some internal features are willingly sacrificed to improve the simulation time, such as temperature dependency, frequency response, average power level, electromagnetic interactions, and load sensitivity [10].

Over the last few decades, various behavioral modeling methods have been introduced. Behavioral models can be classified into three types based on their memory effect: memoryless nonlinear, quasi-memoryless nonlinear, and nonlinear with memory [9]. Memoryless nonlinear models are characterized by their AM/AM characteristics, while quasi-memoryless nonlinear models are characterized by their AM/AM and AM/PM characteristics. When the memory effect is considered, the model's behavior depends on the envelope of the input signal and its frequency. This approach has become increasingly common in modern communication systems due to high bandwidths. Subsequently, models considering memory effects, such as the Volterra series, memory polynomial, and generalized memory polynomial models, have been investigated. However, these models have faced challenges in improving the modeling performance due to the high correlation between polynomial bases. As a result, in recent years, neural network (NN) based predistortion techniques have gained popularity.

3.2 Conventional Memoryless and Memory Nonlinear Models

The evolution of conventional nonlinear models up to NN-based nonlinear modeling is presented in this section. The journey starts with memoryless models such as the Saleh model, Ghorbani model, and polynomial models without memory. These models lack the ability to capture dynamic behaviors and are limited in their applications. Then the memory models were developed, including Volterra based models, memory polynomials, and Wiener-Hammerstein models. These models have the ability to capture dynamic behaviors and memory effects, making them suitable for a wider range of applications.

3.2.1 Saleh Model

The Saleh model was a commonly used PA modeling technique in earlier days, which modeled the AM/AM and AM/PM characteristics of the PA [9]. The Saleh model was specifically designed and optimized for traveling wave tube amplifiers (TWTAs), and, therefore may not be well-suited for modeling solid-state PAs (SSPAs) [2]. The Saleh model can be expressed in either polar or Cartesian form [1]. The polar form is directly related to the AM/AM and AM/PM characteristics which can be expressed as

$$G_A(A) = \frac{\alpha_a}{1 + \beta_a A^2} \tag{15}$$

$$\phi_G(A) = \frac{\alpha_\phi A^2}{1 + \beta_\phi A^2} \tag{16}$$

where $G_A(A)$ and $\phi_G(A)$ denote the AM/AM and AM/PM characteristics of the PA, respectively, and $\alpha_a, \beta_a, \alpha_\phi$ and β_ϕ denote the amplitude and phase coefficients of the model. On the other hand, the cartesian form is related to the I/Q nonlinearities and can be expressed as

$$G_I(A) = \frac{\alpha_I}{1 + \beta_I A^2} \tag{17}$$

$$G_Q(A) = \frac{\alpha_Q A^2}{(1+\beta_Q A^2)^2} \tag{18}$$

where $G_I(A)$ and $G_Q(A)$ denote the in-phase and quadrature nonlinearities of the PA, respectively, and $\alpha_I, \beta_I, \alpha_Q$ and β_Q denote the I/Q coefficients of the model.

3.2.2 Ghorbani Model

The Ghorbani model is an analytically based model designed for SSPAs [1]. It describes the AM/AM distortion $G_A(A)$ and AM/PM distortion $\phi_G(A)$ as

$$G_A(A) = \frac{x_1 A^{x_2-1}}{(1+x_3 A^{x_2})} + x_4 \tag{19}$$

$$\phi_G(A) = \frac{y_1 A^{y_2}}{(1+y_3 A^{y_2})} + y_4 A \tag{20}$$

where A is the magnitude of the input signal. The model parameters x_1, x_2, x_3, x_4 are for the AM/AM distortion, and y_1, y_2, y_3, y_4 are for the AM/PM distortion. The Ghorbani model is well suited for modeling field effect transistor (FET) PAs and low amplitude nonlinearities [2].

3.2.3 Rapp Model

The Rapp model is a memoryless nonlinearity model, especially in the design and analysis of SSPAs. The input-output characteristics of the Rapp model can be expressed as

$$y(t) = \frac{G_{ss}}{[1+\left|\frac{x(t)}{x_{sat}}\right|^{2\sigma}]^{\frac{1}{2\sigma}}} \tag{21}$$

where the signals input to the model are denoted by $x(t)$, while the output signals are represented by $y(t)$. The input saturation value is referred to as x_{sat}, and G_{ss} is the gain for small signals. Additionally, a positive scaling factor is represented by σ.

3.2.4 Memoryless Polynomial Model

AM/AM and AM/PM characterizing complex nonlinearities are commonly used nonlinear PA models [1]. The typical way is to measure them in a static method. However, dynamic AM/AM and AM/PM measurements can obtain more accurate results. The complex input envelope, $x(t)$, and complex output envelope, $y(t)$ can be expressed as

$$y(t) = x(t)G(A) \tag{22}$$

where $A = |x(t)|$ and $G(A)$ is the complex gain of PA, which can be expressed in a complex polynomial power series of finite order N as

$$G(A) = \sum_{k=1}^{N} a_k |x(t)|^{k-1} \tag{23}$$

where a_k are the complex coefficients of the model. Then the model output can be expressed as

$$y(t) = \sum_{k=1}^{N} a_k |x(t)|^{k-1} x(t). \tag{24}$$

3.2.5 Volterra Series Model

As explained in previous chapters, PAs can have memory effects due to multiple factors associated with operating signals and PA characteristics. The memory effects in PAs can be classified into two main categories: electrical memory effects, which are dependent on the signal bandwidth, and thermal memory effects, which are dependent on the power dissipation of the PA [1]. Narrowband signals exhibit minimal electrical memory effects because the PA response remains relatively constant across the signal bandwidth. Hence, electrical memory effects can generally be disregarded for signals with a bandwidth lower than 10 MHz. Similarly, the thermal memory effects can be ignored if the junction temperature variation is a few degrees. If thermal and electrical memory effects can be neglected, the PA can be represented using a memoryless nonlinear static model. However, with the higher bandwidths of modern communication systems, memory impact can not be ignored in PA modeling. Volterra series can accurately model the nonlinearity and different types of memory effects, and its input-output relation can be expressed as

$$y(n) = \sum_{p=1}^{P} \sum_{i_1=0}^{M} ... \sum_{i_p=0}^{M} h_p(i_1, ..., i_p) \prod_{j=1}^{p} x(n - i_j) \tag{25}$$

where $x(n)$ and $y(n)$ represent the input and output of the model, respectively. The kernels of the model are denoted by $h_p(i_1, \ldots, i_p)$, with a memory depth of M and a nonlinearity order of P. The Volterra model is considered a complete model because it captures all possible forms of memory effects and nonlinearities. However, a major drawback of the model is that it requires a large number of coefficients, which increases exponentially with both the nonlinearity order and memory depth. Hence, the Volterra series is used to model lower memory depths and nonlinearity orders. To address the issue of complexity, several modifications have been proposed to the Volterra model. These modifications aim to reduce the number of coefficients required for the model while preserving its ability to capture memory effects and nonlinearities.

3.2.6 Wiener, Hammerstein and Wiener-Hammerstein Models

These models are introduced to simplify the complexity of the Volterra series, by separating the memory component from the memoryless nonlinearity [1]. The Wiener model consists of a filter to capture the memory effect, followed by a memoryless nonlinearity. In contrast, the Hammerstein model has a memoryless nonlinearity followed by a filter. The Wiener-Hammerstein model combines these two approaches, with a memoryless nonlinearity sandwiched between two filters. Due to this structure, these models reduce the number of required coefficients significantly. The structure of these models allows for a significant reduction in the number of required coefficients. They do not require coefficients for many cross-product terms, instead relying only on the filter and memoryless model parameters. However, there are still some limitations that need to be addressed. One of the main drawbacks is that separating memory and nonlinearity does not accurately reflect real-world situations. Additionally, these approaches do not take into account the filtering impact on different power levels.

3.2.7 Memory Polynomial

The memory polynomial model is derived from the Volterra series model by eliminating the diagonal terms [1]. Thus, the memory polynomial output $y(n)$ can be expressed in terms of baseband input $x(n)$ as

$$y(n) = \sum_{m=0}^{M} \sum_{k=1}^{K} a_{mk} x(n-m)|x(n-m)|^{k-1} \tag{26}$$

where K represents the nonlinearity order and M represents the memory depth, and a_{mk} denotes the coefficients of the model. The same equation can be interpreted with matrix notation as

$$y(n) = X(n) \cdot A \tag{27}$$

where $X(n)$ and A are denoted as

$$X(n) = \begin{bmatrix} x(n) \\ \vdots \\ x(n)|x(n)|^{K-1} \\ x(n-1) \\ \vdots \\ x(n-1)|x(n-1)|^{K-1} \\ \vdots \\ x(n-M)|x(n-M)|^{K-1} \end{bmatrix}^T \tag{28}$$

$$A = \begin{bmatrix} a_{01} & \dots & a_{0K} & a_{11} & \dots & a_{1K} & \dots & a_{MK} \end{bmatrix}^T. \tag{29}$$

3.2.8 Generalized Memory Polynomial Model

The generalized memory polynomial (GMP) model is derived by including additional basis functions to the memory polynomial model [1]. It introduces cross terms resulting in a complex signal including leading and lagging terms and can be expressed as

$$\begin{aligned} y(n) = \sum_{m=0}^{M_a} \sum_{p=1}^{P_a} a_{mp} x(n-m)|x(n-m)|^{p-1} + \sum_{m=0}^{M_b} \sum_{p=2}^{P_b} \sum_{p'=1}^{P'} b_{mpp'} x(n-m)|x(n-m-p')|^{p-1} \\ + \sum_{m=0}^{M_c} \sum_{p=2}^{P_c} \sum_{q=1}^{Q} c_{mpq} x(n-m)|x(n-m+q)|^{p-1} \end{aligned} \tag{30}$$

where $x(n)$ and $y(n)$ denote the input and output of the model. The GMP model consists of three polynomial functions. The first function time aligns the input samples with a memory depth of M_a and a nonlinearity order of P_a. The second function uses the input samples and lagging values of its envelope with a memory depth of M_b and a nonlinearity order of P_b. The third function uses the input samples and leading values of its envelope with a memory depth of M_c and a nonlinearity order of P_c.

3.3 Neural Network-based Models

Current literature investigates numerous NN-based approaches, including shallow NNs [11], [12], [13], and deep NNs [14] with multiple hidden layers. The shallow NNs comprise a simple network structure and a simple training process. The authors in [11] introduced a shallow NN called the real-valued time-delay NN (RVTDNN) model. Since it has one or two hidden layers, additional computations are required in the hidden layers to learn complex nonlinear and memory effects of the PA. To overcome that, augmented radial basis function NN (ARBFNN) [13] and augmented real-valued time delay NN (ARVTDNN) [12] architectures were introduced, including the envelope-dependent terms in the input along with the I/Q components of current and past signals. However, as the bandwidth increases, the number of memory taps also increases, resulting in a high input dimension. The suitability of deep NN (DNN) is studied in [14] experimenting over three hidden layers. This study shows that it can approximate intricate nonlinear relationships with relatively low complexity.

Nevertheless, the excessive signal processing resources for high bandwidth situations is still a concern. To address the complexity issue in wideband PAs, [5] introduced a real-valued time delay convolutional NN (RVTDCNN) model with image input data. This approach was refined in [15] by replacing the two-dimensional (2D) convolution kernel with a couple of consecutive one-dimensional (1D) convolution kernels to enhance the computational complexity. The authors in [16] included an analysis of the average power consumption and ambient temperature variation impact on the performance of the same CNN architecture. Another hardware-friendly modular 1D-CNN architecture was proposed in [17] for real-time DPD. Due to the time series nature of the input signal, RNN and LSTM based NNs are introduced to capture the memory effects of PA [18–20]. The vanilla LSTM structure is used in [18] consisting of one LSTM layer and a couple of fully connected layers. Then it is further improved in [20] by including a CNN layer followed by the LSTM layer to build an LSTM-CNN architecture. Sequence-to-sequence modeling nature of LSTM is utilized in [19] using a bidirectional LSTM (BiLSTM) model. Each NN-based model is briefly described below.

3.3.1 Real-Valued Time Delay Neural Network (RVTDNN)

The RVTDNN model takes complex I/Q input data as a real-valued vector, including time-delayed versions [11]. The model is based on a feed-forward NN (FFNN) and tapped delay lines (TDLs), as Figure 3.2 illustrates. The model uses I_{in}, Q_{in} of current and past baseband input samples to predict the current baseband output I_{out}, Q_{out}. TDLs are utilized to capture the previous input samples, which reflect the short-term memory effect of the PA. p past baseband input I_{in} and q past baseband input Q_{in} samples are used in the estimation of the output value. The input of the model X_n can be expressed as

$$\begin{aligned} X_n = [&I_{in}(n), I_{in}(n-1), ..., I_{in}(n-p), \\ &Q_{in}(n), Q_{in}(n-1), ..., Q_{in}(n-q)]. \end{aligned} \tag{31}$$

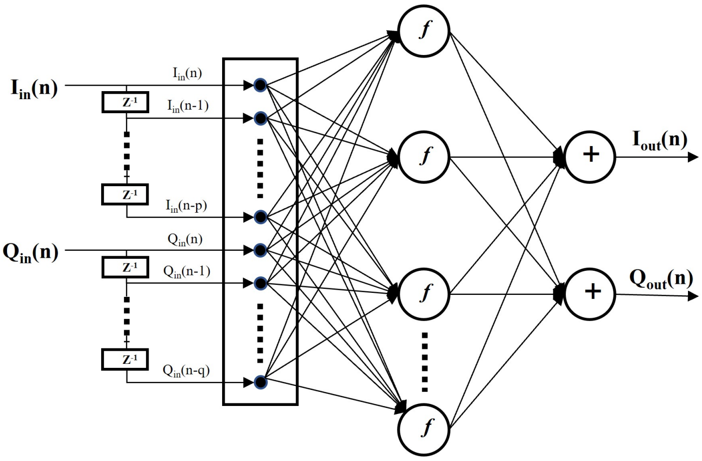


Figure 3.2. RVTDNN model architecture.

3.3.2 Augmented Real-Valued Time Delay Neural Network (ARVTDNN)

The ARVTDNN model incorporates both the I and Q components, including the amplitudes of the envelope terms and their nonlinear versions [12]. Additionally, the current and previous input samples are used to reflect the memory effect as shown in Figure 3.3. The utilization of envelope-dependent nonlinear terms has been shown to enhance both the convergence speed and overall performance of the model. The model input X_n can be expressed as

$$\begin{aligned} X_n = [&I_{in}(n), I_{in}(n-1), ..., I_{in}(n-M), \\ &Q_{in}(n), Q_{in}(n-1), ..., Q_{in}(n-M), \\ &|x_{in}(n)|, |x_{in}(n-1)|, ..., |x_{in}(n-M)|, \\ &|x_{in}(n)|^2, |x_{in}(n-1)|^2, ..., |x_{in}(n-M)|^2, \\ &|x_{in}(n)|^3, |x_{in}(n-1)|^3, ..., |x_{in}(n-M)|^3] \end{aligned} \tag{32}$$

where $x_{in}(n)$ is the current input baseband sample such that $x_{in}(n) = I_{in}(n) + jQ_{in}(n)$. M is the memory depth.

3.3.3 Augmented Radial Basis Function Neural Network (ARBFNN)

In previous models, I and Q data are separated and fed into the model as real values [13]. However, in the ARBFNN model, the complex baseband input is fed in its complex form,

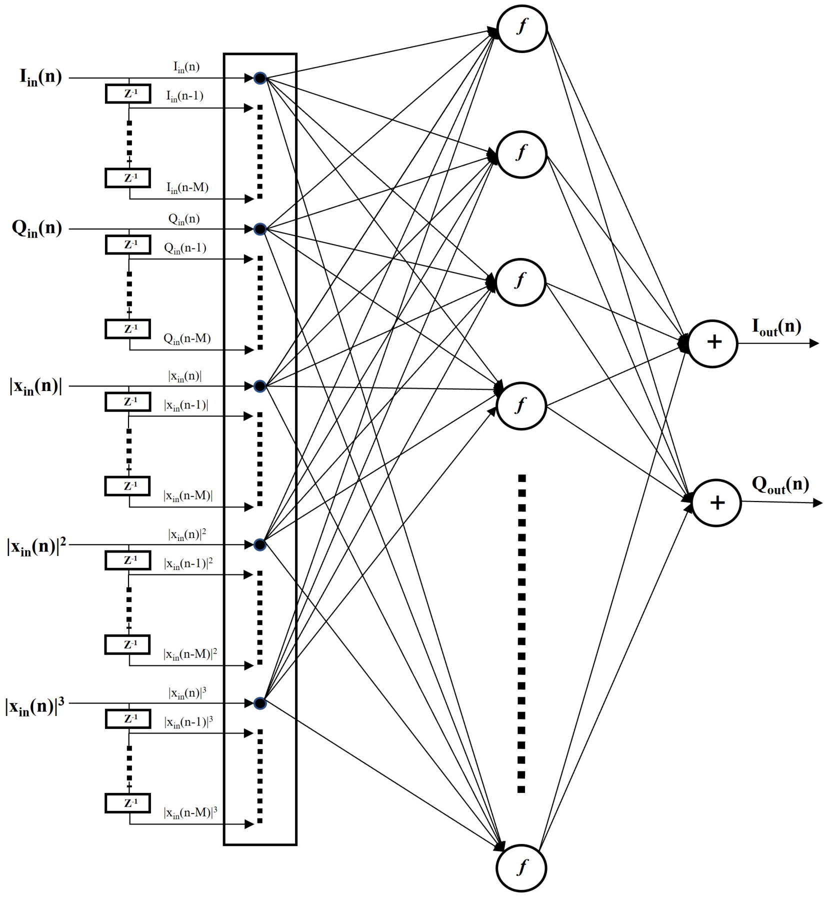


Figure 3.3. ARVTDNN model architecture.

incorporating both current and previous samples, as well as envelope-dependent terms. The model consists of a single hidden layer, and both the input samples and resulting output are in complex form. The input of the model X_n can be expressed as

$$\begin{aligned} X_n = [&x_{in}(n), x_{in}(n-1), ..., x_{in}(n-M), \\ &|x_{in}(n)|, |x_{in}(n-1)|, ..., |x_{in}(n-M)|, \\ &|x_{in}(n)|^2, |x_{in}(n-1)|^2, ..., |x_{in}(n-M)|^2, \\ &|x_{in}(n)|^3, |x_{in}(n-1)|^3, ..., |x_{in}(n-M)|^3]. \end{aligned} \tag{33}$$

3.3.4 Deep Neural Network-based Digital Predistorter (DNN-DPD)

The models previously discussed featured only one or two hidden layers [14]. However, in this DNN-based model, additional hidden layers are incorporated to evaluate the linearization performance. The DNN model shares similarities with the RVTDNN model, except for having more hidden layers as shown in Figure 3.4. The input of the DNN model is the same as that of the RVTDNN model.

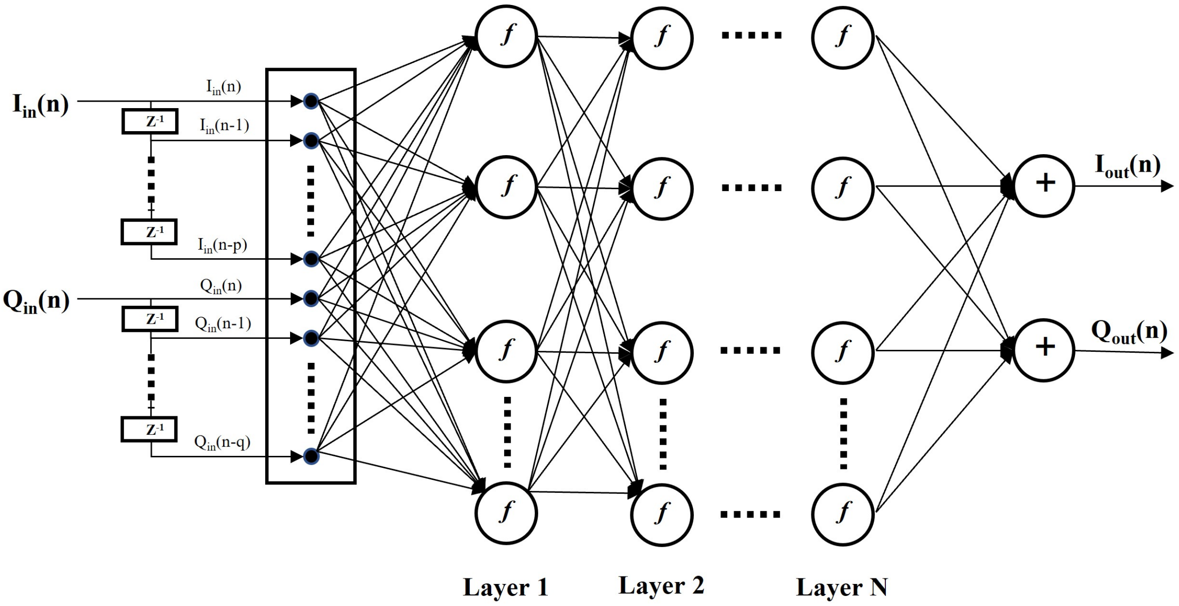


Figure 3.4. DNN model architecture.

3.3.5 Real-Valued Time Delay Convolutional Neural Network (RVTDCNN)

The RVTDCNN model employs a convolutional NN to efficiently extract crucial features [5]. It comprises an input layer, a convolutional layer, a fully connected layer, and an output layer. The diagram of the RVTDCNN model is shown in the accompanying Figure 3.5. One of the primary benefits of this model is its ability to maintain comparable performance while significantly reducing complexity. Moreover, it is an excellent solution for high-bandwidth PAs with more extended memory effects. The input of the model is a 2D matrix with current and previous input I and Q samples along with envelope-dependent nonlinear terms. It can be expressed as

$$X_n = \begin{bmatrix} I_{in}(n) & I_{in}(n-1) & \dots & I_{in}(n-M) \\ Q_{in}(n) & Q_{in}(n-1) & \dots & Q_{in}(n-M) \\ |x_{in}(n)| & |x_{in}(n-1)| & \dots & |x_{in}(n-M)| \\ |x_{in}(n)|^2 & |x_{in}(n-1)|^2 & \dots & |x_{in}(n-M)|^2 \\ |x_{in}(n)|^3 & |x_{in}(n-1)|^3 & \dots & |x_{in}(n-M)|^3 \end{bmatrix}. \qquad (34)$$

3.3.6 Power-Temperature Inclusive Digital Predistortion (PTI-DPD)

The behavior of the PA is sensitive to the average power variation and the ambient temperature [16]. Hence, the PTI-DPD is designed to incorporate those parameters to increase accuracy and robustness. The PTI-DPD model shares the same model structure as the RVTDCNN, with the exception that its input matrix includes an additional row containing temperature and power data. The model input can be expressed as

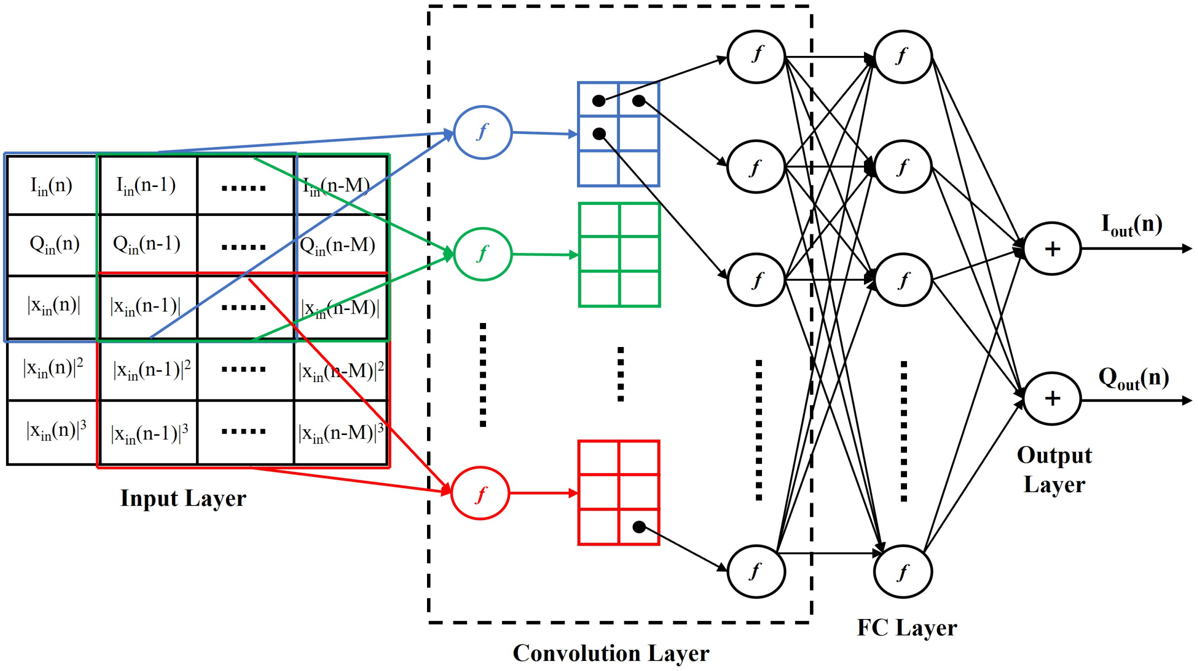


Figure 3.5. RVTDCNN model architecture.

$$X_n = \begin{bmatrix} I_{in}(n) & I_{in}(n-1) & I_{in}(n-2) & \dots & I_{in}(n-M) \\ Q_{in}(n) & Q_{in}(n-1) & Q_{in}(n-2) & \dots & Q_{in}(n-M) \\ \overline{P(n)} & T(n) & \overline{P(n)} & \dots & \overline{P(n)} \\ |x_{in}(n)| & |x_{in}(n-1)| & |x_{in}(n-2)| & \dots & |x_{in}(n-M)| \\ |x_{in}(n)|^2 & |x_{in}(n-1)|^2 & |x_{in}(n-2)|^2 & \dots & |x_{in}(n-M)|^2 \\ |x_{in}(n)|^3 & |x_{in}(n-1)|^3 & |x_{in}(n-2)|^3 & \dots & |x_{in}(n-M)|^3 \end{bmatrix} \tag{35}$$

where $\overline{P}$ and T denote the average power and temperature.

3.3.7 Long Short-Term Memory (LSTM) based Digital Predistortion (LSTM-DPD)

In general, LSTM models can capture time series dependencies [18]. Therefore, the model uses the LSTM layer to capture and leverage the memory effects of the PA. This model comprises an input layer, an LSTM layer, a couple of fully connected layers, and an output layer. The model's input is identical to that of the RVTDNN input, whereby it takes the current and previous I and Q components of the baseband input samples.

4 TRANSFORMER NEURAL NETWORK-BASED BEHAVIORAL MODELING AND PREDISTORTION

In this chapter, the use of transformer-based techniques for behavioral modeling and predistortion is thoroughly examined. The chapter delves into the underlying requirements of these techniques and highlights how they are successful in linearizing even the most complex, nonlinear PAs that would appear in future communication systems.

4.1 Limitations of State-of-the-art Models and Benefits of Transformer-Based Models

The state-of-the-art PA behavioral and linearization models have limitations in addressing the intricate and long-term nonlinear relationships that future high-bandwidth PAs may exhibit. Meeting the high data rate requirements of modern communication systems has become a key concern, and it is expected that future systems will utilize several hundreds of bandwidth systems. The high bandwidths associated with modern communication systems can significantly increase the memory effect of PAs, leading to more distortions that existing models may not be able to resolve.

Moreover, the training and inference of RNN and LSTM models are sequential computational processes that arise challenges in hardware acceleration with graphical processing units (GPUs) and field programmable gate arrays (FPGAs) [21]. In other words, the sequential nature of processing makes it impossible to fully utilize multiple computational resources, even if they are available. These limitations made the foundation for investigating the suitability of transformer-based deep learning solutions for behavioral modeling and linearization of the PA.

The transformer architecture was introduced in 2017 in the paper "Attention is All You Need," enabling high computation parallelism in the training and inference phases [6]. The transformer model structure is designed with encoder-decoder-based architecture. The existing LSTM encoder-decoder architecture presents challenges in terms of training due to its susceptibility to the vanishing gradient problem. The transformer processes the entire data sequence in parallel through the self-attention mechanism, identifying long-term sequence dependencies [21]. Due to the parallel processing, available hardware resources can be utilized to the full extent, which was challenging on LSTM-based models.

In addition to its parallel processing capabilities, transformer-based models can handle complex time series dependencies that are challenging for existing sequence models [22]. Motivated by these capabilities, this study proposes behavioral modeling and linearization of the PA based on the transformer architecture, potentially overcoming the limitations of existing models and improving the design of high-bandwidth PAs for future communication systems.

4.2 Transformer Architecture

This section offers an in-depth explanation of the transformer architecture, providing a comprehensive overview of the entire model. It first presents an overview of the model and then delves into each section of both the encoder and decoder components, providing a detailed explanation of their functions and interactions.

4.2.1 Overall Structure

The transformer architecture is based on the self-attention mechanism, allowing the model to process input data in parallel and learn long-term dependencies more effectively than previous models. This mechanism works by computing the attention weights of each input element based on its similarity to all other elements in the sequence. By weighting the importance of each element based on its relevance to the others, the transformer can capture complex patterns and relationships within the input data.

The transformer model is structured as an encoder-decoder architecture, where the encoder maps the input sequence to an abstract continuous representation, and the decoder generates the output sequence based on this representation. The model architecture is illustrated in Figure 4.1. This architecture is highly adaptable and can be applied to a wide range of sequence-to-sequence problems beyond NLP, including image captioning, speech recognition, music generation, and wireless communication.

The encoder of the transformer consists of multiple layers of self-attention and FFNNs. Each layer of self-attention allows the encoder to focus on different parts of the input sequence, while the FFNNs process the output of the self-attention layers to produce the final encoded representation. The decoder of the transformer also consists of multiple layers of self-attention and FFNNs. The decoder uses the encoded representation from the encoder, along with the previously generated output, to generate the next output in the sequence. The transformer architecture represents a significant improvement over previous models in its ability to learn complex patterns and relationships within sequences, and its highly parallelized structure makes it well-suited for use on modern hardware accelerators. Below is a brief explanation of each submodule of the transformer model.

4.2.2 Input/Output Embedding

NN models cannot directly process words as inputs, therefore, these words are first transformed into a vector representation before being fed into the model. This process can be thought of as a look-up table (LUT), where each word is assigned a vector containing continuous values. This transformation should be applied in both the encoder and decoder sides.

4.2.3 Positional Encoding

Unlike RNN or LSTM, transformers do not have a built-in recurrence mechanism. Instead, the entire sequence is processed simultaneously, and positional information is incorporated into the embeddings using the positional encoding technique. This approach

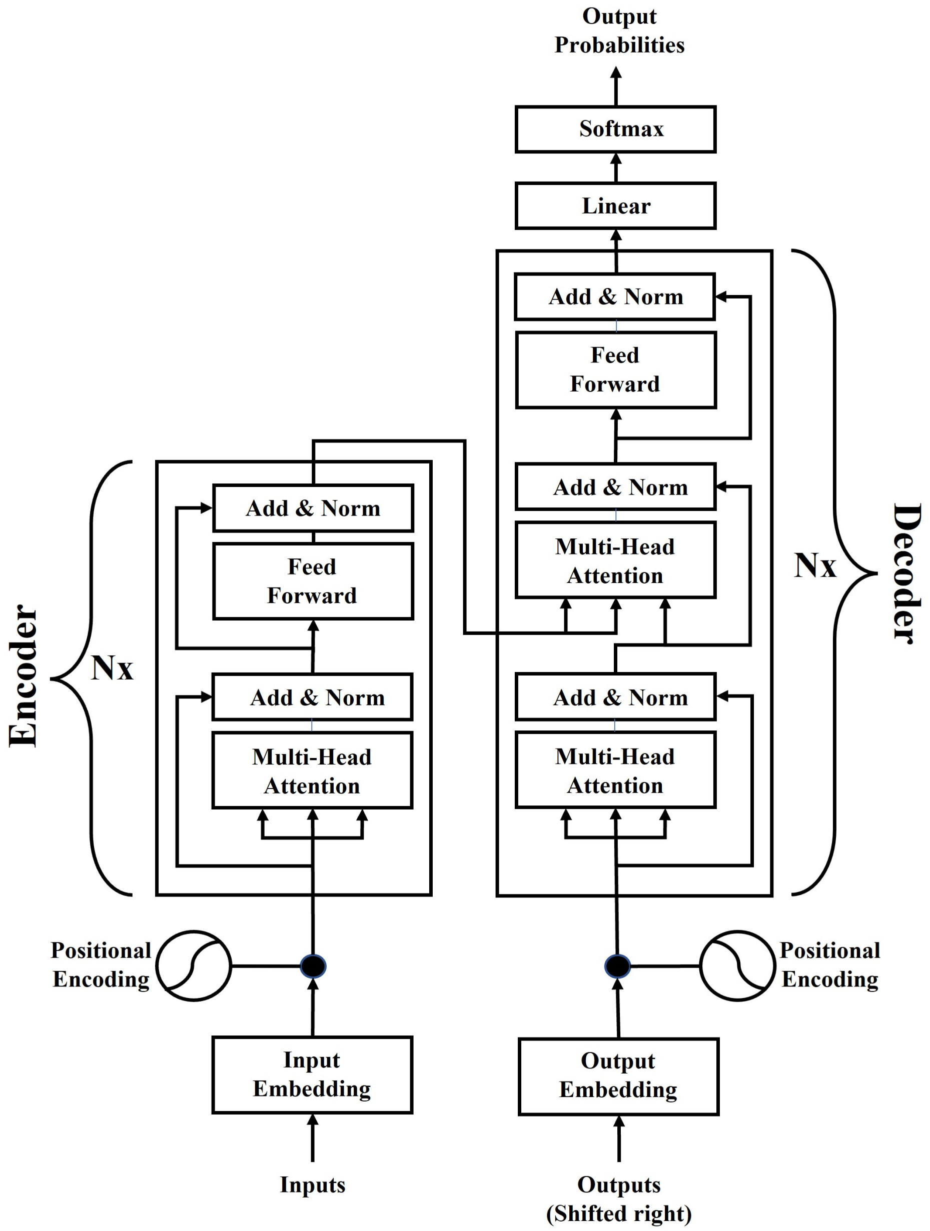


Figure 4.1. Transformer model structure.

involves adding a unique value to represent the position of an element in the sequence. In [6], the authors used a sine and cosine-based scheme, which can be expressed as:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{(2i/d_{model})}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{(2i/d_{model})}). \end{aligned} \tag{36}$$

4.2.4 Attention Mechanism

The attention mechanism maps a query and a set of key-value pairs to an output [6]. The objective is to compute a weighted average of the features of multiple elements in the sequence, with the weights dynamically calculated based on their values. In other words, the attention mechanism dynamically determines which elements in the sequence require more attention than others, and it comprises four parts: the query, key, value, and score function.

The query serves as the feature vector that describes the information of interest within the sequence, whereas the keys correspond to the feature vectors characterizing the input elements and their potential importance. The values refer to the feature vectors that are to be averaged over, while the score function determines the attention weight assigned to each query-key pair, often relying on a straightforward similarity metric.

4.2.5 Scaled Dot-Product Attention

The scaled dot-product attention uses the attention mechanism such that any element in the sequence can attend to any other element in the sequence while maintaining computational efficiency. The matrix form of a set of queries, keys, and values is represented by $Q \in \mathbb{R}^{T \times d_k}$, $K \in \mathbb{R}^{T \times d_k}$, and $V \in \mathbb{R}^{T \times d_v}$, respectively, and is used as inputs. Here, T denotes the sequence length, and d_k and d_v represent the hidden dimensions of queries/keys and values, respectively. The scaled dot product attention can be expressed mathematically as

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{37}$$

The dot product between each possible query-key pair is calculated through the QK^T matrix multiplication. Each row of the resulting matrix QK^T represents the relationships between each element in the sequence and all the other elements in the sequence.

The QK^T matrix multiplication computes the dot product for each query-key pair in a sequence. The rows of QK^T represent the relationship between each element in the sequence and every other element in the sequence. The result of this multiplication is then divided by $\sqrt{d_k}$ to maintain an appropriate variance of attention values after initialization. The output is passed through a $softmax$ function to obtain a weight for each element in the sequence, which is then multiplied with its corresponding value vector to calculate the weighted mean. This process is illustrated in [6] as in Figure 4.2. The masking block is an optional component that is needed when working with sequences of variable lengths.

4.2.6 Multi-Head Attention

As shown in Figure 4.3, the multi-head attention mechanism is an extension of the scaled dot product attention that uses multiple attention heads. This is necessary because complex sequences often require attending to different aspects of the sequence elements. By using multiple heads, the model can learn multiple weighted averages, allowing it to attend to different parts of the sequence simultaneously.

In the multi-head attention mechanism, the initial query, key, and value matrices are divided into h sub-matrices or heads, and each head is used to compute a separate attention vector. Specifically, each sub-query, sub-key, and sub-value matrix is used in a scaled dot product attention block to calculate a corresponding sub-attention vector.

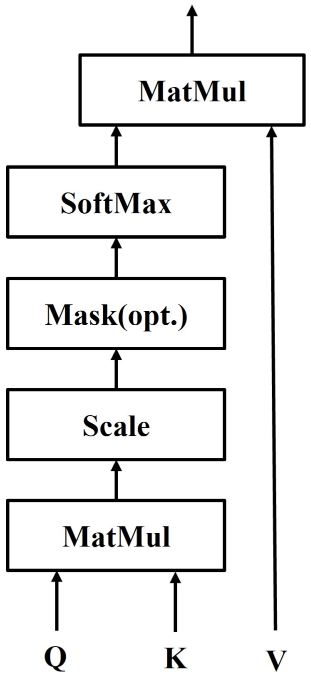


Figure 4.2. Scaled dot-product attention.

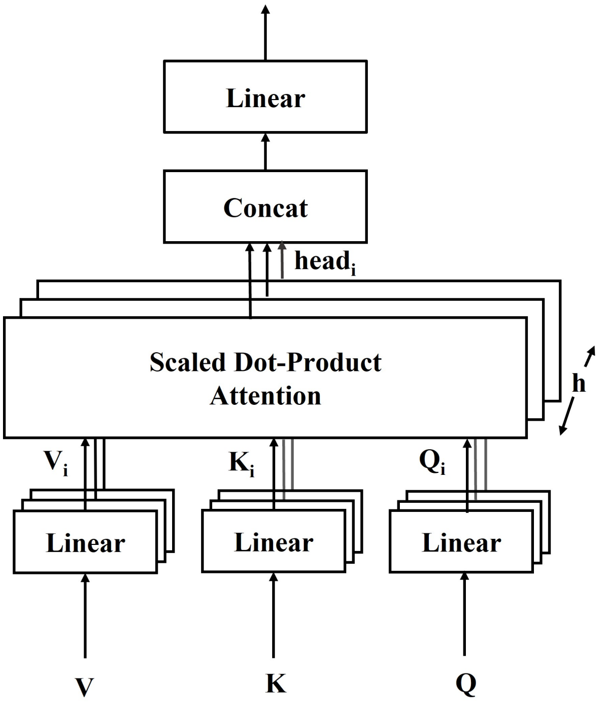


Figure 4.3. Multi-head attention mechanism.

The resulting sub-attention vectors are then concatenated and passed through a linear layer to obtain the final output. The mathematical representation can be denoted as

$$Multihead(Q, K, V) = concat(head_1, head_2, ..., head_h)W^O \tag{38}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{39}$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are projection matrices with learnable parameters.

4.2.7 Feed Forward Network

This refers to an FFNN that is fully connected, comprising two linear layers and a rectified linear unit (ReLU) function in between. The network operates independently on each position, and its mathematical notation can be expressed as

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2x = LayerNorm(x + FFN(x)) \tag{40}$$

where W_1, b_1 and W_2, b_2 are the weight and bias matrices of the first and second linear layers, respectively.

4.2.8 Transformer Encoder

The transformer was originally designed for NLP-related tasks, particularly machine translation, and utilizes an encoder-decoder-based structure. The encoder generates an attention-based representation of the input sequence in the source language, which is then attended to by the decoder to generate the translated output in an autoregressive manner. However, the architecture is not limited to NLP tasks and can be applied to any sequence-to-sequence modeling task with an autoregressive decoding component. In fact, some models have been developed using just the encoder component, such as the BERT family and the vision transformer.

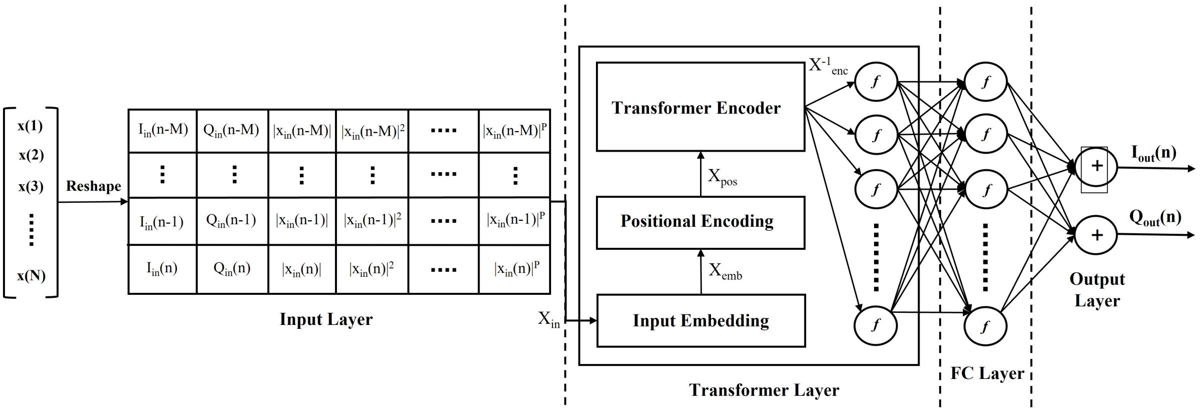
4.2.9 Transformer Decoder

The primary objective of the decoder is to generate text sequences. The decoder has similar sub-layers present in the encoder, such as two multi-head attention layers, a point-wise feedforward layer, and residual connections and layer normalization. However, the first multi-head attention layer in the decoder is referred to as the masked multi-head attention layer, and it has slightly different functionality than the second one. As the decoder generates the output sequence in an autoregressive manner, one word at a time, it is necessary to mask the future tokens when generating the current token to ensure that the model only has access to information available at the current time step. The remaining sub-layers perform the same function as they do in the encoder.

The decoder operates in an autoregressive manner, starting the sequence generation with a special token called $< sos >$, which denotes the "start of the sequence". Using this token and the attention information from the encoder's output, the decoder produces the next possible word. The generated output is then considered an input in the next iteration to produce the subsequent output, and this process repeats until the decoder generates the $< eos >$ token, which denotes the "end of the sequence".

4.3 Proposed Transformer Based Model

The transformers were originally designed for NLP tasks, thus modifications were necessary to handle time series regression data. As a result, a new model called the augmented real-valued time delay transformer neural network (ARVTDTNN) was proposed. This model consists of four layers: an input layer, a transformer layer, a transformer encoder layer, a fully connected layer, and an output layer, as illustrated in Figure 4.4. The functionality of each layer is as follows:



$I_{in}(n-M)$	$Q_{in}(n-M)$	$\lvert x_{in}(n-M)\rvert$	$\lvert x_{in}(n-M)\rvert^2$	····	$\lvert x_{in}(n-M)\rvert^P$
⋮	⋮	⋮	⋮	····	⋮
$I_{in}(n-1)$	$Q_{in}(n-1)$	$\lvert x_{in}(n-1)\rvert$	$\lvert x_{in}(n-1)\rvert^2$	····	$\lvert x_{in}(n-1)\rvert^P$
$I_{in}(n)$	$Q_{in}(n)$	$\lvert x_{in}(n)\rvert$	$\lvert x_{in}(n)\rvert^2$	····	$\lvert x_{in}(n)\rvert^P$

Figure 4.4. Architecture of the proposed ARVTDTNN model.

4.3.1 Input Layer

The input layer processes each I and Q data to generate the input sequence for the transformer layer. The input sequence denoted as X_{in} consists of the I/Q components and the envelope-dependent terms of current and past signals. X_{in} is a 2D matrix where the columns represent features, and the rows represent the temporal variation of the sequence. This can be expressed as follows:

$$X_{in} = \begin{bmatrix} I_{in}(n-M) & \dots & I_{in}(n-1) & I_{in}(n) \\ Q_{in}(n-M) & \dots & Q_{in}(n-1) & Q_{in}(n) \\ |x(n-M)| & \dots & |x(n-1)| & |x(n)| \\ |x(n-M)|^2 & \dots & |x(n-1)|^2 & |x(n)|^2 \\ \vdots & \ddots & \vdots & \vdots \\ |x(n-M)|^P & \dots & |x(n-1)|^P & |x(n)|^P \end{bmatrix}^T, \tag{41}$$

where $I_{in}(n)$ and $Q_{in}(n)$ represent the I/Q components of the complex envelope $x(n)$ of the current PA input signal, respectively; $|x(n)|$ represents the amplitude of the signal; $I_{in}(n-m), Q_{in}(n-m)$, and $|x(n-m)|$, $\forall m$ where $\{m \in \mathbb{Z}|1 \leq m \leq M\}$ denote the corresponding past samples, respectively; M and P represent the memory depth and nonlinearity order, respectively.

4.3.2 Transformer Layer

For a simple illustration, input embedding, positional encoding, and transformer encoder modules are integrated into the transformer layer, as shown in Figure 4.4. The input to this layer is the preprocessed data sequence $X_{in} \in \mathbb{R}^{T \times d_{in}}$ where $T(= M+1)$ is the sequence length and $d_{in}(= P+2)$ is the number of features. Each sequence goes through the transformer layer and outputs another sequence with dimension $d_{model} \times T$, where d_{model} is the dimension used inside the transformer encoder.

Input embedding and positional encoding

The input embedding block uses time embedding where each time instance vector in the sequence X_{in} is transformed into a d_{model} dimensional vector using an FC layer. Embedding output $X_{emb} \in \mathbb{R}^{T \times d_{model}}$ can be written as

$$X_{emb} = X_{in} W_{emb}, \tag{42}$$

where $W_{emb} \in \mathbb{R}^{d_{in} \times d_{model}}$ is the weight matrix of the FC layer. The positional encoding is done similarly to that described in [6], using sine and cosine-based schemes to represent each position and dimension combination. The positional encoder adds $PE_{(i,j)}$ to the $(i,j)^{th}$ element of the X_{emb}, to obtain the $(i,j)^{th}$ element of $X_{pos} \in \mathbb{R}^{T \times d_{model}}$

$$PE_{(i,j)} = \begin{cases} sin(i/10000^{(2j/d_{model})}), & j \text{ is even,} \\ cos(i/10000^{(2j/d_{model})}), & j \text{ is odd,} \end{cases} \tag{43}$$

$$X_{pos}^{(i,j)} = X_{emb}^{(i,j)} + PE_{(i,j)}. \tag{44}$$

Transformer encoder

The transformer encoder employed in this study has the same functionality as the one detailed in [6]. It comprises a multi-head attention block, a fully connected FFN, layer normalization blocks, and residual connections, as illustrated in Figure 4.5. The multi-head attention function concatenates multiple parallel streams of scaled dot product attention outputs. Then it projects it to d_{model} dimensional space to get the final result, as shown in Figure 4.5. The query (Q), key (K), and value (V) matrices can be obtained as follows:

$$Q = X_{pos} W^Q, \tag{45}$$

$$K = X_{pos} W^K, \tag{46}$$

$$V = X_{pos} W^V, \tag{47}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{model} \times d_{model}}$ and $Q, K, V \in \mathbb{R}^{T \times d_{model}}$. Then, these matrices are split and fed into each attention head, as expressed below:

$$Q_i = Q W_i^Q, \tag{48}$$

$$K_i = K W_i^K, \tag{49}$$

$$V_i = V W_i^V, \tag{50}$$

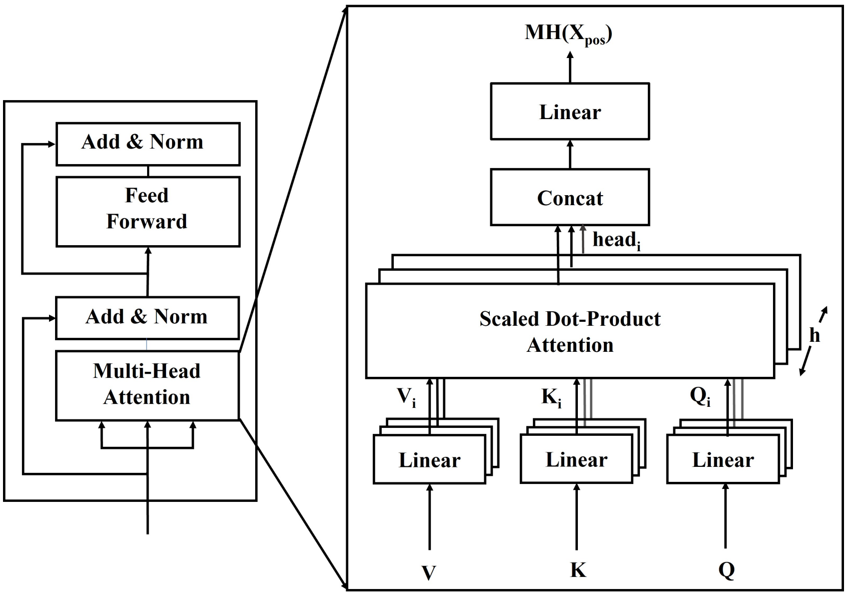


Figure 4.5. Architecture of transformer encoder and multi-head attention.

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_{model}/h}$. Now, i^{th} attention head output, $head_i$ can be computed using the scaled dot product and softmax function as below:

$$head_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_{model}/h}})V_i. \tag{51}$$

The multi-head output (MH) is obtained by aggregating the individual outputs of the scaled dot product attention operation across all heads $(head_i)$ and then passing the resulting combined output through a linear layer as

$$MH(X_{pos}) = concat(head_1, ..., head_h)W^O, \tag{52}$$

where $W^O \in \mathbb{R}^{d_{model} \times d_{model}}$ is the projection matrix of linear output layer. Multi-head attention output and positional encoded output are added with a residual connection and normalized through layer normalization [23] to obtain the input to the feed-forward network (FFN). It can be expressed as

$$X_{ffn} = LN(X_{pos} + MH(X_{pos})) \tag{53}$$

where LN is the layer normalization, as defined in [23]. Then X_{ffn} goes to the FFN comprising two linear transformations with a ReLU activation in between. FFN output can be written as follows:

$$FFN(X_{ffn}) = ReLU(X_{ffn}W_1 + b_1)W_2 + b_2, \tag{54}$$

$$ReLU(x) = max(0, x), \tag{55}$$

where W_i and b_i are the weight and bias matrices of the i^{th} linear layer. The residual connections and layer normalization follow the same procedure as before to obtain the final output of the transformer encoder as

$$X_{enc} = LN(X_{ffn} + FFN(X_{ffn})), \tag{56}$$

where $X_{enc} \in \mathbb{R}^{T \times d_{model}}$ is the final output of the transformer layer. Then the last time-step of this sequence (X_{enc}^{-1}) is selected and fed to the FC network.

4.3.3 Fully Connected Layer and Output Layer

The FC layer is included to improve the performance further. Input to the FC can be denoted as $X_{enc}^{-1} = [t_1, t_2, t_3 ..., t_{d_{model}}]^T$ where t_i is the i^{th} element of the last time step of transformer layer output. Then j^{th} neuron output of the FC layer ($j = 1, 2, 3, ..., J$) can be expressed as

$$fc_j = ReLU\left(\sum_{i=1}^{d_{model}} w_{ij}^{fc} t_i + b_i^{fc}\right), \tag{57}$$

where w_{ji}^{fc} and b_j^{fc} are the weights and biases of FC layer. The output layer is an FC layer consisting of two neurons with no activation function, which maps the final network output into two I and Q values as

$$I_{pred}(n) = \sum_{j=1}^{J} w_{1j}^{out} fc_j + b_1^{out}, \tag{58}$$

$$Q_{pred}(n) = \sum_{j=1}^{J} w_{2j}^{out} fc_j + b_2^{out}, \tag{59}$$

where $w_{1j}^{out}, w_{2j}^{out}$ and b_1^{out}, b_2^{out} are the weights and biases of output layer.

4.4 Training of the ARVTDTNN Model

The proposed model is trained using the PA measurement dataset provided by MathWorks, Inc., MATLAB version 2022a [24]. It was generated using NXP Airfast LDMOS Doherty PA operating at a frequency range of $3.6 - 3.8$ GHz with a gain of 29 dB. The input signal was a 100 MHz 5G-like OFDM waveform, with each subcarrier carrying a 16-QAM symbol. The training, validation, and testing data are normalized to have zero mean and unity standard deviation to improve the network convergence. Adam optimization algorithm is chosen to train the model and minimizes the mean square error (MSE) between the measured and ARVTDTNN predicted output. MSE is represented as

$$E = \frac{1}{2N} \sum_{n=1}^{N} [(I_{pred}(n) - I_{out}(n))^2 + (Q_{pred}(n) - Q_{out}(n))^2], \tag{60}$$

where N is the number of data points in the training set. $I_{pred}(n), Q_{pred}(n)$ and $I_{out}(n), Q_{out}(n)$ denote the predicted and measured I, Q values, respectively. The Adam optimization algorithm is set with the training parameters; $\beta_1 = 0.9, \beta_2 = 0.999, \mu_0 = 0, \nu_0 = 0, \epsilon = 10^{-8}$ and $learning\_rate = 10^{-3}$. A learning rate scheduler is employed, reducing the learning rate by 0.95 times every two epochs. The maximum number of training epochs is 200, and the batch size is 128. Training performance is evaluated using the validation set for every four epochs, and the training loop terminates if the performance fails to improve for five consecutive iterations. Finally, the model's coefficients are updated with the values that result in the lowest validation loss.

4.5 Extension to DPD

The DPD model aims to overcome the nonlinearity and memory effects of the PA; thus, it has the inverse function of the PA's nonlinear characteristics. The indirect learning approach is utilized in this study to determine the DPD function, with the output of the PA serving as the input for the DPD and the input of the PA being used as the output of the DPD during the modeling process. Then, the main path DPD is updated using the trained DPD to linearize the PA, as shown in Figure 4.6. The input matrix (X_n^{DPD}) and label vector (Y_n^{DPD}) of the DPD model can be represented as follows:

$$X_n^{DPD} = \begin{bmatrix} I_{out}(n-M) & \dots & I_{out}(n-1) & I_{out}(n) \\ Q_{out}(n-M) & \dots & Q_{out}(n-1) & Q_{out}(n) \\ |y(n-M)| & \dots & |y(n-1)| & |y(n)| \\ |y(n-M)|^2 & \dots & |y(n-1)|^2 & |y(n)|^2 \\ \vdots & \ddots & \vdots & \vdots \\ |y(n-M)|^K & \dots & |y(n-1)|^K & |y(n)|^K \end{bmatrix}^T, \tag{61}$$

$$Y_n^{DPD} = [I_{in}(n)\ Q_{in}(n)]^T, \tag{62}$$

where $I_{out}(n)$ and $Q_{out}(n)$ denote the I/Q components of the complex envelope $y(n)$ of the current PA output signal, respectively.

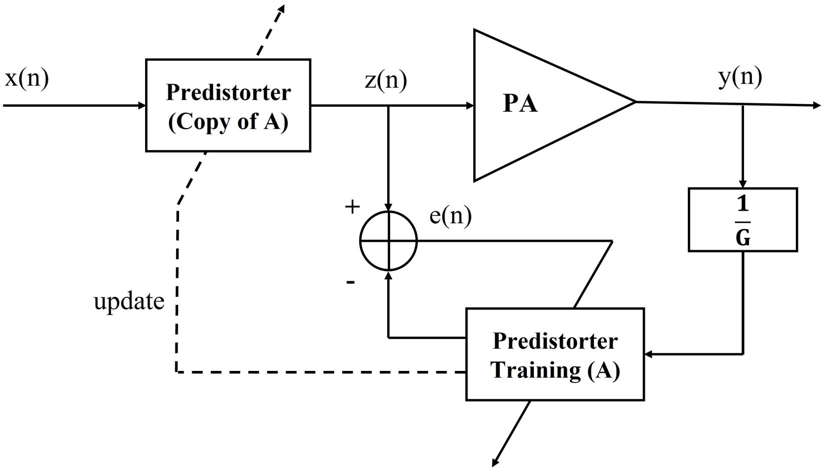


Figure 4.6. Indirect learning architecture.

5 RESULTS AND DISCUSSION

In this chapter, the findings of the research are presented, which involve the utilization of simulation tools to address the research problem and compare the proposed model with existing ones. The primary focus of the chapter is to present the results obtained from the ARVTDTNN model, particularly in terms of its performance in behavioral modeling and linearization.

5.1 Behavioral Modeling Performance

The ARVTDTNN model is used to obtain the behavioral model of the PA, which was illustrated in Figure 4.4. The measured input and output I/Q samples are used to train the ARVTDTNN model. Figure 5.1 depicts the behavioral modeling performance of the ARVTDTNN model, displaying the normalized power spectrum of the measured and modeled PA output. The comparison between the measured output and the output predicted by the ARVTDTNN model indicates that the model is highly accurate in predicting the behavior of the PA for 5G-like OFDM signals.

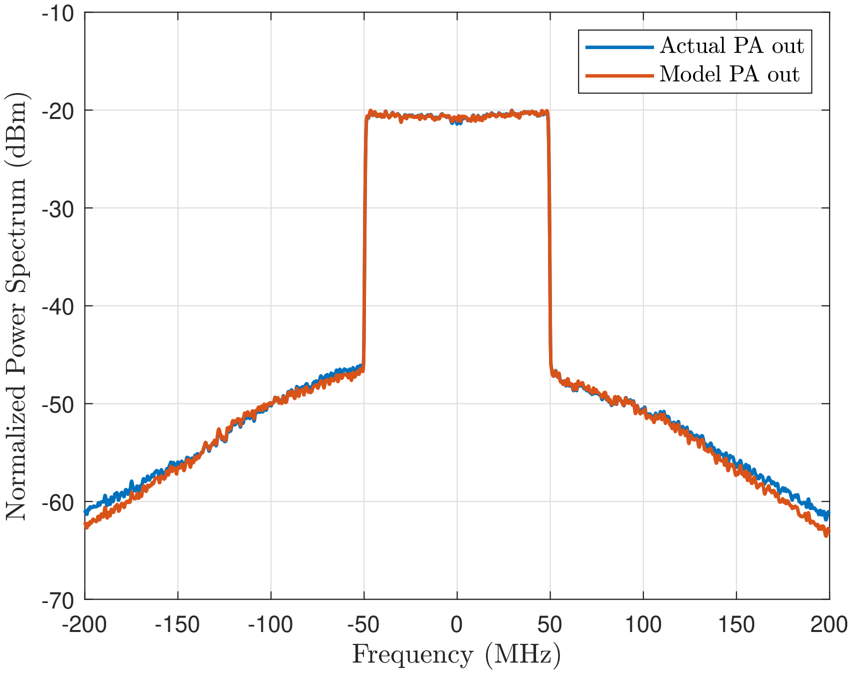


Figure 5.1. Behavioral modeling performance of ARVTDTDNN.

Thus, the proposed ARVTDTNN model is a highly effective approach for the behavioral modeling of PAs in 5G-like OFDM signals. The accuracy of the model is affirmed by the close match between the measured and modeled PA output, indicating that the transformer NNs are capable of accurately predicting the behavior of the PA for this type of signal. These findings have important implications for the design of communication systems, as accurate behavioral modeling of PAs along with faster inference and training time.

The behavioral ARVTDTNN model is obtained with the parameters $d_{model} = 8$, $h = 4$, and $J = 10$. These parameters define the size and complexity of the model, and they are chosen to optimize the accuracy and efficiency of the ARVTDTNN model. The transformer behavioral model is important because it allows the ARVTDTNN model to accurately capture the behavior of the transformer used in the model, which is critical for the accurate prediction of the PA output.

5.2 DPD Linearization Performance

The experimental setup involves evaluating the impact of linearization on the performance of DPD models by cascading different models before the ARVTDTNN behavioral PA model. The assessment of each DPD model's linearization performance is carried out by computing the normalized mean square error (NMSE) between the input to the DPD and the output of the PA, as well as the ACPR at the PA output. ACPR measures the ratio of interfering power in the adjacent channels to the signal power in the main channel.

To visually evaluate the improvement in linearization and the reduction in spectral regrowth, the normalized power spectrum of the PA output is used. The ARVTDTNN model and other state-of-the-art models are constructed using the PyTorch framework. The DPD models utilized in the experiment include LSTM, ARVTDNN, DNN, RVTDCNN, and ARVTDTNN. Table 5.1 provides a summary of the specific parameters for each model, all of which have a memory depth of four and two output layer neurons.

Table 5.2 presents a comparison between the proposed ARVTDTNN model and other state-of-the-art models with respect to their linearization performance in terms of NMSE and ACPR, as well as their complexity in terms of the number of coefficients. The results reveal that the ARVTDTNN model outperforms most existing DPD models, with a significant improvement in both NMSE and ACPR. Specifically, the ACPR is enhanced from -26.8 dB to -41.8 dB, while the NMSE is reduced from -22.3 dB to -37.6 dB. These findings demonstrate the effectiveness of the proposed ARVTDTNN model in linearizing the PA.

The comparison also indicates that the ARVTDTNN model offers superior performance and complexity when compared to other existing DPD models. It is worth noting that the performance of the ARVTDTNN model is comparable to that of the RVTDCNN model. Moreover, the parallel computation nature of transformers offers an advantage in training and inference times for complex non-linear problems, as is typical in 6G transmitters, compared to existing solutions.

Figure 5.2 illustrates a comparison of the normalized power spectrum of the linearized PA output obtained using different DPD models. The results demonstrate that the proposed ARVTDTNN DPD model is highly effective in reducing the spectral regrowth induced by the nonlinearity of the PA when compared to other existing models.

Specifically, the plot corresponding to the ARVTDTNN model shows a significant reduction in spectral regrowth compared to the other models, indicating that the model is highly effective in linearizing the PA. Moreover, the plot for the ARVTDTNN model appears to be following that of the RVTDCNN model, indicating that the ARVTDTNN model's performance is comparable to that of the RVTDCNN model.

Table 5.1. Comparison of performance and complexity.

Model	**Parameter**	**Value**
LSTM	Input features	I, Q
	Number hidden state features	10
	Number neurons in FC layers	[12 10]
	FC layers' activation	ReLU
ARVTDNN	Input features	$I, Q, \|x\|, \|x\|^2 \dots \|x\|^4$
	Number neurons in FC layer	17
	FC layer's activation	Tanh
DNN	Input features	I, Q
	Number neurons in FC layers	[17 17 17]
	FC layers' activation	ReLU
RVTDCNN	Input features	$I, Q, \|x\|, \|x\|^2 \dots \|x\|^4$
	Number conv. out channels	5
	Kernel size	$3 \times 3 \times 1$
	Number neurons in FC layer	10
	Conv. & FC layers' activation	Tanh
ARVTDTNN	Number features in transformer	8
	Number attention heads	4
	FFN dimension	16
	Number neurons in FC layer	10
	FC layers' activation	ReLU

Figures 5.3 and 5.4 provide a comparison of the gain and phase variations of the PA output obtained with and without the proposed ARVTDTNN DPD model. The results clearly demonstrate that the ARVTDTNN model is highly effective in reducing both gain and phase distortions in the PA.

In Figure 5.3, the gain variations of the PA output are plotted for both cases, with and without the ARVTDTNN DPD model. The plot for the PA output without the DPD model shows significant gain variations, whereas the plot for the PA output with the DPD model shows a much smoother curve with reduced gain variations. This indicates that the proposed ARVTDTNN DPD model effectively reduces gain distortions in the PA.

Similarly, in Figure 5.4, the phase variations of the PA output are plotted for both cases, with and without the ARVTDTNN DPD model. The plot for the PA output without the DPD model shows significant phase distortions, whereas the plot for the PA output with the DPD model shows a much smoother curve with reduced phase distortions. This indicates that the proposed ARVTDTNN DPD model effectively reduces phase distortions in the PA.

Table 5.2. Comparison of performance and complexity.

	ACPR (dB) (-/+ 25MHz)	**NMSE(dB)**	**Number of model coefficients**
Without DPD	-26.57/-27.03	-22.35	N/A
LSTM	-39.54/-39.17	-34.22	849
ARVTDNN	-41.48/-39.63	-36.38	563
DNN	-41.72/-40.17	-36.69	835
RVTDCNN	-42.79/-41.25	-37.58	682
ARVTDTNN	-42.49/-41.19	-37.61	768

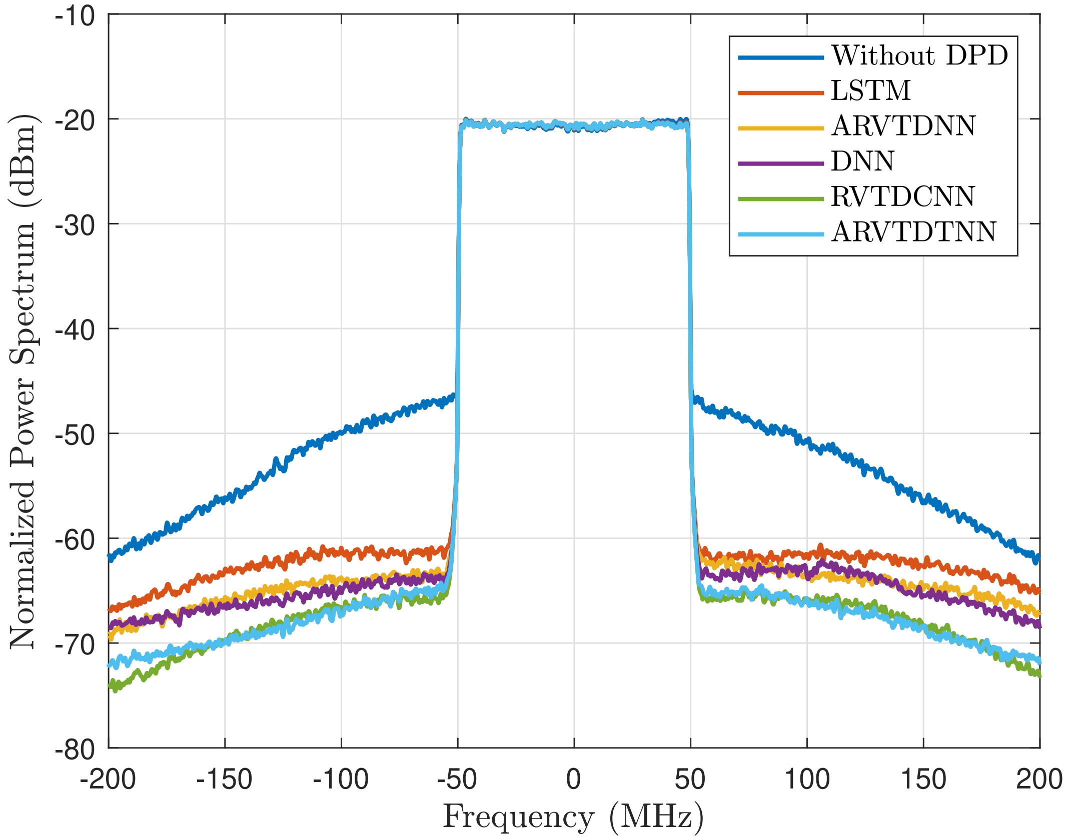


Figure 5.2. Normalized power spectrum comparison with state-of-the-art models.

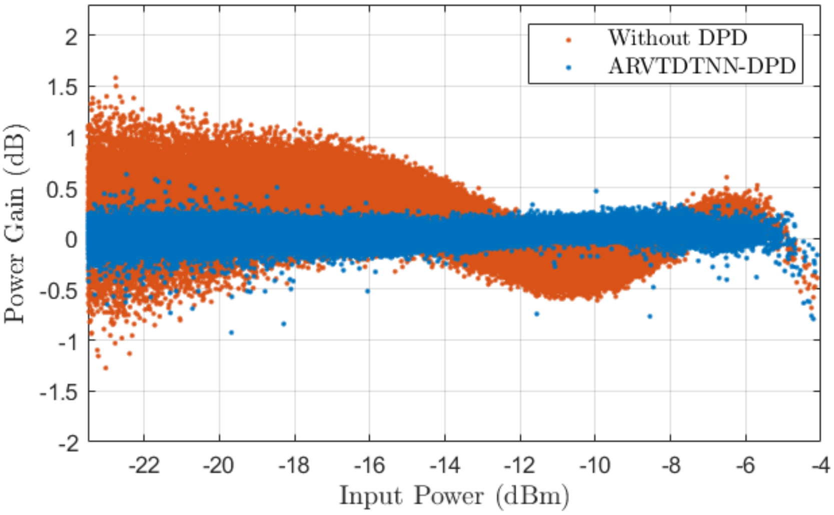


Figure 5.3. AM/AM characteristics of PA with and without the proposed DPD.

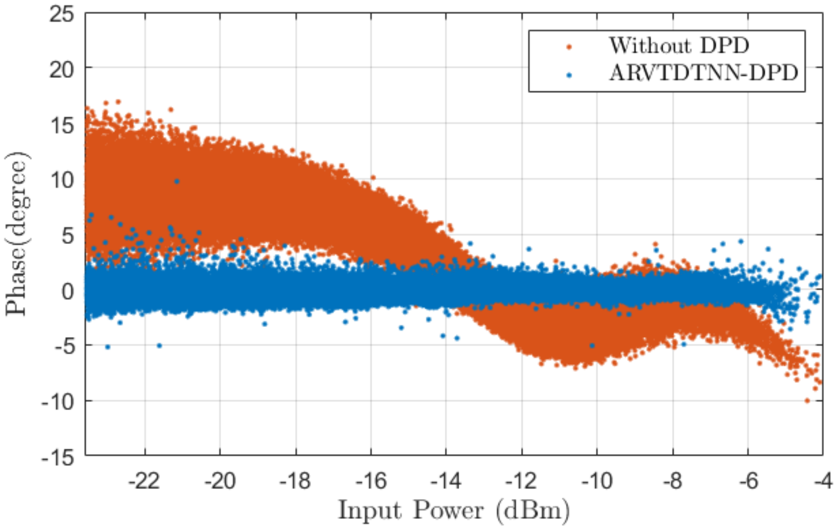


Figure 5.4. AM/PM characteristics of PA with and without the proposed DPD.

6 CONCLUSION AND FUTURE WORK

In this study, we proposed a transformer NN-based behavioral model and DPD linearizer for wideband PAs. This approach utilizes the multi-head attention mechanism to parallelize the computation and model complex nonlinearities and memory effects introduced by the PA. ARVTDTNN is tested with 5G-like 100 MHz OFDM signal, exhibiting an improvement in NMSE and ACPR of up to -37.6 dB and -41.8 dB, respectively. Based on the comparison, ARVTDTNN outperforms most state-of-the-art solutions and delivers comparable results to the RVTDCNN. Additionally, the computation parallelism nature of the transformer mechanism can improve the training and inference time during FPGA implementation. The ability of transformer NNs to model highly complex nonlinear problems and long-term dependencies makes them suitable for solving complex challenges in the upcoming high-bandwidth applications. Our future work aims to replace the PA behavioral model with a real PA to validate the simulation performance. Additionally, we plan to implement our DPD model on an FPGA to assess its linearization performance on actual hardware.

7 REFERENCES

[1] Ghannouchi F., Hammi O. & Helaoui M. (2015) Behavioral Modeling and Predistortion of Wideband Wireless Transmitters.

[2] Teikari I. (2008) Digital predistortion linearization methods for RF power amplifiers. Doctoral thesis. URL: `http://urn.fi/URN:ISBN:978-951-22-9546-3`.

[3] Yadav S.P. & Bera S.C. (2014) Nonlinearity effect of power amplifiers in wireless communication systems. In: 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), pp. 12–17.

[4] Morgan D., Ma Z., Kim J., Zierdt M. & Pastalan J. (2006) A generalized memory polynomial model for digital predistortion of rf power amplifiers. IEEE Transactions on Signal Processing 54, pp. 3852–3860.

[5] Hu X., Liu Z., Yu X., Zhao Y., Chen W., Hu B., Du X., Li X., Helaoui M., Wang W. & Ghannouchi F.M. (2022) Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers. IEEE Transactions on Neural Networks and Learning Systems 33, pp. 3923–3937.

[6] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L. & Polosukhin I. (2017) Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, p. 6000–6010.

[7] Raghavan A. & Srirattana N. (2008.) Modeling and Design Techniques for RF Power Amplifiers. Wiley - IEEE Ser., John Wiley Sons, Incorporated,, Hoboken :, 1st ed. ed. URL: `https://ebookcentral.proquest.com/lib/buse-ebooks/detail.action?docID=331585`.

[8] Van Moer W., Rolain Y. & Geens A. (2001) Measurement-based nonlinear modeling of spectral regrowth. IEEE Transactions on Instrumentation and Measurement 50, pp. 1711–1716.

[9] Sadeghpour Ghazaany T. (2011) Design and implementation of adaptive baseband predistorter for OFDM nonlinear transmitter. Phd thesis, University of Bradford, England. Available at `https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.582943`.

[10] Wood J. (2014) Behavioral modeling and linearization of RF power amplifiers.

[11] Liu T., Boumaiza S. & Ghannouchi F. (2004) Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks. IEEE Transactions on Microwave Theory and Techniques 52, pp. 1025–1033.

[12] Wang D., Aziz M., Helaoui M. & Ghannouchi F.M. (2019) Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters. IEEE Transactions on Neural Networks and Learning Systems 30, pp. 242–254.

[13] Liu T., Hui M., Zhang Y., Yang D., Ye Y., Zhang M., Lin W. & Jiang M. (2016) RF power amplifier modeling and linearization with augmented RBF neural networks. In: 2016 IEEE International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM), pp. 1–3.

[14] Hongyo R., Egashira Y., Hone T.M. & Yamaguchi K. (2019) Deep neural network-based digital predistorter for doherty power amplifiers. IEEE Microwave and Wireless Components Letters 29, pp. 146–148.

[15] Liu Z., Hu X., Xu L., Wang W. & Ghannouchi F.M. (2022) Low computational complexity digital predistortion based on convolutional neural network for wideband power amplifiers. IEEE Transactions on Circuits and Systems II: Express Briefs 69, pp. 1702–1706.

[16] Motaqi A., Helaoui M., Boulejfen N., Chen W. & Ghannouchi F.M. (2022) Artificial intelligence-based power-temperature inclusive digital predistortion. IEEE Transactions on Industrial Electronics 69, pp. 13872–13880.

[17] De Silva U., Koike-Akino T., Ma R., Yamashita A. & Nakamizo H. (2022) A modular 1D-CNN architecture for real-time digital pre-distortion. In: 2022 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), pp. 79–81.

[18] Phartiyal D. & Rawat M. (2019) LSTM-deep neural networks based predistortion linearizer for high power amplifiers. In: 2019 National Conference on Communications (NCC), pp. 1–5.

[19] Sun J., Shi W., Yang Z., Yang J. & Gui G. (2019) Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems. IEEE Transactions on Vehicular Technology 68, pp. 10348–10356.

[20] Wang W., Sun L., Liu H. & Feng Y. (2022) LSTM-CNN for behavioral modeling and predistortion of 5G power amplifiers. In: 2022 IEEE 9th International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications (MAPE), pp. 28–32.

[21] Peng H., Huang S., Geng T., Li A., Jiang W., Liu H., Wang S. & Ding C. (2021) Accelerating transformer-based deep learning models on FPGAs using column balanced block pruning. In: 2021 22nd International Symposium on Quality Electronic Design (ISQED), pp. 142–148.

[22] Wu N., Green B., Ben X. & O'Banion S. (2020), Deep transformer models for time series forecasting: The influenza prevalence case. URL: `https://arxiv.org/abs/2001.08317`.

[23] Ba J.L., Kiros J.R. & Hinton G.E. (2016), Layer normalization. URL: `https://arxiv.org/abs/1607.06450v1`.

[24] MATLAB (2022) version 9.12 (R2022a). The MathWorks Inc., Natick, Massachusetts.