System approach for optimizing a low-complex digital predistortion model for RF power amplifier linearization

Chouaib Kantana

System Approach for Optimizing a Low-Complexity Digital Predistortion Model for RF Power Amplifier Linearization

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale n° d'accréditation, dénomination et sigle
Spécialité de doctorat: voir annexe
Unité de recherche : voir annexe

Thèse présentée et soutenue à l'Université Gustave Eiffel, le 10/12/2021, par

Chouaib KANTANA

Composition du Jury

Patricia DESGREYS Pr. HDR, Telecom Paris	Examinatrice
Daniel ROVIRAS Pr. HDR, CNAM Paris	Examinateur
Smail BACHIR MdC HDR, Université de Poitiers	Rapporteur
Yide WANG Pr. HDR, Université de Nantes	Rapporteur
Geneviève BAUDOIN Pr. émérite, Université Gustave Eiffel	Directrice de thèse
Olivier VENARD Pr. associé, Université Gustave Eiffel	Co-Directeur de thèse
Rui MA Dr., MERL	Invité

Encadrement de la thèse

Geneviève BAUDOIN Pr. émérite, Université Gustave Eiffel	Directrice de thèse
Olivier VENARD Pr. associé, Université Gustave Eiffel	Co-Directeur de thèse

Université Gustave Eiffel

École doctorale MSTIC
Mathématiques, Sciences, et Technologies de l'information et de la communication

Doctoral Thesis

Specialty: Electronics, Optronics, and System

from University Gustave Eiffel

Presented by

Chouaib Kantana

System Approach for Optimizing a Low-Complexity Digital Predistortion Model for RF Power Amplifier Linearization

Supervised by

Geneviève Baudoin and Olivier Venard

December 10, 2021

Jury:

Rapporteurs :	Pr. HDR **Yide WANG**	Université de Nantes
	MdC HDR **Smail BACHIR**	Université de Poitiers
Examinateurs :	Pr. HDR **Patricia DESGREYS**	Télécom Paris
	Pr. HDR **Daniel ROVIRAS**	CNAM Paris
Directeur de thèse :	Pr. HDR **Geneviève BAUDOIN**	Université Gustave Eiffel
Co-Directeur de thèse :	PA **Olivier VENARD**	Université Gustave Eiffel
Invité :	Dr. **Rui MA**	MERL - Cambridge US

Acknowledgment

I would like to express my gratitude to the following people for their help and continuous support throughout my PhD.

First and foremost, I would like to thank my supervisor Professor HDR Geneviève BAUDOIN and my co-supervisor Professor Olivier VENARD, whose guidance and patience guided my work and assisted me in writing this thesis. They ensured my academic development and continuously supported my work and ideas throughout my three years in Université Gustave Eiffel and ESIEE Paris. We have had widespread fruitful discussions since the first week of my PhD life. It was a great joy to have been working with them.

My international internship in Cambridge US has been one of the most exciting, engaging, and prolific experiences I have had. This would not have been possible without Dr. Rui MA, senior principal research scientist and team leader at Mitsubishi Electric Research Laboratories (MERL). Thank him for the trust and support he has shown me during my internship at MERL. I would also like to thank Dr. Mouhacine BENOSMAN, senior scientist at MERL, for his unconventional ideas, guidance, and helpful advice.

I would like to sincerely thank my thesis defense committee members for agreeing to evaluate my work. Thanks to Professor Yide WANG and Professor Smail BACHIR for agreeing to report my dissertation and improve my thesis with your thoughtful advice and valuable suggestions.

Finally, I would like to thank my dear beloved family, to whom I am offering this work.

”,”

Abstract

This Ph.D. work contributes to the digital predistortion linearization technique of power amplifiers. Power Amplifier is one of the most critical elements of radiocommunication systems, which exhibits static nonlinearities and nonlinear memory effects. Achieving a good trade-off between the linearity of the power amplifier and its efficiency is becoming more crucial. Digital predistortion is a powerful linearization technique that aims to compensate for power amplifier distortions and provides linear amplification with good efficiency. The predistortion principle consists of implementing a nonlinear function, the so-called predistorter upstream of the power amplifier. The predistorter ideally has the inverse characteristics of the power amplifier. This operation allows us to consider the memory effects of the power amplifier, and in particular, the long-term memory. Several behavioral models have been used as predistorter. Most of them are classified into two families: global models derived from the Volterra series and models based on the segmentation approach. The comparative and analysis study of these models is one of the focuses of this dissertation, in which three aspects are used for comparison: linearization performance, complexity, and hardware implementation properties. By focusing on models based on the segmentation approach, this dissertation proposes an approach to design an optimal model according to a trade-off between linearization performance and model complexity. This model is used to linearize a dual-input Doherty power amplifier. A global optimization algorithm combined with a control process is proposed to enhance efficiency while maintaining a good linearity level according to a proposed adaptive cost function.

Résumé

Le travail de thèse présenté par ce manuscrit s'intéresse à la linéarisation des amplificateurs de puissance en utilisant la prédistorsion numérique. L'amplificateur de puissance est l'un des modules les plus critiques des équipements de communication radio qui présente des non-linéarités statiques ainsi que des effets de mémoire. Il devient de plus en plus crucial de réaliser un compromis entre la linéarité et le rendement énergétique. La prédistorsion numérique est une technique de linéarisation efficace qui compense les distorsions dues à la non-linéarité en appliquant une déformation sur le signal d'entrée, de manière à ce que le système global réalise une amplification linéaire. Le principe de la prédistortion numérique consiste à mettre en œuvre une fonction non linéaire dite prédistorteur en amont de l'amplificateur de puissance, et qui a idéalement les caractéristiques inverses de l'amplificateur de puissance. Cette opération permet également de prendre en compte les effets mémoire, et en particulier la mémoire à long terme. Plusieurs modèles comportementaux sont proposés pour la prédistorsion qui peuvent être classés en deux catégories : les modèles globaux et les modèles par segmentation. L'étude comparative de ces modèles est l'un des axes de travail traité par cette thèse, dont la comparaison repose sur trois aspects qui sont soulignés : les performances de linéarisation, la complexité et les propriétés de l'implémentation matérielle. En se concentrant sur les modèles basés sur l'approche par segmentation, un algorithme est proposé pour concevoir un modèle optimal en satisfaisant un bon compromis entre les performances de linéarisation et la complexité du modèle. Ce modèle est utilisé pour linéariser un amplificateur de puissance Doherty à double entrée pour lequel un algorithme d'optimisation globale associé à un processus de contrôle est proposé pour améliorer le rendement tout en conservant un bon niveau de linéarité selon une fonction de coût adaptative.

Contents

List of Figures

List of Tables

List of Acronyms

ACPR	Adjacent Channel Power Ratio
ADC	Analog Digital Converter
AWG	Arbitrary Waveform Generator
CPWL	Canonical Piece Wise Linear
CCDF	Complementary Cumulative Distribution Function
CDMA	Code Division Multiple Access
CFR	Crest Factor Reduction
CDF	Cumulative Distribution Function
DAC	Digital Analog Converter
DPD	Digital Predistortion
DUT	Device Under Test
DVR	Decomposed Vector Rotation
DDR	Dynamic Deviation Reduction
ESC	Extremum-Seeking Control
EVM	Error Vector Magnitude
FPGA	Field-Programmable Gate Array
GA	Genetic Algorithms
GMP	Generalized Memory Polynomial
GS	Golden Section
HC	Hill Climbing
LTE	Long Evolution Term
LS	Least Square
MP	Memory Polynomial
NMSE	Normalized Mean Square Error
OFDM	Orthogonal Frequency Division Multiplex
OP	Operating Point
PA	Power Amplifier
PAE	Power Added Efficiency
PAPR	Peak-to-Average Power Ratio
PDF	Probability Density Function
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency
SA	Simulated Annealing
SNR	Signal to Noise Ratio
SDR	Software Defined Radio
VSA	Vector Signal Analyzer

Publications and Patents

Submitted Patents

1. Chouaib Kantana, Rui Ma, and Mouhacine Benosmane, "Proposal linearizer-splitter for multiple-input power amplifiers", submitted for US patent application in April 2021.

2. Chouaib Kantana, Rui Ma, and Mouhacine Benosmane, "Deep Learning-based Online Adaptation of Digital Pre-Distortion and Power Amplifier Systems", submitted for US patent application in December 2020.

3. Chouaib Kantana, Rui Ma, and Mouhacine Benosmane, "Interactive Online Adaptation for Digital Pre-Distortion and Power Amplifier System Auto-Tuning", submitted for US patent application in December 2020.

Journals

1. Chouaib Kantana, Olivier Venard and Genevieve Baudoin, "Thresholds Optimization of Decomposed Vector Rotation Model for Digital Predistortion of RF Power Amplifier", Radioengineering, 30(1), 250-258.

2. Hadi Muhammad Usman, Chouaib Kantana, Pier A. Traverso, Giovanni Tartarini, Olivier Venard, Geneviève Baudoin, and Jean Luc Polleux. "Assessment of Digital Predistortion Methods for DFB-SSMF Radio-Over-Fiber Links Linearization." Microwave and Optical Technology Letters 62 no. 2 (2020): 540-546.

International Conferences

1. Chouaib Kantana, Rui Ma, Mouhacine Benosman, Olivier Venard and Genevieve Baudoin, "A Hybrid Heuristic Search Control Assisted Optimization of Dual-Input Doherty Power Amplifier", accepted to EuMW 2021.

2. C. Kantana, O. Venard and G. Baudoin, "Decomposed Vector Rotation Model Sizing by Hill-Climbing Heuristic for Digital Predistortion of RF Power Amplifiers," 2020 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), San Antonio, TX, USA, 2020, pp. 22-25.

3. C. Kantana, O. Venard and G. Baudoin, "Linearization Performance vs Implementation Properties of Digital Predistorter Modelling for Three-Way Doherty Power Amplifier," 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), 2019, pp. 1-6. (Best paper award).

4. C. Kantana, O. Venard and G. Baudoin, "Comparison of GMP and DVR models" 2018 International Workshop on Integrated Nonlinear Microwave and Millimeter-wave Circuits (INMMIC), Brive La Gaillarde, 2018, pp. 1-3.

5. Kantana C., Venard O., Baudoin G., "Digital Predistortion for RF power amplifiers with high power efficiency and linearity", Workshop International en Technologies de l'Information et de la Communication au service d'une Ville intelligente, 3-4 May 2018, Marrakech, Maroc.

National Conferences and Workshops

1. C. Kantana, O. Venard and G. Baudoin, "La prédistorsion numérique : Etude de comparaison des performances de linéarité et des propriétés d'implémentation", La 21ème édition des Journées Nationales Microondes (JNM), 14-17 mai 2019, Caen, France.

2. Kantana C., Baudoin G., Venard O., "Evaluation and Comparison of Digital Predistorter Modeling for Power Amplifier", 13ème Colloque. du GDR SoC/SiP, 13–15 Juin 2018, Paris, France.

Introduction

Motivation

Modern wireless communication systems have been developed over the past three decades to provide high-speed services such as mobile communications, broadcast TV, WiFi networks, video conferences, etc. With their increasing demands for higher data rates and capacity, the rapid evolution of wireless communications continuously increases the complexity of radiofrequency systems. To meet the demands of future wireless communication systems, extensive research is being conducted to develop an energy-efficient and reconfigurable radio transmitter that can support multiple access technologies and operate at variable frequency bands.

Constrained by limited radio frequency (RF) resources, transmitting modulations, such as orthogonal frequency division multiplexing (OFDM), are used to achieve high spectral efficiency. However, these non-constant envelope signals lead to modulated signals with a high peak-to-power average power ratio (PAPR), which are more sensitive to the transmission channel's nonlinearity and require a high degree of linearity at the transmitter. Besides, carrier aggregation allows for a very high data rate but creates new challenges for transmitters in terms of bandwidth, intermodulation, harmonics, and filtering.

The power amplifier (PA) is one of the essential components of radio transmitters. Besides its role in amplifying the signal, the PA is also the primary source of signal distortion and the major contributor to the radio transmitter chain's energy consumption. To maintain the reliability of the RF system and reduce energy consumption, PA is required to be linear and have high efficiency. Unfortunately, the design of high-efficiency and linear PAs is not easy.

On the other hand, the development of 5G and beyond 5G wireless communication creates new challenges to provide a wide range of frequency bands that 5G wireless communication systems can handle. With the cadence of this development, high efficiency and linearity are two crucial requirements that are not easy to fulfill simultaneously, and achieving a good trade-off between them is becoming more crucial. For high efficiency, PA's operating point (OP) is usually driven towards the saturation region where high nonlinear behavior is exhibited. Besides strong nonlinearities, PAs may have memory effects.

The output of an ideal PA should be linear, but in reality, PA is neither perfectly linear nor entirely instantaneous when a memory effect happens. These nonlinearities generate distortions in the signal band and cause spectral regrowth in the adjacent channels. Unfortunately, a good efficiency is obtained at the price of poor linearity, especially with modern communication waveforms with very high PAPR and large bandwidths. This situation leads to ensures a trade-off between efficiency and linearity. The nonlinear behavior of the PA depends not only on its electronics components but also on the characteristics of the transmitted signal.

The behavioral modeling of PA and its linearization have given many research works

in recent years. Digital predistortion (DPD) is one of the most efficient linearization techniques used to linearize high PAs, pushing the OP of PA towards its high-efficiency region. The principle of DPD consists of applying, upstream of PA, a pre-correction on the signal so that the cascade of DPD and PA produces a linear and memoryless amplification with good efficiency. The purpose of studying the distortions of PA is to design an appropriate DPD function, called predistorter, that will compensate for these distortions and be implemented in a real digital circuit (FPGA, DSP ...). The hardware implementation, which is an important aspect of this dissertation, involves two aspects: identification and predistortion implementation.

Many mathematical models have been proposed in the literature in order to be used to accurately model the dynamic nonlinear systems, including linearity and different types of memory effects. One of the contributions of this Ph.D. work is the comparison of main DPD models in terms of linearization performance, complexity, and hardware implementation properties.

The DPD models are mainly derived from the Volterra series. However, other models are based on the segmentation approach, such as the canonical piecewise linear (CPWL) functions. The segmentation approach is based on splitting the input baseband signal's amplitude range into different regions characterized by different models. The models based on the segmentation approach can represent strong nonlinearities and are less prone to numerical problems as they are divided into segments in which a smaller order of nonlinearity is required. The design of an appropriate DPD model based on the segmentation approach with low complexity and a viable implementation strategy is an essential topic of this dissertation.

On the other hand, maintaining a good linearity level using the DPD with enhanced PA efficiency becomes more complex, especially for wideband signals with high PAPR. Advanced PA architectures have been proposed in the literature to enhance efficiency and avoid wasting excessive power resources. Some of these architectures are used with two RF inputs, so having more degrees of freedom offered by the separate inputs can control the performance and improve the PA efficiency.

Objectives

The above discussion reveals significant challenges in applying existing linearization solutions to improve further the system for both current LTE and future 5G wireless communication systems. This thesis aims to address these questions through the following objectives:

This dissertation mainly focuses on:

- Analysis and comparative study of the DPD models in terms of linearization performances, model complexity, and numerical hardware properties.
- Search of the optimal structure of models with segmentation approach according to a cost function representing a trade-off between modeling accuracy and model complexity. Besides, an approach is proposed to divide the model segments optimally.
- Optimization of the structure of dual-input PA with DPD in which an optimization process is proposed to enhance the efficiency while maintaining the linearity.

Main Contributions

The main contributions of this dissertation are listed as follows:

- An analysis and comparative study of the DPD models. These models have been categorized into two families: global models derived from the Volterra series and models based on the segmentation approach. The models have been evaluated according to a trade-off between:
  - Linearization performances presented by the modeling accuracy measure and the adjacent channel power ratio.
  - Complexity presented by the number of DPD coefficients to be identified.
  - Hardware implementation related to the identification process and predistorter implementation.
- A new approach is proposed to optimally set the thresholds bounding the segments of the DPD models based on the segmentation approach. The impact of the model parameters on optimal segmentation has been investigated.
- A heuristic algorithm based on hill-climbing is proposed and confirmed to determine the optimal structure of the DPD model based on the segmentation model with:
  - A new definition of the neighborhood for the hill-climbing algorithm.
  - A designed cost function is representing the trade-off between modeling accuracy and model complexity.
- A proposed system approach to design an optimal DPD model in terms of modeling accuracy, model complexity, and numerical properties. This contribution is based on:
  - A proposed approach to integrate with reduced system complexity the hill-climbing algorithm and the segmentation optimization process.
  - A proposed strategy for the hardware implementation of the DPD model designed by the system approach.
- A proposed auto-tuning approach to optimize the parameters of the dual-input Doherty PA to enhance its power efficiency with maintaining a good level of linearity. The designed DPD model is used to linearize the PA and confirms our proposed system approach.

Context

The Ph.D. work presented in this dissertation is funded by the FUI22 APOGEES research project (AmPlification recOnfiGurablE multimodES). The APOGEES project aims to solve the problems related to the reconfigurability of amplification systems to meet the different challenges generated by the new uses of the spectrum.

The APOGEES project has been labeled by Aerospace Valley, Image Réseaux, and Elopsys. The industrial partners involved in this project (ARELIS, THALES, TELERAD) aim to implement a reconfigurable system that deals with distortions of power amplifiers to make the transmitter systems more flexible for different applications: drones control, tactical communications, aeronautical communications, etc. The work carried out within this thesis's framework is also done in close collaboration with the other academic partners involved in this project: Centrale Supélec, IMS Bordeaux, XLim Poitiers, and IETR University of Nantes.

International mobility was carried out as an intern at Mitsubishi Electric Research Laboratories (MERL) in Cambridge, USA, under the host Dr. Rui Ma and Dr. Mouhacine

Benosman. The internship is titled *Machine Learning for Digital Predistortion Linearization of High-Efficiency Dual-Input Doherty Power Amplifier*.

Outline

The dissertation consists of five chapters.

Chapter 1 presents general concepts and the background of PA. Figures of merit used to assess the PA distortions are cited. PAPR reduction techniques and linearization techniques are also introduced, which are at the heart of this dissertation.

The principle of DPD is described in Chapter 2. Different DPD models are cited and reviewed. An analysis and comparison of DPD models is achieved according to modeling performances, complexity, and numerical properties.

In Chapter 3, the Decomposed Vector Rotation (DVR) model is investigated. A new approach to optimize the thresholds that determine the boundary of the segments is proposed. The determination of the optimal structure of the DVR model is introduced as well.

A system approach is proposed in Chapter 4 to design an optimal DVR model with reduced complexity by combining the thresholds optimization and the DVR model's optimal sizing. The performances of the designed optimal DVR model are compared with those of the conventional DVR model. An implementation strategy of the optimal model is presented.

Chapter 5 presents the architecture of dual-input Doherty PA, in which a proposed optimization process is presented. The designed optimal DVR model from Chapter 4 is used as a DPD model to linearize the dual-input Doherty PA, where the DPD is optimally integrated into the global optimization process.

Finally, we give the conclusion and perspectives.

Chapter 1

Generalities on Radio Frequency Power Amplifier

1.1 Introduction

Wireless radiofrequency (RF) transmitter is a fundamental block of modern communication systems to transmit information from a given source to a receiver. Figure 1.1 illustrates a conventional architecture of an RF transmitter by highlighting the digital and analog domains.

The information to be transmitted is first processed in digital form based on digital circuits such as FPGA (Field-Programmable Gate Array) or by digital signal processing (DSP). During this step, the information is encoded to be transmitted on two channels: in-phase I and quadrature Q. The IQ data are then passed through a block, so-called digital front-end for the RF processing, which usually contains a digital-up converter (DUC) for the frequency transposition, a crest factor reduction (CFR) to improve the PA efficiency, and digital predistortion (DPD) for the PA linearization. The signal is then converted by a digital-to-analog converter (DAC). Reconstruction filters are often used after this conversion to eliminate unwanted frequency components and limit bandwidth.

The analog part of the transmitter is responsible for transposing the signal from the baseband to the RF domain. Several transmitter architectures are used, such as direct conversion (Zero-IF), which transposes the signal directly to the carrier frequency, and low-IF architecture, which transposes the signal to an intermediate frequency before being transposed to the carrier frequency. Since the power of the RF signal must be of a sufficient level to be transmitted through the channel, an amplification module is required to supply the power needed for the RF signal to ensure its transmission to destination.

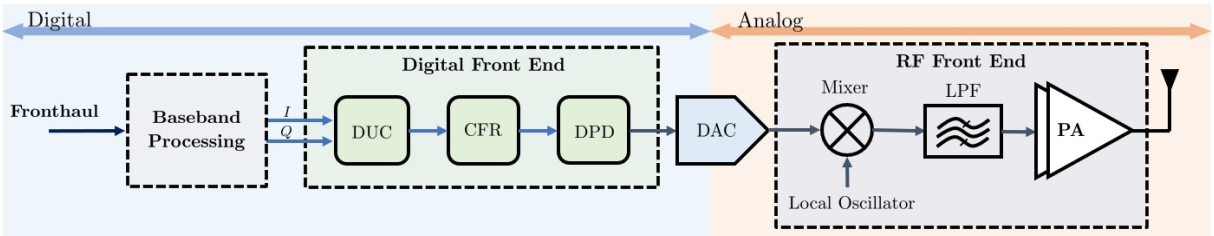


Figure 1.1: Block diagram of RF wireless transmitter

The RF transmitters are expected to conduct modulated signals while maintaining three performance indicators: linearity, bandwidth, and power efficiency. The linearity is affected by the nonlinear behavior of the transmitter caused by its distortions. Ensuring linearity is an essential aspect to preserve the quality of the transmitted signal and to avoid any loss of information during the transmission process. Since the bandwidth of the RF

signal to be transmitted is wider than that of the input signal due to the generation of new frequency components, the bandwidth is a critical aspect to be managed. Moreover, wider bandwidths are needed to accommodate higher data. Power efficiency is an important consideration that reflects the power level of transmitting the information. It depends on the electronic topology of the transmitter, in particular the power amplifier, and affects the costs of deploying and operating the communication infrastructure and the environmental impact. An ideal transmitter must achieve a better compromise between good linearity and high power efficiency in handling wideband signals.

Wireless transmitters consist of several stages, including modulation, DAC, frequency up-conversion, filtering, and amplification. Different wireless transmitter architectures have been introduced in the RF community, and one of them is shown in Figure 1.1. The position of DAC may change in some structures.

The distortions seen in RF transmitters are usually due to hardware imperfections. They have various origins, such as IQ imbalance, oscillator phase noise, DAC imperfections, frequency response distortions, amplitude and phase distortions, etc. The most predominant distortions are those due to the nonlinearity presented in the RF front-end and mainly the RF power amplifier (PA), which is identified as the primary source of nonlinear distortions. Thus, modeling and compensating for nonlinear transmitter distortions are often turned to modeling and compensating for PA nonlinearities.

This chapter presents some generalities related to PAs. The distortions introduced by PA and the metrics to evaluate them are discussed. Two essential aspects concerning the PA, which are linearity and efficiency, and the techniques to improve them will be discussed.

1.2 Waveform Features

To quantify the dynamic of modulated waveform with non-constant envelope, some criteria such as peak-to-average power ratio (PAPR) and effective PAPR are used.

To statistically characterize the modulated waveform's instantaneous power fluctuations, the most used parameters are the crest factor (CF) and its quadratic value, the PAPR, which is defined by the ratio of the peak power P_{peak} and the average power P_{avg} as:

$$\text{PAPR}_{\text{dB}} = 10 \log \left(\frac{\text{P}_{\text{peak}}}{\text{P}_{\text{avg}}} \right) \tag{1.1}$$

The CF is defined as square root of the PAPR:

$$\text{CF} = \sqrt{\text{PAPR}} \tag{1.2}$$

It is important to know that the PAPR of the baseband signal and the RF signal are not the same. However, they are linked by:

$$\text{PAPR}_{\text{RF}_{\text{dB}}} \leq \text{PAPR}_{\text{baseband}_{\text{dB}}} + 3 \text{ dB} \tag{1.3}$$

where it is true if the carrier frequency f_c is much greater than the signal bandwidth, which is met almost all the time. In this dissertation, we will only talk about $\text{PAPR}_{\text{baseband}_{\text{dB}}}$ which will be denoted by PAPR.

Generally, the PAPR is used to define and represent the dynamics of the signal. However, this parameter is not sufficient to characterize the real dynamics of the signal.

Figure 1.2 presents the magnitude of IQ data in dBm of LTE signal, where the PAPR of the LTE signal is shown.

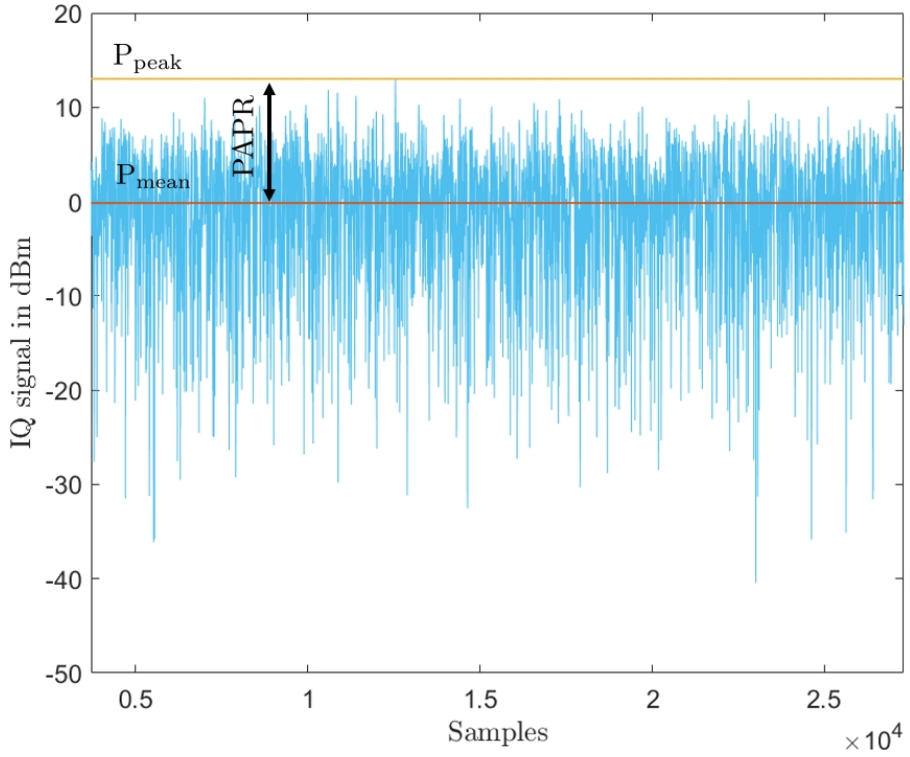


Figure 1.2: IQ data of LTE signal in dBm

In practice, the transmitted signal is random, and it has a PAPR value which is only exceeded with a low α probability, typically 1% or 1‰.

This value so-called effective PAPR, denoted by PAPR_0, is defined using CCDF by

$$p(\text{PAPR} \geq \text{PAPR}_0) = \alpha \tag{1.4}$$

1.3 Overview of Power Amplifiers Characteristics

The PA is an essential component of the transmission block of RF wireless systems. In its conventional definition, a PA is an electronic device used to increase the power of a signal.

The PA is usually the final stage before the antenna, which is designed to gain the input signal and, therefore, the power required to transmit it through the radio channel to reach the receiver. In an ideal context, the amplification operation should not clip the desired signal. However, the PA being a nonlinear device, is a source of distortions that impact the transmitted signal.

The design of PA is always subject to the antagonism between linearity and power efficiency [1]. Figure 1.3 shows a common behavior of output power (blue curve) and power efficiency (red curve) with respect to the input power, where the maximization of efficiency is done by sacrificing linearity by pushing the operating point of the PA towards its saturation zone.

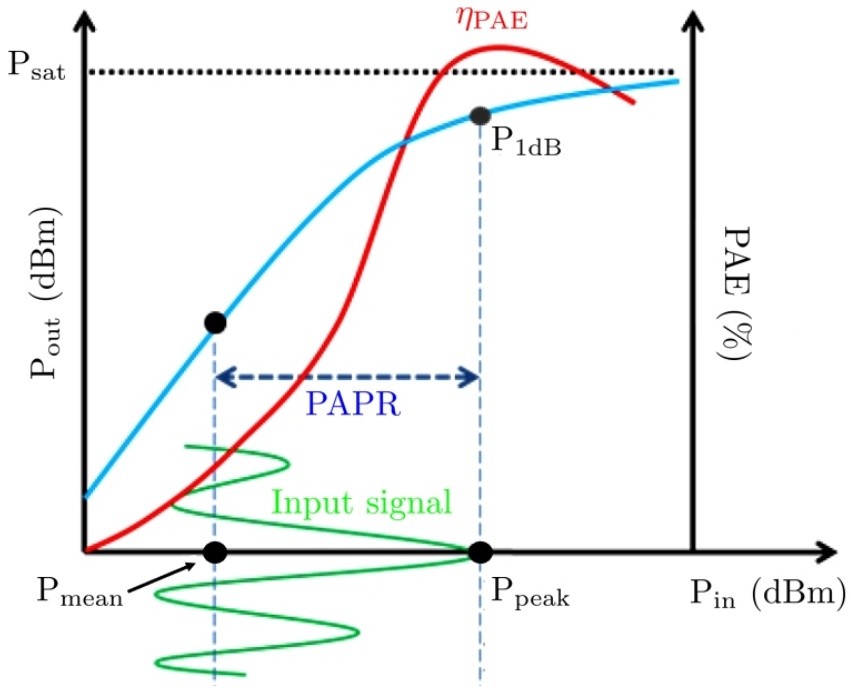


Figure 1.3: Trade-off between the linearity and the efficiency of PA

However, the operating point needs to be backed off from the saturation zone to avoid nonlinear distortions. For this to happen, an approach is required to maximize power efficiency while keeping its distortions at a reasonable level that can be compensated at the system level using linearization techniques such as Feedforward, Feedback, or Predistortion [2].

The behavior and characteristics of a PA depends on its class and topology. In the following, we present some general characteristics of PA.

1.3.1 Gain

The gain is one of the most important characteristics of the PA, which provides a measure of the amplification level of the transmitter system.

The power gain is the ratio between the output power and the input power of the signal, which is given in dB by

$$G_{\mathrm{dB}} = P_{out_{\mathrm{dBm}}} - P_{in_{\mathrm{dBm}}} \tag{1.5}$$

where $P_{in_{\mathrm{dBm}}}$ and $P_{out_{\mathrm{dBm}}}$ are the input and output power, respectively.

1.3.2 AM-AM and AM-PM Characteristics

The nonlinearities can be characterized by curves called Amplitude-to-Amplitude (AM-AM) and Amplitude-to-Phase (AM/PM) curves, as shown in Figure 1.4 that is an example for a Doherty PA.

The blue curve is the AM-AM characteristic, which shows the normalized magnitude of the PA's output signal versus the normalized magnitude of its input signal. The signals are the complex baseband IQ signals. The orange curve is the AM-PM characteristic, which shows the PA output signal's phase deviation versus the normalized input signal magnitude.

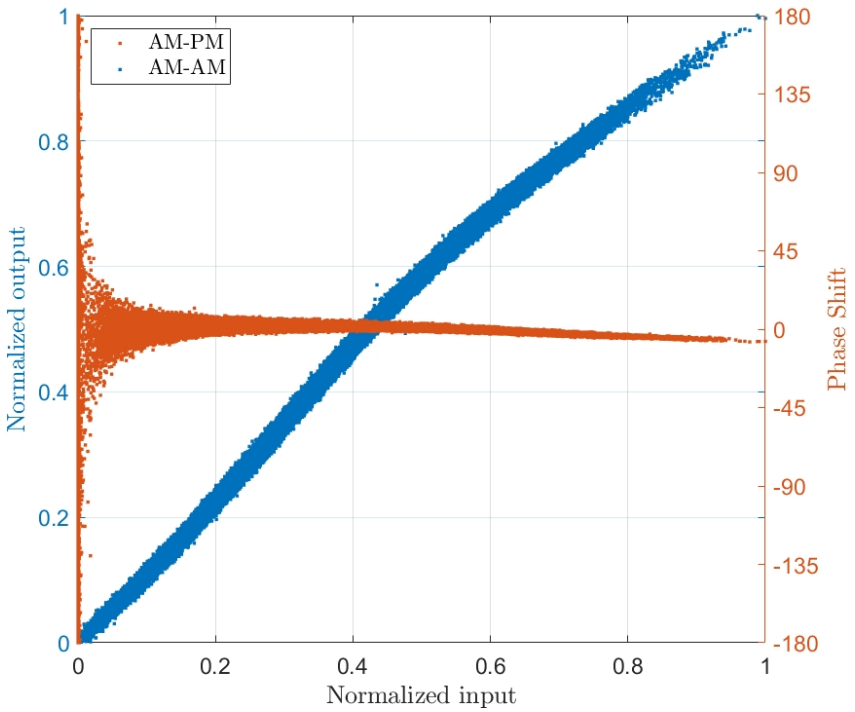


Figure 1.4: AM-AM and AM-PM characteristics of a Doherty PA

Ideally, the PA should have constant gain G over the entire operating range, where the output and input of PA, denoted by $y(n)$ and $x(n)$, respectively, are expressed by

$$y(n) = Gx(n) \tag{1.6}$$

But in reality, the gain is not constant due to the inherent nonlinearities in the PA, which are represented by the characteristic AM-AM [2]. On the other hand, the nonlinear effects, whatever its source (average power, bandwidth, polarization conditions, temperature, etc.), change the phase shift between the input and output signals. This is represented by the AM-PM characteristic.

1.3.3 1-dB Compression Point

The AM-AM and AM-PM characteristics are widely used to characterize the transfer function of PA. It can be seen that the gain is compressed by PA when the input power increases.

By definition, the 1 dB compression (P1dB) is the point where the gain of the PA is compressed by 1 dB against the small-signal gain of the linear region, as shown in Figure. 1.5. The points P_{in_C} and P_{out_C} represent the input and output power levels for which 1 dB of gain compression is obtained, respectively.

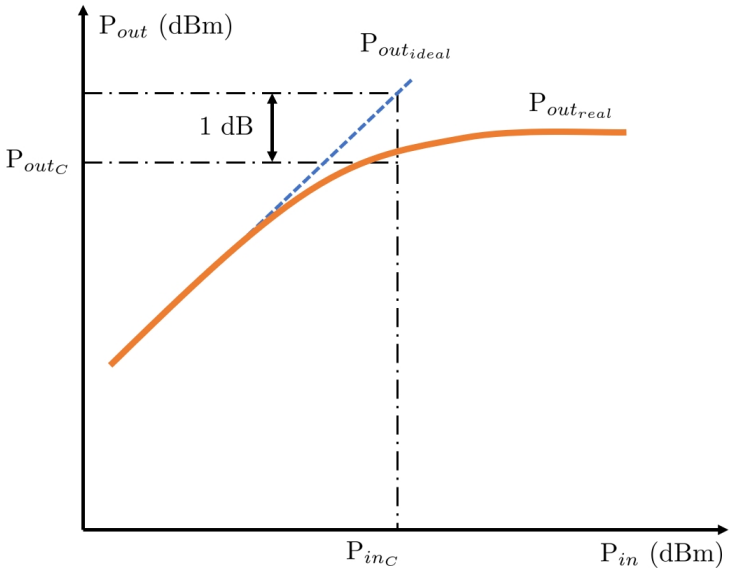


Figure 1.5: 1-dB compression point on AM-AM characteristic

1.3.4 Back-Off

To ensure reliable signal transmission and avoid saturation at the PA's output, it is necessary to keep the operating point backed off from the saturation.

This back-off can be adjusted according to the input or output power at saturation, denoted respectively by $P_{out_{sat}}$ or $P_{in_{sat}}$, by using two indicators: input back-off (IBO) and output back-off (OBO), which are illustrated in Figure 1.6.

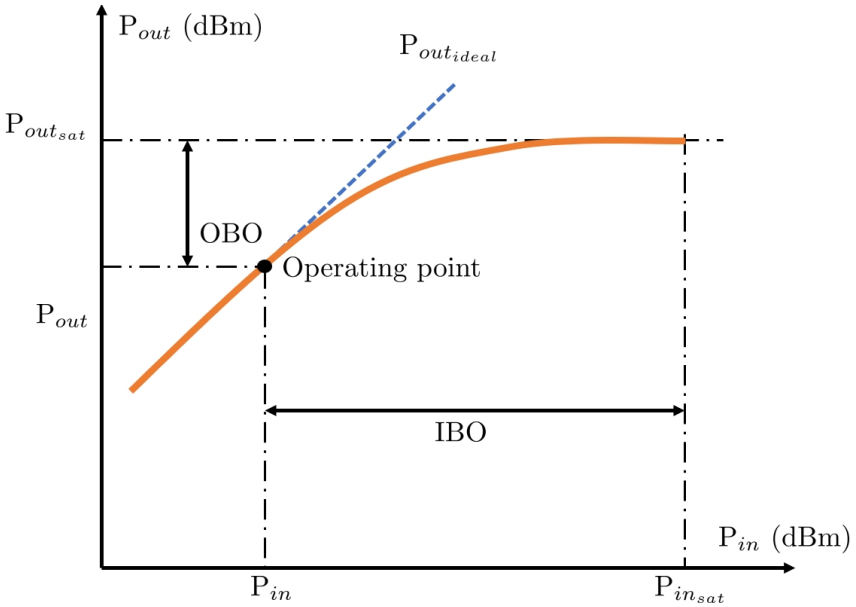


Figure 1.6: 1-dB compression point on AM-AM characteristic

The IBO is defined as the ratio between $P_{in_{sat}}$ and the mean input power P_{in} of the signal and is expressed in dB by

$$IBO_{dB} = P_{in_{sat}}(dBm) - P_{in}(dBm) \tag{1.7}$$

The OBO is defined as the ratio between $P_{out_{sat}}$ and the mean input power P_{out} of the signal and is expressed in dB by

$$OBO_{dB} = P_{out_{sat}}(dBm) - P_{out}(dBm) \tag{1.8}$$

1.3.5 Harmonics and Intermodulation Products

The nonlinear behavior can be analyzed by exciting the PA by signals of different types: continuous wave (CW) signal, two-tone signal and modulated signal, while monitoring saturation power, gain, 1-dB compression point, and intermodulation distortions.

1.3.5.1 CW Signal Test

Stimulating the PA by a CW signal is a typical test that aims to evaluate the PA characteristics in terms of linearity and power efficiency. As shown in Figure 1.7, the CW signal test consists of exiting the PA with a purely sinusoidal signal

$$x(t) = A\cos(2\pi f_0 t)$$

with an amplitude A at a frequency f_0.

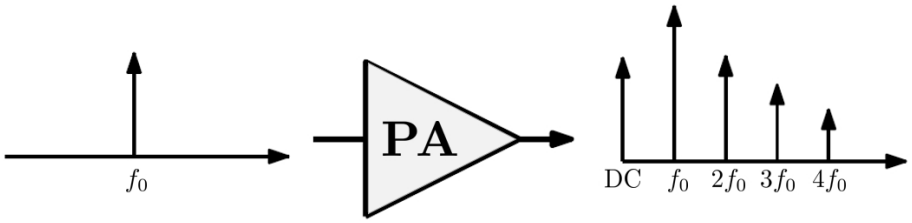


Figure 1.7: PA driven by CW signal

The behavior of the PA in the time domain shows a distortion of the signal at the PA output, which results in the frequency domain by the generation of new frequency components depending on the fundamental frequency f_0: ($2f_0$, $3f_0$, $4f_0$, ...), called harmonic components. The higher the power of these harmonic components, the stronger PA's nonlinearity.

The CW signal test appears to be a good approach to characterize the fundamental effects of the PA nonlinearities. However, this test remains limited for current communication systems and does not make it possible to further characterize the nonlinearity effects for modulated signals.

1.3.5.2 Two-Tone Signal Test

The two-tone signal test is widely used method for more relevant PA characterization and to have better description of the nonlinear behavior of the PA.

The two-tone signal is composed of two closely-spaced sinusoidal frequencies with amplitude A, and frequencies f_1 and f_2:

$$x(t) = A\cos(2\pi f_1 t) + A\cos(2\pi f_2 t) = 2A\cos(2\pi f_m t)\cos(2\pi f_c t)$$

where $f_m = \frac{f_2 - f_1}{2}$ and $f_c = \frac{f_2 + f_1}{2}$.

By driving the PA by the two-tone signal as shown in Figure 1.8, the output contains the frequency harmonics ($2f_1$, $2f_2$, $3f_1$, $3f_2$, ...) and additional frequency components, called intermodulation (IMD) products, generated at frequencies $mf_1 + nf_2$ where m and n are positive or negative integer and $|m|+|n| = k$ where k is the order of intermodulation.

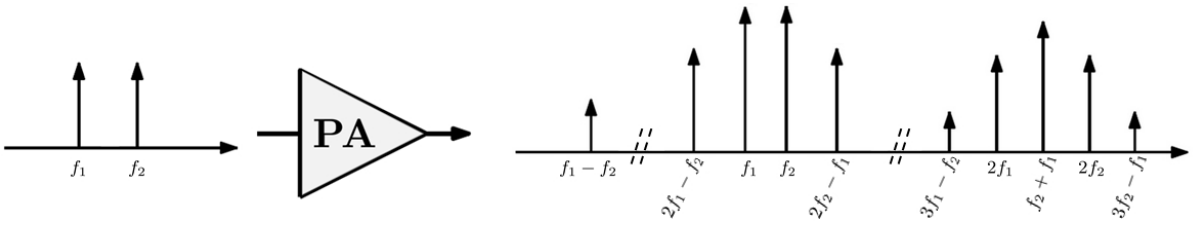


Figure 1.8: PA driven by two-tone signal

The odd-order terms of the IMD product ($k = 3, 5, 7, \dots$) are the most likely to disturb the amplified signal since they are close to the PA's useful band. In general, the 3^{rd} and 5^{th} IMD products have the highest power.

In order to quantify the spectral regrowth at the output of a PA driven by a two-tone, we use a metric called the carrier to inter-modulation distortion ratio (C/IMD), which represents the ratio in a linear scale between the power at the fundamental frequencies $f_{1,2}$ and the power generated at inter-modulation frequencies $mf_1 + nf_2$, as shown in Figure 1.9.

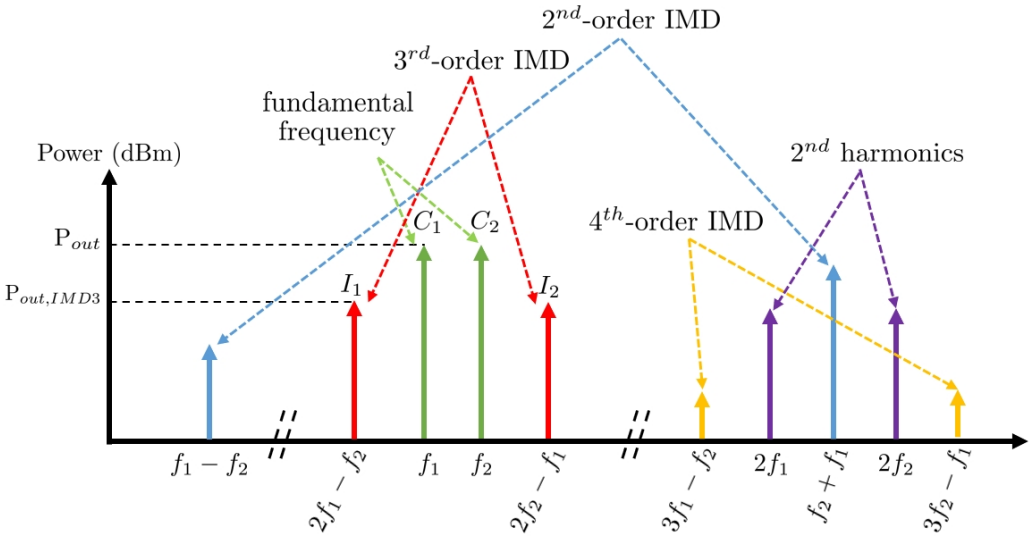


Figure 1.9: IMD products and harmonics of PA driven by two-tone signal

The two-tone test allows a standard evaluation of distortions by quantifying the nonlinear behavior of the PA. However, it is insufficient to represent the nonlinear behavior when the PA is driven by a real complex modulated signal used for communication systems.

1.3.6 Power Efficiency

An efficient PA aims to deliver a certain amount of power to the load without consuming too much power itself [3].

However, the actual DC power consumption, P_{DC} is always more extensive than the output power P_{out}. The PA can dissipate a considerable amount of energy in the form of waste heat. Two parameters are commonly used to characterize the PA efficiency:

- Drain efficiency which is defined as the ratio between the output power P_{out} and the DC power P_{DC}:

$$\eta_{DC} = \frac{P_{out}}{P_{DC}} \tag{1.9}$$

- Power added efficiency (PAE) which takes into account the PA input power P_{in}, and is defined as:

$$\eta_{PAE} = \frac{P_{out} - P_{in}}{P_{DC}} \tag{1.10}$$

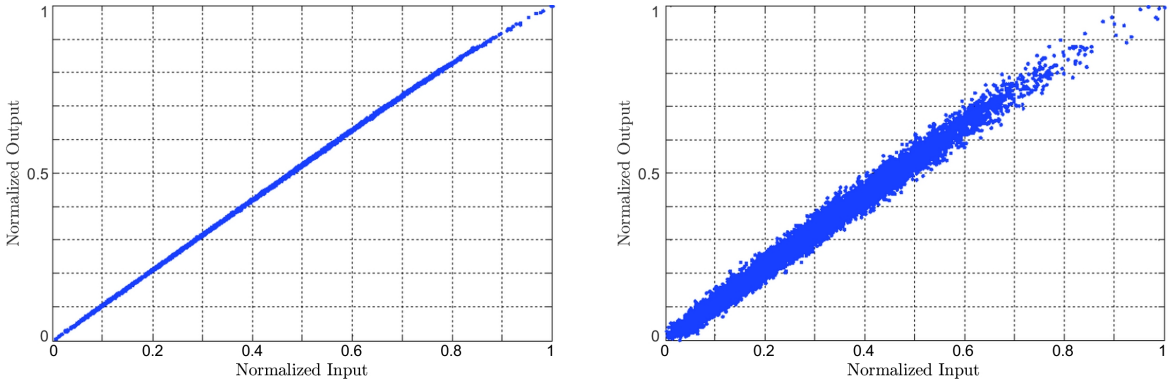
1.3.7 Memory Effect

In some instances, the PA is characterized by considering only its static characteristics.

Mainly, the PA exhibits nonlinear and dynamic dispersive phenomena called memory effects, which means that the system's output at any instants depends not only on the corresponding instantaneous input but also on the inputs at other previous instants. Therefore, the characteristics change with the frequency.

In narrowband wireless communication systems, the PA can be considered memoryless or quasi-memoryless. Wideband wireless communication systems need to identify whether the PA has a memory effect or not [4].

Figure 1.10 presents the AM-AM characteristic of PA excited by a two-tone signal with a bandwidth of 1 MHz and 8 MHz. As shown in Figure 1.10b, the AM-AM characteristic for the wideband scenario exhibits strong dispersion.



(a) Bandwidth of 1 MHz

(b) Bandwidth of 8 MHz

Figure 1.10: Memory effect on AM-AM

For PA design, substantial knowledge on the origins of memory effect and the accurate approaches to measure and quantify their impacts are required [5]. The origin of the memory effect can be either thermal or electrical [6].

The memory effect could be categorized into two types: short-term memory effect, also known as high-frequency (HF) memory effect, and long-term memory effect, also known as low-frequency (LF) memory effect.

The HF memory effects are caused by short time constants phenomena and depend on matching circuits and active components. The LF memory effects are induced by the slow variations of the signal, i.e., the same order of magnitude as the inverse of the envelope bandwidth. These undesirable effects are due to the components' self-heating, the trapping effect due to specific technologies such as GaN, value of the input and/or output load of the transistor at the envelope frequency, automatic gain control, etc.

1.4 Figures of Merit

Stimulus signal may experience substantial distortions due to the transceiver chain's non-linear components, such as the PA, and other imperfections such as the IQ imbalance, oscillator phase noise, and sampling jitter.

As a result of the nonlinearities, in-band and out-of-band distortions are generated. The main effects are the cloud-like shape of constellation points and the out-of-band radiation. The error vector magnitude (EVM) and the adjacent channel power ratio (ACPR) are mainly used to evaluate the distortion introduced by PA [7]. For high PA nonlinearities with many carriers, noise power ratio (NRP) can be used to assess performance.

1.4.1 Adjacent Channel Power Ratio

The imperfections and nonlinearities of PA usually result in some adjacent channel spectral regrowth.

This phenomenon can be quantified with ACPR defined as a bandwidth-limited ratio between the power in the main channel and the power in the adjacent channel as illustrated in Figure 1.11.

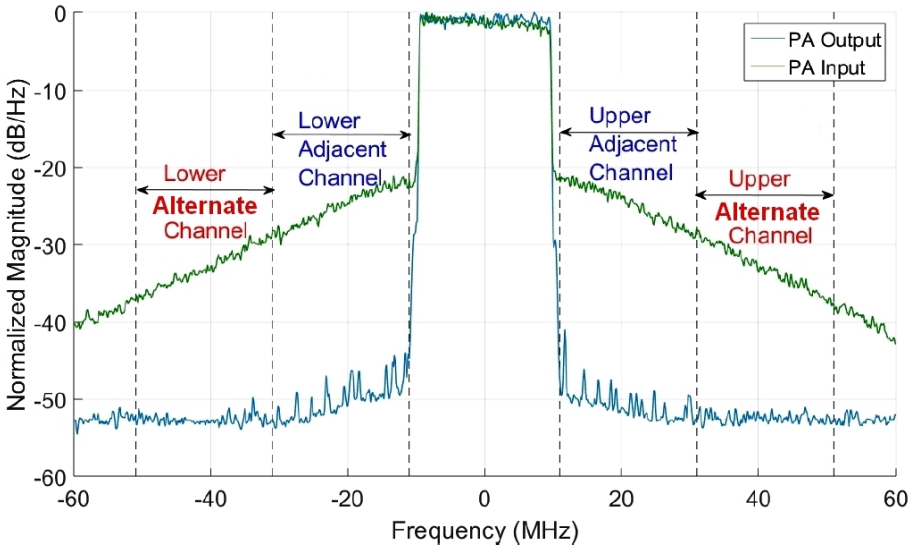


Figure 1.11: Upper and lower adjacent channels

Therefore, this property can be defined for lower (left) and upper (right) adjacent channels, which are expressed as:

$$\begin{aligned} \mathrm{ACPR}_{U,dB} &= 10\log_{10}\frac{\int_{-B/2}^{B/2} P(y(t))df}{\int_{B/2}^{3B/2} P(y(t))df} \\ \mathrm{ACPR}_{L,dB} &= 10\log_{10}\frac{\int_{-B/2}^{B/2} P(y(t))df}{\int_{-3B/2}^{-B/2} P(y(t))df} \end{aligned} \tag{1.11}$$

where B represents the bandwidth of the signal and $P(.)$ is power spectral density.

1.4.2 Error Vector Magnitude

The EVM is a metric that measures the in-band distortion level of the PA driven by a modulated signal. It is defined in the constellation domain and evaluates the deviation between the reference constellation point and the actual constellation point obtained in the presence of distortions.

Analytically, EVM is defined as:

$$\mathrm{EVM}_{\%} = \sqrt{\frac{\frac{1}{N}\sum_{j=0}^{N-1}(\delta I_j^2 + \delta Q_j^2)}{S_{\mathrm{avg}}^2}} \times 100\% \tag{1.12}$$

Where δI and δQ are errors magnitude corresponding to in-phase symbol and quadrature symbol of received data compared with an ideally reconstructed constellation respectively, N is the number of symbols, S^2_{avg} is the average square magnitude. The offset and the rotation of the constellation can also be considered in the definition of EVM.

Figure 1.12 shows an example of a 64-QAM signal constellation from Doherty PA operating at 3.1 GHz. The blue points present the constellation of the input signal. The red points present the constellation of the output signal after demodulation. In this case, the EVM is 10 %.

Some measurements (IQ data) from real PA were collected in this dissertation without any demodulation information. For these waveforms, the value of EVM, denoted by EVM_{app}, has approximately been estimated using NMSE in-band.

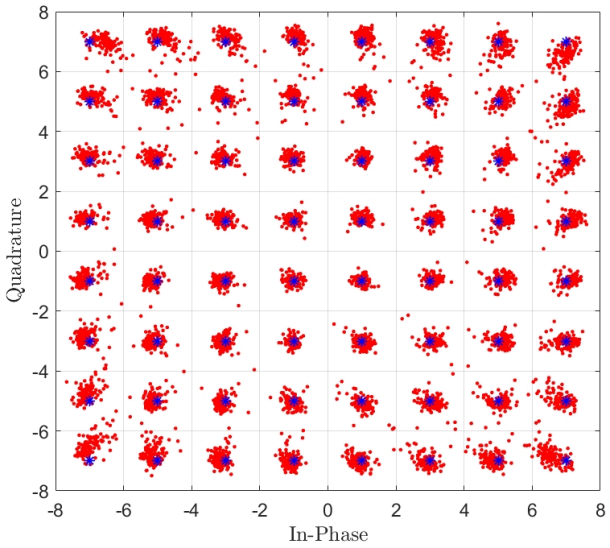


Figure 1.12: Constellation of input-output IQ data with EVM=10%

1.4.3 Noise Power Ratio

Noise power ratio (NPR) is another possible measurement to characterize nonlinear distortions. As EVM it is used to evaluate in-band distortion.

In the NPR measurement procedure, multi-carrier signals with random amplitudes and phases (such as OFDM) will be simulated by an additive white Gaussian noise. A large number of tones with different frequencies with the same magnitudes and random phases with a uniform distribution can generate a Gaussian white noise [8], which is filtered and passed through a narrowband-reject filter to produce a deep notch in the noise pedestal center, then stimulate the PA and produce the IMD product.

The PA nonlinearity will result in the generation of a spurious signal in the notch. The NPR will then be used to evaluate the ratio between the average power of the signal outside the notch and the intermodulation noise in the notch. It is expressed as follows:

$$\text{NRP}_{dB} = 10\log_{10}\Big(\frac{2\int_{B_{notch}} P(y)df}{2\int_{B_{carrier}} P(y)df}\frac{B_{carrier}}{B_{notch}}\Big) \tag{1.13}$$

where B represents the bandwidth and $P(.)$ is power spectral density.

The EVM and NPR are considered practical measurements for quantifying the in-band distortion of the PA. However, NPR measurement does not require transmitted signal demodulation, unlike EVM [9].

1.5 PAPR Reduction Techniques

To improve the PA efficiency, an intuitive solution is to reduce the PAPR of the input signal. PAPR reduction techniques, commonly known as Crest Factor Reduction (CFR), find their interest in improving the PA's efficiency.

Several PAPR reduction techniques have been proposed in the literature. As described in [10], PAPR reduction techniques can be classified into three top categories: coding methods, probabilistic methods, and adding signal methods.

In order to evaluate the performance of every technique, the PAPR reduction is the most significant criterion. However, other metrics could be used to quantify the effectiveness of the approach used to reduce PAPR.

- **Average power variation:** Some PAPR reduction techniques result in a decrease or increase of the transmitted signal's average power. In the literature, most PAPR reduction techniques performance studies do not take into account the average power variation of the transmitted signal [11]. This variation has a strong impact on the quality of the transmission.
- **In and out of band distortions:** Some PAPR reduction techniques introduce in-band and/or out-of-band distortions because of their nonlinear operation. EVM and ACPR are used to evaluate these distortions.
- **Downward compatibility:** A PAPR reduction technique is said to be downward compatible if it does not involve any change on the receiving side. This is the case of tone reservation and clipping technique. The coding methods are not downward compatible as they require post-processing on the receiver side.
- **Data rate loss:** For some methods, the receiver needs additional information to recover useful transmitted data. These methods need an increase in the bandwidth and consequently a decrease in the spectral efficiency. If the bandwidth has to be kept constant, this information transmission involves a data rate loss. This is the case of the selective mapping technique.

1.5.1 Coding Methods

The coding methods consist in reducing the occurrence probability of the same phase value of signals which add up with the same phase. A simple block coding scheme was introduced in [12], and it consists of finding out all possible codewords and then select those codewords of the lowest PAPR. It has been shown that using this technique, the PAPR of the signal can be reduced by 4 dB.

In [13] and [14], the authors used the Golay complementary sequences where more than 3 dB PAPR reduction has been obtained. However, the coding methods is limited by two constraints. The first requires a brute-force approach to find the best appropriate code-word. The second is the complexity of storing in LUTs for encoding and decoding in the transmitter and receiver.

In summary, the actual benefits of coding for PAPR reduction for practical multi-carrier systems are limited, regarding the low coding rate, the intractable required search for a good code, and the prohibitive complexity for many sub-carriers.

1.5.2 Probabilistic Methods

The idea behind the probabilistic methods is to perform several copies of the initial signal by modifying the phase, amplitude, and/or position of sub-carriers and then select the copy with the minimum PAPR.

These methods cannot guarantee the PAPR below a specified level. Moreover, it decreases the spectral efficiency, and the computational complexity increases as the number of sub-carriers increases. The probabilistic methods include selective mapping (SLM) technique and partial transmit sequence (PTS) [15]. The block diagram of SLM technique is shown in Figure 1.13.

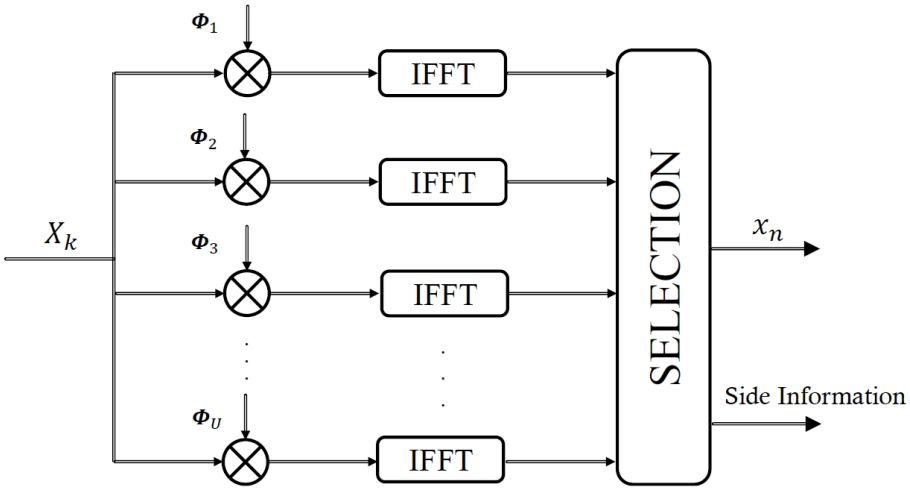


Figure 1.13: Selective mapping technique

In the SLM technique, the input data sequences are multiplied by U different phase sequences to generate alternative input symbol sequences. Each of these alternative input data sequences are then applied to the IFFT operation, and then the one with the lowest PAPR is selected for transmission [16].

Therefore, its performance in reducing the PAPR depends on the number and phase factors design. The corresponding selected phase factor also needs to be transmitted to the receiver as side information to extract the original information properly.

1.5.3 Adding Signal Methods

This category, as its name suggests, includes all techniques of PAPR reduction that can be formulated as:

$$\text{PAPR}(X + C_{\text{papr}}) < \text{PAPR}(X) \tag{1.14}$$

where X refers to the waveform signal and C_{papr} refers to the peak-reduction signal.

1.5.3.1 Clipping

Clipping is one of the simplest techniques to reduce the PAPR [17], where the amplitude peak of a given signal $x(n)$ is clipped according to a certain threshold A_{max}.

$$x(t) = \begin{cases} |x(t)|e^{j\phi(x)} & \text{if } |x(t)| \leq A_{\max} \\ A_{\max}e^{j\phi(x)} & \text{if } |x(t)| > A_{\max} \end{cases} \tag{1.15}$$

This technique generates some unwanted distortions in in-band and out-band, which degrades the system performance, including bit error rate (BER) and spectral efficiency.

1.5.3.2 Tone reservation

Tone reservation for multi-carrier signals proposed in [18] is based on the reservation of sub-carriers that do not carry any useful information, in order to generate the PAPR reduction signal that lowers its peaks when added to the original multi-carrier signal.

1.5.3.3 Active Constellation Extension

Active constellation extension introduced in [19] reduces the PAPR by appropriately extending the outer points of the signal constellations in the frequency domain. The constellation modifications appropriately cancel the time domain peaks of the transmitted signal. This is achieved without any degradation in BER performance and data rate. However, this comes at the cost of a slight increase in the average power of the transmitted signal.

1.6 Linearization Techniques

Many linearization techniques compensate for PA's distortion, such as Feedforward, Feedback, and Predistortion.

1.6.1 Back-Off

The simplest method to deal with PA nonlinearity is to operate the PA with a significant power back-off from its saturation point to achieve linear amplification. This implies that the maximum input power level must be capped so that the entire signal remains within the linear region of the PA. This solution is not exactly a linearization technique. However, it is often used because of its simplicity of realization, but at the cost of poor energy efficiency.

1.6.2 Feedforward

The principle is to extract the error signal (distortions) at the output of PA and then subtract it from the output signal. Therefore, an input signal splits into two paths, as shown in Figure 1.14.

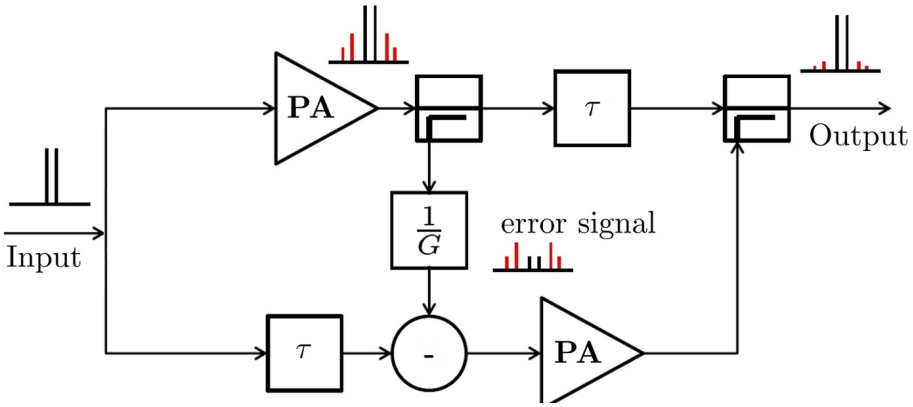


Figure 1.14: Principle of Feedforward

The first path consists of the main PA and time delay with some coupling elements. The second one amplifies the error signal that is the difference between the output of the first PA aligned in time and amplitude with the input signal and the time-delayed input. A coupler subtracts the amplified error signal from the PA's time-delayed distorted output at the output. Thus, the distortions in the out-band are reduced.

1.6.3 Feedback

The output from the feedbacked amplifier is subtracted from the input signal with a specific attenuation. Therefore, the PA is driven by the error signal. The advantage is the

relative simplicity of additional circuitry needed to provide feedback. On the contrary, the main disadvantage of feedback is caused by problems of stability and bandwidth limitation. Feedback linearization is classified into two main groups: RF Feedback and Cartesian Feedback. The basic layout of RF Feedback linearization can be seen in Figure 1.15.

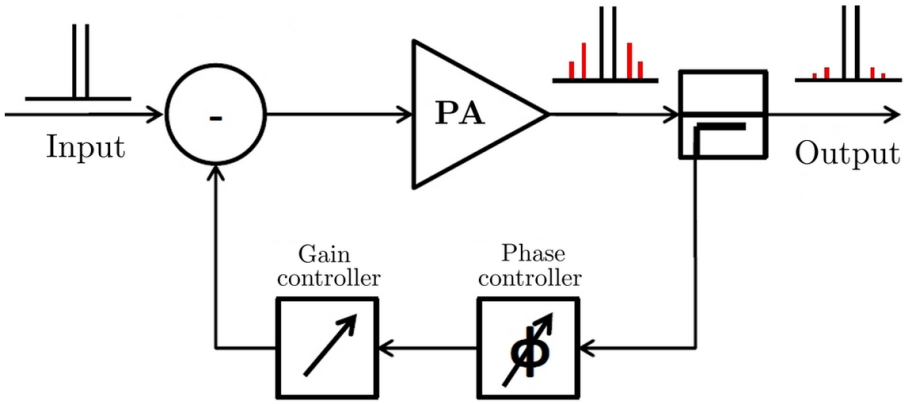


Figure 1.15: Principle of Feedback

The performance of Feedback linearization depends on the quality of the cancellation process linked to the gain loop. The method can only improve the linearity for narrowband signals due to the limitation of the loop-bandwidth product guaranteeing stability.

1.6.4 Predistortion

Predistortion (PD) is an efficient technique for linearizing the PAs. It introduces a module, called predistorter, upstream of the PA, which compensates for the PA's nonlinearities. The predistortion does not require an in-depth knowledge of the PA's physic circuits, and we only need to precisely measure the PA's actual behaviors for the predistortion procedure [20].

The first PD systems for wireless communications were proposed by Nagata [21] in 1989, Cavers [22] - [23] in 1990 and Wright [24] in 1992.

The implementation of PD can be done in analog or digital domain, but currently, the most common implementation is digital, applied to the baseband signal [25] - [26].

In the analog domain, the analog PD is performed at the carrier RF frequency, at the IF frequency, or at the baseband level after the DAC, using analog circuits, based on diodes or transistors [27] - [28]. The implementation of analog PD has been proposed in [29]. In [30], the authors propose an approach where the modulated signal is predistorted. Furthermore, they also proposed an approach to make this analog predistortion adaptive. The analog PD circuits are generally small, facilitating their integration into a power amplification system with reduced power consumption. However, linearity is limited because the elimination of intermodulation is generally sufficient only for third-order IMD. Moreover, the drifts due to the aging of the PD's analog components and the PA cannot be corrected.

When speaking of analog PD, analog processing may be less power-consuming than its digital counterpart. However, it remains limited to reasonably simple predistortion functions like the memory polynomial and does not offer the same processing versatility as digital processing.

Predistortion is widely implemented in the digital domain, which offers better linearization performance. The digital predistortion (DPD) process is done by hardware,

using a programmable logic device (FPGA, ASIC, DSP). Besides, the predistortion system can be adaptive by adding a feedback path to the system. This path takes a part of the PA output in order to update the predistortion block. The DPD is independent of the carrier frequency, which is a property of flexibility, and it remains helpful in a multi-mode or multi-band context, thanks to adaptability.

In 4G and 5G wireless communication systems, very high data rates are achieved thanks to carrier aggregation generating signals with very high bandwidths for which DPD may be challenging to implement (it would require too expensive DAC and ADC converters and high computation load).

On the other hand, the DPD is suitable for wideband applications. Furthermore, it is highly flexible and could be reconfigurable. However, the variation on PA characteristics due to the variation of component tolerances and the drift of temperature should be considered. The adaptation is required.

The core of this dissertation focuses on the PA linearization using the digital predistortion, which will be detailed in the following chapter.

1.7 Conclusion

This chapter has provided an overview of the PA and its characteristics and the distortions that are introduced. The main metrics that evaluate these distortions have been presented. Two main aspects concerning PAs, which are linearity and power efficiency, are discussed. The techniques to improve each aspect are discussed, as well.

The main contributions of these dissertations are thoroughly presented in the following chapters.

Chapter 2

Analysis and Comparative Study of Digital Predistortion Models

2.1 Introduction

The concern to linearize the PA dates back to the beginning of broadcasting and the expansion of telecommunications in the last century [31]. As the transmissions shift to communication with higher spectral efficiency waveforms such as M-QAM, the requirement for linearity becomes more stringent and opens the way for DPD to be developed and implemented. Even if the main goal of DPD is to linearize the power amplifier, it also contributes, in many cases, to improve the power efficiency, which is vital since PAs are responsible for a significant part of the power consumption in base stations of wireless cellular networks [32].

This chapter mainly focuses on the principle of DPD, its architectures, and the different DPD models. The comparison of these models, which presents the first contribution of this dissertation, is achieved in different scenarios and for different types of PA.

2.2 Principle of Digital Predistortion

As mentioned in Section 1.6.4, DPD is an efficient linearization technique of PAs. It consists of generating signal components of inverted proportional amplitude and opposite phase to the distortion products, in order to have a constant complex gain over the entire operating power range of the linearized amplifier, so that the whole system operates as a linear amplification system, as illustrated in Figure 2.1.

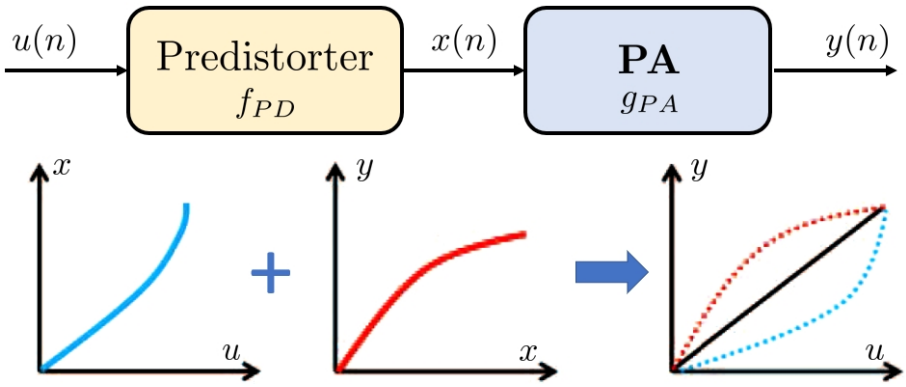


Figure 2.1: Principle of digital predistortion

The characteristic of the predistortion operator f_{PD}, called predistorter, is the inverse

of that of the PA. For example, for a memoryless PA represented by g_{PA}, the linear amplification system for memoryless systems can be expressed by:

$$g_{\mathrm{PA}}(f_{\mathrm{PD}}(u)) = Gu \tag{2.1}$$

where G is the gain of the system.

The strong inverse match between the nonlinear characteristics of the predistorter and that of the PA leads to an effective DPD to cancel the distortions of the PA. Besides, it is required from DPD to achieve high linearization performances and low cost.

The DPD does not require a deep knowledge of the physical circuits of PA [6], but it is necessary to precisely measure the actual behavior of PA for the DPD process [20].

On the other hand, the DPD needs to be made adaptive to maintain a linear amplification system, particularly when the input signal changes characteristics (power level, frequency, and bandwidth) as well as the PA (variation of component tolerances and temperature drift).

Figure 2.2 illustrates the principle of baseband adaptive DPD. The DPD requires a reference signal for its adaption, which could come from path 1 or 2, depending on the approach used [33]. The feedback signal should be acquired of good quality since it is used for correction and synchronization. For this, it is advisable to use the mixers at the lowest possible power level. The DAC and ADC converters are essential components in the blog diagram of Figure 2.2. Their power consumption is not negligible (especially for wideband signals) and can degrade the efficiency, especially if the PA output power is not very high.

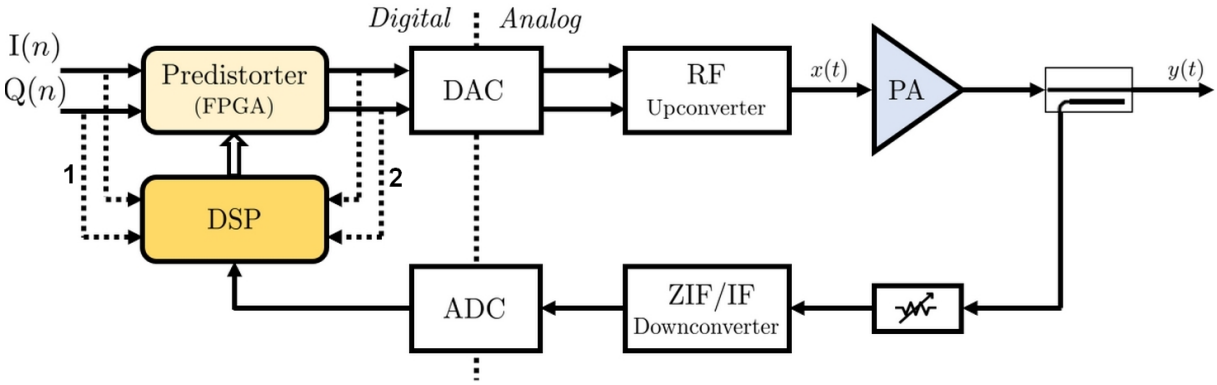


Figure 2.2: Baseband adaptive DPD with PA

2.3 Digital Predistorter Models

The aim of studying PA distortions is to design an appropriate predistorter that will compensate for these distortions and ensure linear amplification of the signal to be transmitted.

Several mathematical models have been proposed in the literature to model the PA and serve as well as predistorter. Most of them are based on Volterra Series [34].

2.3.1 Volterra Series

Volterra series are widely used to represent the input-output relationship of nonlinear dynamical systems. They are formed by a combination of linear convolution and a nonlinear power series so that it can be used to characterize a nonlinear dynamic system accurately [35]. To model the RF behavioral system, the Volterra series can be expressed as:

$$\tilde{y}(t) = \sum_{k=1}^{+\infty} \int_0^{+\infty} \cdots \int_0^{+\infty} \tilde{h}_k(\tau_1, .., \tau_k) \prod_{j=1}^{k} \tilde{x}(t-\tau_j) d\tau_j \tag{2.2}$$

where $\tilde{x}$ and $\tilde{y}$ represent the RF input and RF output signal, respectively, $\tilde{h}_k(\cdot)$ is the real-valued k-th order Volterra kernel.

In baseband, the complex envelope of the signal can be obtained by a low-pass filter. Hence, the relationship between the baseband input and output in discrete time is:

$$y(n) = \sum_{k=0}^{K} \sum_{i_1=0}^{M-1} ... \sum_{i_{2k+1}=i_{2k}}^{M-1} h_k(i_1, ..., i_k) \prod_{j=1}^{k+1} x(n-i_j) \prod_{j=k+2}^{2k+1} x^*(n-i_j) \tag{2.3}$$

where $x(n)$ and $y(n)$ represent the baseband complex input and output, respectively, $h_p(i_1, ...i_p)$ is called the Volterra series kernel, K is the nonlinearity order, and M is the memory depth.

The Volterra series model is robust and can provide greater modeling performance, but it is very complex and slowly converge. Unfortunately, the main drawback of Volterra series is that the number of coefficients increases exponentially with the order of nonlinearity and the memory depth. This drawback makes the Volterra series unattractive for real-time applications. This leads to use particular cases of the Volterra series, which retain only a few terms of these series.

2.3.2 Pruning of Volterra Series

In [34], the authors proposed an efficient approach to pruning of the Volterra series according the physical properties of the PA, which significantly reduces model complexity by removing unnecessary coefficients from the general Volterra series.

The memory polynomial (MP) model and the Generalized Memory Polynomial (GMP) model are the most popular models obtained by direct pruning of the Volterra series.

The MP model proposed in [37] is obtained by reducing the Volterra series model to its diagonal terms. The input-output relationship for the MP can be written as:

$$y(n) = \sum_{k=1}^{K} \sum_{i=0}^{M} a_{ki} x(n-i) |x(n-i)|^{k-1} \tag{2.4}$$

where a_{ki} represent the coefficients of the model, and K and M are the nonlinearity order and memory depth, respectively.

When only odd degrees are used, (2.4) can be rewritten :

$$y(n) = \sum_{k=0}^{K} \sum_{i=0}^{M} a_{2k+1,i} x(n-i) |x(n-i)|^{2k} \tag{2.5}$$

The GMP model is built by augmenting the memory polynomial model with additional basis functions, introducing cross-terms. It is proposed in [38] to better deal with PAs with strong nonlinearities and wideband signals. The GMP model can be written as:

$$\begin{aligned} y(n) = & \sum_{k=0}^{K_a-1} \sum_{l=0}^{L_a-1} a_{kl} x(n-l) |x(n-l)|^k \\ & + \sum_{k=1}^{K_b} \sum_{l=0}^{L_b-1} \sum_{m=1}^{M_b} b_{klm} x(n-l) |x(n-l-m)|^k \\ & + \sum_{k=1}^{K_c} \sum_{l=0}^{L_c-1} \sum_{m=1}^{M_c} c_{klm} x(n-l) |x(n-l+m)|^k \end{aligned} \tag{2.6}$$

where K_a, K_b, K_c are the highest orders of nonlinearity. L_a, L_b, L_c are the highest memory depths. M_b, M_c denote the longest lagging and leading delay tap length, respectively. a_{kl}, b_{klm}, c_{klm} are the coefficients of the general memory polynomial applied on the aligned terms, the lagging and leading cross-term.

2.3.3 Modified or Dynamic Volterra Series

To overcome the complexity of Volterra series, several models have been proposed from modified or dynamic Volterra series. In [39], the authors limit the number of coefficients by separating the static and dynamic parts to model the PA. In [40], the author proposed a derived model from the dynamic Volterra series, called dynamic deviation reduction (DDR).

The DDR model can be further simplified as [41] and [42]. The 1st-order truncated DDR-1 is defined as:

$$\begin{aligned} y(n) = &\sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} b_{2k+1,1} |x(n)|^{2k} x(n-i) \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,2} |x(n)|^{2(k-1)} x^2(n) x^*(n-i) \end{aligned} \tag{2.7}$$

where $b_{2k+1,j}$ is the kernel of the system. A more complex DDR model can be derived by adding some 2nd-order dynamic terms:

$$\begin{aligned} y(n) = &\sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} b_{2k+1,1} |x(n)|^{2k} x(n-i) \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,2} |x(n)|^{2(k-1)} x^2(n) x^*(n-i) \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,3} |x(n)|^{2(k-1)} x(n) |x(n-i)|^2 \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,4} |x(n)|^{2(k-1)} x^*(n) x^2(n-i) \end{aligned} \tag{2.8}$$

The MP model, which contains only the diagonal terms, has limited performance because some off-diagonal terms may be significant. Compared with the MP model, the DDR model includes off-diagonal terms, but has fewer diagonal terms. The Modified Dynamic Deviation Reduction-based Volterra (MDDRV) proposed in [43] and [44] can be viewed as the combining of the MP model and the DDR. By combining (2.4) and (2.7), the first-order MDDRV-1 model can be expressed as:

$$\begin{aligned} y(n) = &\sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} a_{2k+1,1} |x(n)|^{2k} x(n-i) \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,2} |x(n)|^{2(k-1)} x^2(n) x^*(n-i) \\ &+ \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1} |x(n-i)|^{2k} x(n-i) \end{aligned} \tag{2.9}$$

where $a_{2k+1,j}$ and b_{2k+1} are the model coefficients. The simplified second order MDDRV-2 model can be expressed, by combining (2.4) and (2.8), as:

$$\begin{aligned} y(n) = & \sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} a_{2k+1,1}|x(n)|^{2k}x(n-i) \\ & + \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,2}|x(n)|^{2(k-1)}x^2(n)x^*(n-i) \\ & + \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,3}|x(n)|^{2(k-1)}x(n)|x(n-i)|^2 \\ & + \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,4}|x(n)|^{2(k-1)}x^*(n)x^2(n-i) \\ & + \sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1}|x(n-i)|^{2k}x(n-i) \end{aligned} \tag{2.10}$$

2.3.4 Models with Segmentation

Another approach to representing a nonlinear dynamic system is splitting the amplitude range into different regions or segments characterized by piecewise approaches models, such as models using B-splines approximation, vector-switched models, decomposed vector rotation, etc.

These approaches are able to represent strong nonlinearities and may be less sensitive to numerical problems than global polynomial models. Furthermore, these models are based on segments of smaller orders of nonlinearity, which can be identified with short buffers of training data.

In [45], Chua proposed the first compact representation for modeling a nonlinear function by line segments, called canonical piecewise linear (CPWL) functions. The nonlinear basis functions of CPWL are constructed piecewise, making it much more flexible than the Volterra models. The CPWL is expressed as:

$$y(n) = \sum_{i=0}^{M} a_i x(n-i) + \sum_{k=1}^{K} c_k \left| \sum_{i=0}^{M} a_{k,i}x(n-i) - \beta_k \right| \tag{2.11}$$

where K is the number of segments that are bounded by the thresholds β_k. M is the memory depth and a_i, $c_{k,i}$ and $a_{k,i}$ are the coefficients.

The main drawback of CPWL in (2.11) is that the model is nonlinear with respect to its coefficients, which complicates the process of identifying the coefficients $c_{k,i}$ and $a_{k,i}$.

Zhu proposed in [46] an extension and modified form of CPWL that it called decomposed vector rotation (DVR) model. It is based on the vector rotation technique to handle complex signals and nonlinear systems with memory.

The extension from CPWL functions consists of changing the form $|x(n-i) - \beta_k|$ to $||x(n-i)| - \beta_k|e^{j\theta(n-i)}$ through an operation of 4 steps:

- Calculate the magnitude value of the input signal $x(n-i)$.
- Subtract away the threshold β_k.
- Apply an absolute operation $|.|$ to $|x(n-i)| - \beta_k$.
- Restore the phase $\theta(n-i)$ of the signal $x(n-i)$.

The representation $||x(n-i)|-\beta_k|e^{j\theta(n-i)}$ can be rewritten as:

$$||x(n-i)|-\beta_k|e^{j\theta(n-i)} = \begin{cases} x(n-i)-\beta_k e^{j\theta(n-i)}, & \text{if } |x(n-i)| \geq \beta_k \\ [x(n-i)-\beta_k e^{j\theta(n-i)}]e^{j180^\circ}, & \text{if } |x(n-i)| < \beta_k \end{cases} \tag{2.12}$$

where the phase rotation on $|x(n-i)-\beta_k|$ is applied if $|x(n-i)|$ is less than β_k, making of it the nonlinear process introduced into the DVR model.

Furthermore, the DVR model is linear with respect to its coefficients. The simplest version of the model, denoted DVR-2, is defined by combining the linear term and the basis function discussed below, which is expressed as:

$$y(n) = \sum_{i=0}^{M} a_i x(n-i) + \sum_{k=1}^{K}\sum_{i=0}^{M} c_{ki,1}||x(n-i)|-\beta_k|e^{j\theta(n-i)} \tag{2.13}$$

The DVR-2 model refers to 2 terms: the linear term and the 1st-order basis.

The DVR model can be enriched by higher-order extension and variations of basis functions (derived from DDR, for example). We will call DVR-6 model, the DVR model that can be derived by adding some DDR terms.

$$\begin{aligned} y_(n) = &\sum_{i=0}^{M} a_i x(n-i) \\ &+ \sum_{k=1}^{K}\sum_{i=0}^{M} c_{ki,1}||x(n-i)|-\beta_k|e^{j\theta(n-i)} \\ &+ \sum_{k=1}^{K}\sum_{i=0}^{M} c_{ki,21}||x(n-i)|-\beta_k|e^{j\theta(n-i)}.|x(n)| \\ &+ \sum_{k=1}^{K}\sum_{i=1}^{M} c_{ki,22}||x(n-i)|-\beta_k|.x(n) \\ &+ \sum_{k=1}^{K}\sum_{i=1}^{M} c_{ki,23}||x(n-i)|-\beta_k|.x(n-i) \\ &+ \sum_{k=1}^{K}\sum_{i=1}^{M} c_{ki,24}||x(n)|-\beta_k|.x(n-i) \end{aligned} \tag{2.14}$$

An important aspect concerning the DVR model is how the thresholds β_k are distributed. In this chapter, The DVR model is used with a uniform distribution of the thresholds, e.g., $\beta_k = k/K$ for $k = 1, 2, ..., K-1$.

In the next chapter, we will propose an original approach to determine the thresholds' optimal values and determine the terms to be used in the DVR model.

2.4 Predistorter Identification

An essential aspect of digital predistortion is the estimation of the digital predistorter model coefficients.

Several approaches has been proposed in the literature [47] - [48]. The two main techniques are direct learning architecture (DLA) and indirect learning architecture (ILA).

In the DLA, the predistorter is determined in two steps. In the first step, the coefficients of the nonlinear model for the PA are extracted. In the second step, the identified PA model is used to estimate the predistorter by minimizing a criterion based on the difference between the observed PA output and the ideal output, which is equal to the

original input signal multiplied by a reference gain. Several algorithms were proposed to improve the DLA [49].

Unlike DLA, ILA consists of estimating a postdistorter that will be used as a predistorter. Besides, if the model is linear in its coefficients, the ILA requires a linear estimation, while the DLA requires a nonlinear estimation.

In [50], a system-level convergence of DLA and ILA is investigated and compared. The MP model was used for both architectures to asses the robustness of the identification process, where it was shown that the DLA was found to be more robust than the ILA in the presence of noisy measurements.

2.5 Post-Distortion Identification

In this chapter, the DPD model coefficients are identified using postdistortion architecture that is based on ILA, but without copying coefficients to the predistorter model, unlike the ILA as is illustrated in Figure 2.3.

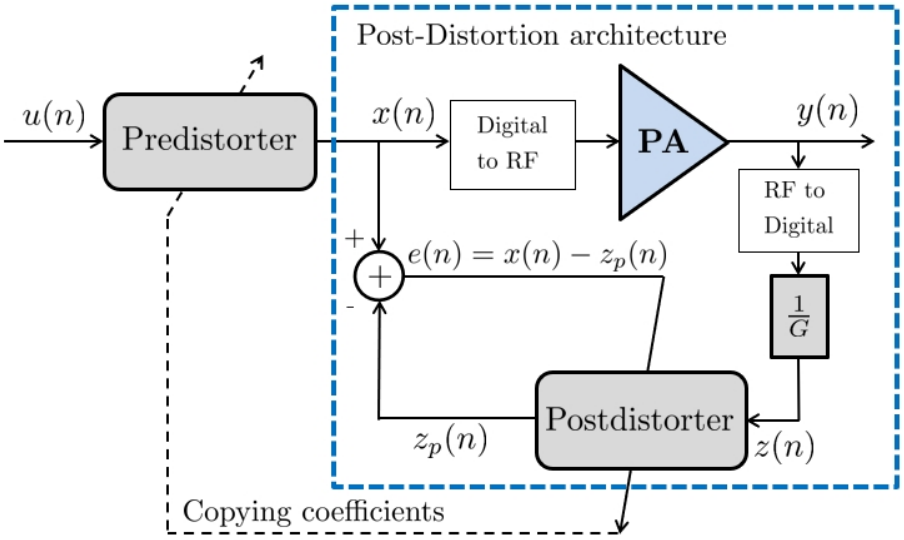


Figure 2.3: ILA and Post-Distortion architecture

In this architecture, only the input-output signals from PA are required to estimate the model coefficients. The principle of Post-Distortion architecture is illustrated in Figure 2.4.

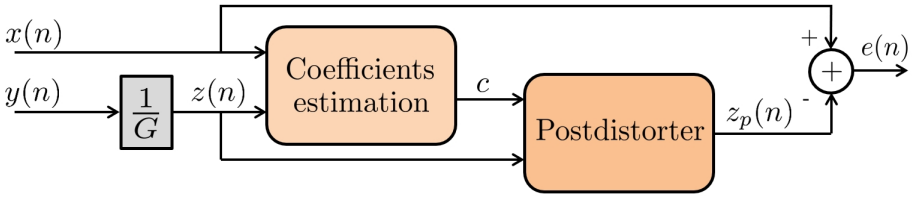


Figure 2.4: Principle of Post-Distortion architecture

The PA output signal $y(n)$ is divided by the amplification gain G. The signals $x(n)$ and $z(n)$ are used to estimate the model coefficients $\mathbf{c}$ using least square (LS) approach in order to minimize the LS criterion built on the difference between $x(n)$ and $z_p(n)$, the model output that is computed using the estimated coefficient $\mathbf{c}$ and $z(n)$. The instantaneous error is defined as:

$$e(n) = x(n) - z_p(n) \tag{2.15}$$

For the DPD models that are linear with respect to their coefficients, the relationship between its input and output can be rewritten using matrix notation. For a block of N samples:

$$\mathbf{z_p} = \mathbf{Zc} \tag{2.16}$$

where $z_p = [z_p(1), \ldots, z_p(N)]^T$, $\mathbf{c}$ is a $C \times 1$ vector containing the set of model coefficients, $\mathbf{Z}$ is $N \times C$ matrix of regressors containing basis functions of z. For example, in the case of a MP model, this matrix is represented as:

$$\mathbf{Z} = \begin{bmatrix} \Phi_{1,1}(z(n)) & \ldots & \Phi_{K,1}(z(n)) & \Phi_{1,2}(z(n)) & \cdots & \Phi_{K,L}(z(n)) \\ \Phi_{1,1}(z(n-1)) & \vdots & \ddots & & \vdots & \Phi_{K,L}(z(n-1)) \\ \vdots & \vdots & & & & \vdots \\ \Phi_{1,1}(z(n-N+1)) & \cdots & & & & \Phi_{K,L}(z(n-N+1)) \end{bmatrix} \tag{2.17}$$

where $\Phi_{k,l}(z(n)) = z(n-l+1)|z(n-l+1)|^{k-1}$.

The least square (LS) solution will be the solution for the following equation for $\mathbf{c}$:

$$[\mathbf{Z}^H\mathbf{Z}]\mathbf{c} = \mathbf{Z}^H\mathbf{x} \tag{2.18}$$

or equivalently using the pseudo-inverse:

$$\hat{\mathbf{c}} = [\mathbf{Z}^H\mathbf{Z}]^{-1}\mathbf{Z}^H\mathbf{x} \tag{2.19}$$

with $\mathbf{x}$ is the input signal, where (2.18) minimizes the LS criterion:

$$\hat{\mathbf{c}} = \min_{\mathbf{c}}(J) \tag{2.20}$$

with:

$$J = \sum_{n=1}^{N} |e(n)|^2 = \sum_{n=1}^{N} |x(n) - z_p(n)|^2$$

Many approaches can be used to solve (2.18) such as LU, LDL, QR decomposition, etc. It should be noticed that the matrix $\mathbf{Z}^H\mathbf{Z}$ is generally ill-conditioned.

2.6 Comparison of DPD models

Many comparative studies have been done to compare the linearization performance of DPD models [51] - [53]. In [53], the paper presented a comparative overview of various nonlinear behavioral approaches for PAs. In [54], the study focused on the linearization performance comparison of 4 DPD models: MP, GMP, Orthogonal Polynomials with Memory (OPM), and DDR.

In [55], a comparative study has been carried out of the MP, GMP, and DVR models for DPD to linearize the radio on fiber laser with distributed feedback. The comparison was made in terms of NMSE, EVM, and ACPR, where the authors show through experimental results The DVR model can achieve reliable performance linearization with fewer coefficients than the MP and GMP models.

In this chapter, besides MP, GMP, and DDR, the DVR-2 and DVR-6 models are also evaluated and compared, which underlines that the main contribution of this part of our work is to compares two different categories of models: global models derived from the Volterra series and models based on segmentation approach [56].

2.6.1 Metrics of Comparison

The studied DPD models are evaluated according to metrics which are divided into two categories. The first one is the metrics of accuracy, which assess the linearity performance. The second one highlights the hardware implementation complexity, which refers to the number of coefficients and numerical properties, which refers to numerical stability to extract the model and the dynamic range of the coefficients. Another complexity feature of a model comes from the construction of the matrix operator, $\mathbf{Z}$ in (2.18) and the auto-correlation matrix $\mathbf{Z^H Z}$, which is controlled by the buffer length N.

2.6.1.1 Linearization Performance

As (2.18) seeks to minimize $e(n)$, a natural measure for the modeling accuracy is the normalized mean square error (NMSE) between the post-distortion output $z_p(n)$ and the PA input signal $x(n)$:

$$\mathrm{NMSE}_{dB} = 10\log_{10}\frac{\sum_{n=1}^{N}|x(n)-z_p(n)|^2}{\sum_{n=1}^{N}|x(n)|^2} \tag{2.21}$$

From a system point of view, the figure of merit is provided by the out-band distortions which are evaluated using adjacent channel power ratio (ACPR) which is defined for the right and left channel as:

$$\begin{aligned}\mathrm{ACPR}_{R,dB} &= 10\log_{10}\frac{\int_{-B/2}^{B/2}P(z_p(t))df}{\int_{B/2}^{3B/2}P(z_p(t))df}\\ \mathrm{ACPR}_{L,dB} &= 10\log_{10}\frac{\int_{-B/2}^{B/2}P(z_p(t))df}{\int_{-3B/2}^{-B/2}P(z_p(t))df}\end{aligned} \tag{2.22}$$

where B represents the bandwidth of the signal and $P(.)$ is power spectral density.

2.6.1.2 Implementation Properties

Predistorters are generally implemented in real digital circuits such as FPGA. Their identification can also be implemented on FPGA or some general-purpose digital processor. For that, it is important to have figures of merit related to hardware implementation. Two critical aspects of the implementation are concerned: the identification and the predistorter implementation.

2.6.1.2.1 Identification

The complexity to solve (2.18) is related to the number of coefficients C of the DPD models, and to the buffer length N [57]. Table 2.1 presents the parameters and number of coefficients of the different DPD models compared in this study.

Table 2.1: Comparison of DPD models complexity

DPD Model	**Parameters**	**Number of coefficients**
MP	K: nonlinearity order M: memory depth	$K \times (M+1)$
GMP	K_a, K_b, K_c: nonlinearity order L_a, L_c, L_c: memory depth M_b, M_c: lagging and leading cross-terms order	$K_a \times L_a$ $+K_b \times L_b \times M_b$ $+K_c \times L_c \times M_c$
DDR-1	K: nonlinearity order M: memory depth	$\frac{K+1}{2} + K \times M$
DDR-2	K: nonlinearity order M: memory depth	$\frac{K+1}{2} \times (M+1)$ $+3M\frac{K-1}{2}$
MDDRV-1	K: nonlinearity order M: memory depth	$\frac{K+1}{2} + (\frac{3K-1}{2}) \times M$
MDDRV-2	K: nonlinearity order M: memory depth	$\frac{K+1}{2} \times (M+1)$ $+2 \times (K-1) \times M$
DVR-2	K: number of segments M: memory depth	$(K+1) \times (M+1)$
DVR-6	K: number of segments M: memory depth	$5 \times K \times M + 2 \times K + M + 1$

The computation to solve (2.18) may exhibit numerical sensitivity depending on the condition number of $\mathbf{Z^H Z}$ which will be used as an indicator of this sensitivity.

On the other hand, according to Table 2.1, we can classify the DPD models in terms of complexity. The complexity of the DPD models, derived from the Volterra series, depends on the nonlinearity order K and the memory depth M. The complexity depends on the number of segment K, the memory depth M, and the number of the basis functions for the DVR model.

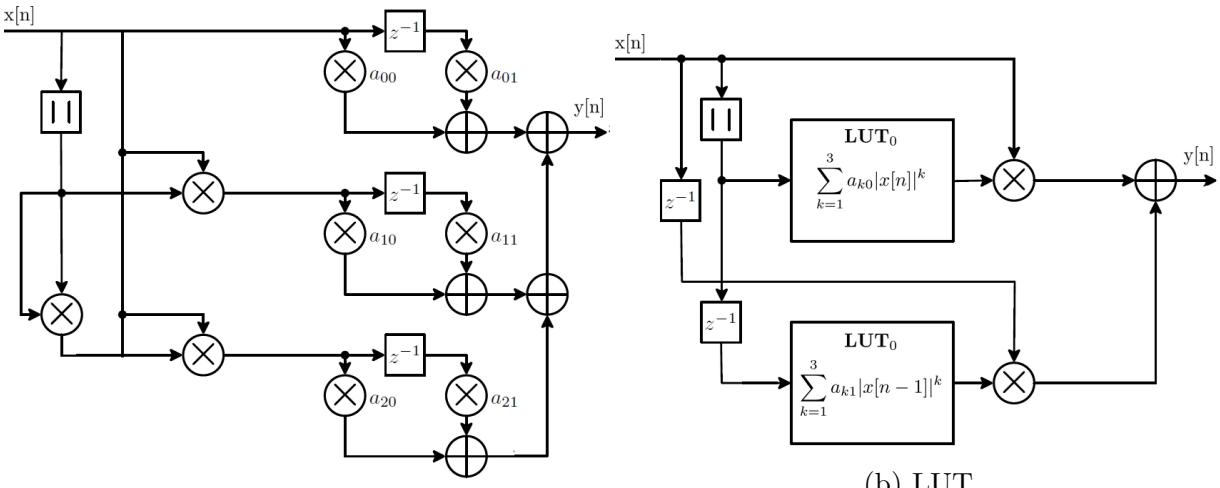
2.6.1.2.2 Predistorter implementation

For the implementation of the predistorter, there are two aspects to consider: the overall complexity related to the number of parameters (Table 2.1) and the number of bits required to encode each of those parameters. This number of bits will depend both on the dynamic range and the resolution required for the coefficients.

The hardware implementation can be carried out by two structures: direct implementation and the implementation by LUT. A comparison study of the direct form and implementation using LUT was presented in [58].

In direct implementation, complexity refers to the number of computational operations directly related to the number of coefficients. In implementation by LUT, complexity refers to the number of LUT, which is only related to the memory depth. It is worth noting that LUT implementation may not provide better implementation efficiency for all kinds of models, such as GMP.

An example implementation of the MP model with $K = 3$ and $M = 1$ is illustrated in Figure 2.5, where the direct method and LUT are respectively shown in Figure 2.5a and Figure 2.5b.



(a) Direct method

(b) LUT

Figure 2.5: Implementation of an MP model with $K = 3$ and $M = 1$

In this work, the dynamic range is considered a relative indicator of the requirement for the number of bits required for each model. We thus make the assumption that every model will require the same resolution, which is certainly rough. The dynamic range of the coefficients is introduced as a comparison element, which can be calculated as:

$$\epsilon = max[real(\mathbf{c}), imag(\mathbf{c}))] - min[real(\mathbf{c}), imag(\mathbf{c})] \tag{2.23}$$

2.6.2 Comparison of MP, GMP, DDR, MDDRV, and DVR

2.6.2.1 Testbench Description

The DPD models are evaluated using the input-output signals from real PA shown in Figure 2.6. The PA line is made of a three-way Doherty PA designed for the base station (BS-PA) with three LDMOS transistors BLF7G22LS-130 from Ampleon, formerly NXP, and its associated driver. This Doherty PA can have a peak output power of 57 dBm (500 W) and has a linear gain of 16 dB. In the following, the PA in Figure 2.6 is denoted by LDMOS-TW-500W PA.

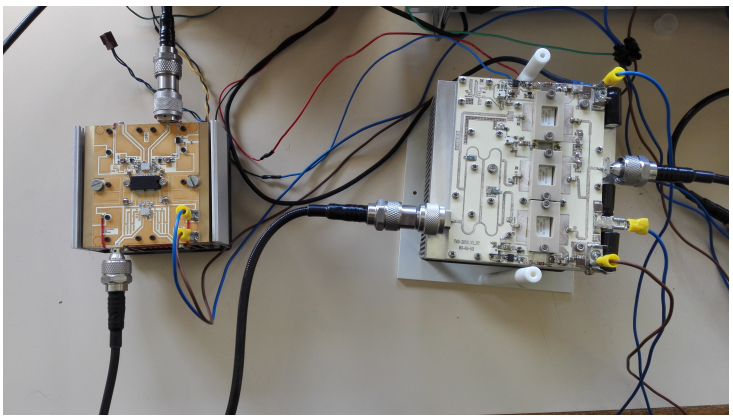


Figure 2.6: Three-way Doherty PA

The LDMOS-TW-500W PA is excited by an LTE signal with 20 MHz bandwidth and a PAPR of 8 dB, and the carrier frequency is 2.14 GHz with a sampling frequency of 200 Ms/s. The AM-AM and AM-PM curves obtained with LDMOS-TW-500W PA are plotted in Figure 2.7.

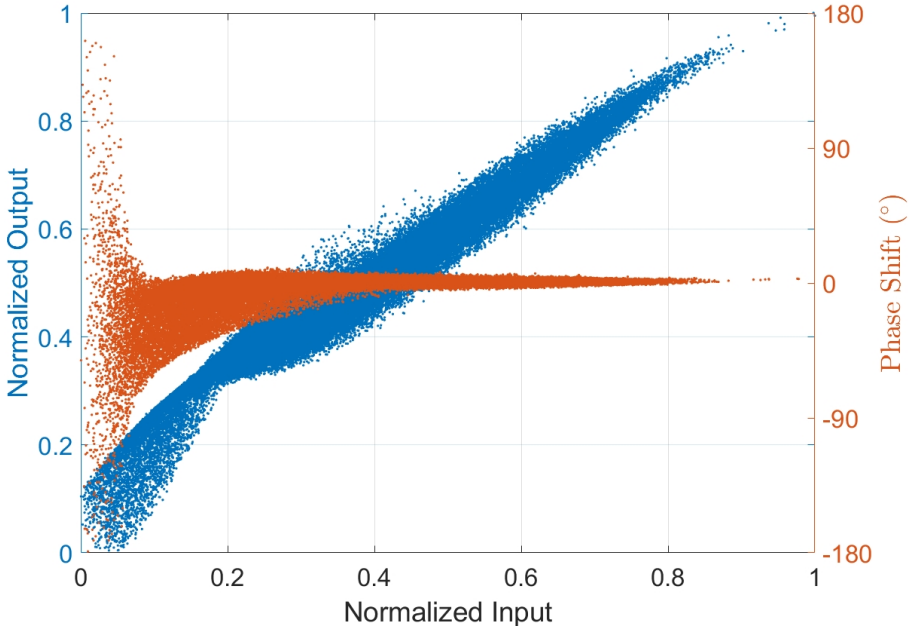


Figure 2.7: AM-AM & AM-PM curves of Doherty PA for an 20 MHz LTE signal

2.6.2.2 Parameters of DPD Models

The structure of the GMP model are optimally determined using a hill-climbing algorithm as proposed in [59] using off-line additive criterion.

For this study, we have chosen to limit the maximum nonlinearity order and memory depth to the same maximum values as the GMP model, *i.e.* $K = 11$ and $M = 4$, for MP, DDR-1, DDR-2, MDDRV-1, and MDDRV-2. The number of coefficients is 43.

We size the DVR model to have a model with a similar trade-off between complexity and performance compared to the other DPD models. For the DVR-2 model, the number of coefficients is 55 with $K = 10$ and $M = 4$ and 71 for the DVR-6 model with $K = 3$ and $M = 4$.

Figures of merit of every model are summarized in Tables 2.2, where it can be seen that the GMP model achieves the best performance regarding NMSE and ACPR even though DVR-6, MP, and DVR-2 also achieve good performances.

Table 2.2: Comparison of linearization performances

DPD Model	NMSE (dB)	ACPR_L (dB)	ACPR_R (dB)
Without DPD	**-10.29**	**-23.32**	**-22.46**
MP	-32.65	-41.23	-38.90
GMP	-35.39	-43.02	-42.46
DDR-1	-29.39	-38.37	-36.26
DDR-2	-31.81	-39.81	-38.32
MDDRV-1	-29.39	-38.37	-36.26
MDDRV-2	-31.82	-39.81	-38.33
DVR-2	-32.84	-41.60	-38.95
DVR-6	-34.94	-41.88	-41.81

Figure 2.8 shows the plot of the NMSE according to the number of coefficients for each DPD model. From this figure, it can be seen that the NMSE for all models, except GMP and DVR-6, converges toward an asymptote around 33 dB while the NMSE for GMP and DVR-6 is still decreasing with the number of coefficients.

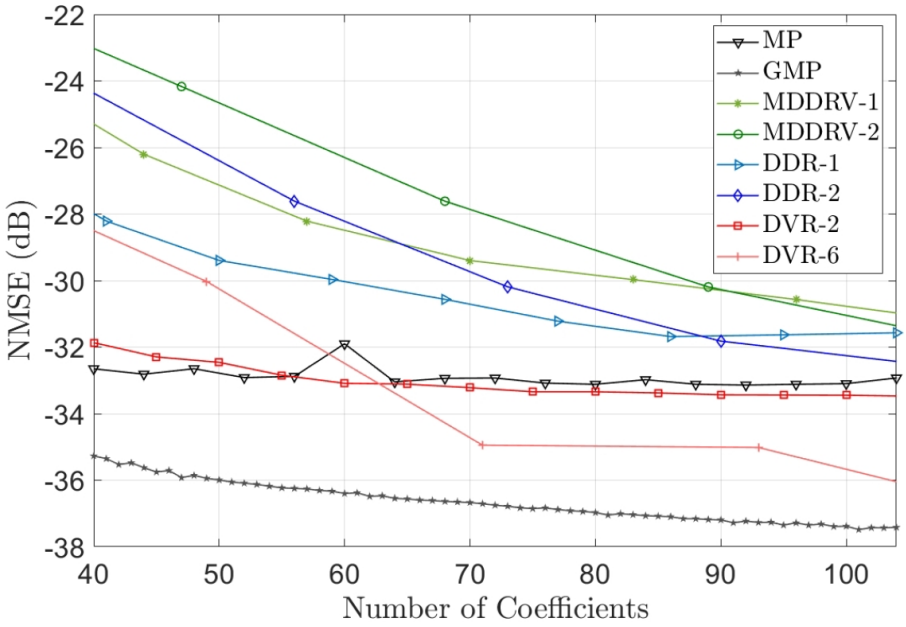


Figure 2.8: Modeling accuracy vs number of coefficients

Table 2.3 shows the number of coefficients and their dynamic range for each model, and the condition number of the matrix $\mathbf{Z^H Z}$ which is involved in the computation of the identification procedure.

Table 2.3: Comparison of implementation properties

DPD Model	Number of coefficients	Condition number	Coefficient dynamic range
MP	55	$1.12e^{27}$	$2.18e^{11}$
GMP	43	$5.24e^{23}$	$1.75e^{10}$
DDR-1	50	$7.52e^{22}$	$4.59e^{8}$
DDR-2	90	$3.16e^{24}$	$2.53e^{9}$
MDDRV-1	70	$7.54e^{22}$	$4.83e^{8}$
MDDRV-2	110	$3.72e^{24}$	$2.53e^{9}$
DVR-2	55	$5.40e^{7}$	1
DVR-6	71	$2.70e^{9}$	11.65

For the LDMOS-TW-500W PA with this waveform, DVR exhibits exceptional values for both dynamic range and condition number as there are more than ten orders of magnitude for both dynamic range and condition number between DVR models and GMP-MP models.

Even if the GMP model reaches the best linearization performance, the DVR-6 has a very similar performance but with nearly twice the number of coefficients. It is the price to pay to benefit from the outstanding properties of the DVR-6 model for digital hardware implementation in this scenario.

In order to validate the effectiveness of this comparative study, the GMP and DVR-6 models will be specifically highlighted and compared in the following section. The comparison will be made in different scenarios with different bandwidths, and at different power levels [60].

2.6.3 Comparison of GMP and DVR

2.6.3.1 Testbench

The experiments are carried out using an LDMOS PA 50 W at 1.78 GHz center frequency. In the following, this PA is denoted by LDMOS-50W PA. The testbench used for these tests consists of software-defined radio (SDR) from ARELIS for generating the IQ signals and a vector signal transceiver (VST) NI for the data acquisition at 200 Ms/s.

The tests are made with a set of LTE signals of 5, 10, and 20 MHz bandwidth and a maximum PAPR of 12dB, at four output power levels: 34, 38, 42, and 44 dBm.

The nonlinearities and the memory effect of the LDMOS-50W PA can be seen from the AM-AM and AM-PM curves in Figure 2.9 for 20 MHz bandwidth case at 44 dBm.

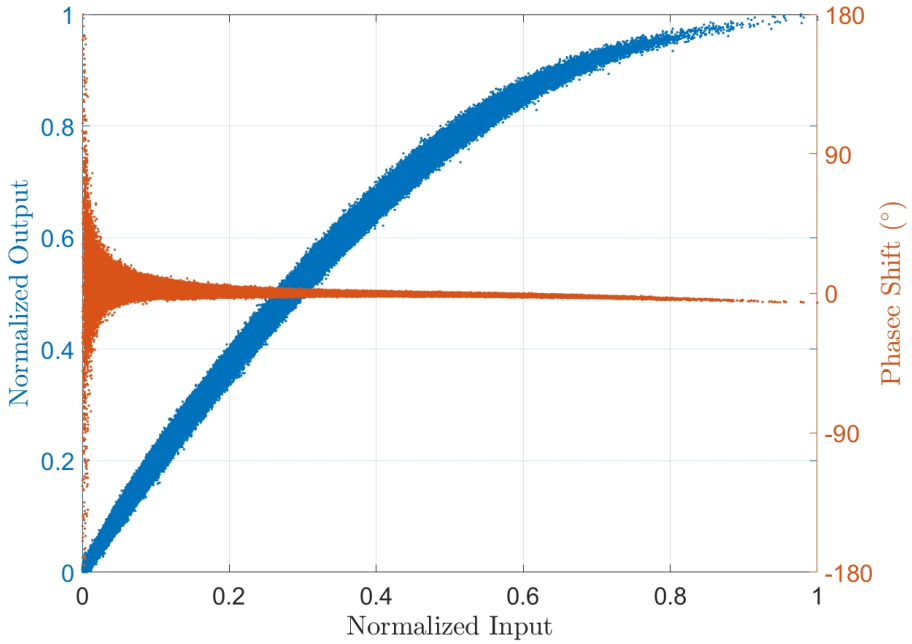


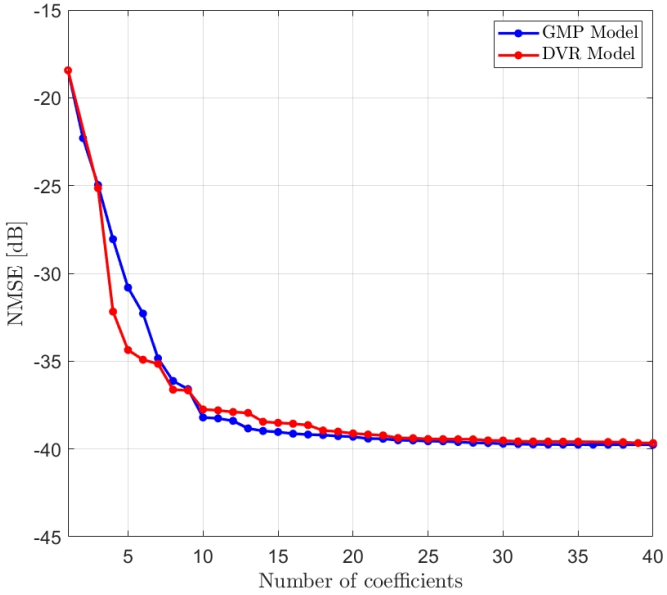
Figure 2.9: AM-AM & AM-PM curves of LDMOS PA for a 20 MHz LTE signal

2.6.3.2 Parameters of Models

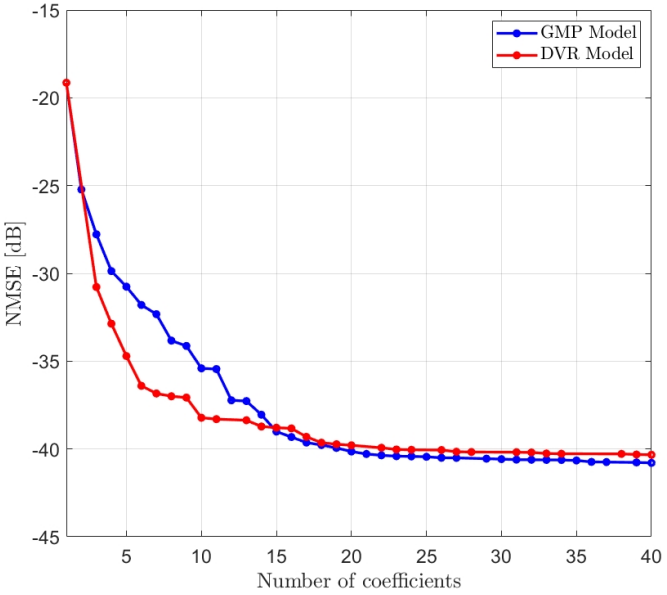
The structure determination of the GMP model is done using the hill-climbing algorithm, which is proposed in [59]. With this algorithm, we size the structure of the GMP model using the criterion which makes a trade-off between modeling accuracy and complexity. For the DVR-6 model, an exhaustive search is carried out by exploring all combinations when the number of segments K ranges from 2 to 10 and the memory depth M ranges from 0 to 10.

The thresholds β_k for the set of partitions are optimally selected by a proposed approach which will be the subject of the next chapter.

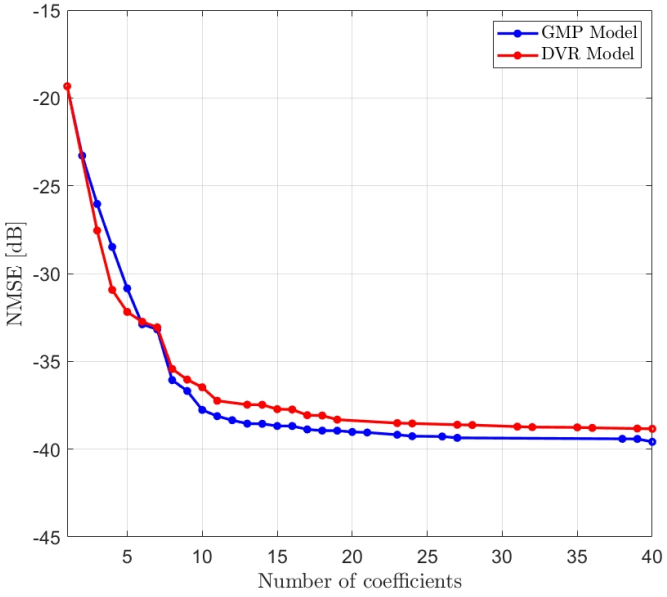
Figure 2.10 shows the plot of the modeling accuracy presented by NMSE versus the number of coefficients for 5, 10, and 20 MHz bandwidths. Theses results are achieved using postdistortion architecture. The output power is 44 dBm.



(a) 5 MHz bandwidth



(b) 10 MHz bandwidth



(c) 20 MHz bandwidth

Figure 2.10: NMSE vs number of coefficients at 44 dBm output power

In Figure 2.10a, the NMSE behavior versus the number of coefficients of the two DPD models has a similar form of convergence where they start from the same point and converge towards an NMSE asymptote around 40 dB. However, the DVR-6 model has faster convergence since its NMSE between 3 and 8 coefficients is better than that of the GMP model. This is clearly seen in Figure 2.10b, where the DVR-6 model has fast convergence to join the 40 dB asymptote, while the GMP model requires 15 coefficients. However, the GMP model convergence is slightly better than that of the DVR-6 model by about 0.2dB of NMSE, as shown in figure 2.10c.

To study the linearization performance and the numerical properties of the two DPD models, we have chosen to compare the two models with the same number of coefficients. In this case, the number of coefficients is set to 39 coefficients.

Table 2.4 summarizes the parameters of structure found by the hill-climbing algorithm for the GMP model and the structure parameters found by the exhaustive search for the DVR-6 model.

Table 2.4: Parameters of GMP and DVR model for each bandwidth at 44 dBm output power

Bandwidth	DVR-6		GMP							
	K	M	K_a	L_a	K_b	L_b	M_b	K_c	L_c	M_c
5 MHz	3	2	1	14	5	1	1	10	2	1
10 MHz	3	2	4	2	1	6	3	13	1	1
20 MHz	3	2	1	3	2	2	3	8	1	3

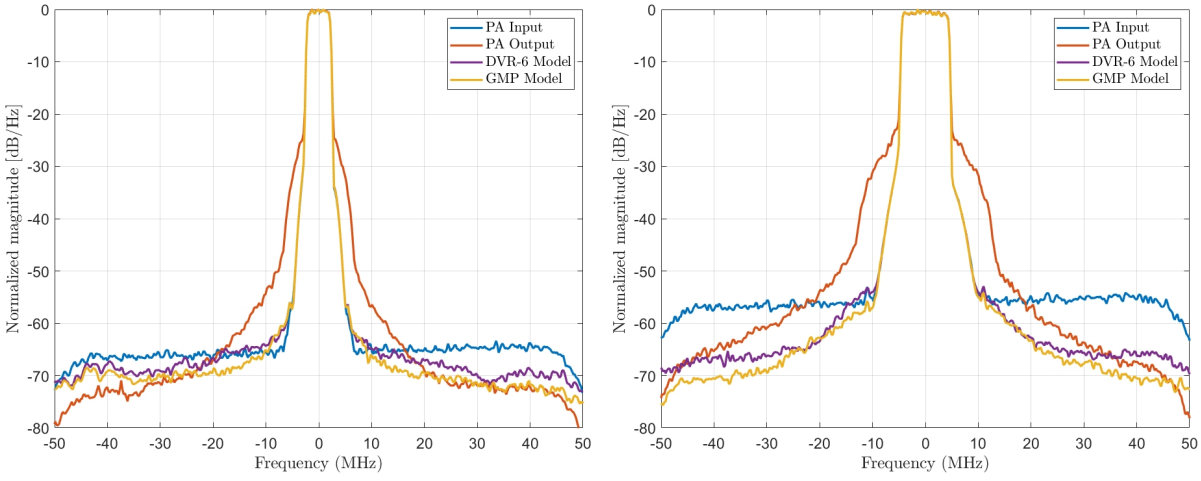
The linearization performances are summarized in Table 2.5 in terms of NMSE and ACPR.

Table 2.5: Linearization performance of GMP and DVR-6 with 39 coefficients

Bandwidth	NMSE (dB)		ACPR_R (dB)		ACPR_L (dB)	
	DVR-6	GMP	DVR-6	GMP	DVR-6	GMP
5 MHz	-39.66	-39.75	-42.43	-42.53	-40.81	-40.84
10 MHz	-40.31	-40.76	-44.78	-44.48	-43.43	-43.57
20 MHz	-38.82	-39.41	-49.38	-49.76	-48.80	-49.94

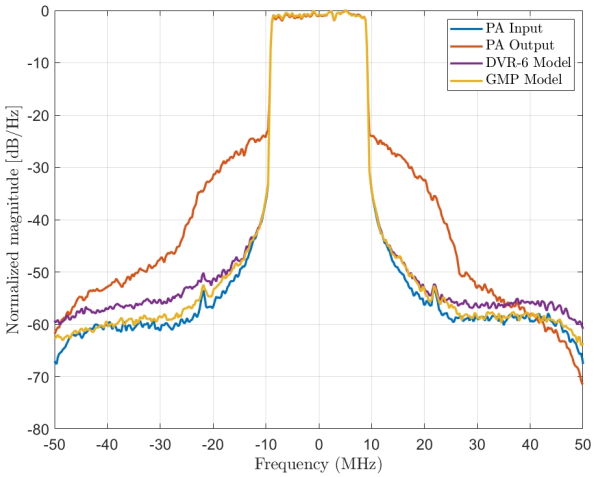
By comparing the results in Figure 2.10, the values collected in Table 2.5 confirm that the GMP model provides slightly better accuracy than the DVR model, which can be tolerable since the results are collected using postdistortion scenario.

Figure 2.11 shows the power spectrum of the postdistorter for the GMP and DVR-6 model with 5, 10, and 20 MHz bandwidth and 44 dBm output power. As it can be seen, both DPD models have close linearization performances.



(a) 5 MHz

(b) 10 MHz



(c) 20 MHz

Figure 2.11: Spectra of PA input-output, GMP, and DVR-6 postdistorter at 44 dBm output power

Table 2.6 presents the dynamic ranges of the coefficients and the condition number of the matrix involved in the identification procedure. The results in Table 2.5 and Table 2.6 are given with the same complexity for both DPD models.

Table 2.6: Numerical properties of GMP and DVR-6 with 39 coefficients

Bandwidth	$\log_{10}$(Condition Number)		Dynamic range	
	DVR-6	GMP	DVR-6	GMP
5 MHz	13	17	109	34887
10 MHz	14	18	327	19700
20 MHz	13	18	307	16278

From Table 2.6, it can be seen that the condition number for DVR is lower than for GMP, even though it is still significant. The dynamic range of the coefficients is the other figure of merit that assesses the numerical properties of the model. It can be seen that the DVR model exhibits a better behavior for this feature than the GMP model.

On the other hand, the performance of the GMP and DVR-6 models is investigated with the variation of output power level. For this study, we choose the signal with 20 MHz bandwidth by varying the output power at four levels: 34, 38, 42, and 44 dBm.

The parameters for the DVR-6 model are $K = 4$ and $M = 1$. The parameters of the GMP model are obtained with the hill-climbing algorithm using the additive criterion and

summarized in Table 2.7, for each output power level. The number of model coefficients for each power level is 30 coefficients.

Table 2.7: GMP structure obtained by hill-climbing algorithm for each output power

GMP parameters	34 dBm	38 dBm	42 dBm	44 dBm
K_a	2	1	2	3
L_a	6	6	5	4
K_b	4	2	2	8
L_b	1	1	1	1
M_b	3	6	3	1
K_c	3	6	7	10
L_c	2	2	2	1
M_c	1	1	1	1

Table 2.8 shows the linearization performance and numerical properties for each power level.

Table 2.8: Comparison for each output power level of GMP and DVR models

Output Power	34 dBm		38 dBm		42 dBm		44 dBm	
	GMP	DVR-6	GMP	DVR-6	GMP	DVR-6	GMP	DVR-6
NMSE (dB)	−32.58	−32.34	−36.28	−35.45	−36.34	−35.36	−39.25	−37.73
ACPR_L (dB)	−50.41	−49.93	−50.44	−49.43	−46.77	−45.70	−48.99	−47.46
ACPR_R (dB)	−49.38	−49.27	−49.85	−49.11	−46.03	−45.52	−48.35	−47.86
$\log_{10}$(Condition Number)	9	11	11	10	13	10	17	10
Dynamic range	8	42	75	31	474	8	$1.18e^5$	104

The three first rows of Table 2.8 allow making some comparison between both models on linearization performances. The two last rows present the numerical properties results. The dynamic range and condition number of the GMP model increase with the output power. On the other hand, they remain bounded for the DVR model.

The GMP model performs better than DVR when the PA exhibits lower nonlinearities, but the DVR model remains suitable for strong nonlinearity. On the other hand, the DVR-6 model confirms its outstanding numerical properties for different bandwidths and output power.

2.7 Conclusion

This chapter introduces the principle of digital predistortion. The Post-Distortion architecture has been presented and discussed. The DPD models are presented and described according to 2 categories: global models derived from Volterra or modified Volterra series and models based on segmentation approach. These DPD models are compared according to different criterion, and their linearization performance versus implementation properties is presented.

The GMP model provides a good trade-off between accuracy performance and complexity. The DVR model may achieve similar performances but with an increase in the number of coefficients.

On the other hand, the DVR model has remarkable numerical properties that make it a good choice for hardware implementation. For that purpose, the following chapter will focus on the design and the optimization of the DVR model, determining its structure and proposing its implementation architecture.

Chapter 3

Optimization and Sizing of DVR Model

3.1 Introduction

Volterra series have good performances for the modeling of nonlinear dynamic systems. Besides, they are linear with respect to their coefficients, which simplifies their identification using the least-square method. However, their complexity is very high because their number of coefficients increases exponentially with nonlinearity order and memory depth. Moreover, the Volterra series have destructive numerical properties to identify their coefficients for a high order of nonlinearity, which can be justified because the model is built with a non-orthogonal basis.

Models derived from Volterra series such as MP, GMP, or DDR have proven their effectiveness to serve as a predistorter or PA model. However, PA's advanced architecture with good efficiencies, such as Doherty and envelope tracking, exhibits a nonlinear behavior, which is more challenging to be linearized. On the other hand, the needs for higher data rates require the use of efficient spectral modulation techniques such as orthogonal frequency division multiplexing (OFDM). As a result, the modulated signals have a high PAPR, which stimulates PA nonlinearities. Besides, 5G and MIMO systems allow a very high data rate, which is a real challenge for DPD in bandwidth, nonlinearity, and dynamic behavior. Considering all these facts, it becomes more challenging to design an efficient DPD system to achieve accurate modeling with good efficiency and low computational complexity.

This has pushed the interest of research to find a different approach for DPD. One of these approaches is the segmentation approach, where the input space is divided into several subspaces represented by small models. One of these models based on the segmentation approach is the DVR model presented in the previous chapter as a DPD model in the comparative study.

The study of the DVR model raises different questions such as:

- how to segment the input space optimally,
- how to determine the best model structure,
- how to determine the best model structure jointly with its optimal segmentation.

This chapter focuses on the DVR model, mainly on how to optimize its boundary thresholds and find its optimal structure. We propose a new approach for the optimization of thresholds in which the optimization problem is decomposed into a set of unimodal sub-problems that allow using a unidirectional minimization [61]. The sizing of the DVR model is based on hill-climbing (HC) algorithm to determine an optimal structure of the

DVR model according to a cost function [62] representing a trade-off between modeling accuracy and model complexity.

3.2 DVR Model

As mentioned in Section 2.3.4, the DVR model is based on a segmentation approach to serve as behavioral PA modeling or a predistorter model.

According to (2.8), the nonlinear basis function is constructed by vector decomposition and phase rotation. The absolute operation from vector rotation is used to generate the nonlinearity. This property allows fitting many shapes of nonlinear behavior. The structure of the model basis function

$$\sum_{k=1}^{K} c_k||x(n)| - \beta_k|e^{j\theta(n)}$$

is presented in Figure 3.1.

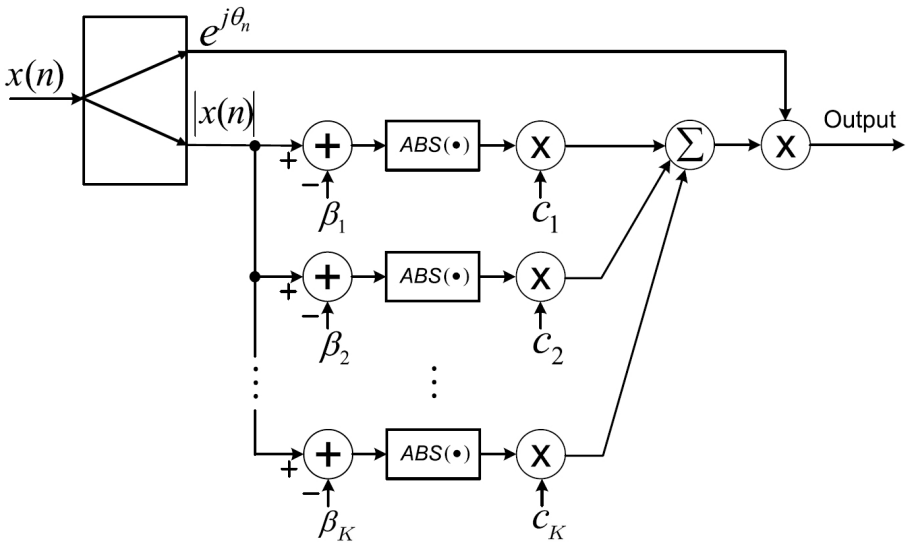


Figure 3.1: DVR basis function structure

In [63], a modified CPWL based model is proposed for modeling wideband PA. The proposed model has a structure similar to that of the DVR model but without decomposed vector rotation operation introduced in Section 2.3.4, in which the first-order basis function

$$\sum_{i=0}^{M}\sum_{k=1}^{K} c_{ki}||x(n-i)| - \beta_i|e^{j\theta(n-i)}$$

is replaced by a k-order function

$$\sum_{i=0}^{M}\sum_{k=1}^{K} c_{ki}|x(n-i)|^k e^{j\theta(n-i)}$$

In another research work, the authors of [64] proposed a low-complexity sideband distortion suppression model based on the DVR model concept to construct the distortion component and eliminate the unwanted transmitter leakage.

In [65], another modification has been made to the DVR model by borrowing the GMP models concept into the structure of the DVR model for dual-band envelope tracking PA. The authors of [65] have proposed a behavioral model by combining the GMP model in (2.6) and the DVR model presented in (2.14).

The authors of [66] proposed a hardware implementation strategy with low complexity and reduced power dissipation, where it was achieved by modifying the DVR model into a sub-decomposed representations, which significantly reduce the computational complexity of the model extraction. The proposed model is expressed as:

$$\begin{aligned} y(n) = & \sum_{i=0}^{M} a_i x(n-i) \\ & + \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,11} F(|x(n-i)|, k) e^{j\theta(n-i)} \\ & + \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,21} F(|x(n-i)|, k) e^{j\theta(n-i)} . |x(n)| \\ & + \dots \end{aligned} \tag{3.1}$$

where $F(|x(n)|, k)$ is a constructed segment expressed by

$$= \begin{cases} 0.5(\beta_k - |x(n)| + |\beta_k - |x(n)||), & \text{for odd values of } k \\ 0.5(|x(n)| - \beta_k + ||x(n)| - \beta_k|), & \text{for even values of } k \end{cases} \tag{3.2}$$

In this chapter, we adopt a general presentation of the DVR model which can be expressed by:

$$y(n) = \sum_{i=0}^{M_{lin}} a_i x(n-i) + \sum_{T_t \in S} T_t \tag{3.3}$$

where $x(n)$ and $y(n)$ are the input and output of the model, M_{lin} is the memory depth for the linear term, a_i are the complex coefficients of the linear term, and S is the set of the terms T_t which are used in the model with $S \subset \mathbf{T}$ and $\mathbf{T} = [T_{1,[0,...,P]}, T_2, T_3, T_4, T_5, T_6, T_7]$ where $T_{1,[0,...,P]}$ is the set $T_{1,0}, ..., T_{1,P}$ whose elements are defined hereafter along with $T_2, ..., T_7$.

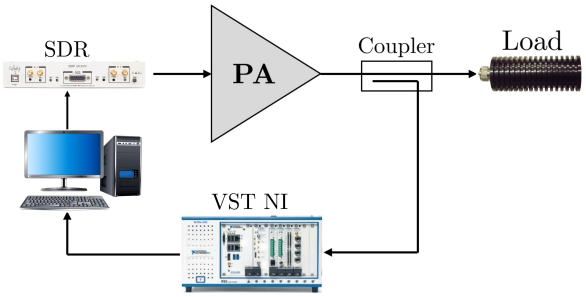
$$\begin{aligned} T_{1,p} &= \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1p} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \cdot |x(n)|^p \\ T_2 &= \sum_{k=1}^{K} \sum_{i=i_2}^{M} c_{ki,2} ||x(n-i)| - \beta_k| \cdot x(n) \\ T_3 &= \sum_{k=1}^{K} \sum_{i=i_3}^{M} c_{ki,3} ||x(n-i)| - \beta_k| \cdot x(n-i) \\ T_4 &= \sum_{k=1}^{K} \sum_{i=i_4}^{M} c_{ki,4} ||x(n)| - \beta_k| \cdot x(n-i) \\ T_5 &= \sum_{k=1}^{K} \sum_{i=i_5}^{M} c_{ki,5} ||x(n-i)| - \beta_k| \cdot x^2(n) \cdot x^*(n-i) \\ T_6 &= \sum_{k=1}^{K} \sum_{i=i_6}^{M} c_{ki,6} ||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \\ T_7 &= \sum_{k=1}^{K} \sum_{i=i_7}^{M} c_{ki,7} ||x(n-i)| - \beta_k| \cdot x^*(n) \cdot x^2(n-i) \end{aligned} \tag{3.4}$$

where K is the number of segments, β_k are the bounds of the segments, p is the nonlinearity order, M the memory depth, and c_{ki} are the complex coefficients of the model for each segment. The indexes $i_2, i_3, ..., i_7$ are equal to 0 or 1 depending on the set of selected T_i terms. The cardinal of $\mathbf{T}$ is thus $P + 7$. The number of coefficients, denoted by C, depends on the parameters of the DVR model structure.

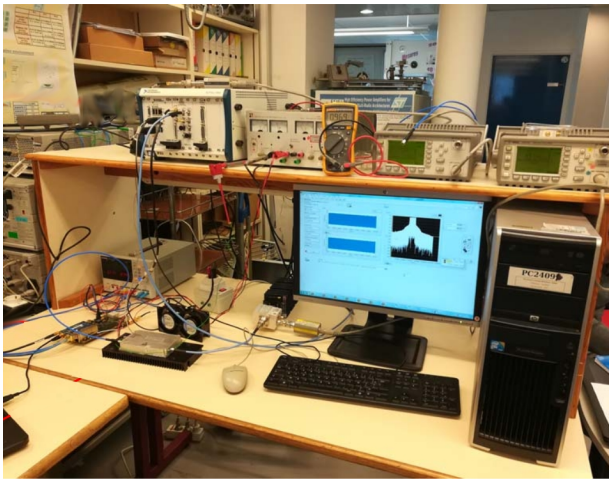
3.3 Experimental Testbench

Experimental acquisitions from real PA have been carried out.

The testbench is represented in Figure 3.2, which includes a PC MATLAB software, an SDR card, a Vector Signal Transceiver (VST) from National Instruments (NI), and an LDMOS PA of 50W.



(a) Block diagram of testbench



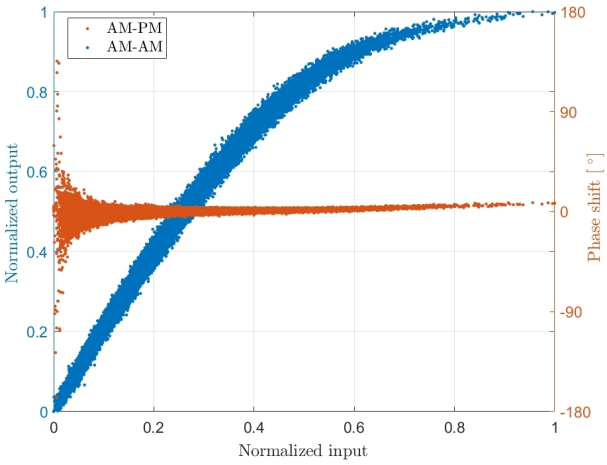
(b) Testbench of LDMOS PA

Figure 3.2: Testbench for experimental implementation

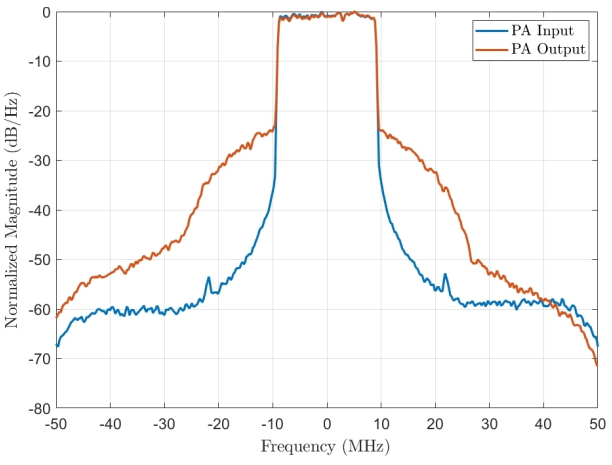
The baseband IQ signal is fed from PC MATLAB software to the PA chain through an SDR card. The SDR up-converts the baseband signal to the carrier frequency. The signal at the output of the PA is then down-converted to the baseband by the VST NI, which provides to the PC workstation the baseband signal digitized with a maximum sampling frequency of 200 MHz.

The input and output baseband signals are then synchronized in time to be used by the identification algorithm (2.18). Around 64000 IQ samples were used for training. A 20 MHz LTE signal with 12 dB of PAPR was used for the test, and the average output power of the PA was 44 dBm.

The nonlinearities of the PA can be seen from the AM-AM and AM-PM characteristics for the tested 20 MHz LTE signal in Figure 3.3a. The power spectrum of input and output of PA is shown in Figure 3.3b.



(a) AM-AM and AM-PM curves



(b) Spectra of PA input-output

Figure 3.3: The PA characteristics with 20MHz LTE signal

The ACPR and approximate EVM of this PA are:

$$\text{ACPR}_{\text{L1}} = -29.54 \text{ dB} \; ; \; \text{ACPR}_{\text{R1}} = -30.58 \text{ dB}$$
$$\text{EVM}_{\text{app}} = 9.2 \; \%$$

3.4 Optimization of DVR Model Thresholds

3.4.1 State of the Art

In (3.4), the thresholds β_K that define the boundaries of segments can be selected using uniform or optimized segmentation. Although many works in the literature are dedicated to models with uniform segmentation, e.g., $\beta_k = k/K$ for $k = 1; 2; ...; K-1$, few of them deal with the optimal segmentation.

The first work for DPD application about optimal spacing was conducted for memoryless LUT DPD [67] - [68], which is a challenge closely related to the optimization of segmentation for a piecewise model. In [67], the author has proposed a non-uniform systematic spacing using a companding function into table indexing, which reduces the intermodulation power. In [68], an iterative procedure is proposed to optimize non-uniform LUT spacing, which is based on an IMD criterion and the input signal statistics.

The benefit of optimal segmentation compared to uniform segmentation has already been established in [69] where the authors suggest reducing the complexity of the algorithm by considering the memoryless version of the actual DVR model. The authors used Gauss-Newton-based methods such as Levenberg-Marquardt [70] to jointly optimize the thresholds and identify the model coefficients. Nevertheless, even with the reduced complexity algorithm proposed in [69], the problem to solve is still a nonlinear least square problem. The authors claim that the nonlinear minimization problem behaves well close to uniform segmentation.

Generally speaking, setting the thresholds optimally for piecewise models such as the DVR model requires global optimization, and in [71] the authors proposed an approach based on genetic algorithm (GA) to optimize the thresholds of the CPWL model for radio-over-fiber. The authors used the ACPR as the fitness function for GA. The GA generations used in the GA were not mentioned in the optimization process, meaning that the complexity is not a significant concern in this research.

3.4.2 Structure of DVR Model

In this section, we use the set $T_S = [T_{1,0} \; T_{1,1} \; T_2 \; T_3 \; T_4]$ with $P = 1$, which corresponds to the DVR-6 model presented in Chapter 2, in which it has achieved an excellent modeling performance. Therefore, the DVR model is expressed as:

$$
\begin{aligned}
y(n) = & \sum_{i=0}^{M_{lin}} a_i x(n-i) \\
& + \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,10} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \\
& + \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,11} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} . |x(n)| \\
& + \sum_{k=1}^{K} \sum_{i=1}^{M} c_{ki,2} ||x(n-i)| - \beta_k| . x(n) \\
& + \sum_{k=1}^{K} \sum_{i=1}^{M} c_{ki,3} ||x(n-i)| - \beta_k| . x(n-i) \\
& + \sum_{k=1}^{K} \sum_{i=1}^{M} c_{ki,4} ||x(n)| - \beta_k| . x(n-i)
\end{aligned}
\tag{3.5}
$$

where a_i and c_{ki} are the model coefficients which are identified using ILA.

In the following, we set M_{lin} to 2 and M to 1.

3.4.3 Motivation

The benefits of an optimized segmentation are highlighted in Figure 3.4 and Figure 3.5 which compares uniform and optimal segmentation for a different number of segments in terms of NMSE and ACPR for a 20 MHz bandwidth LTE signal to drive the LDMOS PA.

According to Figure 3.4, the linearization performance in terms of NMSE for 8 segments using uniform segmentation could be achieved with only 3 segments using optimal segmentation. This is confirmed in Figure 3.5 in terms of ACPR, where the optimal segmentation requires only 4 segments to achieve the same linearization performance that the uniform segmentation with 10 segments.

The NMSE and ACPR of the DVR model with optimal segmentation converge towards an asymptote from 4 segments, while uniform segmentation requires more than 10 segments to reach the best linearization performance.

Thus, the optimal segmentation significantly reduces the complexity since the number of coefficients to be estimated is reduced. For the scenario at hand, only 43 coefficients for optimized segmentation are required to achieve the same linearization performance with 93 coefficients for uniform segmentation. This confirms the interest in optimizing the thresholds of the DVR model.

In the sequel, we propose an approach to optimize the thresholds β_K and how the global optimization problem could be divided into a set of unimodal sub-problems requiring only a unidirectional minimization. This approach will be compared to the brute-force and GA in terms of linearization performance and complexity.

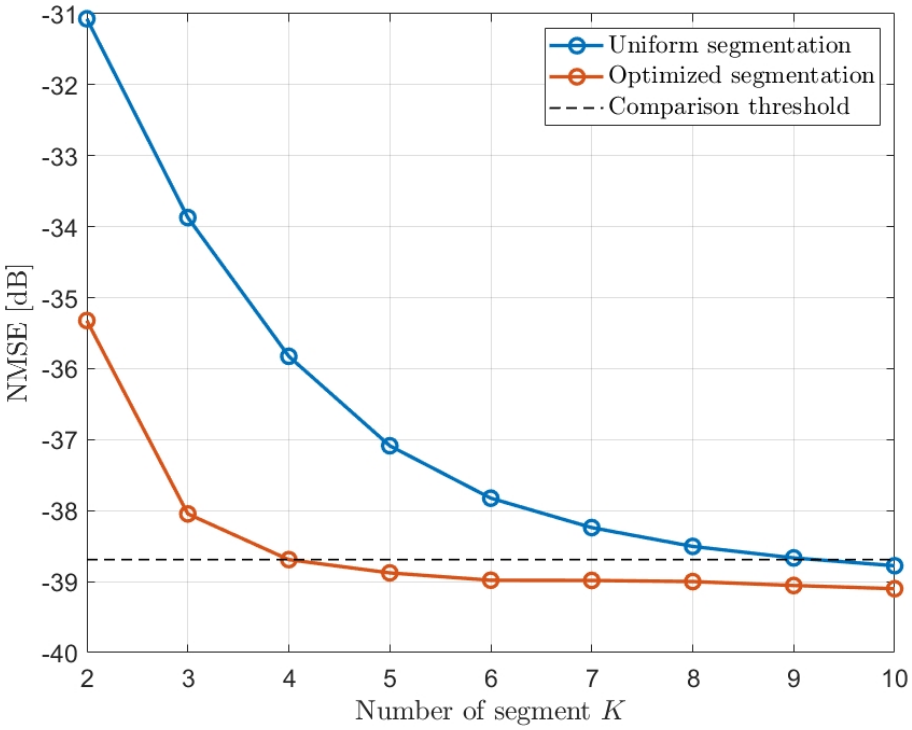


Figure 3.4: NMSE versus K with $M_{lin} = 2$ and $M = 1$

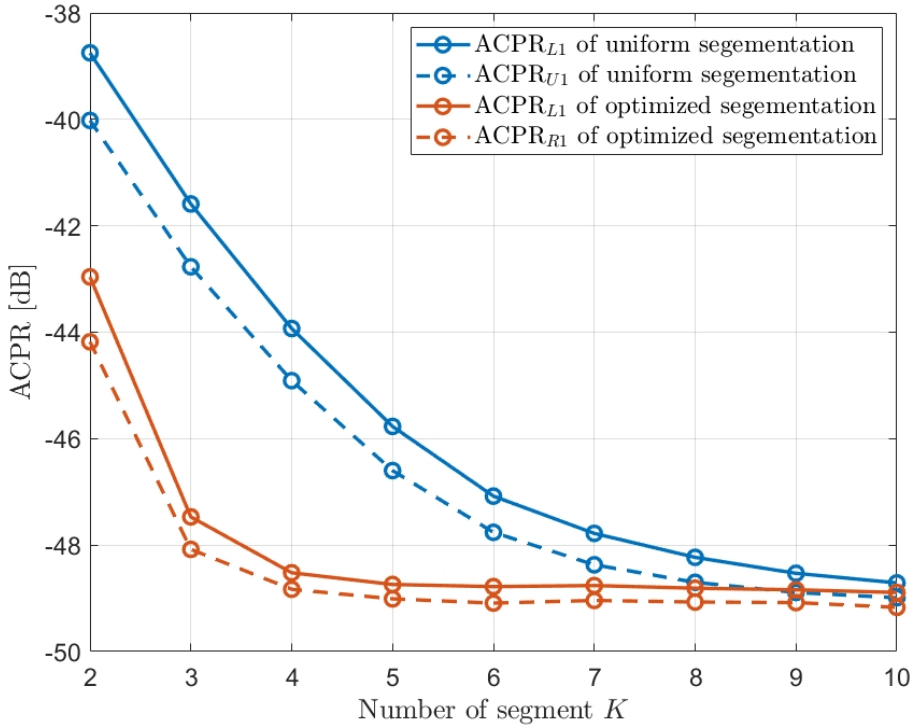


Figure 3.5: ACPR$_{\mathrm{L1}}$ and ACPR$_{\mathrm{R1}}$ versus K with $M_{lin} = 2$ and $M = 1$

3.4.4 Brute-force

Brute force is an exhaustive search over the normalized input range with a given precision. It consists of testing all the possible threshold positions over the normalized input in which for each segmentation corresponding to thresholds allocation, the linearization performance metrics are computed (NMSE or ACPR or both). the threshold positions tested are located on a grid containing N positions which are uniformly spaced.

Once brute-force is finished, it will be easy to find the threshold allocation with the best linearization performance. Brute-force is seen as an intuitive solution to find an optimum for an optimization problem, but it is not a realistic approach since a massive computation is required.

Regarding the optimization of DVR model thresholds, brute-force divides the normalized input range into $\mathcal{N}$ steps. For a given number of segments K with $K-1$ thresholds: $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \ldots\ \beta_{K-1}]$, the search explores all possible combinations of the position of the thresholds over the normalized input range. The brute-force returns the optimal segmentation $\boldsymbol{\beta}_{opt}$ with the best (minimized) NMSE among all segmentation $\boldsymbol{\beta}_{(q)}$ tested in the search space $\mathcal{V}$ where $q \in \mathcal{V}$.

The brute-force returns the optimal segmentation with the best (minimized) NMSE among all segmentation tested in the search space V with Q.

For a given combination, each threshold β_i in the threshold vector $\boldsymbol{\beta}$ must be located on a unique position. The number of combinations is $\binom{\mathcal{N}}{K-1}$, which corresponds to NMSE (or ACPR) computations, where the search grid is divided into $\mathcal{N}$ values. The algorithm of brute-force is presented in Algorithm (1).

Algorithm 1: Algorithm of brute-force

Given K, M_{lin} and M for the DVR model
Divide the search grid to $\mathcal{N}$ steps
Set loop counter $q = 1$
for $\beta_1 = [\frac{1}{\mathcal{N}}\ \frac{2}{\mathcal{N}}\ \ldots\ \frac{\mathcal{N}-1}{\mathcal{N}}]$ **do**
  for $\beta_2 = [\frac{1}{\mathcal{N}}\ \frac{2}{\mathcal{N}}\ \ldots\ \frac{\mathcal{N}-1}{\mathcal{N}}]$ **do**
    if $\beta_2 \neq \beta_1$ **then**
      ⋮
      for $\beta_{K-1} = [\frac{1}{\mathcal{N}}\ \frac{2}{\mathcal{N}}\ \ldots\ \frac{\mathcal{N}-1}{\mathcal{N}}]$ **do**
        if $\beta_{K-1} \neq \beta_{K-2} \& \ldots \& \beta_{K-1} \neq \beta_1$ **then**
          $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \ldots\ \beta_{K-1}]$
          Evaluate NMSE($\boldsymbol{\beta}$)
          $q = q + 1$
        end
      end
    end
  end
end
$\boldsymbol{\beta}_{opt} = \underset{q \in V}{argmin}\left(\text{NMSE}(\boldsymbol{\beta}_q)\right)$

In our study, the results of brute-force will be taken as a reference for the comparison subject.

3.4.5 Genetic Algorithm

GA is an efficient heuristic algorithm that is inspired by the theory of the Darwinian principle of natural evolution.

This algorithm reflects the process of natural selection where the individuals (thresholds) are selected for reproduction to produce offspring (optimal segmentation) of the next generation using some genetic operators such as selection, crossover, and mutation.

The GA begins by creating a random initial population of individuals characterized by a set of so-called genes. The GA then generates a sequence of new populations using

the current population to create the children that make up the next generation. The GA selects the parents from individuals in the current population who have better fitness function and bring their genes to their children.

The GA uses three main techniques at each generation to produce the next generation from the current population :

- Elite selection in which the parents are chosen for the next generation based on their fitness function. An individual with a high fitness function can be selected more than once as a parent, contributing its genes to more than one child.
- Crossover children are created by combining pairs of parents P in the current population. At each coordinate of the child vector, the default crossover function randomly selects a gene at the same coordinate from one of the two parents and assigns it to the child.
- Mutation children are created by randomly changing the genes of individual parents, aiming to maintain diversity within the population and prevent premature convergence.

After several generations, the GA converges to the best solution, ideally the optimal threshold segmentation.

In the problem at hand, the population's size has been chosen as 50 and the maximum number of generation Q=25, but GA can stop if it has already converged towards the same solution. NMSE is used as a fitness function.

The motivation behind the use of GA in our study is to add, along with the brute-force, another process from different optimization approach, which will be used for the comparison subject to validate the effectiveness of the proposed approach, which will be discussed in the next section.

3.4.6 Proposed Approach

3.4.6.1 Principle of Unidirectional Minimization

In the proposed approach, a unidirectional minimization is achieved using the golden section (GS) search.

The GS search is an optimization technique used to find the optimum for a strictly unimodal function of a single variable over a search interval without using derivatives [72].

For a given unimodal function $f(x)$, the principle of GS search consists of finding the optimum x_{opt} which corresponds to the minimum value of $f(x)$, by iteratively narrowing the range of the search interval $[a\ b]$ containing that optimum until a specified accuracy is reached.

The GS search derives its name from the fact that the points determining the search interval are computed using the golden ratio τ, which is equal to $\frac{1+\sqrt{5}}{2}$ and verifies $\tau - 1 = \frac{1}{\tau}$. The golden ratio τ is intimately involved in the Fibonacci sequence, in which the ratios of successive terms of the Fibonacci sequence possess an impressive proportion, e.g., 1.618, or its inverse 0.618.

Therefore, the length of the search interval is iteratively reduced by a factor τ. At q^{th} iteration the search interval is noted $[a_q\ b_q]$. Denoting $\varphi = \frac{1}{\tau} = 0.618$, two intermediate points are used to update the range of the search interval. They are determined by:

$$\begin{cases} x_1 = a_q + \varphi^2(b_q - a_q) \\ x_2 = a_q + \varphi(b_q - a_q) \end{cases} \tag{3.6}$$

The functions $f(x_1)$ and $f(x_2)$ are computed and compared as described in Algorithm (2).

Algorithm 2: Algorithm of GS search

$q = 0$; $a_0 = a$; $b_0 = b$
Compute x_1 and x_2
while $|b_q - a_q| > \epsilon$ **do**
  Compute $f(x_1)$ and $f(x_2)$
  if $f(x_1) \leq f(x_2)$ **then**
    $a_{q+1} = a_q$
    $b_{q+1} = x_2$
    $x_2 = x_1$
    Compute x_1
  end
  if $f(x_1) > f(x_2)$ **then**
    $a_{q+1} = x_1$
    $b_{q+1} = b_q$
    $x_1 = x_2$
    Compute x_2
  end
  $q = q + 1$
end
$x_{opt} = \frac{a_q + b_q}{2}$

The computation process of $f(x_1)$ and $f(x_2)$ is iteratively executed and the search interval $[a\ b]$ is narrowing until the accuracy $|b_q - a_q| < \epsilon$ is reached where ϵ is a specified small value, then takes $x_{opt} = \frac{a_q + b_q}{2}$ as an approximate optimum point and $f(x_{opt})$ is the approximate optimum value.

In this work, the GS search considers NMSE as the function $f(x)$ and the range of the normalized magnitude of the baseband input IQ as the search interval $[a\ b]$.

Another advantage of the GS search is that it requires only one computation of $f(x)$ at each iteration, except for the first one where $f(a_0)$, $f(x_1)$, $f(x_2)$ and $f(b_0)$ have to be evaluated.

3.4.6.2 optimization of a single threshold

Let us consider the case of 2 segments. In this case, we have only a single threshold to determine.

Figure 3.6 presents the evolution of GS search in terms of NMSE over the intermediate points $x_{1_{(q)}}$ and $x_{2_{(q)}}$ in the two first iterations and the final iteration. The blue curve presents the NMSE of postdistortion for every position of the threshold between 0 and 1. The red curve presents the AM-AM characteristic of the PA.

Initially, the search interval $[a\ b] = [0\ 1]$. At the first iteration, the search interval $[a\ b]$ is updated with the narrower interval $[a_1\ 1]$. After 34 iterations, $[a_{34}\ b_{34}]$ fulfills the condition $|a_{34} - b_{34}| < \epsilon$ that allows the GS search to stop and return $\beta_{opt} = \frac{a_{34} + b_{34}}{2}$ as the optimized threshold. According to Figure 3.6, the optimal threshold is located at 0.8.

Thus, the NMSE from uniform segmentation to optimal segmentation is improved by nearly 4 dB, as shown in Figure 3.6.

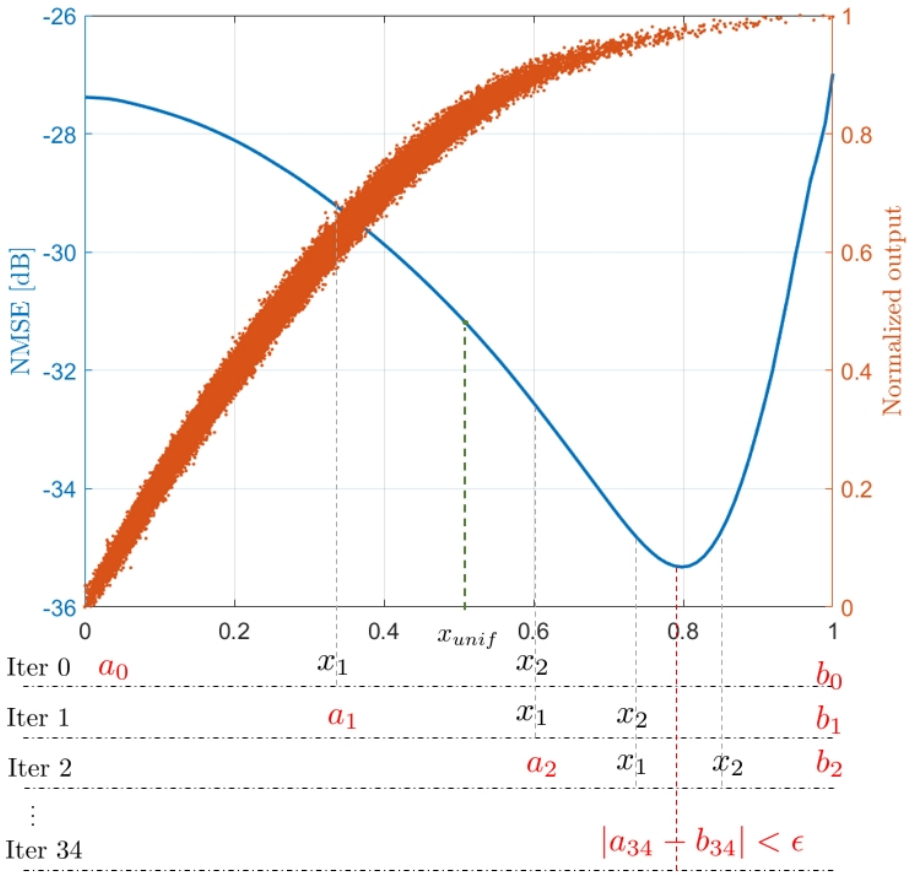


Figure 3.6: Evolution of GS search in terms of NMSE for $K = 2$ with ϵ=0.1

On the other hand, Figure 3.7 shows the behavior of ACPR_{L1} and ACPR_{R1} according to the variation of the threshold position over the normalized input interval. As we can see, the behavior of ACPR_{L1} and ACPR_{R1} are unimodal in which the optimal threshold is located at 0.79, which is close to the optimal thresholds from Figure 3.6.

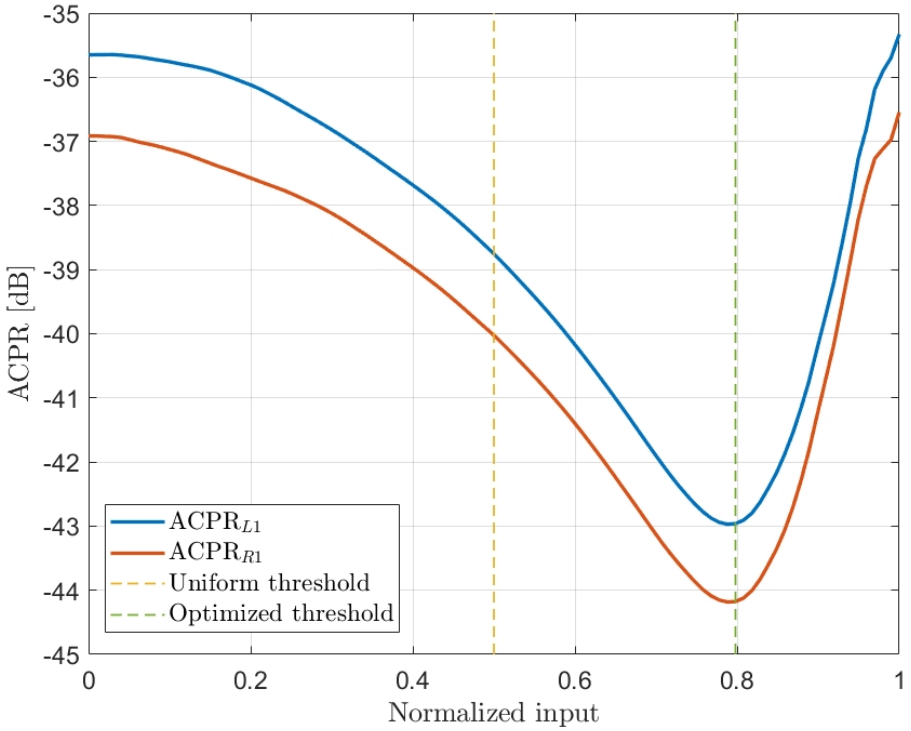


Figure 3.7: Behavior of ACPR_{L1} and ACPR_{R1} over the normalized input

3.4.6.3 General case with K segments and K-1 thresholds

In the general case for K segments, we have $K-1$ thresholds to optimize: $\boldsymbol{\beta} = \beta_1,\beta_2,...,\beta_{K-1}$. We propose to optimize the thresholds successively one by one, starting with an initial uniform segmentation. The optimization interval I_i of a selected β_i is $[\beta_{i-1}\ \beta_{i+1}]$. In a normalized interval, $\beta_0 = 0$ and $\beta_K = 1$ by definition.

Each β_i contributes to the overall NMSE, furthermore as β_i is the bound of the optimization interval for β_{i-1} and β_{i+1} so the optimization of β_i will in turn impact the optimal position of β_{i-1} and β_{i+1} that should be re-evaluated. That is why we propose a heuristic to handle this situation: after having optimized the last threshold β_{K-1}, the algorithm is run in reverse order back to β_1, then again up to β_{K-1} and so on until the segmentation has converged.

Algorithm 3: Algorithm of the proposed approach

Given K, M_{lin} and M for the DVR model
Initialization β_i, ϵ
Set loop counter q=1
while *(1)* **do**
  for $i = 1, 2, ..., K-2, K-1$ **do**
    $\beta_{i(q)} = \underset{\beta_i \in [\beta_{i-1}(q), \beta_{i+1}(q)]}{argmin} \text{NMSE}(\beta_i)$
  end
  $q = q + 1$
  $\beta_{K-1}(q) = \beta_{K-1}(q-1)$
  for $K-1, K-2, ..., 2, 1$ **do**
    $\beta_{i(q)} = \underset{\beta_i \in [\beta_{i-1}(q), \beta_{i+1}(q)]}{argmin} \text{NMSE}(\beta_i)$
  end
  $q = q + 1$
  $\beta_1(q) = \beta_1(q-1)$
  Evaluate NMSE($\beta_{(q)}$)
  if $NMSE(q) > NMSE(q-1) - \epsilon$ **then**
    end while loop
  end
end

Given the DVR model in (3.5) defined by K, M and the thresholds $\beta_1, \cdots, \beta_{K-1}$. The function NMSE($\beta_i$) corresponds to the different value of NMSE, obtained when varying β_i from β_{i-1} to β_{i+1}. The minimum of this function is obtained with $\underset{\beta_i \in [\beta_{i-1}, \beta_{i+1}]}{\text{argmin}} \text{NMSE}(\beta_i)$.

The algorithm for the proposed approach is described in Algorithm (3) in which the steps $\beta_{K-1}(q) = \beta_{K-1}(q-1)$ and $\beta_1(q) = \beta_1(q-1)$ are introduced for notation consistency.

3.4.6.4 Complexity of GS search

The complexity of the GS search is an important aspect to be considered in the thresholds optimization process. In this work, the search complexity is assessed by three features:

- The number of NMSE computations, which are required to identify the model coefficients.
- The number of interval updates, which is defined by the number of times the search interval is updated, except at the first iteration when one interval update corresponds to one NMSE evaluation.

- The number of GS iterations, which is defined by the number of times the thresholds vector $[\beta_1, \cdots, \beta_{K-1}]$ is optimized. It is represented by the parameter q in Algorithm (3).

3.4.6.5 Implementation Strategy of GS search

The proposed approach based on GS search is seen as an offline process to determine the DPD model for the PA linearization.

In the case of the DVR model, we determine the model parameters, we optimize the thresholds $\boldsymbol{\beta}$, then, the DVR model is ready to be inserted as a DPD model according to ILA, in which the model is used in 2 blocks: predistorter and postdistorter, from which both must contain the optimized thresholds $\boldsymbol{\beta}_{opt}$.

For real-time DPD adaptation, GS search is only required to be performed at the initial training when the global characteristics of the PA are determined. The joint architecture of DPD adaptation, which is based on ILA and the proposed approach to optimize the thresholds, is shown in Figure 3.8.

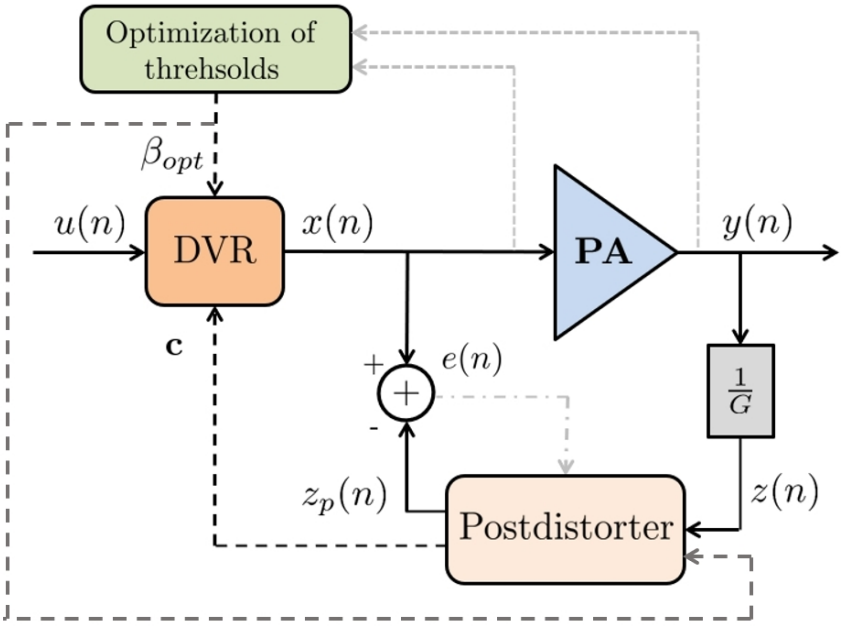


Figure 3.8: Joint architecture of ILA and thresholds optimization process

Moreover, this joint architecture is described in Algorithm (4).

Algorithm 4: Algorithm of ILA and the proposed approach

Determine K, P, M_{lin}, and M for the DVR model
Get input x_0 and output y_0 of PA
Define the PA gain G
Optimize $\boldsymbol{\beta}$ using the proposed approach
Determine DPD iterations Q
for $i = Q$ **do**
    $z_i = y_i/G$ Define regressor matrix $\mathbf{Z}_i$
    Estimate coefficient
    $\mathbf{c}_i = [\mathbf{Z}_i^H \mathbf{Z}_i]^{-1} \mathbf{Z}_i^H x_i$
    $z_p = \mathbf{Z}_i \mathbf{c}_i$
    Compute NMSE between x_i and z_p
    Copy c_i to DVR model
    Get x_i and y_i
    Evaluate linearization performance
end

However, it is recommended to rerun the proposed approach once the input u or the PA changes any of its characteristics, especially output power, signal bandwidth, center frequency, etc.

3.4.7 Comparison of Linearization Performances

In the following, we set K to 4, $M_{lin} = 2$ and $M = 1$.

The results of the proposed approach using GS search are discussed and compared to those of a uniform segmentation and the optimal segmentations obtained from brute-force search and GA.

The results are summarized in Table 3.1 and discussed in terms of NMSE and ACPR. The algorithm complexity is also discussed and evaluated by the number of computations of the objective function.

Table 3.1: Comparison of linearization performances, optimal solutions and complexity

		Uniform segmentation	Brute-force	GA	GS search
Segmentation		[0.25 0.5 0.75]	[0.58 0.81 0.92]	[0.59 0.81 0.92]	[0.59 0.81 0.92]
NMSE (dB)		-35.82	-38.69	-38.69	-38.69
ACPR(dB)	L1	-43.93	-48.52	-48.51	-48.49
	U1	-44.91	-48.83	-48.83	-48.82
NMSE comp.		1	161700	2404	341

The first column corresponds to the uniform segmentation. The next column represents brute-force search which has been performed by dividing the input range into 100 steps. The third column represents the results of GA, which has been performed with a population size of up to 50 and a maximum number of generations up to 25. The last column presents the results of the proposed approach.

In the scenario at hand, the proposed approach has converged towards the same optimal segmentation returned by brute-force search and GA, i.e., $\boldsymbol{\beta}_{opt}$ = [0.59 0.81 0.92]. According to Table 3.1, only 341 objective function evaluations are performed to optimize the thresholds, while GA requires 2404 evaluations and brute-force requires 161700 evaluations to find the exact optimal results.

Compared to uniform segmentation, the NMSE of the proposed approach has been significantly improved by nearly 3 dB. The ACPR is improved by more than 4 dB as well. This is can be confirmed in Figure 3.4 and Figure 3.5 for $K = 4$.

The linearization performances of the optimal segmentation $\boldsymbol{\beta}_{opt}$ are presented in Figure 3.9 and Figure 3.10 where they are compared to that of the uniform segmentation $\boldsymbol{\beta}_{unif} = [0.25\ 0.5\ 0.75]$.

Figure 3.9 presents the spectrum of input and output of PA and the spectrum of the output of predistorter with uniform and optimized segmentation.

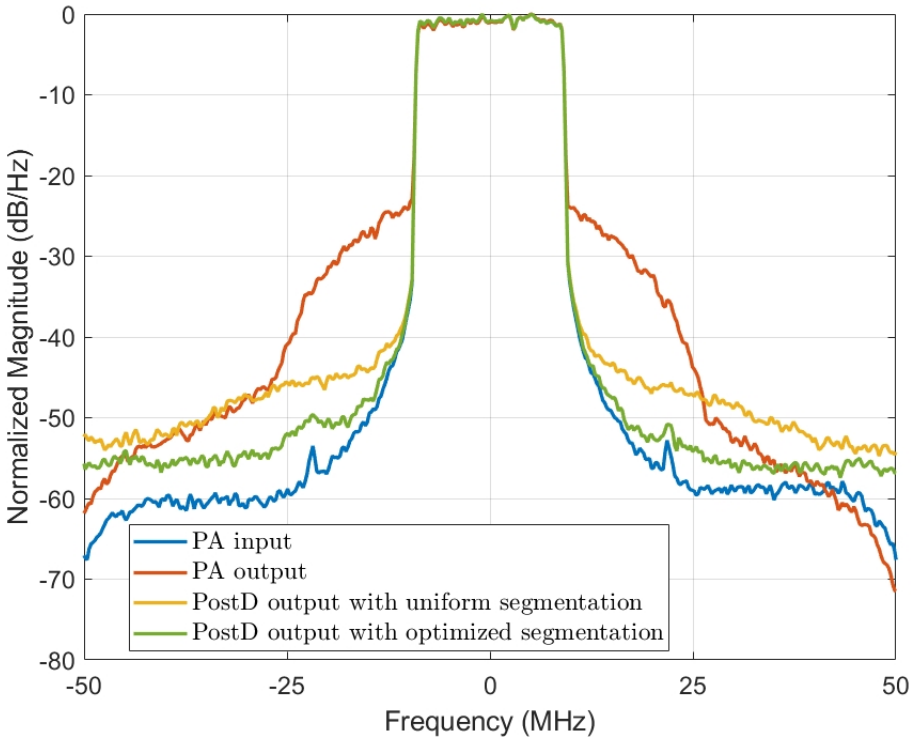


Figure 3.9: Spectra of PA input-output and postdistorter output with uniform and optimized segmentation

Figure 3.10 presents the AM-AM curve of the PA and the AM-AM of the predistorter with uniform and optimized segmentation, which are plotted as well.

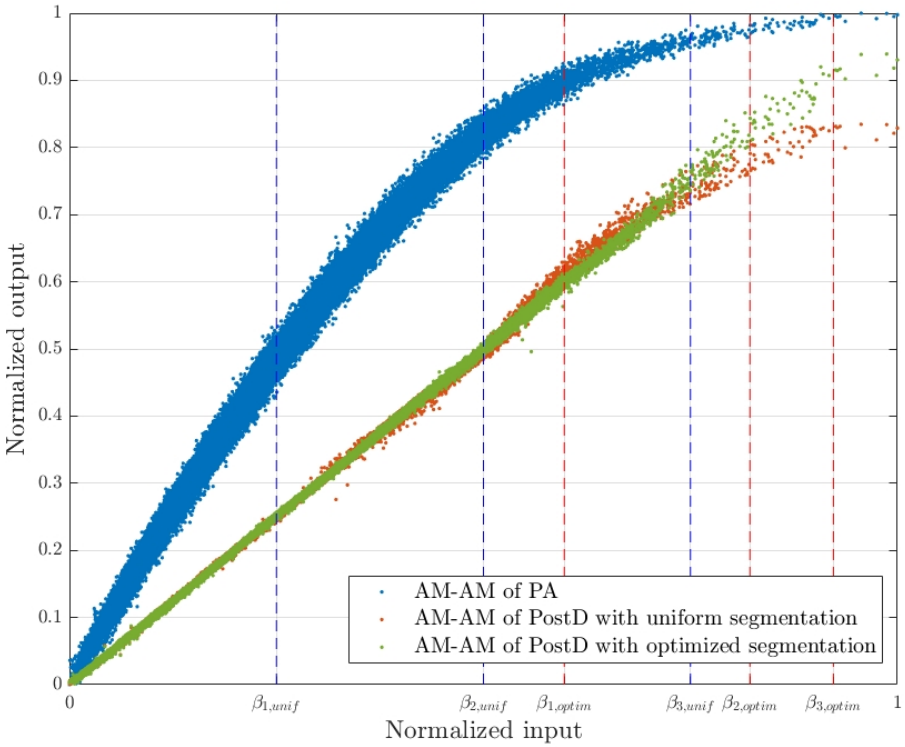


Figure 3.10: AM-AM of PA and postdistorter with uniform and optimized segmentation

3.4.8 Analysis and Improvement of the Proposed Approach

This section highlights the behavior of the proposed approach throughout its convergence and how the optimization problem is decomposed into a set of unimodal sub-problems. Furthermore, a stochastic process is proposed to select the initial segmentation that allows accelerating the convergence of the proposed approach.

3.4.8.1 Behavior of the Proposed Approach

The initial uniform segmentation is

$$[\beta_{1(0)}\ \beta_{2(0)}\ \beta_{3(0)}] = [0.25\ 0.5\ 0.75]$$

The optimization interval vector is

$$I_{(0)} = [I_{1(0)}; I_{2(0)}; I_{3(0)}] = [[0\ 0.5]; [0.25\ 0.75]; [0.5\ 1]]$$

Figure 3.11 presents the behavior of NMSE(β_i) on each interval.

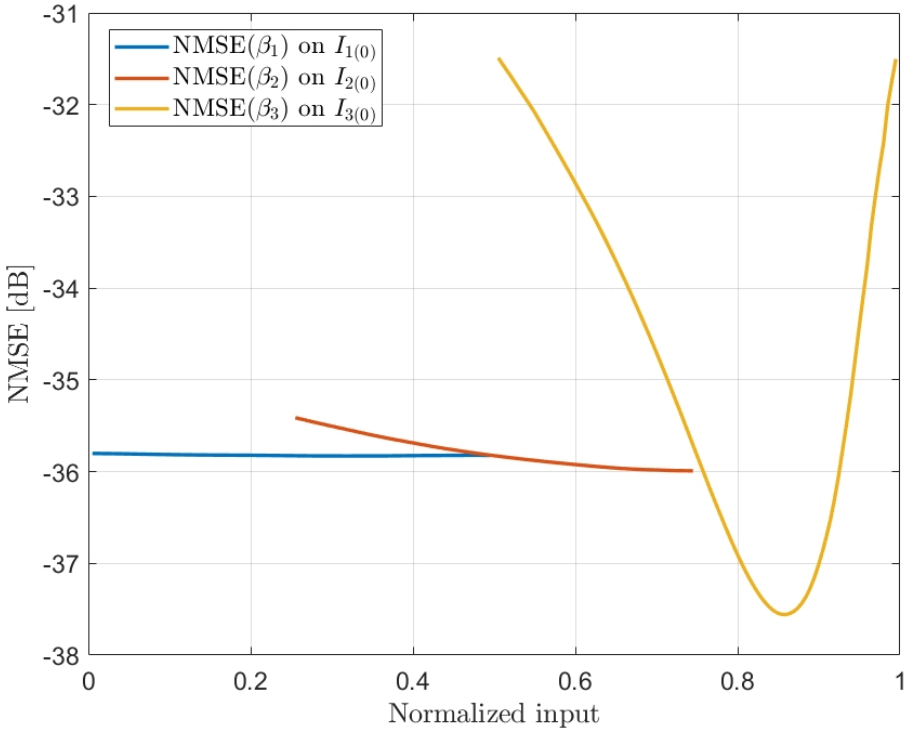


Figure 3.11: NMSE(β_i) on $I_{(0)}$

It can be seen from Figure 3.12 that NMSE(β_1) is not unimodal on $I_{1(0)}$.

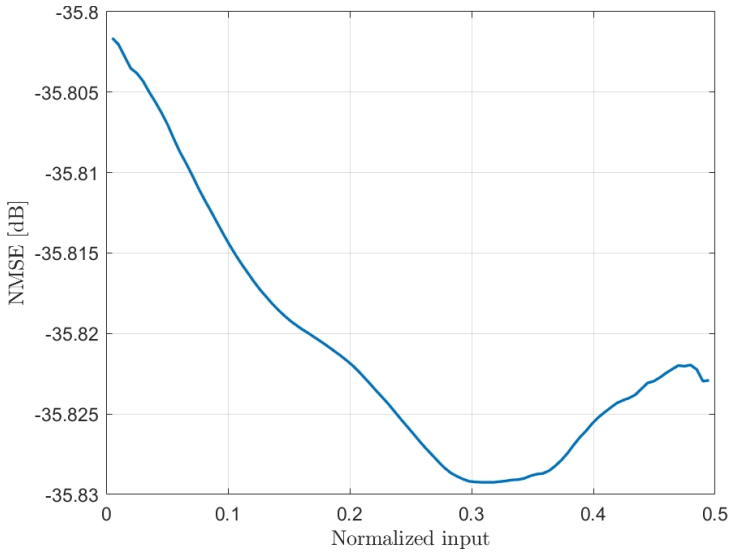


Figure 3.12: Zoom of NMSE(β_1) on $I_{1(0)}$

After the optimization of $[\beta_{1(0)}\ \beta_{2(0)}\ \beta_{3(0)}]$ to

$$[\beta_{1(1)}\ \beta_{2(1)}\ \beta_{3(1)}] = [0.31\ 0.74\ 0.90]$$

the behavior of NMSE on the updated optimization interval vector

$$I_{(1)} = [I_{1(1)}; I_{2(1)}; I_{3(1)}] = [[0\ 0.74]; [0.31\ 0.90]; [0.74\ 1]]$$

is presented in Figure 3.13, which shows that the behavior of NMSE(β_1) on $I_{1(1)}$ becomes unimodal.

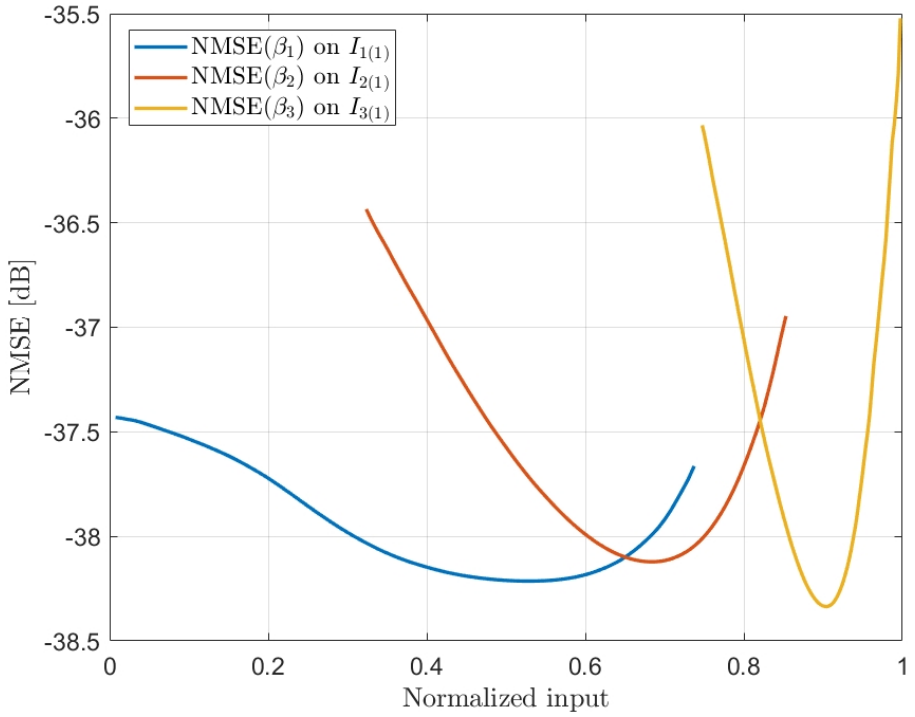


Figure 3.13: NMSE(β_i) on $I_{(1)}$

After the optimization of $[\beta_{1(1)}\ \beta_{2(1)}\ \beta_{3(1)}]$ to

$$[\beta_{1(2)}\ \beta_{2(2)}\ \beta_{3(2)}] = [0.52\ 0.72\ 0.90]$$

for which the functions NMSE(β_i) are shown in Figure 3.14.

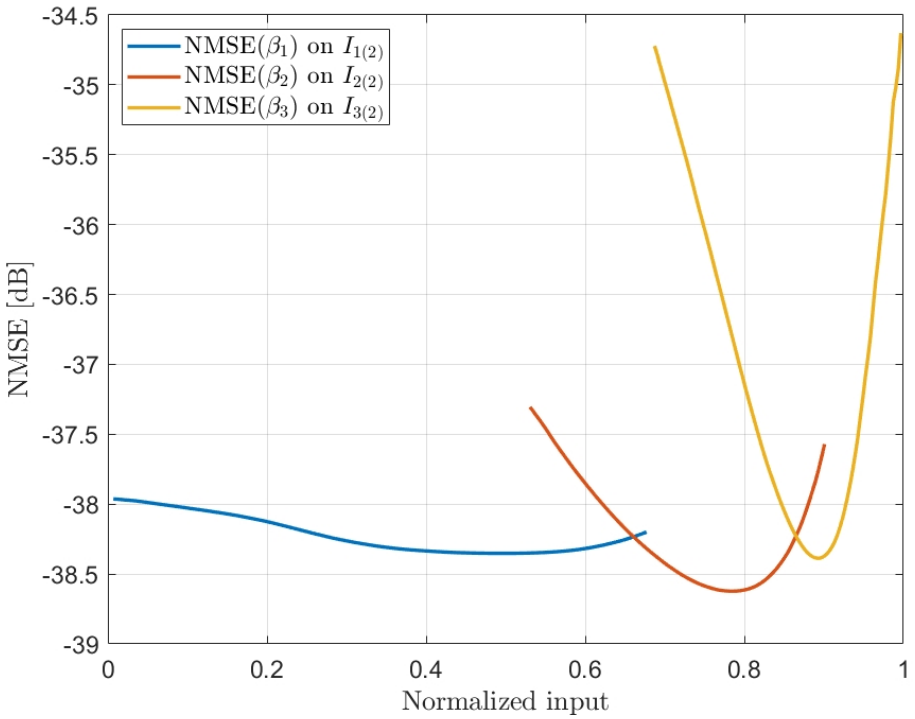


Figure 3.14: NMSE(β_i) on $I_{(2)}$

Hence this example highlights the following interesting behavior, even if each sub-problem is not unimodal initially, they become unimodal as each sub-problem converges towards its optimum.

Figure 3.15 presents the evolution of thresholds β_1, β_2, β_3 and NMSE where the optimization of thresholds has been performed in direct order:

$$[\beta_1; \beta_2; \beta_3; \beta_2; \beta_1; \beta_2; ...]$$

until the algorithm has converged.

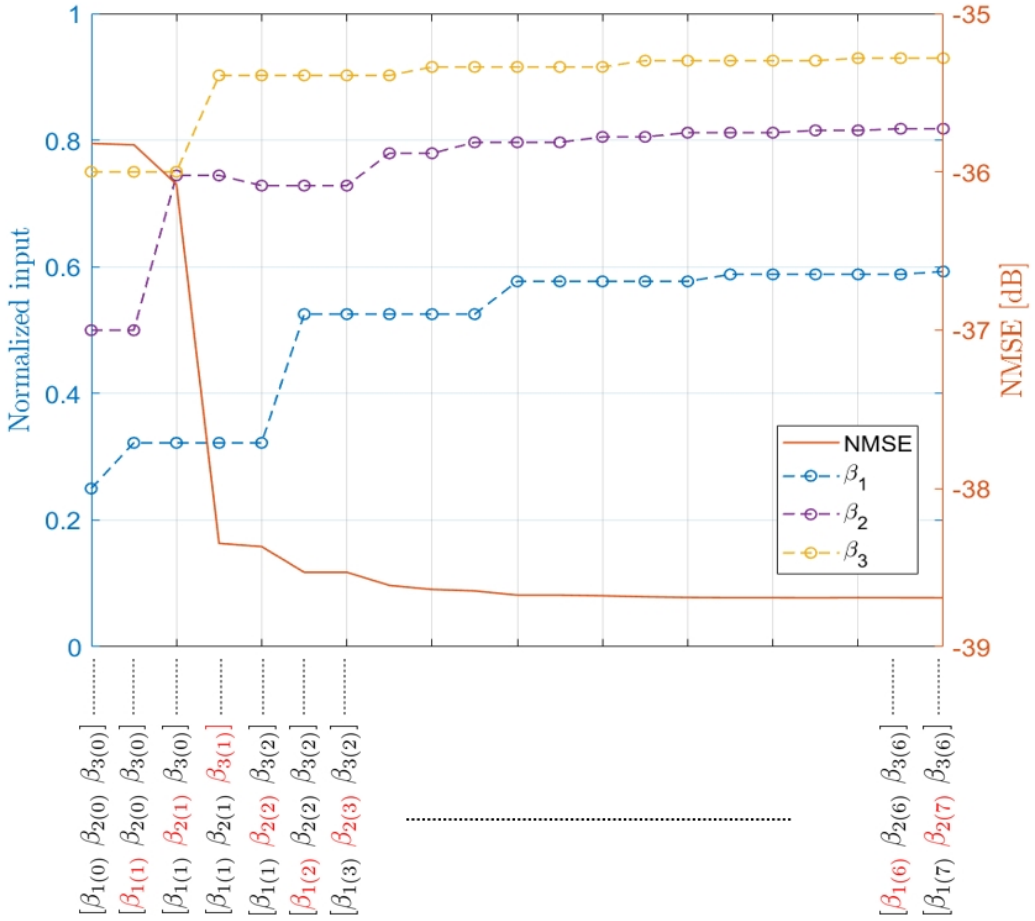


Figure 3.15: Evolution of thresholds and NMSE versus iterations

3.4.8.2 Random Selection

In the previous section, it has been shown how the optimization problem has been decomposed into a set of unimodal sub-problems on each interval, starting from a non-unimodal problem given by the uniform segmentation.

This leads to the fact that the initial segmentation choice may be desirable to start with a segmentation that can give a unimodal criterion on each interval from the first iteration, speeding up the algorithm's convergence and decreasing the execution time.

For that, a stochastic selection process is proposed: N trials of random segmentation are performed, and the best one in terms of NMSE is retained as the initial segmentation for the proposed algorithm. To validate the effectiveness of the stochastic process, 100 experiments of threshold optimization with a random selection have been performed for N=50.

Figure 3.16 presents the 100 tests, which are plotted in colors. Each test starts with uniform segmentation, e.g., $\beta_{initial}$ = [0.25 0.5 0.75], with NMSE = -35.82 dB, and perform 50 trials with random segmentation. The NMSE is evaluated at each trial, and the best is retained as long as no better NMSE has been found.

As shown in Figure 3.16, this stochastic process could significantly improve NMSE, thanks to a good initial segmentation. Out of the 100 tests performed, the worst NMSE is -37.69 dB, which is still better than NMSE from the uniform segmentation.

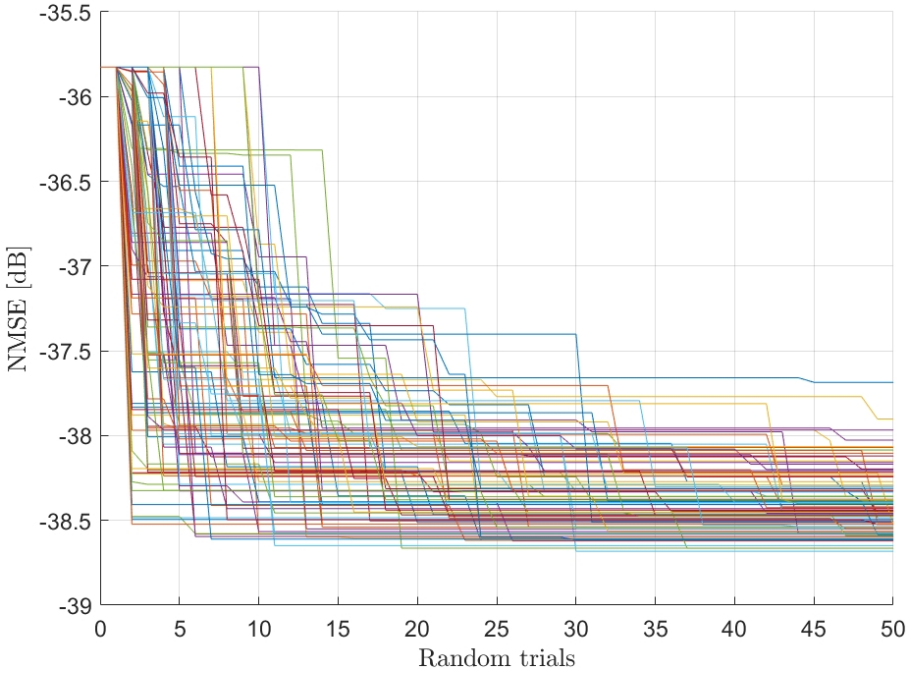


Figure 3.16: Behavior of NMSE during random selection for 100 times

On the other hand, the starting point of the NMSE set plotted in Figure 3.16 is NMSE from the uniform segmentation, and the convergence point is on the NMSE interval of [-37.69 -38.68] dB. This makes it possible to roughly conclude that the number of random trials sufficient for the NMSE to converge to the NMSE interval [-37.69 -38.68] dB is 30 trials, allowing it to be taken a stochastic indicator to determine the number of random tests required in the random selection.

Furthermore, Table 3.2 Highlights that the GS search requires 4 GS iterations with 341 NMSE computations starting from the uniform segmentation. In comparison, it requires only 1 or 2 GS iterations with a total number of 135 or 219 NMSE computations (including the 50 NMSE computations from the random trials) for the best and worst case of the 100 experiments considered.

Table 3.2: Comparison of linearization performances, optimal solutions, and complexity with random selection

		Uniform segmentation	50 trials of random segmentation	
			Best case	Worst case
Segmentation before starting GS		[0.25 0.5 0.75]	[0.61 0.82 0.93]	[0.64 0.76 0.82]
NMSE (dB)		-35.82	-38.68	-37.68
Apply GS search				
Optimal Segmentation		[0.59 0.81 0.92]	[0.59 0.81 0.92]	[0.59 0.81 0.92]
NMSE (dB)		-38.69	-38.69	-38.69
ACPR (dB)	L1	-48.49	-48.49	-48.49
	U1	-48.82	-49.82	-49.82
Complexity	GS iter.	4	1	2
	Interval upd.	171	43	85
	NMSE comp.	341	135(=50+85)	219(=50+169)

As the number of NMSE computations becomes a random variable using stochastic initialization, it is necessary to look for the probability distribution function that best fits the experimental data to determine an upper bound on the expected number of NMSE computations.

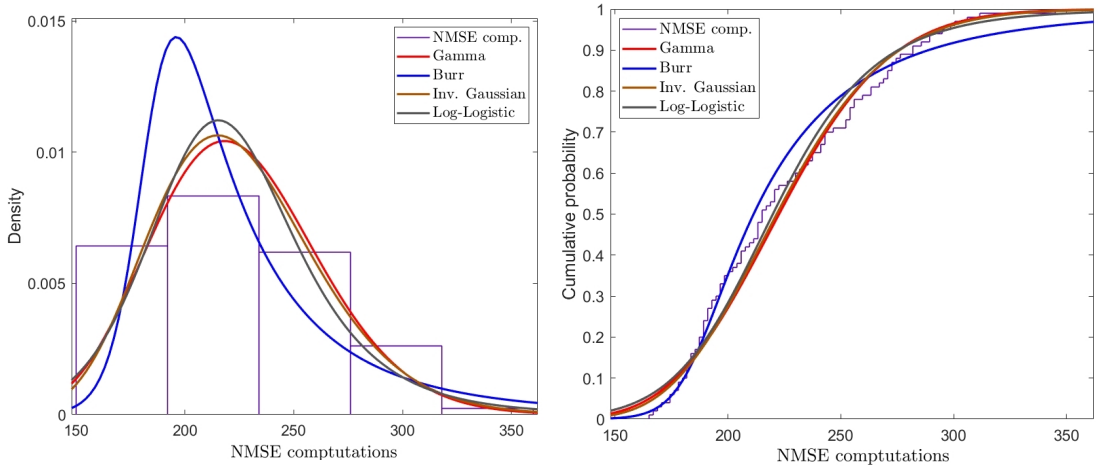
We have investigated the probability $p(N < X)$ that the number of NMSE computations N required for the GS search to be less than an upper bound X.

Several experiments have been carried out with different random trials as follows:

- Experiment 1: (Random selection with 10 trials + GS search) $\times$ 100 times
- Experiment 2: (Random selection with 30 trials + GS search) $\times$ 100 times
- Experiment 3: (Random selection with 50 trials + GS search) $\times$ 100 times

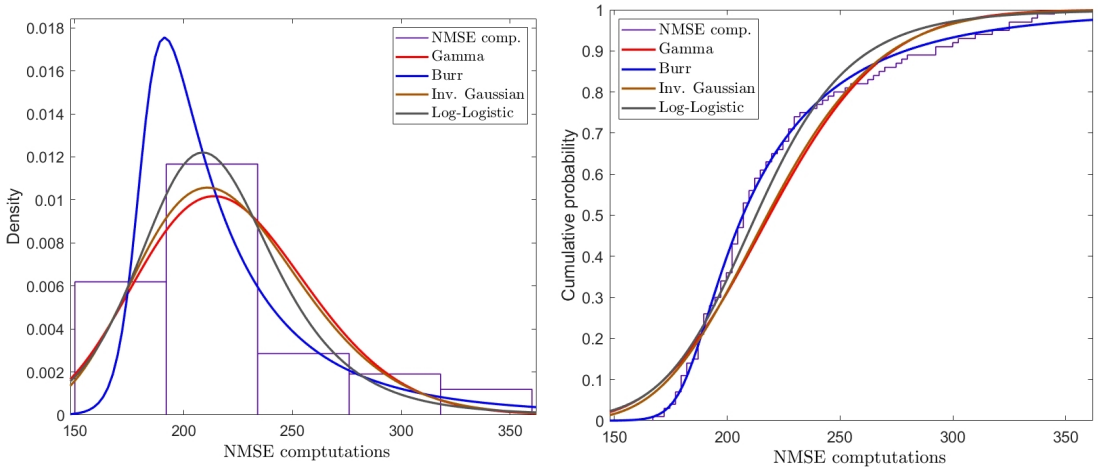
In this study, we present two figures of merit: the average number of NMSE computations and the confidence bound X, where some probability distributions are used to fit the number of NMSE computations, such as gamma distribution [73], Burr distribution [74], inverse Gaussian [75], and log-logistic distribution [76].

Figure 3.17, 3.18, and 3.19 present the probability distribution fitting to the number of NMSE computations for the experiment 1, experiment 2, and experiment 3, respectively, where probability density function (PDF) and cumulative distribution function (CDF) are shown.



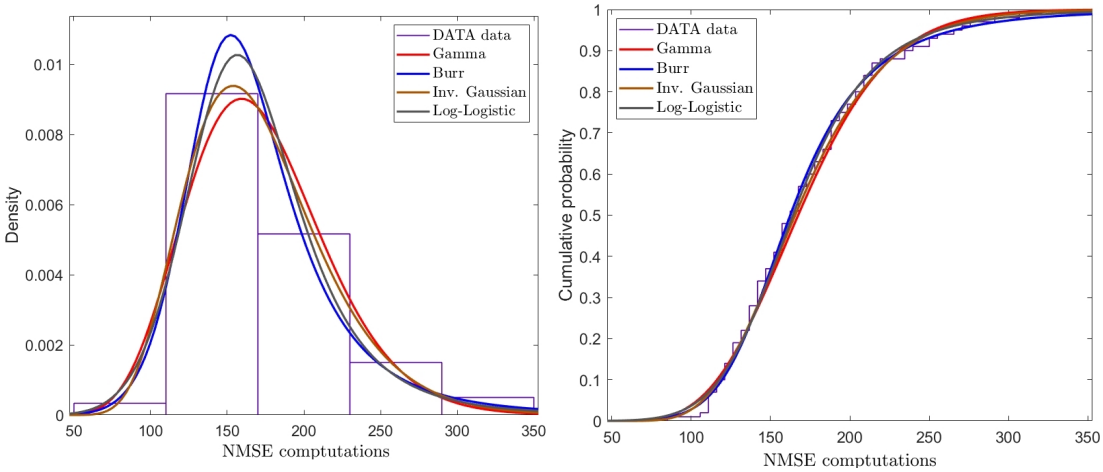
(a) PDF (b) CDF

Figure 3.17: Fit probability distribution to data from experiment 1



(a) PDF (b) CDF

Figure 3.18: Fit probability distribution to data from experiment 2



(a) PDF (b) CDF

Figure 3.19: Fit probability distribution to data from experiment 3

Table 3.3 presents in the two first columns the average number of NMSE computations for each experiment and according to each distribution and its variance. The fourth and

fifth columns present the probability of the required number of NMSE computations below or equal to the given value.

Table 3.3: Probability distribution fitting to number of NMSE computations

	Mean	Var	X|p(N< X)=0.99	X|p(N< X)=0.999
Experiment 1: 10 random trials × 100				
Gamma	225	1495	325	364
Burr	227	3426	451	701
Inv. Gaussian	225	1502	330	375
Log-Logistic	224	1801	352	446
Experiment 2: 30 random trials × 100				
Gamma	221	1580	324	365
Burr	222	2825	427	654
Inv. Gaussian	221	1527	328	373
Log-Logistic	216	1506	333	417
Experiment 3: 50 random trials × 100				
Gamma	171	2078	295	348
Burr	172	2942	362	559
Inv. Gaussian	171	2121	306	371
Log-Logistic	170	2421	330	468

As shown in Figure 3.17, 3.18, and 3.19, it turns out that a Burr distribution fits well the number of NMSE computations with a mean of 227 and 222 and a variance of 3426 and 2825 for experiments 1 and 2, while for experiment 3, Burr distribution and log-logistic distribution have similar behavior in fitting the number of NMSE computations.

With these parameters, the upper bound X on the number of NMSE computations is 451 with a probability of 0.99, and 701 with a probability of 0.999 for experiment 1, where 10 random trials are performed before GS search. For experiment 2 with 30 random trials, the upper bound X is 427 with a probability of 0.99 and 654 with a probability of 0.999.

Regarding experiment 3, it has an upper bound X of 362 with a probability of 0.99, and 559 with a probability of 0.999 through Burr distribution, while with log-logistic distribution, it has an upper bound X of 330 with a probability of 0.99, and 468 with a probability of 0.999

Therefore, we can conclude that the interval [30 50] makes a sufficiently good range for choosing the number of random trials before applying the GS search.

Besides, using a stochastic initialization with 30 trials, the average number of NMSE computations will be 222 and below 427 with a probability of 0.99 or 654 with a probability of 0.999.

After the random selection process, the behavior of each NMSE(β_i) on their interval $I_{i(0)}$ are likely to be unimodal before starting the GS algorithm, as we can see in Figure 3.20. Experiments have shown that some tens of random selection trials are likely to provide a starting point that significantly improves the initial NMSE in comparison with the uniform segmentation.

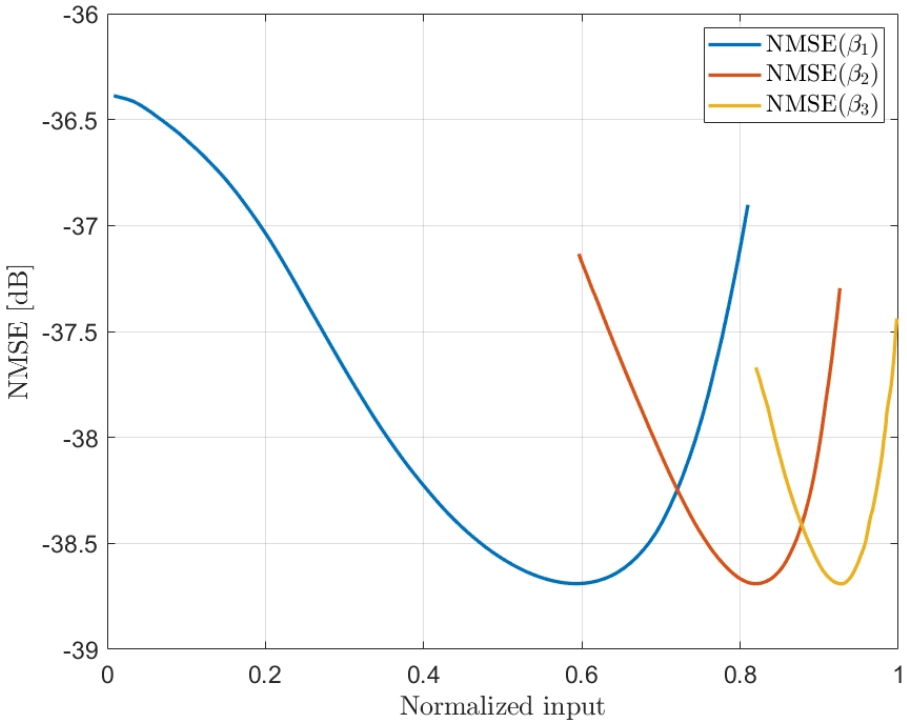


Figure 3.20: NMSE versus variation of β_i over I_i after random selection process where each NMSE(β_i) are unimodal

It is worth noting that for every 100 experiments of the stochastic initialization, the proposed optimization approach has converged towards the same segmentation. This illustrates its convergence behavior.

3.4.9 Sensitivity of Thresholds to Number of Segments

Nevertheless, the proposed approach to optimizing $\boldsymbol{\beta}^{(K)}$ is only applied when the number of segments K is not changing. However, in some particular processes, such as sizing the DVR model or when the characteristics of the input signal are modified, it may be necessary to change the number of segments K along with the optimization process's running.

Once K changes, the proposed approach must be executed to find the new optimal solution starting from uniform or random segmentation since it has been shown that both initial conditions converge towards the exact optimal solution.

Figure 3.22 compares the position of the optimal segmentation for different K from 2 to 10 with $M_{lin} = 2$ and $M = 1$. Note that for each K, the proposed approach to optimize $\boldsymbol{\beta}$ is started from the uniform segmentation.

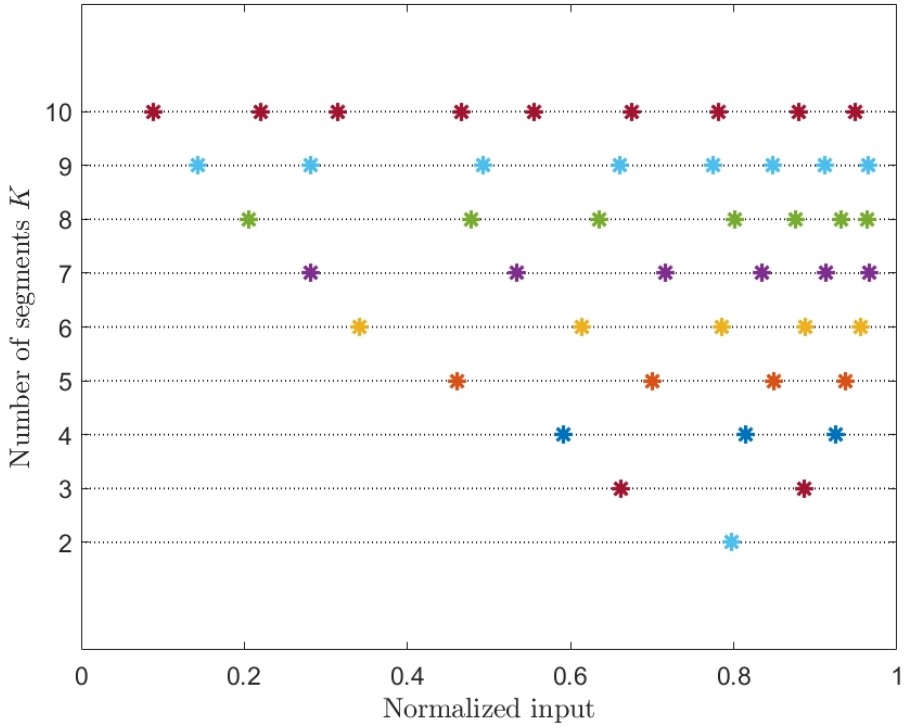


Figure 3.21: Optimal segmentation versus number of segments K

As can be seen, the optimal thresholds tend to be concentrated towards the nonlinear region of the PA. This empirical method carried out in Figure 3.21 allows to observe a repeated behavior, where when we go from K to $K+1$, the optimal position of the first threshold $\beta_{1_{opt}}^{(K+1)}$ is located in the interval $[0\ \beta_1^{(K)}]$, and vice versa, when going from K to $K-1$, the optimal segmentation $\boldsymbol{\beta}_{opt}^{(K-1)} = [\beta_{1_{opt}}^{(K-1)}, ..., \beta_{K-2_{opt}}^{(K-1)}]$ is located in the interval $[\beta_1^{(K)}\ 1]$.

On the other hand, it is essential to emphasize the complexity of the proposed approach for threshold optimization, which is presented in Table 3.4 for each K. The complexity is represented in terms of NMSE computation, interval updates, and GS iterations.

Table 3.4: Complexity of proposed approach for each K

Number of segments K	2	3	4	5	6	7	8	9	10
GS iterations	2	2	4	4	5	5	7	8	9
Interval update	21	56	171	221	346	407	642	912	1235
NMSE computations	41	111	341	441	691	813	1283	1823	2469

The complexity consistently increases with the number of segments K. The complexity required to optimize a given $\beta_k^{(K)}$ of K can be reduced by using the information of the optimal segmentation $\boldsymbol{\beta}_{opt}^{(K-1)}$ of $K-1$ or $\boldsymbol{\beta}_{opt}^{(K+1)}$ of $K+1$.

We run the GS search to optimize $\beta_k^{(K)}$ for each K starting from 2 to 10. We propose to define the initial segmentation of K in terms of the optimal segmentation of $K-1$, which can be defined as

$$\boldsymbol{\beta}_{initial}^{(K)} = [\beta_1^{(K)}; \cdots ; \beta_{K-1}^{(K)}] = [\beta_r; \beta_{1_{opt}}^{(K-1)}; \cdots ; \beta_{K-2_{opt}}^{(K-1)}]$$

where β_r is a thresholds that could be located randomly in the interval $[0\ \beta_{1_{opt}}^{(K-1)}]$. How-

ever, we propose to define it as

$$\beta_r = \frac{\beta_{1_{opt}}^{(K-1)}}{2}$$

As this process starts from $K = 2$, its initials segmentation is the uniform segmentation, i.e,

$$\beta_{initial}^{(2)} = [0.5]$$

.

Since the GS search always converges towards the optimal solution, we will focus on the impact of this approach on the complexity. Table 3.5 summarizes the complexity results of this process.

Table 3.5: Complexity of GS search for increasing K from 2 to 10

Number of segments K	2	3	4	5	6	7	8	9	10
GS iterations	2	2	2	3	2	1	1	1	1
Interval update	21	57	85	162	126	72	75	90	88
NMSE computations	41	113	169	323	256	143	149	179	175

Compared to the complexity results reported in Table 3.4, where the GS search optimizes each K starting from a random initial segmentation, the complexity in Table 3.5 confirms the interest of this method, where the complexity is significantly reduced.

In reverse order, we run the GS search starting from $K = 10$ to 2. The idea here is at K, the initial segmentation $\boldsymbol{\beta}_{initial}^{(K)}$ will be defined in terms of the optimal segmentation $\boldsymbol{\beta}_{opt}^{(K+1)}$ of $K + 1$, where we remove the first thresholds $\beta_{1_{opt}}^{(K+1)}$ from $\beta_{opt}^{(K+1)}$ and we define the rest as an $\boldsymbol{\beta}_{initial}^{(K)}$ of K as

$$\boldsymbol{\beta}_{initial}^{(K)} = [\beta_1^{(K)}; \cdots ; \beta_{K-1}^{(K)}] = [\beta_{2_{opt}}^{(K+1)}; \cdots ; \beta_{K_{opt}}^{(K+1)}]$$

Table 3.6 summarizes the results of complexity. As starting from $K = 10$ with random segmentation, its complexity is similar to that reported in the last row in Table 3.4.

As can be seen, GS iterations and NMSE computations are drastically reduced even though the process is started from $K = 10$ with significant complexity.

Table 3.6: Complexity of GS search for decreasing K from 10 to 2

Number of segments K	10	9	8	7	6	5	4	3	2
GS iterations	9	3	1	2	2	3	3	2	2
Interval update	1235	309	89	151	129	162	126	57	21
NMSE computations	2469	617	177	301	257	323	251	113	41

The application of this method to define the initial segmentation is confirmed by the results presented on complexity. Additionally, it can be a good alternative to the random selection process if the information of $\boldsymbol{\beta}_{opt}^{(K-1)}$ or $\boldsymbol{\beta}_{opt}^{(K+1)}$ is available.

Moreover, the interest of this method can be valuable in processes where K changes regularly, such as the sizing process of the DVR model, which will be illustrated in the next section.

3.4.10 Sensitivity of Thresholds to Memory Depth

Figure 3.22 compares the optimal position of each threshold for different memory depths M with $K = 4$ and $M_{lin} = 2$.

Figure 3.22 highlights an interesting behavior, showing that the optimal segmentation is not very sensitive to memory depth M. The optimization can be performed for the DVR model without memory to reduce the algorithm complexity, making it an important aspect to design an optimal DVR model.

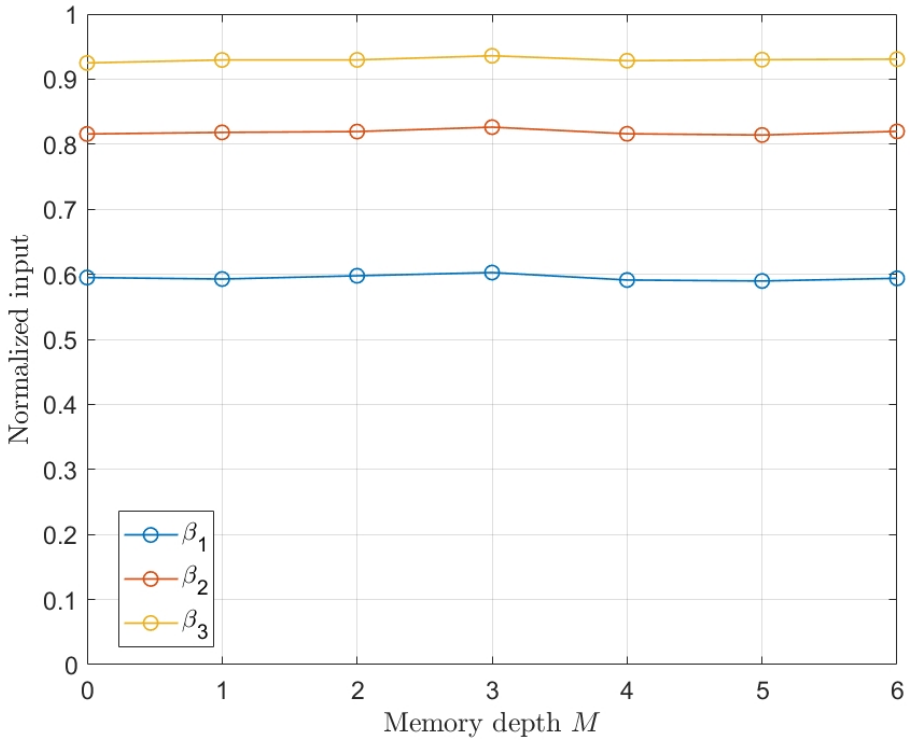


Figure 3.22: Optimal segmentation versus memory depth M

However, the impact of memory depth has also been noted in [69] without this being illustrated by data such as in Figure 3.22.

3.4.11 Sensitivity of Thresholds to Model Terms

In order to study the sensitivity of the optimal segmentation to the model terms T_S, we propose to optimize the thresholds for the following set:

$$
\begin{aligned}
T_{S_1} &= [T_{10}] \\
T_{S_2} &= [T_{10}\ T_{11}] \\
T_{S_3} &= [T_{10}\ T_{11}\ T_2] \\
T_{S_4} &= [T_{10}\ T_{11}\ T_2\ T_3] \\
T_{S_5} &= [T_{10}\ T_{11}\ T_2\ T_3\ T_4]
\end{aligned}
$$

with $K = 4$, $M_{lin} = 2$, and $M = 1$.

Figure 3.23 present the optimal position of each threshold for different T_{S_i}. Unlike the sensitivity of thresholds to the memory depth, the model terms T_S slightly impact the optimal segmentation. This impact is seen particularly over the variation of the first threshold β_1, the optimal position of which is situated in the interval of [0.47 0.61]. The optimal position of β_2 is less sensitive than that of β_1. The optimal position of β_3 remains almost stable with the model terms.

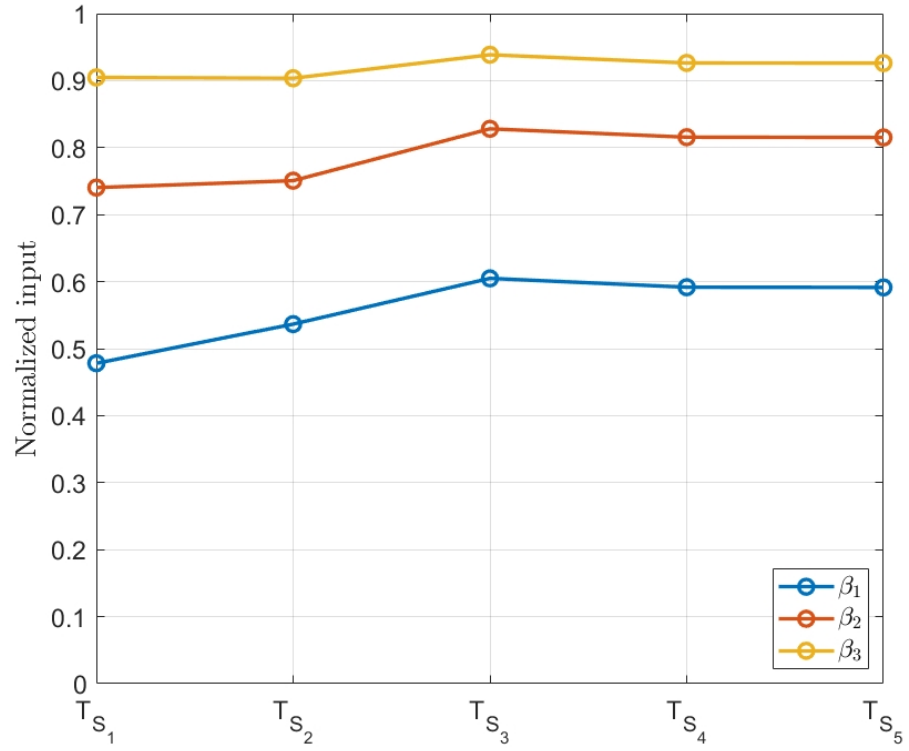


Figure 3.23: Optimal segmentation versus model terms T_S

On the other hand, Figure 3.24 shows the variation of NMSE for each model term T_{S_i}, for which the set of the optimal segmentation $\beta_{opt}(T_{S_i})$ from Figure 3.23 is applied.

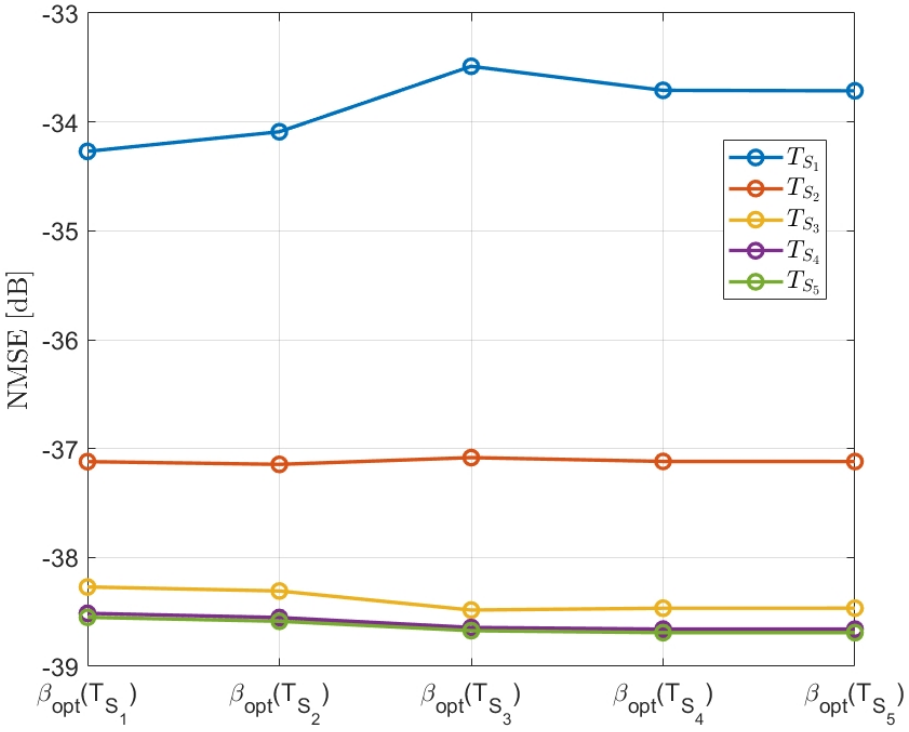


Figure 3.24: Optimal segmentation vs Terms

Except for T_{S_1}, NMSE does not vary significantly with the different optimal segmentation. For T_{S_1}, applying the optimal segmentation $\beta_{opt}(T_{S_3})$ slightly worsens the NMSE by around 0.8 dB, which can be considered negligible in some scenarios.

This analysis allows us to consider that the optimal segmentation is not firmly insensitive to the model terms used in T_S, even if their value remains strongly correlated.

Considering that the optimal segmentation is independent of all the model terms, a rough approximation is established to be used with caution to accelerate the convergence of an optimization process proposed in the next section, aiming to design an optimal DVR model.

3.4.12 Conclusion of Thresholds Optimization

In this framework, we showed that the optimal segmentation could significantly improve the linearization performances compared to the uniform segmentation for the DVR model at the same number of coefficients.

We have proposed a new approach for optimizing the thresholds based on the decomposition of the global optimization problem into a set of sub-problems where the objective function is unimodal. The GS search is used to solve each unimodal sub-problem. The proposed approach has been compared with GA and brute force. Both algorithms converge to the same optimal thresholds. However, our approach presents a much lower complexity than the GA algorithm making it an appropriate candidate to design the DVR model. Once the thresholds have been optimized for a given number of thresholds K, an important aspect arises as to whether this parameter K, as well as the other parameters of the DVR model, are optimal or not. Thus, it is necessary to determine the optimal structure of the DVR model that presents the best compromise between modeling accuracy and complexity. This will be the topic of the following section.

3.5 Sizing of DVR Model Structure

The DVR model sizing consists in finding an optimal model structure that leads to a promising modeling accuracy with reduced complexity.

The DVR model structure can be determined using an exhaustive search (brute force) by exploring all possible parameters order combinations. Setting the maximum order of K and M to K_{max} and M_{max}, respectively, the exhaustive search requires $K_{\max} \cdot M_{\max} \cdot L$ tests, where L presents the total number of possible combination of model terms, $L = \sum_{k=1}^{P+7} \binom{P+7}{k} = 2^{P+7} - 1$, according to (3.3) and (3.4).

Limiting the maximum number of coefficients to be tested could significantly reduce the search space. However, the exhaustive search remains an heavy computation load, and an optimization approach can significantly reduce that load.

3.5.1 State-of-the-Art

In the literature relating to behavioral modeling, it is reported that it is not easy to determine a given behavioral model structure which has reduced complexity and high linearization performance. This operation is known as model sizing, which can be carried out using mainly two methods: Selection of basis function or applying optimization algorithms to determine the model structure. According to a criterion or a cost function, these methods return the best model structure, which refers to a trade-off between the modeling accuracy and model complexity.

As is reported in Section 2.6.1.2, the complexity to solve (2.17) depends on the number of coefficients related to the number of basis functions. Some pruning techniques to reduce the number of basis functions while keeping the same modeling accuracy have been studied in [77] and [78]. For the MP model, a pruning technique is proposed in [79], where the total number of model kernels, which is determined by M and K, is minimized according to the difference between the measured and predicted signal.

Optimization algorithms can also determine the model sizing according to a good trade-off between modeling performance and model complexity. The modeling performance is represented mainly by a quadratic measure such as NMSE or ACPR and the model complexity by the model's number of coefficients. Different heuristics could be applied to determine the optimal structure of the DPD model. This optimization process is seen as a nonlinear integer optimization problem. These algorithms could be deterministic or stochastic with local or global search and aim to find an optimal solution in a reduced runtime. Among these heuristics, we can cite hill-climbing (HC), simulated annealing (SA), tabu search, evolutionary algorithms (such as GA, ant colony ...), particle swarm optimization (PSO).

The first work in finding the optimal structure for DPD models has been conducted for the GMP model using an integer GA [80]. In this algorithm, the fitness function is an additive weighted combination of NMSE and the number of coefficients. Compared to the exhaustive search, a solution close to the optimal model structure can be found much faster.

In [57], the authors proposed an algorithm based on HC heuristic aiming to find the optimal structure of the GMP model with a reduced number of computations. The searching criterion used in this study controls the HC heuristic to ensure a good trade-off between complexity and modeling accuracy of the GMP model. The HC heuristic provides an efficient approach to minimize the proposed criterion.

In [81], a comparative study of two optimization algorithms, HC algorithm, and genetic algorithm, has been done in terms of convergence speed and accuracy of the obtained solution. The two algorithms were performed using the same criterion and had very similar performances, which converge towards the global optimum quickly compared to an exhaustive search.

3.5.2 Hill-Climbing

The HC is an optimization algorithm that belongs to the family of iterative algorithms [82] - [83].

The principle of the HC algorithm consists of starting the search process from a given initial solution, determining its neighbors in a defined space, and choosing the best solution among these neighbors. This solution will be the initial solution in the next iteration, and the HC algorithm continues this process until no better solution is found. The definition of the neighborhood is a very important aspect of the HC algorithm.

The principle of the HC algorithm is described in Algorithm (5).

Algorithm 5: Algorithm of hill-climbing

Choose the initial element $x_{0(1)}$
Evaluate the initial cost function $f(x_{0(1)})$
while *(1)* **do**
  Determine the neighbors of $x_{0(q)}$: $x_{i(q)} \in V_{(q)}$
  Evaluate the cost function $f(x_{i(q)})$
  $x_{s(q)} = \text{argmin}_{x_{i(q)} \in V_{(q)}} \left(f(x_{i(q)})\right)$
  if $f(x_{s(q)}) < f(x_{s(q-1)})$ **then**
    $q = q + 1$
    $x_{0(q)} = x_{s(q-1)}$
  else
    end while loop
  end
end
Take the last solution $x_{s(q)}$ as the best element

In this study, The proposed approach based on the HC algorithm in [84] is extended to the DVR model to find an optimal structure with reduced complexity and high modeling accuracy.

According to Algorithm (5), at the first iteration, the HC algorithm starts from a given initial element $x_{0(1)}$ which represents a DVR model structure, determines the neighbors of $x_{0(1)}$ in the neighbor's subspace V, and evaluates the cost function of these neighbors. The best cost function $f(x_{s(1)})$ of the solution $x_{s(1)}$ is compared with $f(x_{0(1)})$. If $f(x_{s(1)})$ is lower than $f(x_{0(1)})$, the solution $x_{s(1)}$ becomes $x_{0(2)}$ at the next iteration, and the HC repeats this search procedure. Moreover, the HC algorithm has been sufficiently described in [85] to determine the optimal structure of the GMP model.

Figure 3.25 presents an example of the state of HC at q^{th} iteration, which shows how the neighbors are evaluated according to f.

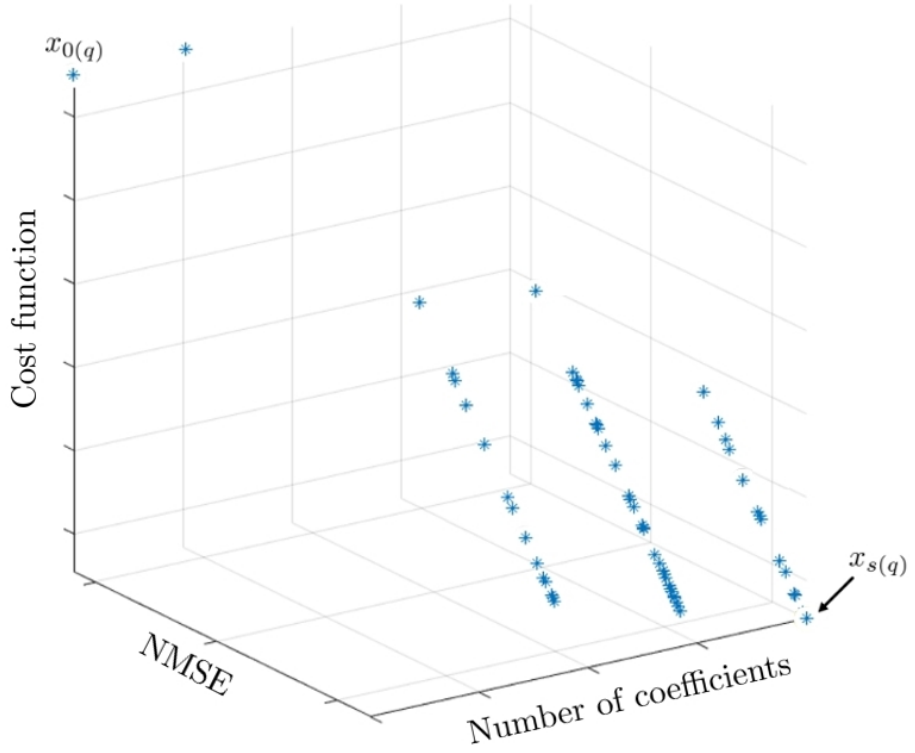


Figure 3.25: Neighbors of $x_{0(q)}$ at q^{th} iteration

The neighbors are denoted by $x_{i(q)}$, where $x_{i(q)} \epsilon V_{(q)}$. They are presented in Figure 3.25 by blue asterisks. The cost function $f(x_{i(q)})$ associated to $x_{i(q)}$ is evaluated. The solution $x_{s(q)}$ is the element of the neighborhood $V_{(q)}$ with the minimum cost function value.

There are two definitions to consider. Firstly, the cost function f that achieves a trade-off between the model's complexity and modeling accuracy. Secondly, the definition of the neighborhood that is used for HC search.

3.5.3 Cost Function

The cost function is an essential aspect of the HC algorithm. It refers to a particular criterion that leads to a trade-off between modeling accuracy and model complexity. NMSE is used to evaluate the modeling accuracy, which is denoted by N. The model complexity is presented by the number of coefficients of the model, denoted by C.

The cost function f is defined by combining N and C in a single criterion. In [80], the fitness function of GA is defined as a weighted additive combination of accuracy and model complexity.

In [84], several approaches have been proposed to define the cost function using two different methods to combine N and C in one single criterion: additive and multiplicative combinations.

In this work, the cost function is defined using the additive combination, which is expressed by:

$$f(x_i) = N(x_i) + \alpha C(x_i) \tag{3.7}$$

where α is a positive real value and considered as a weighting coefficient (or the NMSE tolerance per coefficient).

In [84], the authors proposed two methods to determine the weighting coefficient using offline computation and online computation.

3.5.4 Definition of Neighborhood

A new definition of the neighborhood is proposed for the DVR model in this work, allowing the HC algorithm to converge to the optimal structure.

For the DVR model, there are four parameters to size: the number of segments K, the memory depth M_{lin} for the linear term, the memory depth M for the set T_S, and which model terms will be selected from the set $\mathbf{T}$.

In the discrete space U, the element x_i that represents a DVR model structure is characterized by 3 integers: K_i, $M_{lin,i}$, M_i and by T_{S_i} which is a k combination of the set $\mathbf{T}$ with $k = \{1, ..., P+7\}$. These parameters can have their values changed independently, and they compose a 4-dimension discrete space of DVR model structures.

The neighbor of element x_i is defined as an 4-tuple $(K_i + \delta_1,\ M_{lin,i} + \delta_2,\ M_i + \delta_3,\ \mathcal{T}_{S,i} + \delta_4)$, where $\delta_{1,2,3} \in [0, \pm 1]$.

The process involved behind δ_4 corresponds to either add to T_{S_i} a term from its relative complement in $\mathbf{T}$, to remove a term from T_{S_i}, or to neither remove or add a term.

Figure 3.26 presents the discrete space V of the neighbors of x_i in terms of K, M_{lin}, and M for a given set of model term T_{S_i}. The blue point is the node $\{K_i, M_{lin,i}, M_i\}$ of x_i. The red points are the model parameters K, M_{lin}, and M of neighbors.

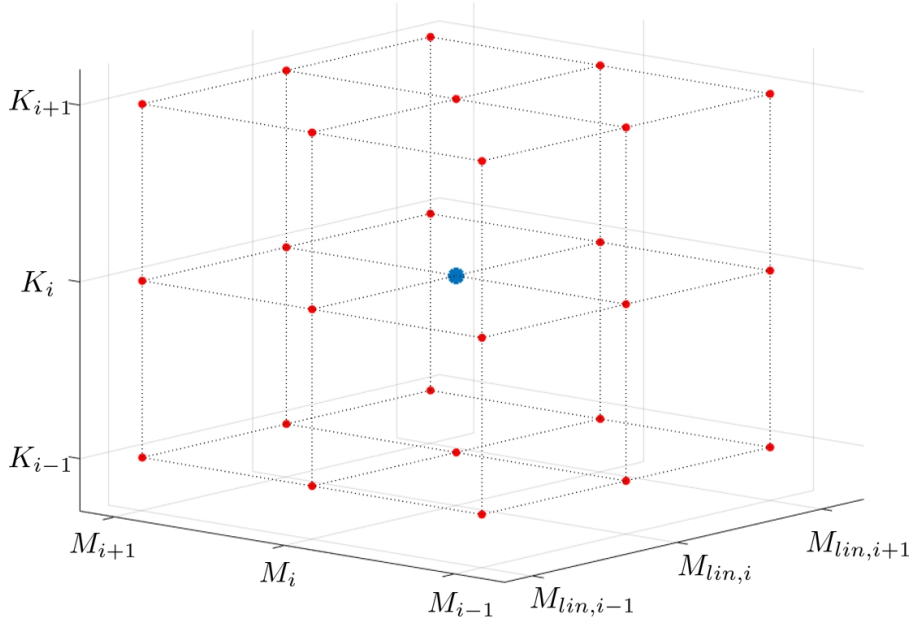


Figure 3.26: Discrete Space S of Neighbors of x_i in terms of K, M_{lin}, and M with a given T_{S_i}

If we have set P to 9 in (3.4) and consider that the current model has $T_{S_i} = [T_{1,0}\ T_{1,6}\ T_2\ T_5]$, then each model represented in red from Figure 3.26 can have the following T_{S_i}

$$T_{S_i} = \begin{cases} [T_{1,6}\ T_2\ T_5] & [T_{1,0}\ T_{1,1}\ T_{1,6}\ T_2\ T_5] & [T_{1,0}\ T_{1,2}\ T_{1,6}\ T_2\ T_5] \\ [T_{1,0}\ T_{1,3}\ T_{1,6}\ T_2\ T_5] & [T_{1,0}\ T_{1,4}\ T_{1,6}\ T_2\ T_5] & [T_{1,0}\ T_{1,5}\ T_{1,6}\ T_2\ T_5] \\ [T_{1,0}\ T_2\ T_5] & [T_{1,0}\ T_{1,6}\ T_{1,7}\ T_2\ T_5] & [T_{1,0}\ T_{1,6}\ T_{1,8}\ T_2\ T_5] \\ [T_{1,0}\ T_{1,6}\ T_5] & [T_{1,0}\ T_{1,6}\ T_{1,9}\ T_2\ T_5] & [T_{1,0}\ T_{1,6}\ T_2\ T_3\ T_5] \\ [T_{1,0}\ T_{1,6}\ T_2] & [T_{1,0}\ T_{1,6}\ T_2\ T_4\ T_5] & [T_{1,0}\ T_{1,6}\ T_2\ T_5\ T_6] \\ [T_{1,0}\ T_{1,6}\ T_2\ T_5\ T_7] & & \end{cases}$$

so the model in blue has $17 \times 3^3 - 1$ neighbors to be evaluated.

3.5.5 Complexity Reduction of Hill-Climbing Algorithm

The complexity is an essential aspect of optimization algorithms. In this study, we link the complexity of the HC algorithm with four features:

- The total number of elements (neighbors) evaluated by the HC algorithm, which corresponds automatically to the number of f evaluations. It includes:
  - The number of neighbors evaluated per one HC iteration, which is an indicator of the distribution of the total number of neighbors over the HC iterations. The average number of neighbors per HC iteration will be presented as well.
- The total number of iterations required for the HC algorithm to perform, where one HC iteration is done when the cost functions of neighbors of an initial solution are computed, and the optimal solution is determined.
- The execution time (runtime) of the HC algorithm is also taken into account as a relative indicator of the complexity of the HC algorithm.
- The number of model coefficients of each neighbor's structure is introduced since it reflects the computational complexity of the identification process. Besides, it refers

to the model complexity, which is essential for sizing the DVR model. Two features will be highlighted and linked to the computational complexity:

- Condition number of the regressor matrix $\mathbf{Z}^H\mathbf{Z}$ in (2.18).
- Dynamic range ϵ of the model coefficients defined in (2.23).

These four criteria assess the complexity of the HC algorithm and the numerical properties of the models trained during the search.

The complexity of the HC algorithm can be reduced by pruning the neighborhoods by adding some constraints on the number of coefficients.

In the HC algorithm, and according to the proposed definition of neighborhood, each element x_i can have up to $((P+1)+7)\times 3^3-1$ neighbors.

An element $x_{0(q)}$ at q^{th} iteration can have common neighbors with $x_{0(q-1)}$ from the previous iteration. Therefore, common neighbors will not be re-evaluated, which partially reduces the complexity of the HC search.

The search space U can also be limited by setting a maximum number of coefficients C_{max} of the model structure to be tested.

As well, the model parameters K, M_{lin}, and M can be limited by setting an interval for each one:

$$\begin{aligned} 2 &\leqslant K \leqslant K_{max} \\ 0 &\leqslant M_{lin} \leqslant M_{max} \\ 0 &\leqslant M \leqslant M_{max} \end{aligned} \tag{3.8}$$

where K_{max} and M_{max} are the maximum number of segments and the maximum memory depth, respectively.

The parameter K must be greater or equal to 2, which is the minimum number of segments. In a DVR structure, at least two segments bounded by one threshold are necessary to always remain in the concept of the segmentation approach.

In some cases, the neighbors may have a considerable variation of many coefficients. Therefore, a variation limit d_C is added to the HC algorithm to control the maximum variation of the number of coefficients of neighbors.

With these constraints on C_{max} and d_c, the original neighborhood space $V_{(q)}$ of the element $x_{0(q)}$ at q^{th} iteration is reduced to the elements $x_{i(q)}$ with $C_{i(q)} \leq C_{max}$ and $|C_{i(q)} - C_{0(q)}| \leq d_C$.

3.5.6 Stopping Condition

The sizing algorithm of the DVR model can be stopped once the cost function of a given solution is not better than that of the previous one. However, in our case, the stop condition of the HC algorithm is activated according to two levels. For the first level, if a solution $x_{s(q)}$ has no neighbors whose number of coefficients is less than C_{max} at the following iterations. This condition immediately leads to the stop of the HC algorithm. Otherwise, the HC algorithm checks the second level for which a positive tolerance $\Delta\nu$ is determined, making it possible to choose a solution $x_{s(q)}$ whose $N(x_{s(q)})$ satisfies the following conditions:

$$\begin{aligned} N(x_{s(q)}) &\leq N(x_{s(q-1)}) + \Delta\nu \\ C(x_{s(q-1)}) &< C(x_{s(q)}) \leq C_{max} \end{aligned} \tag{3.9}$$

where the integration of this constraint in the HC algorithm makes it possible to have a continuity of the search up to C_{max} to explore other neighbors who can have better cost functions.

These stopping levels are achieved to give more flexibility to the HC algorithm for finding good solutions properly. Once the HC algorithm is finished, the optimal solution returned is the one that has the best value of f among the solutions of each HC iteration.

3.5.7 Algorithm of DVR Model Sizing

By combining the proposed definition of neighborhood and the complexity reduction techniques into the HC algorithm, the optimal structure of the DVR model can be determined by the HC algorithm in Algorithm (6).

Algorithm 6: Algorithm of DVR model sizing

Initialization
Get the PA input x and PA output y
Set C_{max} and d_C
Set $\Delta\nu$
Define cost function $f = N + \alpha C$
Set the iteration counter $q = 1$
Set the initial element $x_{0(1)}$ with $K_0 = 2$, $M_{lin,0} = M_0 = 0$ and $T_{S_0} = []$
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  for $K_i \pm \delta_1$ **do**
    for $M_{lin,i} \pm \delta_2$ **do**
      for $M_i \pm \delta_3$ **do**
        for $T_{S_i} \pm \delta_4$ **do**
          Determine the neighbor $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$ under d_C and C_{max} constraints
          Compute the cost function $f(x_{i(q)})$
        end
      end
    end
  end
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    $x_{s(q)} = \underset{x_{i(q)} \in V_{(q)}}{argmin} \left(f(x_{i(q)}) \right)$
    Save $x_{s(q)}$ in solution space $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q-1)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin} \left(f(x_s) \right)$

3.5.8 Experiments and Results

3.5.8.1 Parameters of HC Algorithm

According to Algorithm (6), the evolution of the sizing process of the DVR model towards the optimal solution is mainly controlled by:

- C_{max}: the maximum number of model coefficients.
- K_{max}: the maximum number of segments to be assigned to a tested DVR structure.
- M_{max}: the maximum memory depth to be assigned to a tested DVR structure.
- P: the nonlinearity order of the basis functions in the DVR model.
- α: the weighting coefficient of the cost function.
- d_C: variation limit constraints on the neighbor's coefficients.
- $\Delta\nu$: the NMSE tolerance of the HC algorithm.

The HC algorithm aims to find the optimal structure of the DVR model by taking into account three properties: the high performance of the optimal solution, the short path search towards the optimal solution, and the reduced HC complexity given by the fewest computations of f, which in turn depends on the number of coefficients of each evaluated models.

Setting the HC algorithm parameters should be carefully done to achieve these properties and provide an efficient HC algorithm.

First, we set C_{max} to 50 coefficients. This makes it possible to have a well-extended behavior of the HC algorithm, of which the impact of C_{max} is relatively weak in order to be able to focus on the other HC parameters. We set $\Delta\nu$ to 0.2 dB.

We set the maximum number of segments K_{max} to 10 and the maximum memory depth M_{max}, including the linear term's memory depth to 10. K_{max} and M_{max} considerably controls the complexity of the HC algorithm according to the number of neighbors tested over the search.

The parameter P presents the order of the nonlinearity of the basis function in $T_{1,p}$. To study its impact in the DVR model, we propose to set P up to 1, then to 9, to efficiently analyze its global impact on the optimal solution and the DVR model.

The parameters α and d_C influence the HC algorithm's convergence towards the optimal solution. The parameter d_C limits the number of neighbors of a given element, which may potentially not choose neighbors with a better cost function value, while the parameter α controls directly the cost function f in (3.7).

It is important to emphasize that the DVR model's segmentation is uniform throughout the DVR model sizing.

In this study, we use the PA's experimental measurements presented in Section 3.3 using the same 20 MHz LTE signal.

3.5.8.2 Sizing DVR Model with $P = 1$

In the first step, we perform the HC algorithm to determine the optimal structure of the DVR model by setting P in T_S to 1 (P represents the nonlinearity order of the DVR basis function).

In fact, we seek to analyze the behavior and impact of P on the DVR model through the HC algorithm. Moreover, when the model terms $T_S = [T_{1,0}\ T_{1,1}\ T_2\ T_3\ T_4]$ are selected

by the HC algorithm, this corresponds to the DVR-6 model presented in the previous section.

The DVR model with the optimal structure returned by the HC algorithm is indicated by the DVR$-Q_P$ model, where Q is the number of model terms presented in the optimal structure, including the linear term.

3.5.8.2.1 Exhaustive Search

As mentioned before, the exhaustive search could be the spontaneous solution to find the optimal structure of the DVR model, but this approach requires a very high computational complexity.

However, we carried out an exhaustive search in this study to get all the possible structures, and we take them as a reference. The conditions of this exhaustive search are identical to that of the HC algorithm, where K_{max} and M_{max} were set to 10, C_{max} to 50, and $P = 1$ with $\mathbf{T} = [T_{1,0}\ T_{1,1}\ T_2\ T_3\ T_4\ T_5\ T_6\ T_7]$.

Inspection of the results allows us to determine the DVR model's best structure for each number of coefficients from 1 to $C_{max} = 50$. It is essential to accentuate that the DVR models tested during the exhaustive search were performed using uniform segmentation.

Figure 3.27 presents the exhaustive search results, which are plotted in blue dots, in terms of N and C.

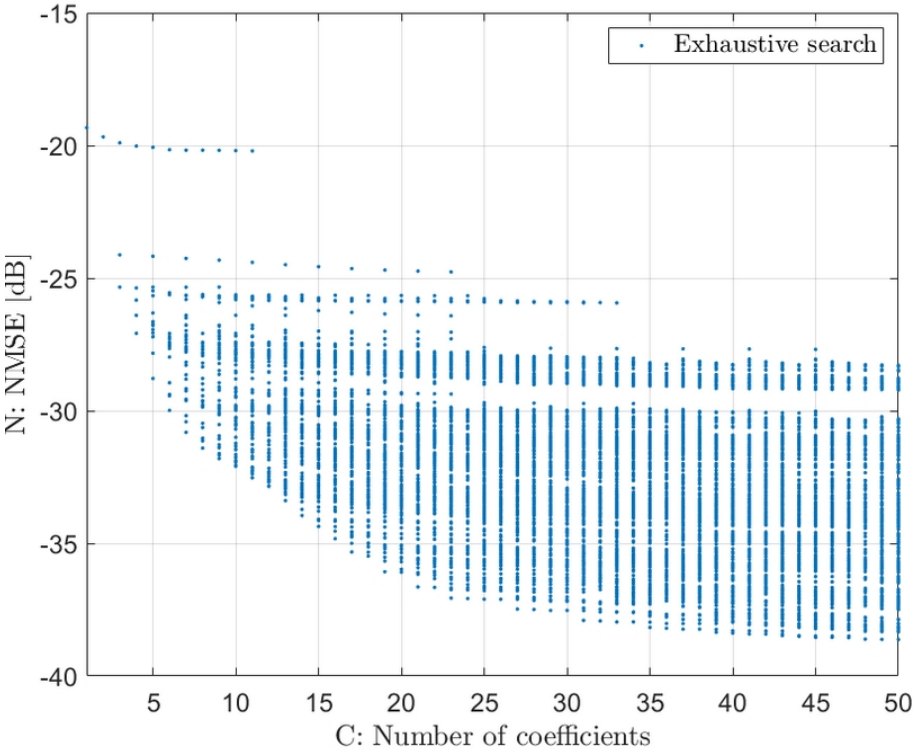


Figure 3.27: Exhaustive search of DVR model with $P = 1$ in terms of C and N

The exhaustive search requires almost 19 hours, in which there are a total of 24357 structures tested. The optimal solution that the HC algorithm seeks to find is located on the Pareto envelope of the exhaustive search.

3.5.8.2.2 HC Algorithm with α=0.1 and d_C=C_{max}

We set α to 0.1, which represents the NMSE tolerance per coefficient. We set d_C to C_{max} coefficients, which makes it possible not to introduce the impact of d_C on the behavior of the HC algorithm.

3.5.8.2.2.1 Optimal Solution

Figure 3.28 presents the HC algorithm results for the 20 MHz LTE signal. The results are presented in 3D regarding the fitness (value of the cost function f), the number of coefficients C, and NMSE N.

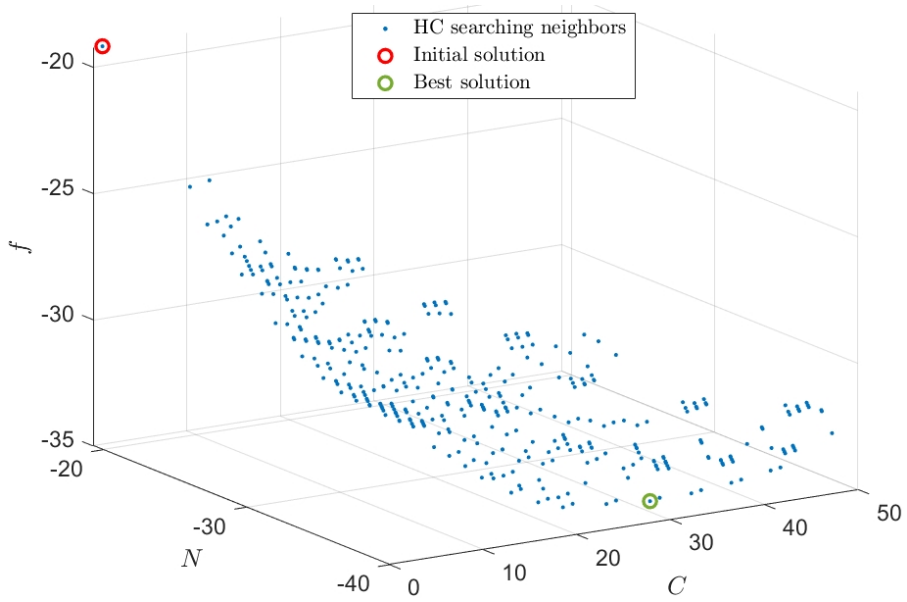


Figure 3.28: HC algorithm results in function of C, N, and f with α=0.1

The blue dots present the tested structures during the HC algorithm, which presents all tested neighbors. The red dot presents the first point of the HC algorithm where its structure is:

$$
\begin{aligned}
&K = 2 \\
&\beta = [0.5] \\
&M_{lin} = 2 \ ; \ M = 0 \\
&T_S = [\ ]
\end{aligned}
$$

The green dot presents the optimal solution where its structure has the best (minimum) f value:

$$
\begin{aligned}
&K = 4 \\
&\beta = [0.25 \ 0.5 \ 0.75] \\
&M_{lin} = 2 \ ; \ M = 1 \\
&T_S = [T_{1,0} \ T_3 \ T_6]
\end{aligned}
$$

with $C = 27$ coefficients and $N = -38.02$ dB.

3.5.8.2.2.2 Evolution of HC Algorithm

Figure 3.29 presents the HC algorithm's behavior in 2D in terms of C and N, where the exhaustive search results are plotted in the background.

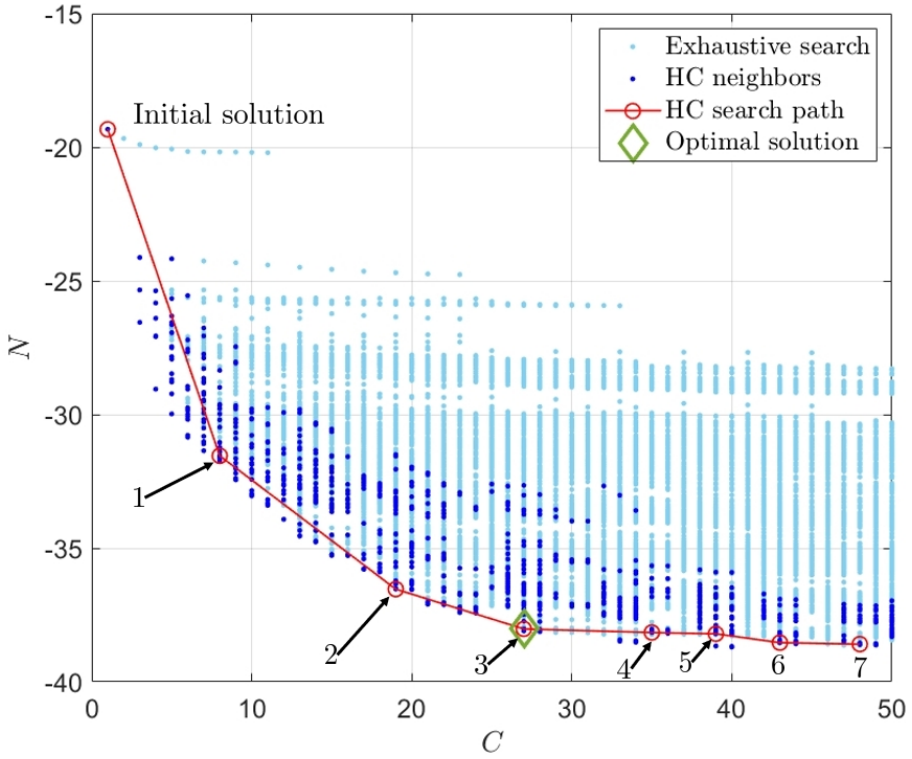


Figure 3.29: HC algorithm and exhaustive search in terms of C, N

As we can see, the HC algorithm determines the neighbors located near the Pareto envelope. The search path of the HC algorithm almost follows the envelope.

In this search, the HC algorithm has 7 HC iterations, and there are 1101 different DVR model structures tested in 4.3 minutes. The evolution of the HC algorithm is depicted in Figure 3.30 for the first iteration.

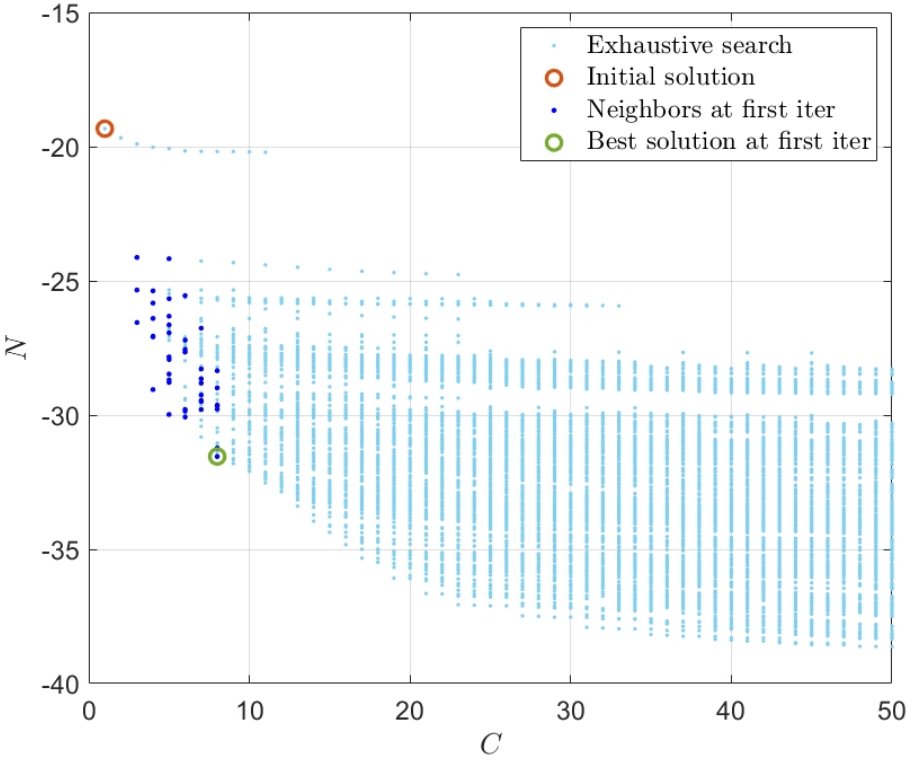


Figure 3.30: Evolution of HC algorithm at first iteration

The results of the exhaustive search are plotted in the background. The red circle is

the initial element. The blue dots present its neighbors, which are here 65. The green circle is the best neighbor of the first iteration selected by its minimum f value among the neighbors.

The structure of the best neighbors at first iteration is:

$$
\begin{aligned}
&K = 3 \\
&\beta = [0.33\ 0.66] \\
&M_{lin} = 1\ ;\ M = 1 \\
&T_S = [T_6]
\end{aligned}
$$

with $C = 8$ coefficients and $N = -31.54$ dB.

At the second iteration, the best neighbor from the first iteration is taken as the initial solution, and the neighbors of this solution are shown in Figure 3.31.

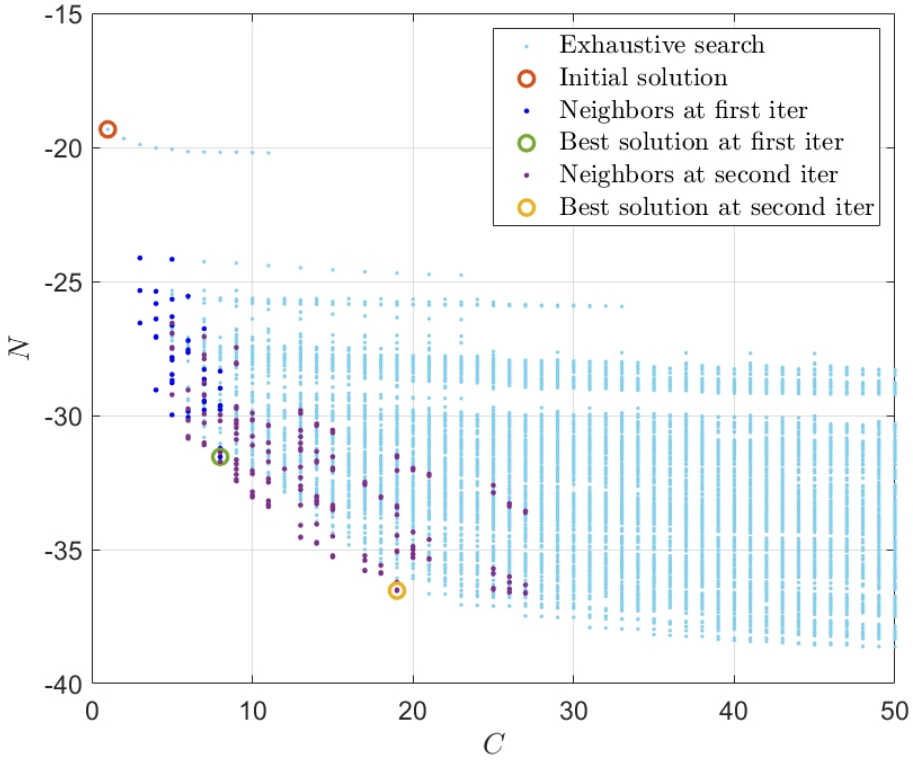


Figure 3.31: Evolution of HC algorithm at second iteration

The purple dots present the neighbors at the second iteration. The best solution at second iteration is highlighted as well by an orange circle, where the parameters of its structure are:

$$
\begin{aligned}
&K = 4 \\
&\beta = [0.25\ 0.50\ 0.75] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_3\ T_6]
\end{aligned}
$$

with $C = 19$ coefficients and $N = -36.53$ dB.

In the next iteration, the best solution at the second iteration will be taken as the initial solution. The HC algorithm iteratively continues until it converges.

At the 7^{th} iteration, the parameters of the best solution are:

$$K = 5$$
$$\beta = [0.20\ 0.40\ 0.60\ 0.80]$$
$$M_{lin} = 2\ ;\ M = 2$$
$$T_S = [T_{1,0}\ T_2\ T_7]$$

with $C = 48$ coefficients and $N = -38.59$ dB.

After the 7^{th} iteration, 91 neighbors of this solution, plotted by purple dots in Figure 3.32, are evaluated by the HC algorithm.

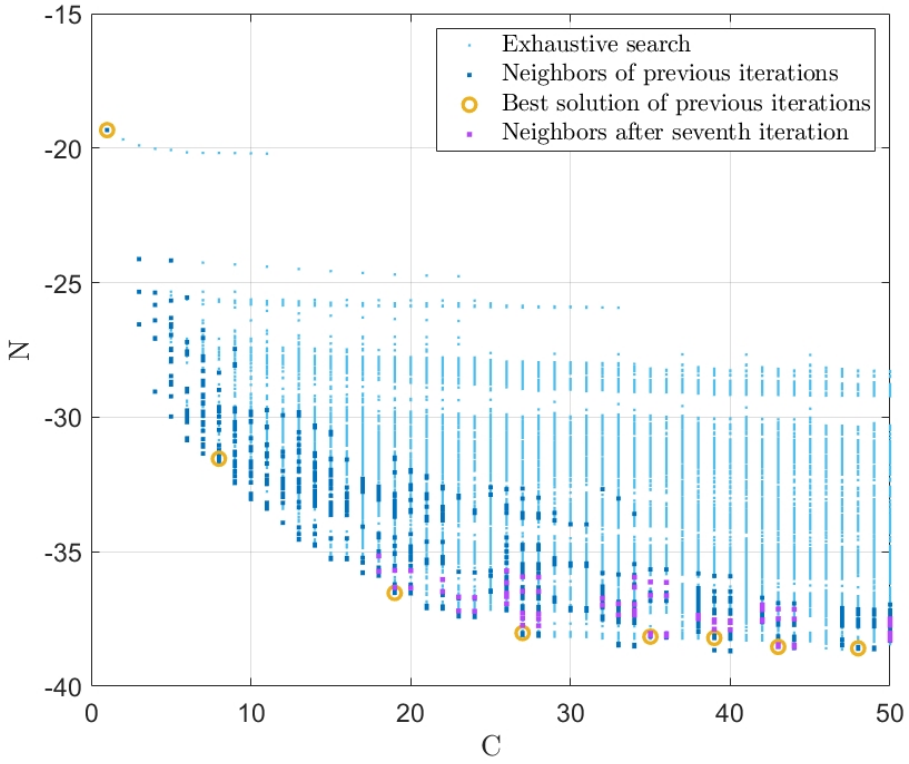


Figure 3.32: Evolution of HC algorithm after seventh iteration

By investigating the stop condition of the HC algorithm, it was shown in Figure 3.33 that after the seventh HC iteration, no solution found among the 91 neighbors satisfies the conditions of (3.9), which leads to stopping the HC algorithm and returning the optimal DVR model.

Table 3.7 presents the number of coefficients C, NMSE N, and the cost function f of the best solution at each HC iteration. The optimal DVR model is the solution of the third iteration, which has the best (minimized) cost function.

Table 3.7: Number of coefficients, NMSE, and cost function of the best DVR model at each HC iteration

HC iteration	Initial	1	2	3	4	5	6	7
C	1	8	19	27	35	39	43	48
N	-19.33	-31.54	-36.53	-38.02	-38.15	-38.20	-38.53	-38.59
f	-19.23	-30.74	-34.63	-35.32	-34.65	-34.3	-34.23	-33.79

After the 3^{th} iteration, an asymptotic behavior is observed around -38 dB, which can be confirmed in Figure 3.29.

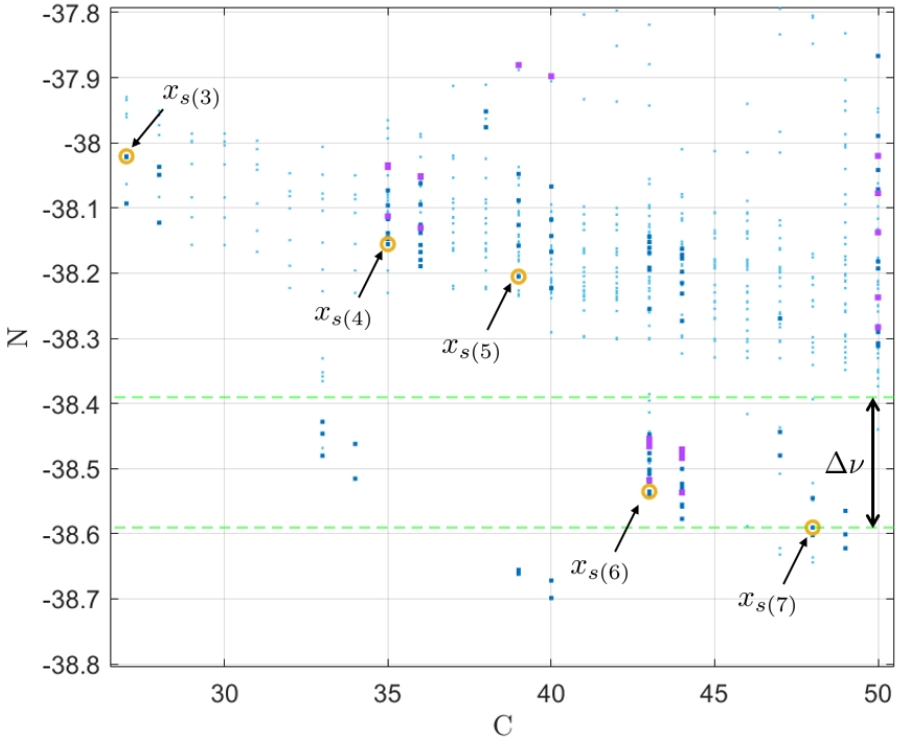


Figure 3.33: Evolution of HC algorithm with stopping conditions

3.5.8.2.2.3 HC Complexity

Table 3.8 presents the parameters of the best structures at each HC iteration.

Table 3.8: Parameters of DVR model structure (initial and best) at each HC iteration

HC iteration		1	2	3	4	5	6	7	-
Initial solut.	K	2	3	4	4	4	4	5	5
	M_{lin}	0	1	2	2	2	2	2	2
	M	0	1	1	1	1	2	1	2
	T_S		T_6	T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_2 T_3 T_6	$T_{1,0}$ T_2 T_6	$T_{1,0}$ T_2 T_6 T_7	$T_{1,0}$ T_2 T_7
Num. of Neigh.		64	208	217	152	126	116	127	91
Best solut.	K	3	4	4	4	4	5	5	HC stops
	M_{lin}	1	2	2	2	2	2	2	
	M	1	1	1	1	2	1	2	
	T_S	T_6	T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_2 T_3 T_6	$T_{1,0}$ T_2 T_6	$T_{1,0}$ T_2 T_6 T_7	$T_{1,0}$ T_2 T_7	

Each column presents the initial solution structure, the number of the neighbors tested by the HC algorithm, and the structure of the best solution according to its cost function f.

After the 7^{th} HC iteration, the HC algorithm evaluates 91 neighbors of the best structure from the 7^{th} HC iteration, but without finding any structure with the best cost function, which leads to stopping the algorithm.

The memory depth of the linear term M_{lin} is different from that of the other terms M, which confirms the interest of specifying two different memory depths.

It can be seen that M_{lin} remains stable at 2 after the second iteration. From 2^{th} to 5^{th} iteration, the model terms are changed, while at the 6^{th} iteration, T_S and M are changed, and at the 7^{th} iteration, K, M, and T_S are changed.

On the other hand, we see that the number of evaluated neighbors at every iteration decreases after the third iterations. This refers to the fact that there are common neighbors between iterations that are not re-evaluated.

Since the HC algorithm's complexity is an important aspect, we summarize in Table 3.9 the total number of DVR models tested during the HC algorithm, where the average number of coefficients estimated by LS is 25 coefficients with a standard deviation of 13.

Table 3.9: Number of coefficients during the HC algorithm

Total number of DVR models	mean	standard deviation
1101	25	13

3.5.8.2.2.4 Comparison with Exhaustive Search

By investigating the best DVR model at 27 coefficients from the exhaustive search, it has been shown that the parameters of its structure are:

$$
\begin{aligned}
&K = 4 \\
&\beta = [0.25\ 0.5\ 0.75] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_{1,0}\ T_2\ T_6]
\end{aligned}
$$

with $C = 27$ coefficients and $N = -38.09$ dB.

The HC algorithm returns a very close structure. In fact, by examining the behavior of the HC algorithm illustrated in Figure 3.29, it was shown that the neighbors explored by the HC algorithm do not contain any structure with the model terms $T_S = [T_{1,0}\ T_2\ T_6]$, which explains why the HC algorithm did not converge towards the solution given by the exhaustive search.

Nevertheless, the HC algorithm is confirmed and validated through this scenario, where the proposed neighborhood definition demonstrates its effectiveness in determining the optimal DVR model.

3.5.8.2.3 Variation of d_C with $\alpha = 0.1$

The parameter d_C controls the space of the neighbors in the HC algorithm. Reducing d_C leads to reducing the runtime of the search, but better solutions may be skipped.

The choice of d_C raises three major concerns:

- From which d_C, the optimal solution returned does not change.
- The impact of the d_C variation on the HC complexity, especially the number of HC iterations and neighbors per iteration.
- The impact of the d_C variation on the search path of the HC algorithm.

3.5.8.2.3.1 Impact of d_C on Optimal Solution

In this study, the HC algorithm was carried out by varying d_C from 3 to C_{max}. Table 3.10 summarizes the parameters of the optimal structure of the DVR model for each d_C. The parameters C, N, and f are presented as well.

Table 3.10: Parameters of optimal DVR model structure for d_C variation

d_C		3	4	5	6	7	8	9	10	15	20	30	40	50
Structure	K	3	4	4	4	4	4	4	4	4	4	4	4	4
	M_{lin}	2	2	2	2	2	2	0	2	2	2	2	2	2
	M	1	1	1	1	1	1	1	1	1	1	1	1	1
	T_S	$T_{1,0}$ $T_{1,1}$ T_5	$T_{1,0}$ T_4 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ $T_{1,1}$ T_6	$T_{1,0}$ $T_{1,1}$ T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_4 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6
C		21	27	27	27	27	27	25	27	27	27	27	27	27
N		-36.60	-38.06	-38.02	-38.02	-37.93	-37.93	-37.04	-38.06	-38.02	-38.02	-38.02	-38.02	-38.02
f		-34.50	-35.36	-35.32	-35.32	-35.23	-35.23	-34.55	-35.36	-35.32	-35.32	-35.32	-35.32	-35.32

From Table 3.10, we can see that the HC algorithm returns the same optimal DVR model when d_C exceeds a distance of 15 coefficients. On the other hand, all the optimal structures have the same number of coefficients (27 coefficients) except when $d_C = 3$ and $d_C = 9$.

Regarding the structure parameters, from $d_C = 4$, the optimal solutions has the same K, M_{lin}, and M, unlike T_S for which the structures are different for the different values of d_c.

Notwithstanding, the model term $T_{1,0}$ is present in all structures, which presents the basis function of the DVR model and corresponds to:

$$\sum_{k=1}^{K}\sum_{i=0}^{M} c_{ki,10}||x(n-i)| - \beta_k|e^{j\theta(n-i)}$$

.

Therefore, selecting the best solution depends on the cost function, on its parameter α. The parameter d_C is involved as well, which controls the size of the neighborhood space.

3.5.8.2.3.2 Impact of d_C on HC complexity

The complexity of the HC algorithm for each d_C is investigated. Table 3.11 presents the total number of neighbors tested for each d_C and the number of neighbors per HC iteration. The first column presents d_C. The second one presents the total number of neighbors tested by the HC algorithm. The third column shows in detail the number of neighbors for each HC iterations.

Table 3.11: Number of neighbors versus HC iteration for each d_C

d_C	Total number of neighbors	HC iter.	Neighbors per each HC iteration	Runtime (min)
3	443	16	{24;35;40;47;19;36;20;47;48;18;18;18;18;18;18;1}	1.20
4	431	16	{40;67;44;29;32;25;32;18;18;18;18;18;18;18;17;1}	1.09
5	427	6	{48;81;79;75;44;48;52}	1.01
6	693	16	{56;95;88;58;91;44;26;26;26;26;26;26;24;4;28;14;35}	2.06
7	782	14	{64;145;143;107;44;49;35;44;44;41;14;14;14;14;10}	2.69
8	798	14	{64;145;147;108;49;44;44;43;47;41;14;14;14;14;10}	2.41
9	458	7	{64;152;83;65;26;48;19;1}	1.20
10	739	11	{64;159;133;105;40;40;40;40;40;40;37;1}	2.54
15	913	7	{64;187;183;136;108;102;71;62}	3.24
20	1009	7	{64;208;193;149;125;111;90;69}	3.68
30	1101	7	{64;208;217;152;126;116;127;91}	4.30
40	1101	7	{64;208;217;152;126;116;127;91}	4.35
50	1101	7	{64;208;217;152;126;116;127;91}	4.31

According to Table 3.11, the HC algorithm requires seven iterations to converge from $d_C = 9$, except when $d_C = 10$ in which the HC algorithm requires 11 iterations. In a general way, we recognize that the total number of neighbors evaluated by the HC algorithm increases with the increases of d_C and the runtime taken by the HC algorithm under the MATLAB environment, while the HC iterations are decreasing.

The number of neighbors evaluated remains 1101 neighbors after a distance of 30 coefficients. This allows us to conclude that beyond $d_C = 30$, the HC algorithm's behavior, particularly its complexity, does not change.

On the other hand, when $d_C = 5$, the total number of neighbors is reduced, and the HC iterations and the HC algorithm runtime. It can be explained by the fact that a distance of 5 coefficients allows appropriately to the search to perform only 6 HC iterations in significantly reduced runtime, in which it returns the same optimal DVR model returned when $d_C = 50$.

3.5.8.2.3.3 Conclusion on d_C Impact

In conclusion, d_C has a significant impact on the complexity of the HC algorithm, unlike the linearization performances where the optimal solutions for each d_C have an NMSE with a tolerance of 1 dB around.

When $d_C = 5$, the HC algorithm exhibits a good behavior in terms of complexity, particularly the number of neighbors, the computational complexity for identifying the coefficients, and the runtime of the search.

In the following, we set d_C to 5.

3.5.8.2.4 Variation of α with $d_C = 5$

The determination of weighting coefficients for the multi-objective optimization problems is an essential aspect of building a cost function. The cost function f controls the evolution of the HC algorithm to determine an optimal DVR model with the best compromise, determined by α, between the number of coefficients and modeling accuracy.

The impact of the weighting coefficient α (unit tolerance) is investigated in terms of the optimal solution and the complexity of the HC algorithm. In this study, The HC

algorithm is performed for each α from the vector

$$[0.01\ ;\ 0.025\ ;\ 0.05\ ;\ 0.075\ ;\ 0.1\ ;\ 0.15\ ;\ 0.2\ ;\ 0.3\ ;\ 0.4\ ;\ 0.5]$$

3.5.8.2.4.1 Impact of α on Optimal Solution

Table 3.12 summarizes the results of the HC algorithm where the parameters of the optimal structure, C, N and $f = N + \alpha C$ are presented for each α.

It is essential to remind that the more significant α, the more one seeks to limit the number of coefficients to the detriment of NMSE.

Table 3.12: Parameters of the optimal DVR model, C, N, and f with the variation α

α		0.01	0.025	0.05	0.075	0.1	0.15	0.2	0.3	0.4	0.5
Structure	K	7	6	4	4	4	4	3	4	3	2
	M_{lin}	7	4	2	2	2	2	2	2	0	1
	M	1	1	1	1	1	1	1	1	1	1
	T_S	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	$T_{1,0}$ T_3 T_6	T_3 T_6	T_3 T_6	T_3 T_6
C		50	41	27	27	27	27	21	19	13	10
N		-38.88	-38.70	-38.02	-38.02	-38.02	-38.02	-37.01	-36.53	-34.52	-33.04
f		-38.38	-37.67	-36.67	-36.01	-35.32	-33.97	-32.81	-30.83	-29.33	-28.04

According to Table 3.12, the variation of α influences the optimal solutions of the HC algorithm, in particular in terms of the number of segments K and memory depth of the linear term M_{lin}.

The parameters K and M_{lin} decrease swiftly with increasing α, while M is much less sensitive to the variation of α, which remains at 1. Regarding T_S, the set $[T_3\ T_6]$ is present in all best structures.

The behavior of how K and M_{lin} change with the variation of α leads to an essential link between the design of the cost function f and the complexity of the optimal solution. If α increases, K and M_{lin} decrease. Consequently, the number of coefficients C decreases, which reduces the complexity.

On the other hand, on the interval, [0.05 0.15], the optimal solution is not very sensitive to the value of α, and the HC algorithm returns the same DVR model.

3.5.8.2.4.2 Impact of α on HC Complexity

On the other hand, it is worth investigating the complexity of the HC algorithm with the variation of α. Table 3.13 summarizes the total neighbors evaluated by the HC algorithm and the HC iteration. The number of neighbors tested at each HC iteration and the total number of model coefficients are shown.

Table 3.13: Complexity of HC algorithm for each α

α	Total number of neighbors	HC iter.	Neighbors per HC iter.	Runtime (min)	Total number of coeff.
0.01	548	15	{48;74;84;49;61;58;53; 27;13;17;4;4;8;8;5}	2.20	11619
0.025	588	17	{48;74;84;49;44;48;56; 35;22;23;13;17;17;17; 17;17;6;1}	2.44	13571
0.05	399	6	{48;74;84;49;44;48;52}	1.18	5850
0.075	399	6	{48;74;84;49;44;48;52}	1.20	5850
0.1	427	6	{48;81;79;75;44;48;52}	1.10	6119
0.15	427	6	{48;81;79;75;44;48;52}	1.15	6110
0.2	697	19	{48;81;79;75;44;67;47; 21;26;26;26;26;26;26; 24;4;31;5;14;1}	2.35	14589
0.3	697	19	{48;81;79;75;44;67;47; 21;26;26;26;26;26;26; 24;4;31;5;14;1}	2.43	14589
0.4	303	5	{48;81;79;20;35;40}	0.64	3250
0.5	303	5	{48;81;79;20;35;40}	0.65	3250

According to Table 3.13 and when $\alpha = 0.05$, the HC algorithm evaluates the lowest total number of neighbors and the total number of coefficients. Prominently, the HC algorithm with $\alpha = 0.05$ provides an optimal DVR model that exhibits a good trade-off between the modeling performance, the model complexity, and the HC complexity.

3.5.8.2.5 HC Algorithm with $\alpha = 0.05$ and $d_C = 5$

By setting α to 0.05 and d_C to 5, the search path of the HC algorithm is shown in Figure 3.34, which almost follows the Pareto envelope of the exhaustive search and stops at 27 coefficients, with an NMSE of -38.02 dB.

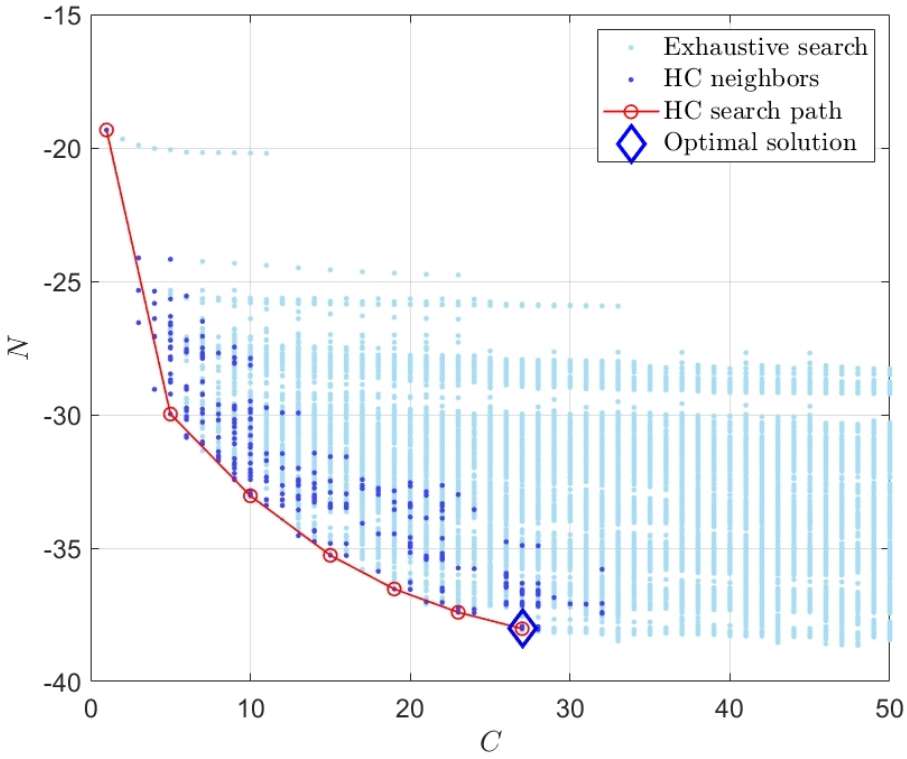


Figure 3.34: HC algorithm with $d_C = 5$ and $\alpha = 0.05$

The structure of the optimal solution is:

$$K = 4$$
$$\beta = [0.25\ 0.5\ 0.75]$$
$$M_{lin} = 2\ ;\ M = 1$$
$$T_S = [T_{1,0}\ T_3\ T_6]$$

where it is denoted by the optimal DVR-4_1 model.

In this search, there are a total of 399 elements tested in 1.18 minutes and 6 HC iterations. These results are summarized in Table 3.13 for α=0.05. The parameters of the best DVR model at each HC iteration are summarized in Table 3.14.

Table 3.14: Parameters of DVR model structure (initial and best) and the number of neighbors, at each iteration

HC Iteration		1	2	3	4	5	6	-
Initial solut.	K	2	2	2	3	4	5	4
	M_{lin}	0	1	2	2	2	2	2
	M	0	1	1	1	1	1	1
	T_S	-	T_6	T_3 T_6	T_3 T_6	T_3 T_6	T_3 T_6	$T_{1,0}$ T_3 T_6
Num. of neighbors		49	74	84	49	44	48	52
Best solut.	K	2	2	3	4	5	4	HC stop
	M_{lin}	1	2	2	2	2	2	
	M	1	1	1	1	1	1	
	T_S	T_6	T_3 T_6	T_3 T_6	T_3 T_6	T_3 T_6	$T_{1,0}$ T_3 T_6	

According to Table 3.14, the number of segments K has a significant impact on the evolution of the search path, while M_{lin} and M remain at 2 and 1, respectively. On the other hand, the model term T_6, which corresponds to

$$\sum_{k=1}^{K}\sum_{i=0}^{M} c_{ki,6}||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2$$

is present in the structure of the best neighbors over the HC iterations.

3.5.8.3 Sizing DVR Model with $P = 9$

Once the HC algorithm has been validated for the DVR model with $P = 1$, we carried out the HC algorithm to size the DVR model using the terms in (3.4) and setting P to 9.

3.5.8.3.1 Exhaustive Search

With $\mathbf{T} = [T_{1,[0,\ldots,9]}\ T_2\ T_3\ T_4\ T_5\ T_6\ T_7]$, and by setting K_{max} to 10, M_{max} to 10, and C_{max} to 50, the exhaustive search requires 339341 tests to be performed, which could take almost seven days to achieve, making of it too heavy to implement in this study.

3.5.8.3.2 HC Algorithm with $\alpha = 0.05$ and $d_C = 5$

As shown previously, setting α to 0.05 and d_C to 5 exhibits a good trade-off between the modeling accuracy of the optimal solution and the algorithm complexity.

We set α to 0.05 and d_C to 5 coefficients.

3.5.8.3.2.1 Optimal Solution

Figure 3.35 presents the DVR model sizing with $P = 9$ using the HC algorithm in terms of C and N. The blue dots present the neighbors evaluated over the search. The red circles present the best solution at each HC iteration.

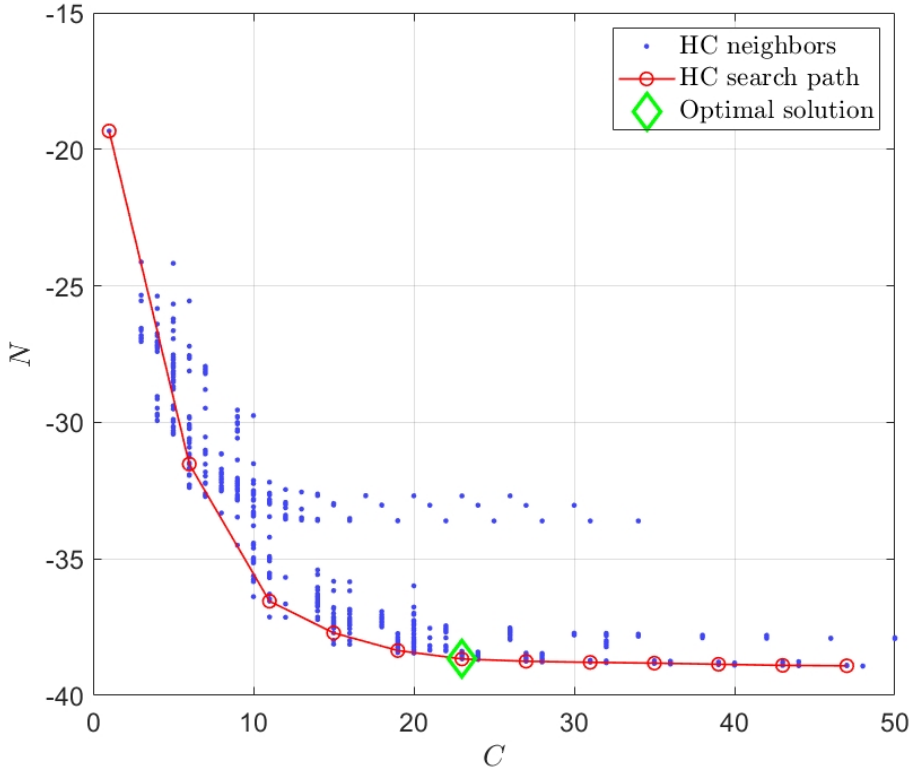


Figure 3.35: HC algorithm and exhaustive search in terms of C, N

The optimal solution shown in Figure 3.35 corresponds to the following structure:

$$
\begin{aligned}
&K = 2 \\
&\beta = [0.50] \\
&M_{lin} = 2 \ ; \ M = 1 \\
&T_S = [T_{1,5} \ T_{1,7} \ T_{1,8} \ T_{1,9} \ T_2]
\end{aligned}
$$

with $C = 23$ coefficients and $N = -38.67$ dB. In the following, this optimal structure is presented as the DVR-6_9.

3.5.8.3.2.2 HC Complexity

In this search to size the DVR model with $P = 9$, the HC algorithm has 11 HC iterations, and there are a total of 1067 different DVR model structures tested in 3.73 minutes.

Table 3.15 presents the parameters of the best structures at each HC iteration. Similarly to Table 3.8, each column presents the structure of the initial solution, the number of the neighbors tested by the HC algorithm, and the structure of the best solution.

Table 3.15: Parameters of DVR model with $P = 9$ structure at each HC iteration

HC iteration		1	2	3	4	5	6	7	8	9	10	11	12
Initial solution	K	2	2	2	2	2	2	2	2	2	2	2	2
	M_{lin}	0	1	2	2	2	2	2	2	2	2	2	2
	M	0	1	1	1	1	1	1	1	1	1	1	1
	T_S		$T_{1,5}$	$T_{1,5}$ $T_{1,8}$	$T_{1,5}$ $T_{1,8}$ T_2	$T_{1,5}$ $T_{1,8}$ $T_{1,9}$ T_2	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_6	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,4}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,3}$ $T_{1,4}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6
Num. of Neigh.		96	146	156	140	112	105	84	69	58	45	31	25
Best solution	K	2	2	2	2	2	2	2	2	2	2	2	HC stops
	M_{lin}	1	2	2	2	2	2	2	2	2	2	2	
	M	1	1	1	1	1	1	1	1	1	1	1	
	T_S	$T_{1,5}$	$T_{1,5}$ $T_{1,8}$	$T_{1,5}$ $T_{1,8}$ T_2	$T_{1,5}$ $T_{1,8}$ $T_{1,9}$ T_2	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_6	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,4}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	$T_{1,2}$ $T_{1,3}$ $T_{1,4}$ $T_{1,5}$ $T_{1,6}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2 T_5 T_6	

As shown in Table 3.15, the parameters K, M_{lin}, and M remain at 2, 2, and 1, respectively, while the model terms change from one optimal structure to another over the HC iterations, which means that the terms bring a significant impact on the convergence of the HC algorithm towards the optimal solution.

Table 3.16 presents C and N of the best solution at each HC iterations. After the 5^{th} iteration, an asymptotic behavior is observed around -38.7 dB, which can be confirmed in Figure 3.35.

Table 3.16: C and N of the best DVR model with $P = 9$ at each HC iterations

HC iteration	Initial	1	2	3	4	5	6	7	8	9	10	11
C	1	6	11	15	19	23	27	31	35	39	43	47
N	-19.33	-31.52	-36.55	-37.71	-38.35	-38.66	-38.75	-38.78	-38.81	-38.85	-38.9	-38.91

The total number of coefficients of all DVR models tested during the HC algorithm is 20728, with a mean of 20 coefficients and a standard deviation of 11.

3.5.8.4 Impact of P on DVR Model Sizing

3.5.8.4.1 Motivation

In the light of the behavior of the HC algorithm for $P = 1$ and $P = 9$, in this section, we present a comparison of both cases, and we also highlight the impact of P on the DVR model through its sizing using the HC algorithm.

The PAs are inherently nonlinear systems, in which many baseband nonlinear dynamic models, which can represent nonlinear systems with memory effect, can be used (MP, GMP, ...) to ensure linear amplification.

As mentioned previously, the DVR model is based on the segmentation approach, which aims to reduce the nonlinearities piecewise, unlike the global models derived from the Volterra series, which consists of deploying a high nonlinearity order to compensate for the nonlinearity effects, which leads to unsatisfactory numerical properties for identification process and complexity for hardware implementation.

In this study, we introduce the nonlinearity order p in (3.5) in the second term

$$T_{1,p} = \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1p} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \cdot |x(n)|^p$$

in order to enhance the DVR model ability to represent strong nonlinearities. However, dealing with p should be carefully since it presents the same properties that the nonlinearity order for the global model derived from the Volterra series. In fact, this can be confirmed in Figure 3.36 and Figure 3.37, which show the impact of p on the modeling accuracy and numerical properties of the DVR model performance.

Figure 3.36 presents the variation of the NMSE versus p, for p from 0 to 9. In this scenario, the NMSE is computed between the PA output and the DVR model output

$$y(n) = \sum_{i=0}^{M_{lin}} a_i x(n-i) + \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1p} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \cdot |x(n)|^p$$

with $K = 4$, $M_{lin} = 2$, and $M = 2$, which are selected arbitrary.

For this scenario at hand, the number of coefficients C is defined as

$$C = M_{lin} + 1 + K \times (M + 1)$$

which remains equal to 15 coefficients with the variation of p.

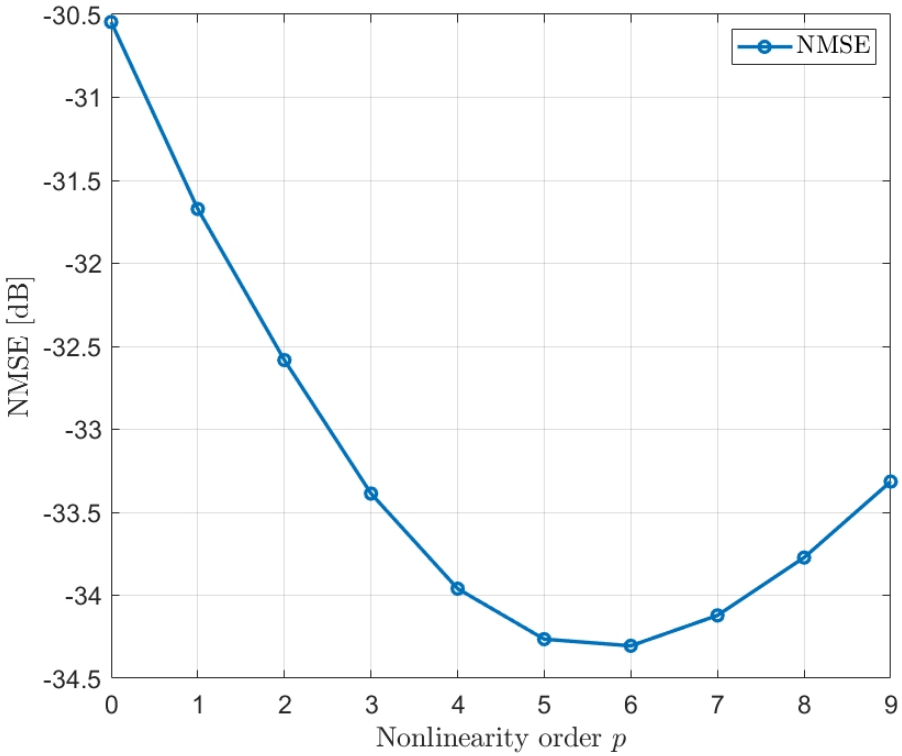
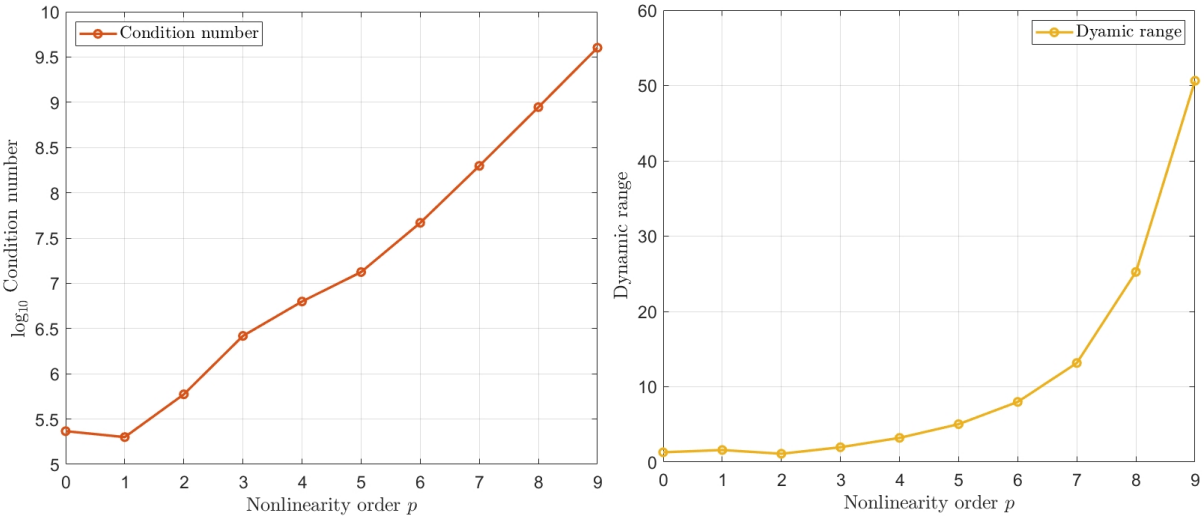


Figure 3.36: Variation of p versus NMSE

As can be shown in Figure 3.36, the variation of p significantly influences the ability of the DVR model to model accurately the behavior of the PA. In this scenario, an optimal structure is found in terms of NMSE when the nonlinearity order $p = 6$, which corresponds to

$$\sum_{i=0}^{2} a_i x(n-i) + \sum_{k=1}^{4} \sum_{i=0}^{2} c_{ki,1p} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \cdot |x(n)|^6$$

On the other hand, Figure 3.37 highlights the numerical properties of the DVR model with the variation of p. These numerical properties are represented by the condition number of the regressor matrix introduced in (2.18) in Figure 3.37a, and the dynamic range ϵ of the estimated model coefficients defined in (2.23) in Figure 3.37b.



(a) Variation of condition number versus p (b) Variation of dynamic range versus p

Figure 3.37: Impact of p on numerical properties

The impact of p on the numerical properties of the DVR model is weighty through

Figure 3.37. Figure 3.37a shows that the condition number of the regressor matrix evolves exponentially with the increase of p, while the dynamic range of the model coefficients evolves exponentially in Figure 3.37b.

This motivates the need to take into consideration the numerical properties of the DVR model. By taking the optimal structure when $p = 6$, it is not an optimal choice if the condition number or the dynamic range is involved in the criterion to choose this optimal structure, driving to make a trade-off between modeling accuracy and the numerical properties.

In the light of the results reported in Figure 3.36 and Figure 3.37, the interest of having a high nonlinearity order in the DVR model is advantageous to represent strong nonlinearities but must be chosen with care in order not to worsen the numerical properties.

Given that the HC complexity depends essentially on the complexity and the numerical properties of the trained DVR models, we propose to focus on the impact of the nonlinearity order on the DVR model and the HC algorithm by comparing and analyzing the results of the HC algorithm with $P = 1$ and $P = 9$ in terms of optimal structure, DVR model complexity, DVR model numerical properties, and HC complexity.

3.5.8.4.2 Comparison of Optimal Solution

Table 3.17 summarizes the optimal structures of the DVR-4_1 and DVR-6_9 models.

Table 3.17: Comparison of DVR-4_1 and DVR-6_9 model structures

		DVR-4_1	DVR-6_9
Model structure	K	4	2
	β	[0.25 0.5 0.75]	[0.5]
	M_{lin}	2	2
	M	1	1
	T_S	$T_{1,0}$ T_3 T_6	$T_{1,5}$ $T_{1,7}$ $T_{1,8}$ $T_{1,9}$ T_2

As shown in Table 3.17, the structure of DVR-4_1 has a significant number of segments K, while the structure of the DVR-6_9 model is determined by the model terms. This is justified by the limitation of the nonlinearity order p to $P = 1$, which leads the HC algorithm to explore new neighbors by increasing K. In contrast, for a limitation of $P = 9$, the HC algorithm explores several combinations of model terms, leading to an optimal structure composed of 6 model terms, including the linear term.

Replacing the parameters of the optimal structure in (3.3), the DVR-4_1 model is expressed by:

$$\begin{aligned} y_{DVR-4_1}(n) = & \sum_{i=0}^{2} a_i x(n-i) \\ & + \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,10}||x(n-i)| - \beta_k|e^{j\theta(n-i)} \\ & + \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,3}||x(n-i)| - \beta_k| \cdot x(n-i) \\ & + \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,6}||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \end{aligned}$$

The DVR-6_9 model is expressed by:

$$\begin{aligned} y_{DVR-6_9}(n) = & \sum_{i=0}^{2} a_i x(n-i) \\ & + \sum_{k=1}^{2}\sum_{i=0}^{1} c_{ki,15}||x(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |x(n)|^5 \\ & + \sum_{k=1}^{2}\sum_{i=0}^{1} c_{ki,17}||x(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |x(n)|^7 \\ & + \sum_{k=1}^{2}\sum_{i=0}^{1} c_{ki,18}||x(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |x(n)|^8 \\ & + \sum_{k=1}^{2}\sum_{i=0}^{1} c_{ki,19}||x(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |x(n)|^9 \\ & + \sum_{k=1}^{2}\sum_{i=0}^{1} c_{ki,2}||x(n-i)| - \beta_k| \cdot x(n) \end{aligned}$$

Their performances are summarized in Table 3.18 in terms of the number of coefficients, NMSE, the dynamic range of the model coefficients, and the condition number.

Table 3.18: Comparison of DVR-4_1 and DVR-6_9 in terms of modeling accuracy, complexity, and numerical properties

	DVR-4_1	DVR-6_9
NMSE	-38.02	-38.67
Number of coeff.	27	23
Dynamic range	29	1499
Condition number	10^9	10^{13}

The HC algorithm returns the DVR-4_1 and DVR-6_9 models according to the same setting of α and d_C.

Both models have almost similar modeling accuracy (NMSE) but with different complexity (number of coefficients). The DVR-6_9 model has achieved roughly the modeling accuracy of the DVR-4_1 model only with 23 coefficients, while the DVR-4_1 model requires 27 coefficients.

In contrast, the DVR-4_1 model exhibits much better numerical properties compared to the DVR-6_9 model. This can be seen in the dynamic range of the DVR-4_1 model coefficients, which presents only 2% of the dynamic range required by the DVR-6_9 model

coefficients, making it an impressive property to be taken into consideration. Further, the condition number of the DVR-4_1 model is significantly reduced compared to the DVR-6_9 model reflecting that there is a meaningful difference in the computational complexity to identify the coefficients of both models. Besides, the complexity also corresponds to the realization of the basis functions, and each order of nonlinearity corresponds to an additional multiplication.

3.5.8.4.3 Comparison of Postdistortion Results

Using the postdistortion architecture, Table 3.19 summarizes the linearization performance of both models, which are presented in terms of NMSE, ACPR, and EVM_{app}. This confirms the rough similarity of the linearization performance of both models.

Table 3.19: Comparison of DVR-4_1 and DVR-6_9 in terms of linearization performance

	DVR-4_1	DVR-6_9
NMSE (dB)	-38.02	-38.67
ACPR_{L1} (dB)	-47.12	-47.91
ACPR_{L1} (dB)	-47.59	-48.26
EVM_{app} (%)	0.97	0.95

Figure 3.38 shows the power spectrum of the DVR-4_1 and DVR-6_9 models, which are compared to the PA output.

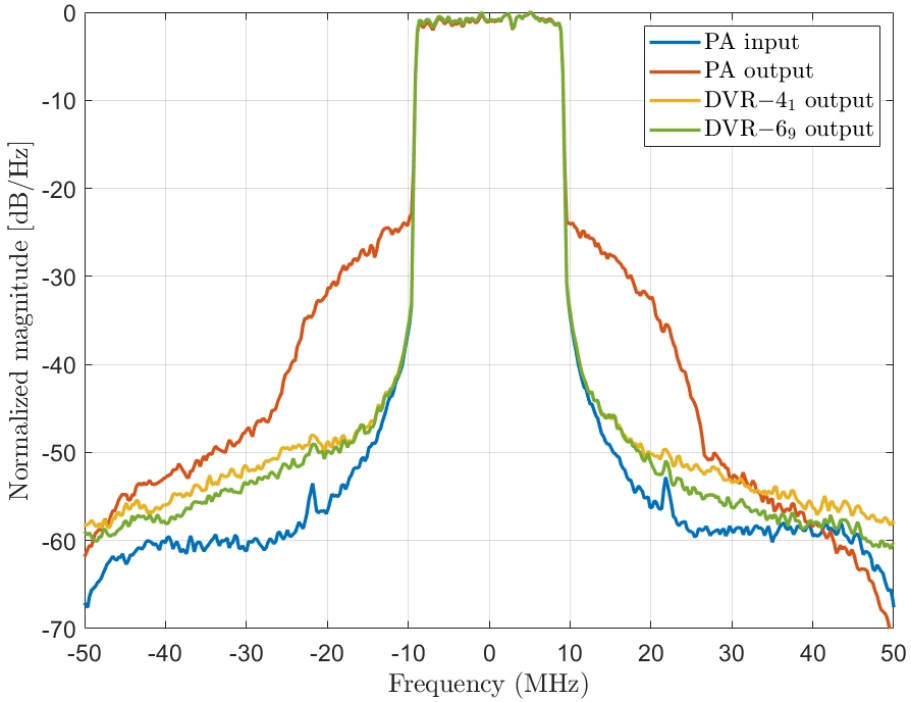


Figure 3.38: Spectra of PA input-output and postdistorter output using DVR-4_1 and DVR-6_9 models

As can be seen, the DVR-4_1 and DVR-6_9 models significantly improve ACPR.

3.5.8.4.4 Comparison of HC Complexity

In Table 3.20, we summarize the results of the HC algorithm reported in Section 3.5.8.2.5 and Section 3.5.8.3.2.2, where the HC complexity is highlighted in terms of the

total number of DVR models evaluated during the search, the HC iterations, the execution time taken by the HC algorithm to perform, and the total number of coefficients by summing the coefficients of all the neighbors tested.

Table 3.20: Comparison of HC complexity for $P = 1$ and $P = 9$

	$P = 1$	$P = 9$
Total number of neighbors	399	1067
HC iterations	6	11
Execution time	1.18 min	3.73 min
Total number of coefficients	5850	20728

Once again, when $P = 1$, this characteristic has a substantial impact on the HC complexity. By taking a general view of the results in Table 3.20, we can conclude that the HC complexity increases by around 300% if we change P from 1 to 9.

The distribution of the neighbors over the HC iterations is depicted in Figure 3.39. According to the HC iteration, the curves present the number of neighbors for $P = 1$ in blue and $P = 9$ in red.

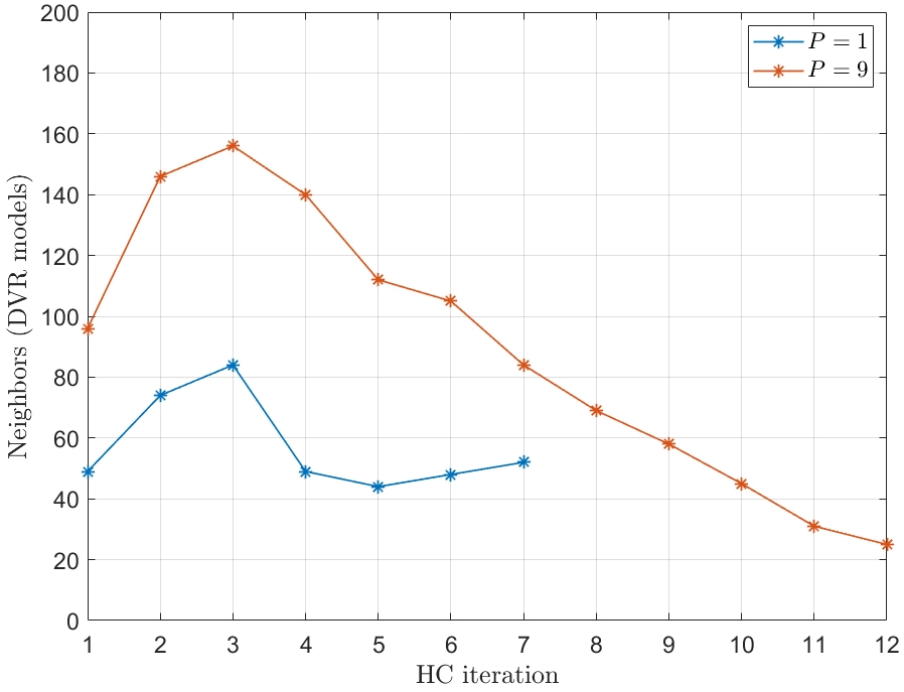
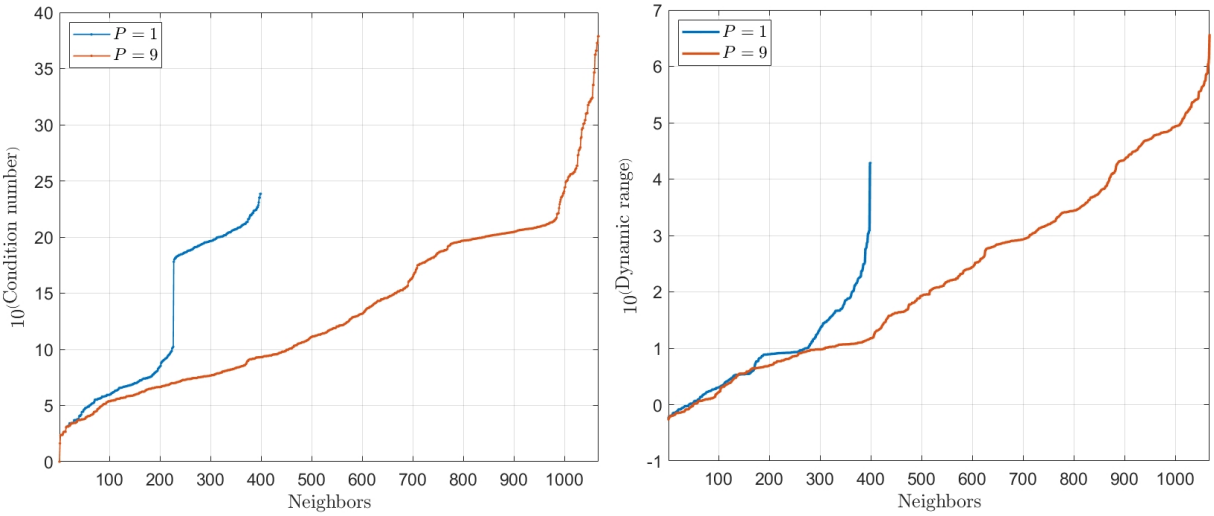


Figure 3.39: Neighbors per each HC iteration

The two curves behave roughly in the same way since the number of neighbors evaluated at the 3^{rd} HC iteration is the highest number. This can be justified because the HC algorithm explores many new neighbors compared to the previous HC iteration. The number of neighbors decreases from the 3^{rd} HC iteration, which amounts to the fact that the HC algorithm finds common neighbors between previous HC iterations.

3.5.8.4.5 Comparison of Numerical Properties

The condition number can be a quantitative factor to represent the computational complexity as it is involved in identifying the model coefficients. Their dynamic range is also discussed. For this, we propose to present for each neighbor the condition number and the dynamic range, which are sorted in ascending order and represented in Figure 3.40.



(a) Condition number of regressor matrix (b) Dynamic range of model coefficients

Figure 3.40: Numerical properties of all neighbors evaluated during the search sorted in ascending order

As shown in Figure 3.40, the numerical properties of the DVR models tested during the search are reduced when P is set to 1, unlike when $P = 9$. In fact, this can be interpreted by the shape of the curves. For example, in Figure 3.40a, the blue curve that shows the results with $P = 1$ has nearly 220 neighbors whose condition number does not exceed 10^{10}. These neighbors have a reduced computational sensitivity to identify their model coefficients.

A particular behavior is observed in the blue curve, where a discontinuous form occurs at neighbor number 220 to the next (in a sorted way), where the condition number suddenly moves from 10^{10} to 10^{18}. This demonstrates that when $P = 1$, the HC algorithm evaluates neighbors of a given element with different numerical properties, even if they are subject to the definition we proposed for the neighborhood, which consists of adding or removing one function to the model parameters. Unlike the blue curve, the red curve representing the HC algorithm with $P = 9$ has a roughly progressive linear shape over nearly 1000 neighbors, which causes an enormous calculation, which is highlighted in terms of execution time in Table 3.20.

Regarding the dynamic range of the model coefficients depicted in Figure 3.40b, the blue curve behaves as an exponential shape, which means that the majority of neighbors tested have a reduced dynamic range between 0.5 and 316, while the red curve ($P = 9$) has a progressive shape that goes up to around 10^6.

3.5.8.4.6 Discussion

The maximum nonlinearity order P for the DVR model significantly impacts the HC algorithm to determine an optimal structure of the DVR model. Setting P to 1 drive the HC algorithm to determine an optimal structure with multiple segments K and remarkable numerical properties, while the choice of P equal 9 leads the HC algorithm to choose an optimal structure dominated by the model term $T_{1,p}$ with higher nonlinearity order of p which causes wicked numerical properties. In contrast, the number of segments K remains at two segments. On the other hand, the computational complexity behind determining the optimal structure is exploded by 300 % when P vary from 1 to 9.

3.5.9 Conclusion of DVR Model Sizing

In this section, the HC algorithm proposed for the GMP model has been extended to the DVR model to find the optimal DVR model structure in a minimal number of computations.

Two important aspects are highlighted in this framework: the new definition of the neighborhood proposed for the HC algorithm and the definition of a cost function to achieve the best trade-off between modeling accuracy and complexity. The HC algorithm parameters such as coefficient distance and the weighting coefficient of the cost function are discussed.

The impact of the nonlinearity order present in the DVR model has been studied through the model sizing using the HC algorithm.

3.6 Conclusion

In this chapter, two processes are discussed and investigated.

The design of the optimal DVR is divided into two processes: The first process investigates how to optimally set the thresholds that define the boundary of each segment for a given structure, while the second process studies the determination of the optimal structure of the DVR model for a uniform segmentation.

The first process to optimize the thresholds of the DVR model is done by proposing an approach based on GS search. The proposed approach is based on decomposing the global optimization problem into a set of sub-problems where the objective function is unimodal. GS search is used to solve each unimodal sub-problem. The results illustrate the effectiveness of the proposed approach and show that the optimization of the thresholds significantly improves the linearization performances compared to conventional DVR with uniform segmentation.

For the second process, we extend the HC algorithm proposed for the GMP model to the DVR model by introducing a new neighborhood definition to determine the neighbors of the evaluated structure. The convergence of the HC algorithm is ensured by a cost function that exhibits the best trade-off between the linearization performance and the model complexity. A comparison between the HC algorithm and exhaustive search is carried out. The HC algorithm validates its efficiency by returning an optimal structure close to the best structure resulting from the exhaustive search. Besides, the HC algorithm was performed with minimal execution time compared to the exhaustive search. One of the advantages of the HC algorithm is that the searching path can be controlled by configuring some parameters such as the coefficients distance d_C and the weighting coefficient of the cost function.

Chapter 4

A System Approach to Design and Implement Optimal DVR Model

4.1 Introduction

In the previous chapter, the optimal structure of the DVR model is determined according to a cost function representing the trade-off between modeling accuracy and complexity. The sizing process of the DVR model is based on the HC algorithm, where the neighborhood definition we proposed allows us to converge towards the optimal solution.

On the other hand, the thresholds that bound the segments of the DVR model are optimized over the normalized input range. The approach to optimize the thresholds consists of decomposing the global optimization problem into sub-problems that require the application of GS search.

Designing an optimal DVR model involves determining its optimal structure with optimal segmentation according to the same criteria as those used in the HC algorithm to size the DVR model. This process presents a combination of the HC algorithm and the proposed approach to optimize the thresholds in a particular way.

In this chapter, we propose system approaches (SysApp) to perform this joint optimization to design the optimal DVR model. These SysApp are developed through different versions, discussed, and compared to the HC algorithm regarding the optimal solution, linearization performance, and complexity.

Once the optimal DVR model has been designed, it is essential to emphasize its numerical properties for hardware implementation. Indeed, the purpose of designing a predistorter is to implement it in a hardware device such as an FPGA.

In this chapter, we investigate and propose an implementation strategy by focusing mainly on the optimal DVR model designed by our system approaches.

4.2 Design of Optimal DVR Model

The optimal DVR model design consists of determining its optimal structure with the optimal segmentation according to the same criteria used in the HC algorithm to size the DVR model that was discussed in Section 3.5.

4.2.1 System approaches

In this study, we propose different SysApp to design the optimal DVR model, which is developed through different versions, and presented as follows:

- Direct approach, which is based on integrating the proposed approach in Section 3.4 into the HC algorithm developed to size the DVR model. This approach is indicated by D-SysApp.
- Simplified approach where the thresholds of the HC envelope are optimized. It is indicated by S-SysApp.
- Optimal approach based on the junction of the thresholds optimization and the HC algorithm, of which the running of the proposed approach to optimize the thresholds is controlled by the number of segments and the model terms. It is indicated by O-SysApp.

These approaches are discussed and compared to the HC algorithm with uniform segmentation in terms of linearization performance of the optimal solution and implementation complexity, including computational complexity and numerical properties.

4.2.1.1 Direct Approach (D-SysApp)

The first system approach called direct approach (D-SysApp) can be seen as an intuitive approach to design the optimal model by combining the HC algorithm described in Algorithm (6) and the proposed approach based on GS search for optimizing the thresholds β described in Algorithm (3).

Indeed, the optimization of β_k is performed for each neighbor through the GS search. The cost function f is computed according to C and N of each model with its optimized segmentation $\boldsymbol{\beta}_{opt}$.

At first sight, this approach has the highest computational complexity. Its complexity will be primarily evaluated by the total number of NMSE computations, including those required for the proposed approach to optimizing $\boldsymbol{\beta}$.

The principle of D-SysApp is described in Algorithm (7).

Algorithm 7: Algorithm of D-SysApp

Choose the initial element $x_{0(1)}$
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  Determine the neighbors $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    Optimize $\boldsymbol{\beta}$ for each $x_{i(q)}$
    $x_{s(q)} = \underset{x_{i(q)} \in V_{(q)}}{argmin} \left(f(x_{i(q)}) \right)$
    Save $x_{s(q)}$ in solution space $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin} \left(f(x_s) \right)$

4.2.1.2 Simplified Approach (S-SysApp)

The direct approach can be simplified in complexity by optimizing the thresholds only for specific DVR model structures.

These structures are determined according to 2 approaches:

- Senv-SysApp: The structures which are on the envelope of the HC algorithm.
- Sitr-SysApp: The structures which are determined at the end of each HC iteration.

The principle of Senv-SysApp is to apply the threshold optimization process using GS search once the HC algorithm is finished. The complexity of Senv-SysApp is mainly evaluated by the complexity of the HC algorithm, with the complexity of the GS search multiplied by the number of structures found on the envelope.

On the other hand, Sitr-SysApp consists of inserting the GS search in the HC algorithm, which will be activated once the HC iteration is finished and applied to the best neighbor of this HC iteration. Similar to Sitr-SysApp, the complexity of Senv-SysApp is primarily rated by the HC complexity, with the complexity of the GS search multiplied by the number of HC iterations.

Since the threshold optimization of the best neighbor at a given HC iteration improves its cost function, it may likely force the HC algorithm to stop earlier, mainly if no neighbor in the next HC iteration is found with a better cost function than the previous one. However, Sitr-SysApp remains a relatively limited approach where determining the

solution at each HC iteration relies on comparisons between a structure with optimal segmentation and others with uniform segmentation.

The algorithm of Senv-SysApp is described in Algorithm (8).

Algorithm 8: Algorithm of Senv-SysApp

Choose the initial element $x_{0(1)}$
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  Determine the neighbors $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    $x_{s(q)} = \underset{x_{i(q)} \in V_{(q)}}{argmin} \left(f(x_{i(q)}) \right)$
    Save $x_{s(q)}$ in solution space $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
for $x_s \in S_{(q)}$ **do**
  Optimize $\boldsymbol{\beta}$ of x_s
  Update $f(x_{s(i)})$ with $\boldsymbol{\beta}_{opt}$ in $S_{(q)}$
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin} \left(f(x_s) \right)$

The algorithm of Sitr-SysApp is described in Algorithm (9).

Algorithm 9: Algorithm of Sitr-SysApp

Choose the initial element $x_{0(1)}$
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  Determine the neighbors $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    $x_{s(q)} = \text{argmin}_{x_{i(q)} \in V_{(q)}} \left(f(x_{i(q)}) \right)$
    Optimize $\boldsymbol{\beta}$ of $x_{s(q)}$
    Update $f(x_{s(i)})$ with $\boldsymbol{\beta}_{opt}$ in $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin} \left(f(x_s) \right)$

4.2.1.3 Joint Optimal Approach (O-SysApp)

The motivation behind the joint optimal approach, indicated by O-SysApp, is to execute the HC algorithm to size the DVR model on a set of neighbors with optimal segmentation, i.e., the optimal segmentation of the trained DVR models is available and stored in LUT.

The size of the LUT depends on the number of times the optimization process on the thresholds $\boldsymbol{\beta}$ has been executed. The optimal thresholds are stored in the LUT along with the search for the optimal solution by the HC algorithm, which makes it a joint approach. Although the perfect case is where the optimal segmentation of all structures is available and stored, it cannot be performed because it requires an exhaustive search, which leads to D-SysApp.

As seen in Section 3.4.9, Section 3.4.10, and Section 3.4.11, the parameters K, M, and T_S have different impacts on the thresholds optimization process. The property of each parameter impact could be used to built O-SysApp through two versions:

- Ored-SysApp: we consider that the optimal segmentation is not sensitive to parameters M and T_S. The thresholds $\boldsymbol{\beta}$ are optimized only when the number of segments K changes over the HC search. The optimal segmentation corresponding to each K is saved in LUT.

- Ogen-SysApp: we consider that the optimal segmentation is not sensitive to the

parameter M. The thresholds $\boldsymbol{\beta}$ are optimized when the number of segments K or model terms T_S change through the HC algorithm.

The principle of Ored-SysApp is to optimize the thresholds of a given model structure only when K changes. The optimal segmentation of each K is stored in LUT. The size of LUT depends on the maximum number of segments explored by the HC algorithm.

An optimal segmentation $\boldsymbol{\beta}_{opt}^{(K)}$ for a given K covers all structures that have the same K, regardless of the other parameters (M, T_S).

On the other hand, the threshold optimization process is based on the methods proposed in Section 3.4.9 in order to speed up the convergence of the GS search.

The algorithm of Ored-SysApp is described in Algorithm (10).

Ogen-SysApp is an upgraded version of Ored-SysApp, where the algorithm is similar, except, when the model terms change, the thresholds optimization process is called. The Ogen-SysApp is more general than Ored-SysApp since the model terms T_S are considered as a sensitive parameter in addition to K. The parameter M remains insensitive as indicated in Section 3.4.10.

The algorithm of Ogen-SysApp is described in Algorithm (11).

Algorithm 10: Algorithm of Ored-SysApp

Choose the initial element $x_{0(1)}$
Optimize $\boldsymbol{\beta}$ of $x_{0(1)}$
Store $\boldsymbol{\beta}_{opt}^{(K_0)}$ in LUT
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  Determine the neighbors $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    if $\boldsymbol{\beta}_{opt}^{(K_i)} \notin LUT$ **then**
      if $\boldsymbol{\beta}_{opt}^{(K_i-1)} \in LUT$ **then**
        $\boldsymbol{\beta}_{init} = [\frac{\beta_{1_{opt}}^{(K_i-1)}}{2}; \beta_{1_{opt}}^{(K_i-1)}; \cdots ; \beta_{K_i-2_{opt}}^{(K_i-1)}]$
      else if $\boldsymbol{\beta}_{opt}^{(K_i+1)} \in LUT$ **then**
        $\boldsymbol{\beta}_{init} = [\beta_{2_{opt}}^{(K_i+1)}; \cdots ; \beta_{K_i{opt}}^{(K_i+1)}]$
      else
        $\boldsymbol{\beta}_{init} = [\frac{1}{K_i}, ..., \frac{K_i-1}{K_i}]$
      end
      Optimize $\boldsymbol{\beta}_{opt}^{(K_i)}$ of $x_{i(q)}$ with initial segmentation $\boldsymbol{\beta}_{init}$
      Store $\boldsymbol{\beta}_{opt}^{(K_i)}$ in LUT
    else
      Get $\boldsymbol{\beta}_{opt}^{(K_i)}$ from LUT
    end
    Update $x_{i(q)}$ with $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})}$
    $x_{s(q)} = \text{argmin}_{x_{i(q)} \in V_{(q)}} \left(f(x_{i(q)})\right)$
    Save $x_{s(q)}$ in solution space $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin}\,(f(x_s))$

Algorithm 11: Algorithm of Ogen-SysApp

Choose the initial element $x_{0(1)}$
Optimize $\boldsymbol{\beta}$ of $x_{0(1)}$
Store $\boldsymbol{\beta}_{opt}^{(K_0,T_{S_0})}$ in LUT
Evaluate the initial cost function $f(x_{0(1)})$
$x_{s(0)} = x_{0(1)}$
Initialize $q = 1$
while *(1)* **do**
  Determine the neighbors $x_{i(q)} \in V_{(q)}$ of $x_{0(q)}$
  if $V_{(q)} = \emptyset$ **then**
    end while loop
  else
    if $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})} \notin LUT$ **then**
      if $\boldsymbol{\beta}_{opt}^{(K_i-1)} \in LUT$ **then**
        $\boldsymbol{\beta}_{int} = [\frac{\beta_{1_{opt}}^{(K_i-1)}}{2}; \beta_{1_{opt}}^{(K_i-1)}; \cdots; \beta_{K_i-2_{opt}}^{(K_i-1)}]$
      else if $\boldsymbol{\beta}_{opt}^{(K_i+1)} \in LUT$ **then**
        $\boldsymbol{\beta}_{init} = [\beta_{2_{opt}}^{(K_i+1)}; \cdots; \beta_{K_{iopt}}^{(K_i+1)}]$
      else
        $\boldsymbol{\beta}_{init} = [\frac{1}{K_i}, ..., \frac{K_i-1}{K_i}]$
      end
      Optimize $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})}$ of $x_{i(q)}$ with initial segmentation $\boldsymbol{\beta}_{init}$
      Store $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})}$ in LUT
    else
      Get $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})}$ from LUT
    end
    Update $x_{i(q)}$ with $\boldsymbol{\beta}_{opt}^{(K_i,T_{S_i})}$
    $x_{s(q)} = \text{argmin}_{x_{i(q)} \in V_{(q)}} \left(f(x_{i(q)})\right)$
    Save $x_{s(q)}$ in solution space $S_{(q)}$
    if $C(x_{s(q-1)}) < C(x_{s(q)}) \leq C_{max}$ **then**
      if $N(x_{s(q)}) \leq N(x_{s(q)}) + \Delta\nu$ **then**
        $q = q + 1$
        $x_{0(q)} = x_{s(q-1)}$
      else
        end while loop
      end
    else
      end while loop
    end
  end
end
Return the optimal solution $x_s = \underset{x_s \in S}{argmin}\,(f(x_s))$

4.2.2 Experimental Results

The effectiveness of the proposed approaches is validated using the same experimental conditions as those reported in Section 3.3.

The HC algorithm part of the system approaches is performed with $C_{max} = 50$, $\alpha = 0.05$, and $d_C = 5$ coefficients. The nonlinearity order P is set to $P = 1$. The different system approaches have the same stop condition as the HC algorithm with uniform segmentation.

4.2.2.1 Search Behavioral

In this section, we present the convergence behavior of each system approach where the HC algorithm results with uniform segmentation are plotted in the background.

We remind that the optimal structure returned by the HC algorithm with uniform segmentation, with $C_{max} = 50$, $\alpha = 0.05$, $d_C = 5$, and $P = 1$ is:

$$
\begin{aligned}
&K = 4 \\
&\boldsymbol{\beta} = [0.25\ 0.5\ 0.75] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_{1,0}\ T_3\ T_6]
\end{aligned}
$$

with $C = 27$ coefficients and $N = -38.02$ dB.

4.2.2.1.1 D-SysApp

Figure 4.1 presents the results of the direct approach (D-SysApp), where the neighbors and the envelope are plotted in blue.

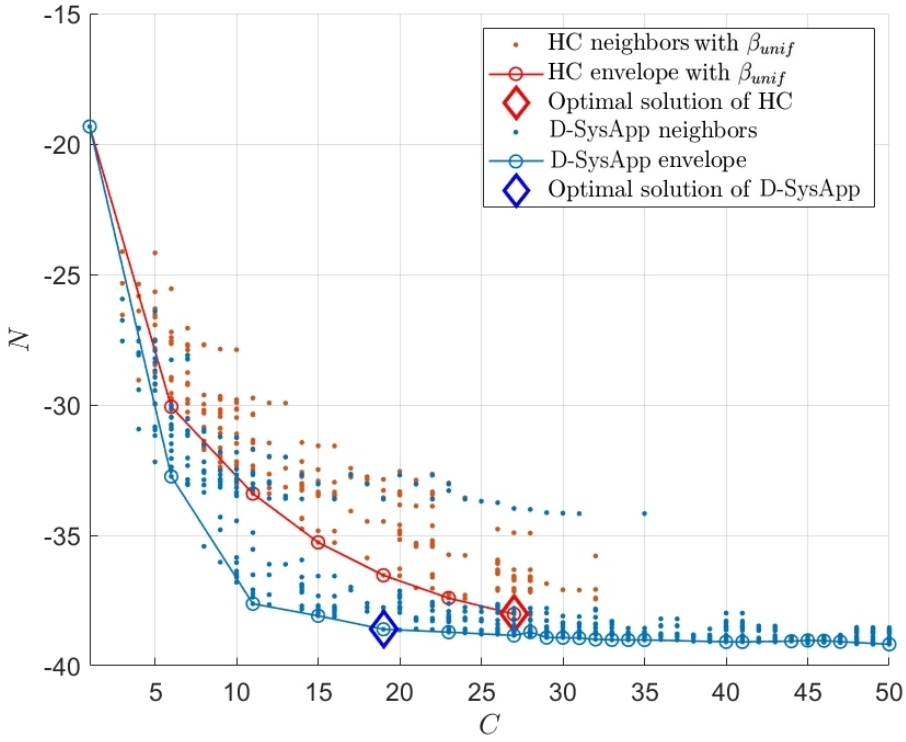


Figure 4.1: Behavior search of D-SysApp

The structure of the optimal solution returned by D-SysApp is:

$$K = 4$$
$$\boldsymbol{\beta}_{opt} = [0.59\ 0.82\ 0.93]$$
$$M_{lin} = 2\ ;\ M = 1$$
$$T_S = [T_2\ T_6]$$

with $C = 19$ coefficients and $N = -38.6$ dB.

Compared to the optimal solution of the HC algorithm, the number of segments and the memory depth are similar in both architectures. The model terms T_S are different. On the other hand, the position of the optimal thresholds is almost similar to what has been shown in Section 3.4 when $K = 4$.

4.2.2.1.2 S-SysApp

4.2.2.1.2.1 Senv-SysApp

Figure 4.2 presents the results of Senv-SysApp, which consists of applying the GS search on the envelope of the HC algorithm.

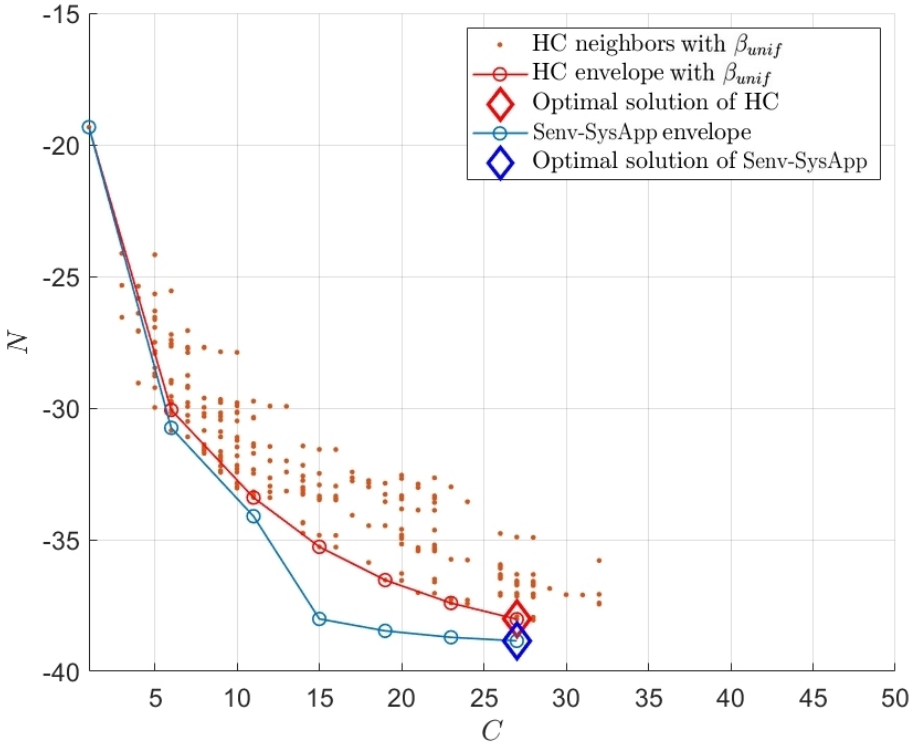


Figure 4.2: Behavior search of Senv-SysApp

As it can be seen from Figure 4.2, the thresholds optimization of the HC envelope significantly improves NMSE.

The structure of the optimal solution returned by Senv-SysApp is:

$$K = 4$$
$$\boldsymbol{\beta}_{opt} = [0.55\ 0.78\ 0.92]$$
$$M_{lin} = 2\ ;\ M = 1$$
$$T_S = [T_{1,0}\ T_3\ T_6]$$

with $C = 27$ coefficients and $N = -38.85$ dB.

In this approach, the optimal solution is chosen based on the cost function, i.e., the solution with the best f is the optimal solution.

4.2.2.1.2.2 Sitr-SysApp

Figure 4.3 presents the results of Sitr-SysApp, which consists of optimizing the structure of the best neighbor at each HC iteration.

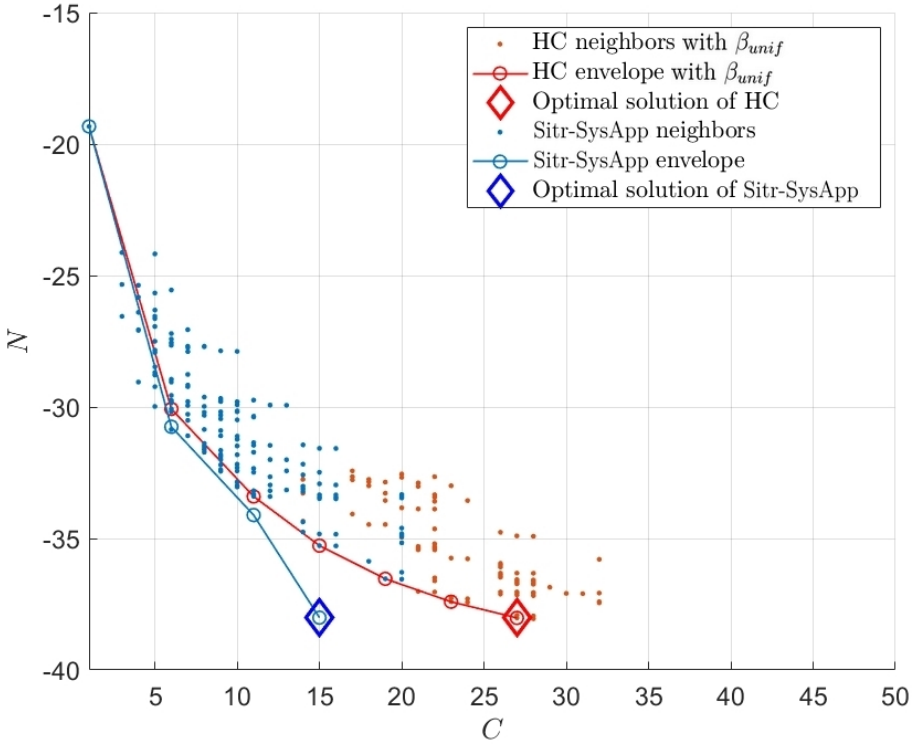


Figure 4.3: Behavior search of Sitr-SysApp

As shown in figure 4.3, Sitr-SysApp is stopped at the third HC iteration, which is justified by the fact that the GS search optimizes best neighbor thresholds, which significantly improves NMSE, hence the cost function. This impacts the HC algorithm, for which the probability of finding neighbors with better cost functions in the successive iterations remains low, causing the algorithm to stop.

The structure of the optimal solution returned by Sitr-SysApp is:

$$
\begin{aligned}
&K = 3 \\
&\boldsymbol{\beta}_{opt} = [0.26\ 0.88] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_3\ T_6]
\end{aligned}
$$

with $C = 15$ coefficients and $N = -38.01$ dB.

4.2.2.1.3 O-SysApp

4.2.2.1.3.1 Ored-SysApp

Figure 4.4 presents the results of Ored-SysApp described in Algorithm (10), where the structure of the optimal solution returned is:

$$
\begin{aligned}
&K = 4 \\
&\boldsymbol{\beta}_{opt} = [0.66\ 0.83\ 0.93] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_{1,0}\ T_{1,1}\ T_4\ T_5]
\end{aligned}
$$

with $C = 35$ coefficients and $N = -38.80$ dB.

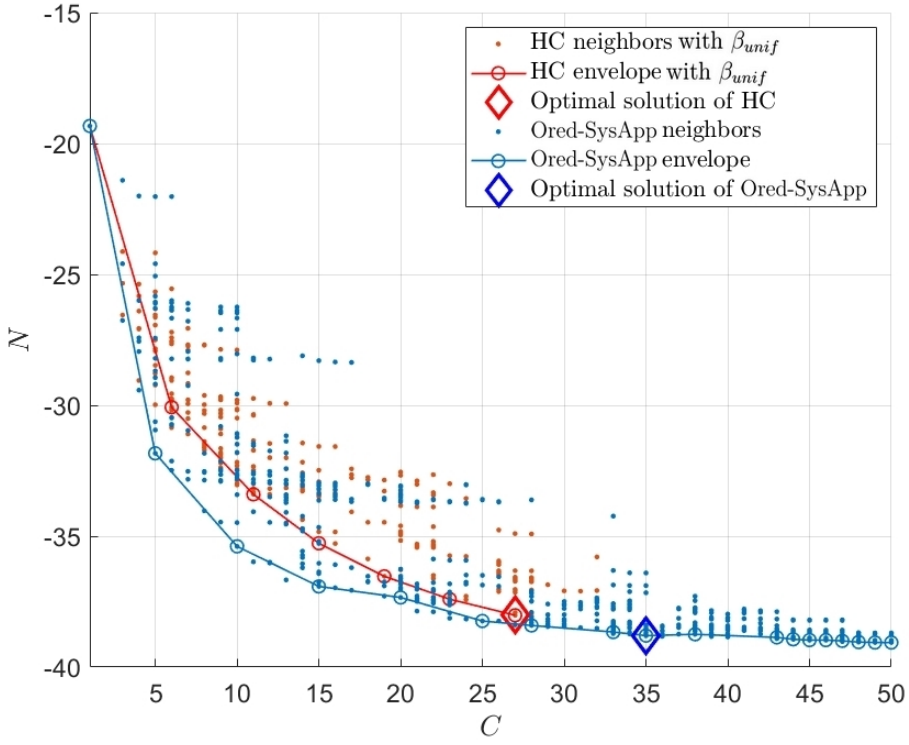


Figure 4.4: Behavior search of Ored-SysApp

Table 4.1 presents the LUT created by Ored-SysApp, where the optimal segmentation is saved according to the variation of K. In this scenario, the maximum number of segments K explored by Ored-SysApp is 7 segments.

Table 4.1: Optimal segmentation saved through Ored-SysApp

K	Optimal segmentation $\boldsymbol{\beta}_{opt}$
2	[0.84]
3	[0.59 0.87]
4	[0.65 0.83 0.93]
5	[0.36 0.66 0.82 0.93]
6	[0.04 0.24 0.55 0.83 0.95]
7	[0.11 0.27 0.51 0.65 0.83 0.95]

4.2.2.1.3.2 Ogen-SysApp

As described in Algorithm (11), the results of Ogen-SysApp to design an optimal DVR model are depicted in Figure 4.5.

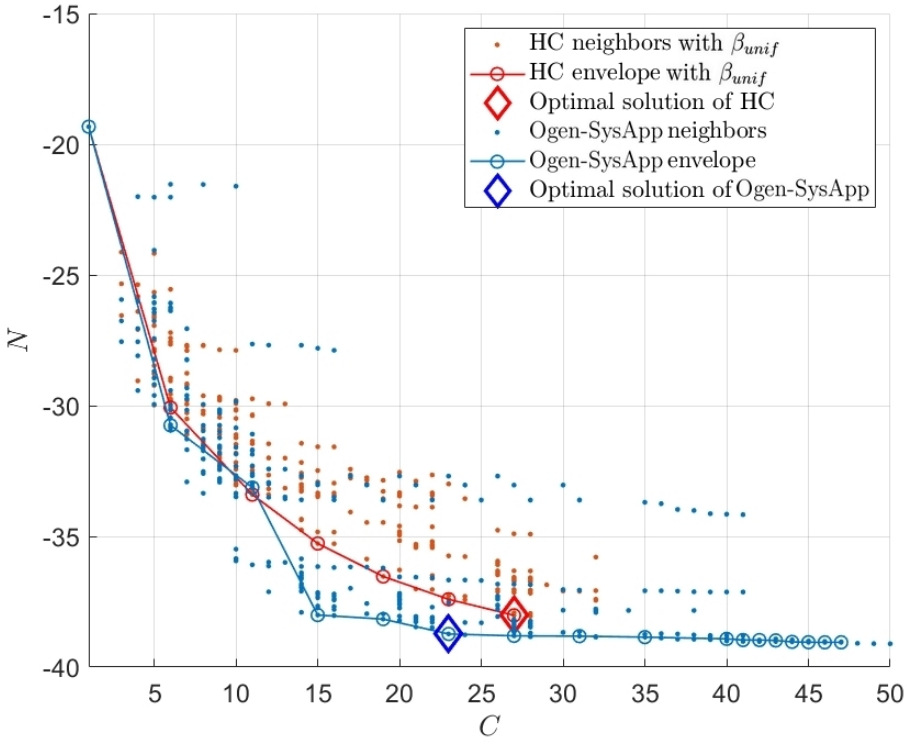


Figure 4.5: Behavior search of Ogen-SysApp

The structure of the optimal solution returned by Ogen-SysApp is:

$$
\begin{aligned}
&K = 5 \\
&\boldsymbol{\beta}_{opt} = [0.11\ 0.67\ 0.84\ 0.94] \\
&M_{lin} = 2\ ;\ M = 1 \\
&T_S = [T_3\ T_6]
\end{aligned}
$$

with $C = 23$ coefficients and $N = -38.93$ dB.

During Ogen-SysApp, the thresholds optimization process is performed 106 times, where the optimal segmentation is saved in 2D according to K and T_S.

4.2.2.2 Comparison of Search Behavioral

Figure 4.6 compares the search path of the envelope of each system approach, where the envelope of the HC algorithm with uniform segmentation is plotted in red. As can be seen, Ored-SysApp is sub-optimal from 10 to 30 coefficients, while Senv-SysApp, Sitr-SysApp, and Ogen-SysApp achieves nearly the same performance as D-SysApp from 15 coefficients. And as expected, the direct approach (D-SysApp) achieves the best performance.

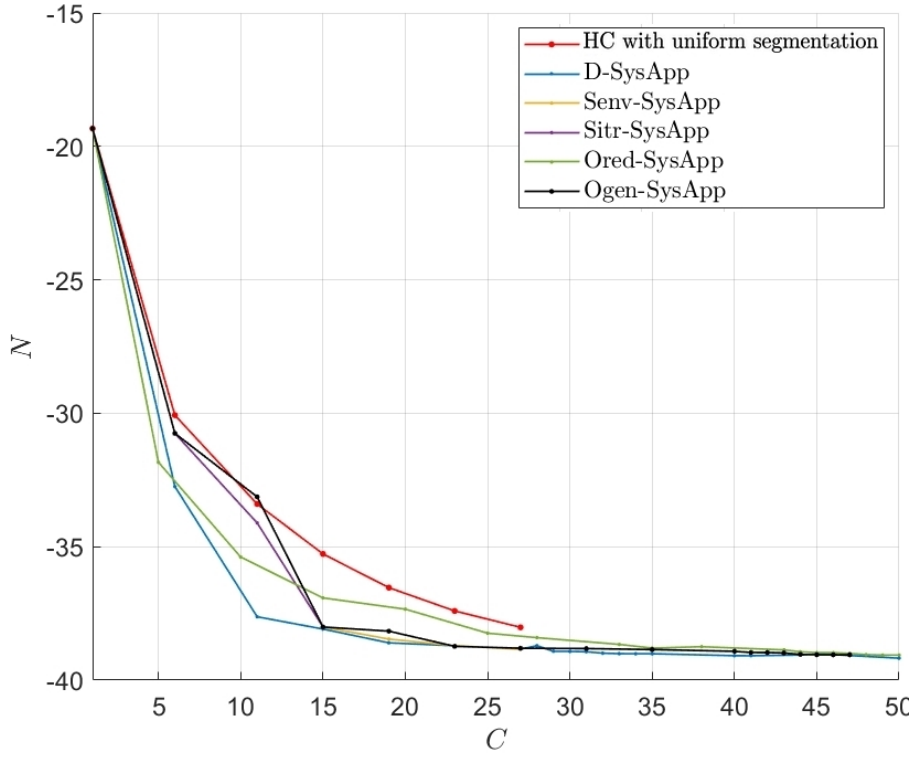


Figure 4.6: Behavior search comparison of different SysApp

4.2.2.3 Optimal Solution

Table 4.2 summarizes the optimal solutions returned by the different system approaches to design the DVR model.

Table 4.2: Comparison of optimal solutions of system approaches

		HC algorithm	D-SysApp	Senv-SysApp	Sitr-SysApp	Ored-SysApp	Ogen-SysApp
Model structure	K	4	4	4	3	4	5
	$\boldsymbol{\beta}$	[0.25 0.5 0.75]	[0.59 0.82 0.93]	[0.55 0.78 0.92]	[0.26 0.88]	[0.66 0.83 0.93]	[0.11 0.67 0.84 0.94]
	M_{lin}	2	2	2	2	2	2
	M	1	1	1	1	1	1
	T_S	$T_{1,0}$ T_3 T_6	T_2 T_6	$T_{1,0}$ T_3 T_6	T_3 T_6	$T_{1,0}$ $T_{1,1}$ T_4 T_5	T_3 T_6
C		27	19	27	15	35	23
N		-38.02	-38.6	-38.85	-38.01	-38.80	-38.93
f		-36.32	-37.65	-37.5	-37.26	-37.05	-37.78

As can be shown, the optimal solutions have the same model structure in terms of memory depths (M_{lin} and M), while the number of segments K and model terms T_S are different from an optimal solution to another.

In contrast, the optimal segmentation when the optimal solutions share the number of segments is roughly located on the same region, regardless of the model terms.

On the other hand, the optimal solutions have the same linearization performance but with different complexity. By computing the cost function with $\alpha = 0.05$, the best optimal solution is the one returned by Ogen-SysApp followed by D-SysApp, which is plausible since it is based on the full integration of the GS search within the HC algorithm. However, it is important to emphasize the complexity of each system approach to conclude on their effectiveness.

4.2.2.4 Complexity

The complexity of the proposed system approaches to design the DVR model is assessed in the same way as the complexity reduction of the HC algorithm described in Section 3.5.5.

Table 4.3 presents the complexity of the system approach, which is evaluated by the total number of models evaluated over the system approach, the HC iterations, the total number of NMSE computations including those of the threshold optimization process, the execution time (runtime), and the total number of model coefficients identified along with the system approach.

Table 4.3: Complexity comparison of system approaches

	D-SysApp	Senv-SysApp	Sitr-SysApp	Ored-SysApp	Ogen-SysApp
Evaluated DVR model	892	399	256	750	506
HC iterations	21	6	3	17	16
NMSE computations	51979	910	356	1100	3866
Runtime (min)	294	2.1	0.6	4.9	8.54
Model coefficients	74511	5952	2575	19185	28537

According to Table 4.3, we can see that the behavior of each approach impacts the complexity of its convergence towards the optimal solution.

The runtime of D-SysApp is very important with a high number of NMSE computations, so it cannot be a realistic candidate to design the DVR model, especially for real-time DPD adaptation. Senv-SysApp and Sitr-SysApp remain candidates in terms of the runtime and number of NMSE computations. Ored-SysApp and Ogen-SysApp exhibit superior complexity behavior compared to Sens-SysApp and Sitr-SysApp.

4.2.3 Discussion and Conclusion

According to the results in Table 4.2 and Table 4.3, and since D-SysApp is excluded from our choice due to its computational complexity, we can conclude that Ogen-SysApp is an excellent candidate to design the DVR model since it presents a good trade-off between linearization performance and computational complexity.

On the other hand, Sitr-SysApp exhibits a similar linearization performance compared to Ogen-SysApp at reduced complexity (15 model coefficients) and with the same model terms, which cannot be removed from our choice. However, the stop criterion of Sitr-SysApp can be subject to improvement to make it more superimposed with Ogen-SysApp in terms of the search path. Another viewpoint that can be deepened and elaborated is to develop a system approach by combining Ored-SysApp and Sitr-SysApp. When Ored-SysApp allows having a particular slope of the searching behavior, we switch to Sitr-SysApp to refine the search path.

4.3 Hardware Implementation Strategy of Optimal DVR Model

The design of an appropriate predistorter based on a software environment such as MATLAB is suitable for laboratory usage but not for real-time DPD applications. Besides, software solutions provide difficulty in implementing the adaptive algorithm. Therefore, to meet the industrial requirements, the DPD system should be implemented in hardware such as FPGA, which has many advantages in digital signal processing, including high-speed processing, flexible implementation, and high reliability.

Two critical points to be considered during the translation from software to hardware are: accuracy and speed. Parallel operation mechanisms can improve these points.

In the hardware implementation of DPD, there are mainly two processes:

- The application of DPD on the input signal needs to be sampled at a sufficient sampling rate to meet the growing demand for wideband signals in the current telecommunication systems and to be able to accommodate the widening of the signal bandwidth by the highest nonlinearity order of the DPD.

- The identification and update of the DPD coefficients, which can be done using block per block or sample per sample approaches.

As is mentioned in Chapter II, there are two structures to implement the DPD model in hardware: the LUT method and the direct structure with multipliers and adders. The LUT method stores the precomputed value of the gain functions, which are indexed by the input magnitude. On the other hand, the direct structure with multipliers and adders, the so-called direct multiple and add method, usually takes advantage of the pipeline architecture of the FPGA to increase the throughput in which more logic resources are involved compared to the LUT method.

In this section, we propose a hardware strategy to implement the optimal DVR model designed by the joint approach.

4.3.1 State-of-the-Art

According to the literature related to this research line, many works have been reported to implement the DPD models on hardware.

In [88], the authors describe the platform for the software-hardware architecture of DPD implementation for 3G transceivers, where the authors used a curve-fitting function to generates a smooth version of the AM-AM and AM-PM curves. In [58], the authors proposed hardware implementation of DPD, which is based on a nonlinear auto-regressive moving average structure, and mapped into a set of scalable LUTs.

Another work dedicated to implementing models based on the Volterra series is reported in [89], where an FPGA design is proposed and considers the memory depth. In [90], the authors proposed a hardware methodology with high-level synthesis for FPGA devices to implement the arithmetic representations of the models based on the Volterra series. A combination between FPGA and ARM processors is proposed in [92] to implement the MP model, where the authors used the ARM processor for its advantage in floating-point computations. In [93], the authors proposed a scalable LUT architecture to implement the GMP model using high-level synthesis.

Regarding the DVR model, few research works have been reported in the literature devoted to hardware implementation. In [66], the authors proposed a hardware structure by dividing the DVR model into a sub-composed format to reduce the implementation complexity. However, the parameters of the DVR model have been set arbitrarily without raising any concerns about the model complexity reduction. In [94], a model derived from the DVR model is proposed, where the hardware implementation has been emphasized.

4.3.2 Direct Multiply and Add Method

The direct method is based on the use of multipliers and adders to implement predistortion. For polynomials such as

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

the complexity of the direct method to evaluate the polynomial expression can be reduced by using the Horner's rule by formulating y as

$$a_0 + x(a_1 + x(\cdots + x(a_{n-2} + x(a_{n-1} + a_n x))))$$

The Horner's rule allows to evaluate a polynomial of degree n with only n multiplications and additions and it can be implemented with modular architectures using pipeline.

By extending this concept to the polynomial with memory (MP model, for example), the function y consists of several stages of $a + b|x(n-i)|$ in which n pipelined stages are required in the circuit. This block only deals with multiplication and summation algorithms, and it is simple to implement in FPGA.

However, the complexity of implementation depends on the structure of the model (in the case of the MP model, memory depth, and nonlinear order) since the multipliers are one of the most complex and expensive components in FPGA. In addition, it is necessary to take into account the dynamics of the model coefficients and the computation noise introduced by each elementary operation.

For the optimal DVR model designed by Ogen-SysApp and according to (3.3), it can be expressed by:

$$\begin{aligned} y(n) = & \sum_{i=0}^{2} a_i x(n-i) \\ & + \sum_{k=1}^{5} \sum_{i=0}^{1} c_{ki,3} ||x(n-i)| - \beta_k| \cdot x(n-i) \\ & + \sum_{k=1}^{5} \sum_{i=0}^{1} c_{ki,6} ||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \end{aligned}$$

which gives

$$
\begin{aligned}
y(n) = {} & a_0 \cdot x(n) + a_1 \cdot x(n-1) + a_2 \cdot x(n-2) \\
& + c_{10,3} \cdot ||x(n)| - 0.11| \cdot x(n) \\
& + c_{20,3} \cdot ||x(n)| - 0.67| \cdot x(n) \\
& + c_{30,3} \cdot ||x(n)| - 0.84| \cdot x(n) \\
& + c_{40,3} \cdot ||x(n)| - 0.94| \cdot x(n) \\
& + c_{50,3} \cdot ||x(n)| - 1| \cdot x(n) \\
& + c_{11,3} \cdot ||x(n-1)| - 0.11| \cdot x(n-1) \\
& + c_{21,3} \cdot ||x(n-1)| - 0.67| \cdot x(n-1) \\
& + c_{31,3} \cdot ||x(n-1)| - 0.84| \cdot x(n-1) \\
& + c_{41,3} \cdot ||x(n-1)| - 0.94| \cdot x(n-1) \\
& + c_{51,3} \cdot ||x(n-1)| - 1| \cdot x(n-1) \\
& + c_{10,6} \cdot ||x(n)| - 0.11| \cdot x(n) \cdot |x(n)|^2 \\
& + c_{20,6} \cdot ||x(n)| - 0.67| \cdot x(n) \cdot |x(n)|^2 \\
& + c_{30,6} \cdot ||x(n)| - 0.84| \cdot x(n) \cdot |x(n)|^2 \\
& + c_{40,6} \cdot ||x(n)| - 0.94| \cdot x(n) \cdot |x(n)|^2 \\
& + c_{50,6} \cdot ||x(n)| - 1| \cdot x(n) \cdot |x(n)|^2 \\
& + c_{11,6} \cdot ||x(n-1)| - 0.11| \cdot x(n) \cdot |x(n-1)|^2 \\
& + c_{21,6} \cdot ||x(n-1)| - 0.67| \cdot x(n) \cdot |x(n-1)|^2 \\
& + c_{31,6} \cdot ||x(n-1)| - 0.84| \cdot x(n) \cdot |x(n-1)|^2 \\
& + c_{41,6} \cdot ||x(n-1)| - 0.94| \cdot x(n) \cdot |x(n-1)|^2 \\
& + c_{51,6} \cdot ||x(n-1)| - 1| \cdot x(n) \cdot |x(n-1)|^2
\end{aligned}
$$

where 27 adders and 33 multipliers are required to achieve this hardware implementation. Implementing the optimal DVR model by direct method is explicitly illustrated in Figure 4.7, where the number of multipliers and adders used has not been optimized.

4.3.3 LUT Method

LUT method is an efficient solution to implement the gain function of predistorter in FPGA. Figure 4.8 depicts the basic cell of the LUT method, where the magnitude of the signal is used as an index.

The corresponding value of the gain function is precomputed and retrieved in the memory with the address associated with the quantified magnitude of the input sample.

Conventionally, for the polynomial models, the LUT is built only for each polynomial term [86] [58]. This method is more flexible since once the DPD coefficients are identified, the contents of LUT are updated without changing their size. However, more memory is needed to implement LUTs, and additional multipliers and adders are needed to combine all of the polynomial terms into the final result.

On the other hand, all terms with the same memory depth M can be calculated and saved in one LUT, allowing only $M + 1$ LUTs. This technique considerably reduces the number of multipliers and adders.

Figure 4.9 presents an example architecture of this LUT method to compute the polynomial function

$$\sum_{i=0}^{M} \sum_{k=0}^{K} a_{k,i} x(n-i)|x(n-i)|^k$$

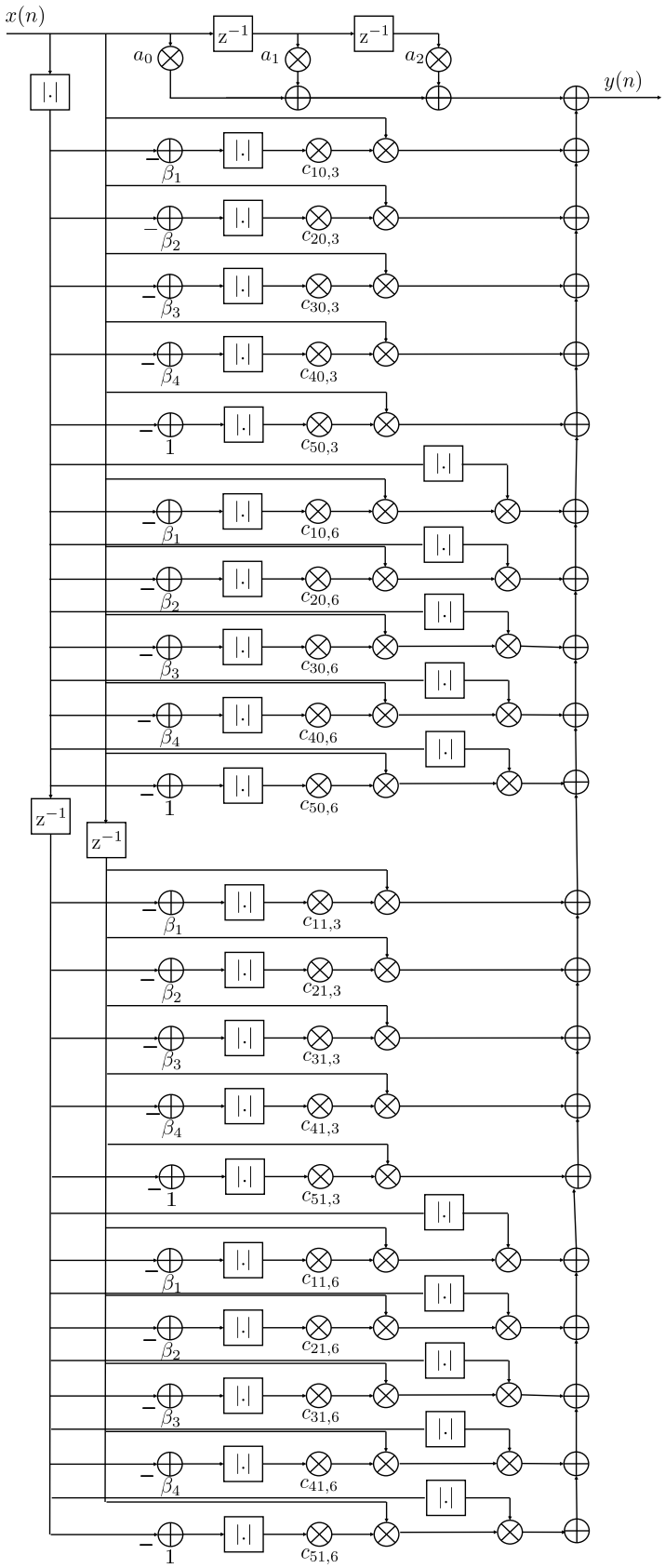


Figure 4.7: Implementation of optimal DVR model by direct method

which can be presented as

$$\sum_{i=0}^{M} x(n-i)(\sum_{k=0}^{K} a_{k,i}|x(n-i)|^k) = \sum_{i=0}^{M} x(n-i)G_i$$

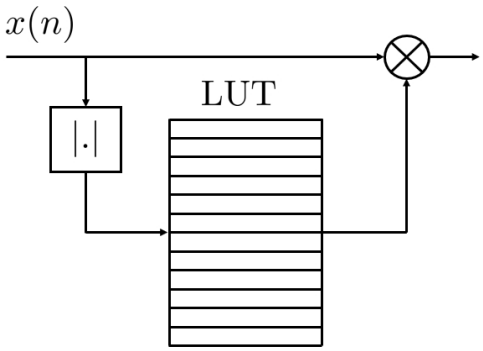


Figure 4.8: Basic cell of LUT method for polynomial

where the values stored in the LUT gains for $x(n-i)$ are

$$G_i = \sum_{k=0}^{K} a_{k,i}|x(n-i)|^k$$

and the number of LUT is controlled by the memory depth M.

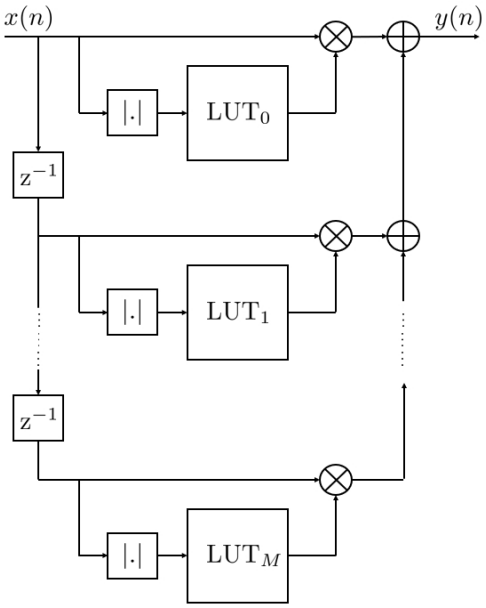


Figure 4.9: Architecture of LUT method to calculate polynomial

The LUT entries can be made with two methods of spacing: uniform and non-uniform [87] [67]. The non-uniform spacing based on the companding function has been proven to provide the best modeling accuracy. However, the probability density function of the input signal is required, which is generally of high computational complexity. Therefore, uniform spacing is often chosen to space the LUT due to its relatively low complexity and sufficiently good results compared to non-uniform spacing.

For the DVR model, the value stored in LUT corresponds to the function $G(|x(n-i)|)$ such as

$$G(|x(n-i)|) = \sum_{k=1}^{K} c_{ki}||x(n-i)| - \beta_k|$$

It should be noted that the precomputation is performed when updating the DPD coefficients.

For the optimal DVR model returned by Ogen-SysApp, its implementation using LUT is depicted in Figure 4.10.

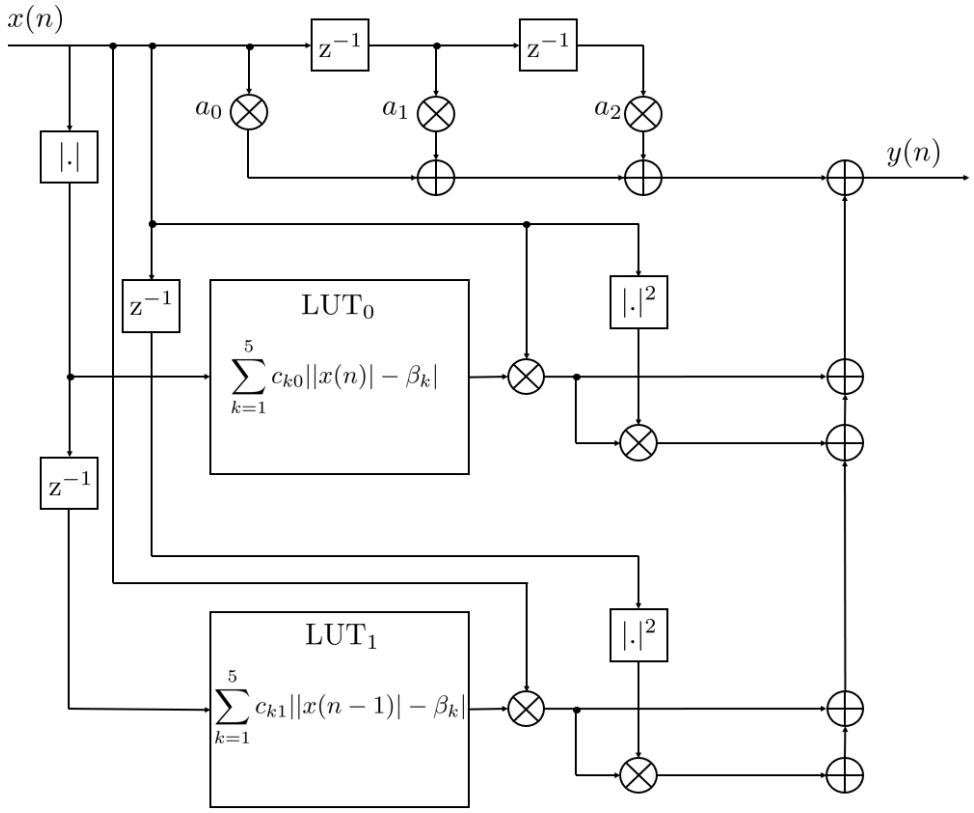


Figure 4.10: Implementation of optimal DVR model by LUT

It is seen that the number of adders and multipliers is reduced compared to the direct implementation, where 7 multipliers, 7 adders, and 2 LUTs are needed to achieve this implementation. Once the DPD coefficients are estimated, the two LUTs are updated with the precomputed values of their respective functions

$$\sum_{k=1}^{5} c_{k0}||x(n)| - \beta_k|$$

and

$$\sum_{k=1}^{5} c_{k1}||x(n-1)| - \beta_k|$$

The accuracy of the function

$$\sum_{k=1}^{K} c_{ki}||x(n-i)| - \beta_k|$$

depends on the number of bits of the LUT data, while the size of LUT depends on the number of bits used in the quantization of the signal and the interpolation used for the LUT values.

The complexity of the LUT implementation can be evaluated by the number of LUT (e.g., memory depth M of the DVR model), the quantization bits, and the number of bits required for each memory slot which indicates the resolution of the value stored in LUT.

Having 2 LUTs in Figure 4.10 refers to the fact that the optimal structure of the DVR model has $M = 1$, which is optimized by the joint approach we proposed to design the DVR model.

The length of the LUT, denoted by L, is defined by the number of bits N_Q used in the quantization of the signal by

$$L = 2^{N_Q} \tag{4.1}$$

In order to find the optimal L, we propose to evaluate the linearization performance of the optimal DVR model designed by the joint approach using the LUT implementation structure by varying N_Q from 2 to 20 bits.

Figure 4.11 presents the behavior of linearization performance in terms of NMSE of the DVR model according to the variation of N_Q from 2 bits to 20 bits. The NMSE from the direct approach (NMSE $= -38.93dB$) is presented as well.

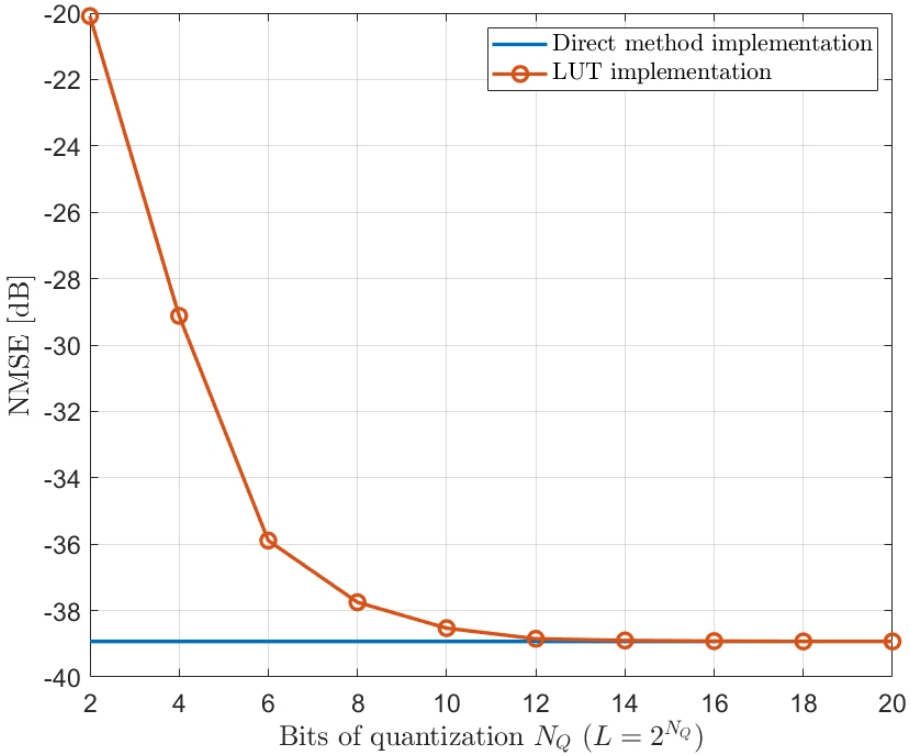


Figure 4.11: NMSE according to variation of N_Q

According to Figure 4.11, we see that the linearization performance by the LUT implementation converges to that by the direct implementation and that from $N_Q = 12$ bits that corresponds to $L = 4096$ rows, the linearization performances are identical.

4.4 Conclusion

In this chapter, we proposed a system approach combining the HC algorithm and GS search to design the optimal DVR model. The system approach is developed through different versions, where the concept is to determine the optimal structure of the DVR model with its optimal segmentation. These system approaches are discussed and compared in terms of the optimal solution and computational complexity.

LUT-based hardware implementation for optimal DVR model is proposed and optimally sized according to the linearization performances compared to the direct implementation, making it a solid strategy to implement the DVR model on hardware such as FPGA. On the other hand, it is essential to emphasize that some model parameters, such as the memory depth M, which impact the complexity of implementation (number of LUTs), could replace the number of coefficients in the cost function design since the number of coefficients has no impact on the implementation complexity by LUT. Nevertheless, the number of coefficients has an impact on the identification process.

The study of the optimal DVR model design is shown to be robust in this Chapter and will be applied as a DPD model to linearize a dual-input Doherty PA in the next chapter.

Chapter 5

Linearization and Efficiency Enhancement of Dual-Input Doherty Power Amplifier

5.1 Introduction

The introduction of non-constant amplitude modulated signal makes the enhancement of PA efficiency with maintaining an adequate level of linearity more challenging. Starting from wideband code division multiple access (W-CDMA) in 3G, the PAPR of signals increases with the use of OFDM in 4G LTE and 5G. Besides, the requirements for higher transmission rates should be satisfied. Wider signal bandwidths are needed to accommodate higher data rates.

For modulated signals with high PAPR, PA should operate at large power back-off, which decreases PA efficiency. Advanced architectures of PA based on dynamic load or supply modulation have been proposed in the literature to avoid wasting excessive power resources. Some of the most popular solutions are Doherty PA [95], envelope elimination and restoration [96], envelope tracking [97], and outphasing PA [98]. These highly efficient topologies require linearization techniques such as DPD to ensure the specified level of linearity according to the communication standards. Introducing DPD to the amplification system becomes more necessary to meet the linearity requirement, especially with increased signal bandwidth.

The amplification architectures based on active load modulation, one of the most common PA efficiency enhancement techniques, rely on the nonlinear interaction between the main and auxiliary transistors for modulated signals with a significant dynamic. Although these architectures can be designed with a single RF input to be used in the transmitter, several studies have been reported in the literature to highlight the benefits in maintaining separate inputs [99] [100] [101], and the advantages of dual-input Doherty PA compared to single-input has been studied in [102]. Therefore, additional degrees of freedom, so-called free parameters offered by the separate inputs, can be used to control the performance or to enhance the PA efficiency [103].

Focusing on dual-input PA architecture, searching for optimal free parameters guaranteeing high performance requires experimental cross-validation or exhaustive search. These processes are usually costly and computationally significant, especially when the search space is enormous and not limited.

Setting these free parameters to their optimal values within a defined interval can be viewed as a global optimization problem. Several techniques have been proposed in the literature to find the optimal set of free parameters among large tunable ranges considered as search intervals [104]. Typically, the global optimization techniques could be catego-

rized into three classes: the deterministic methods (Bayesian search algorithm, branch and bound algorithms, interval methods, etc.), the stochastic methods (simultaneous perturbation, simulated annealing, swarm algorithms, etc.), and the heuristic methods (GA, ant colony, tabu search, particle swarm optimization, etc.).

In this chapter, we propose a new architecture to enhance the power efficiency of the dual-input Doherty PA while meeting the linearity requirement. The proposed architecture is based on an auto-tuning approach consisting of optimizing the free parameters by a proposed hybrid optimization algorithm according to an adaptive cost function. This cost function adaptively indicates the trade-off between power efficiency and linearity. The free parameters to be optimized in the auto-tuning approach cover PA biasing, baseband calibration process such as the power level, and PAPR reduction. Besides, a DPD linearization is jointly optimized and integrated into our approach to meet the linearity requirements.

The remainder of this chapter is organized as follows.

Section II presents the system-level aspects, including the dual-input Doherty PA and the experimental testbench.

Section III presents the free parameters to be optimized.

In Section IV, we describe the proposed auto-tuning approach-based architecture to optimize the performance of the dual-input Doherty PA with the DPD linearization technique.

Section V presents the experimental results of the proposed approach.

Finally, the conclusion is given in Section VI.

5.2 System-Level Aspects

5.2.1 Dual-Input Doherty Power Amplifier

The PA, which is based on active load modulation (Doherty, outphasing) with separate RF inputs, can be viewed, by generalization, as the block diagram depicted in Figure 5.1.

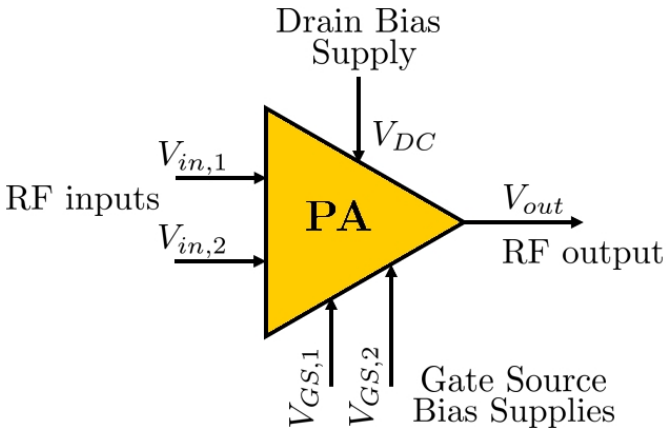


Figure 5.1: Block diagram of dual-input PA

The dual-input PA has two RF inputs, a drain bias V_{DC}, and two gate-source voltages $V_{GS,1}$ and $V_{GS,2}$ to control the transistor's terminal independently. A typical example of PA with independent V_{GS} is Doherty, where the main (carrier) amplifier is biased in class B and the auxiliary amplifier (typically named peaking amplifier) in class C. The instantaneous amplitude and phase of each input in baseband, as well as the V_{GS} gate bias voltages, can be controlled and adjusted separately, which allows having a significant degree of freedom for these parameters to improve the performance of the PA, in particular its efficiency.

At a given operating back-off (BO), the main amplifier reaches its maximum output voltage and becomes maximally efficient. From this power level, the auxiliary amplifier turns on and injects current into the common node, increasing the output power and modulating the load seen by the main amplifier.

In this chapter, the dual-input Doherty PA presented in [105] is used as the device under test (DUT). In [105], the authors have presented a 3.0 - 3.6 GHz wideband GaN Doherty PA with a frequency dependency compensating circuit. The architecture of the dual-input Doherty PA is presented in Figure 5.2.

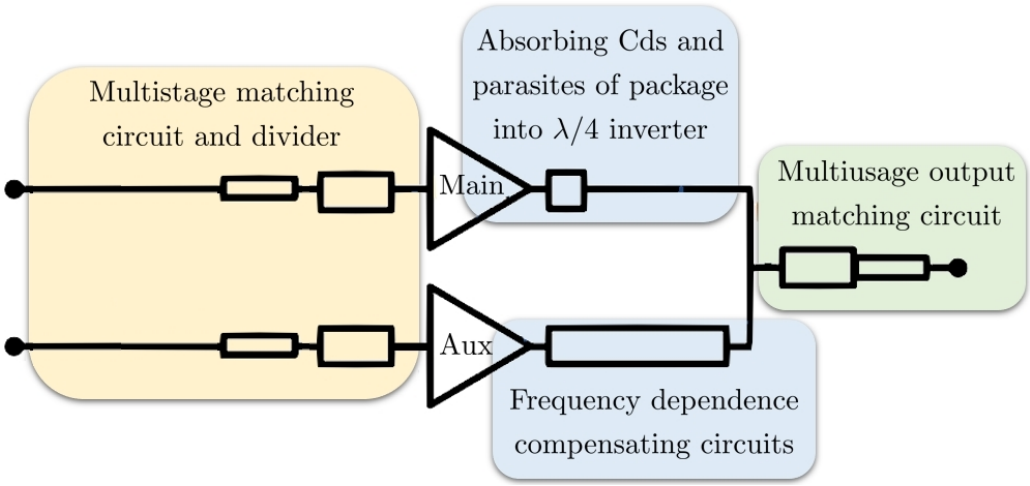


Figure 5.2: Architecture of Doherty PA with frequency dependency compensating circuit

Figure 5.3 shows the photo of the assembled dual-input Doherty PA. The dual-input Doherty PA contains two GaN HEMTs inside one package. The matching networks and the output hybrid couplers are based on micro-strip networks with SMD capacitors and resistors. The circuit is mounted on an aluminum fixture, and SMA coaxial launchers are used for the RF ports. The input and output circuits are fabricated with RO4350B substrates.

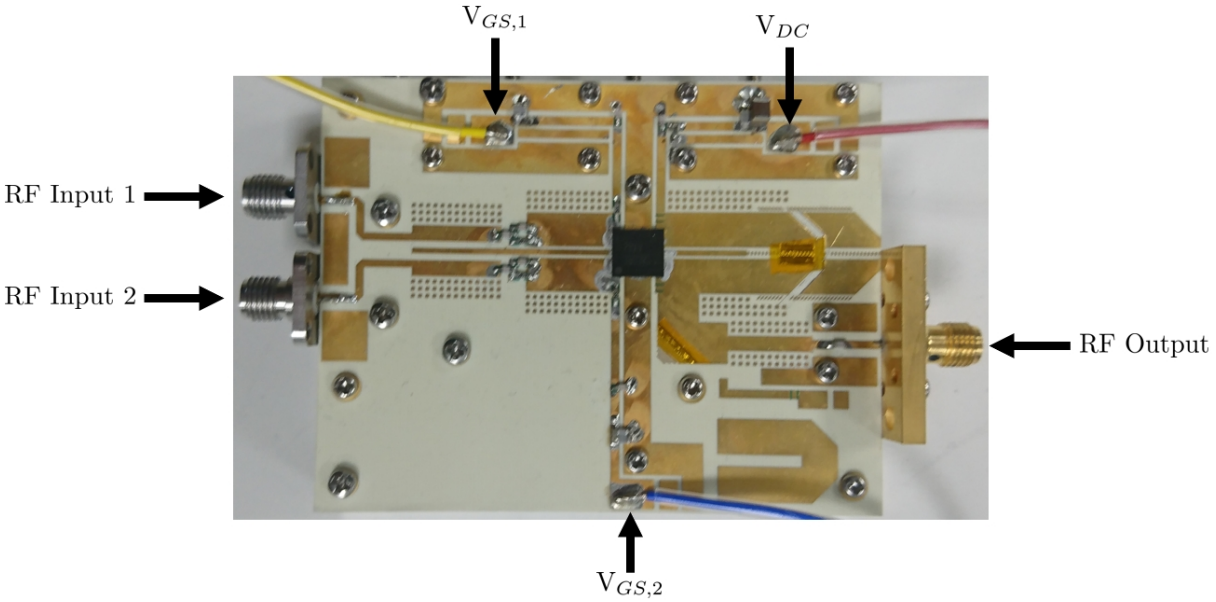


Figure 5.3: Photo of dual-input Doherty PA

In Figure 5.3, the upper path leads to a transistor operating as the main amplifier, and the lower one leads to a transistor operating as a peaking amplifier.

This dual-input Doherty PA can produce a peak output power of 40 dBm (10 W) and has a linear gain of 15 dB. The measurements reported in [105] have been performed under a modulated signal with 20 MHz channel LTE signal and showed, over 3.0 − 3.6 GHz frequency range with an efficiency exceeding 40%. The linearizability of this PA has been studied as well, where the ACPR has been improved by 10 - 15 dB using DPD linearization.

Nevertheless, the two RF inputs of Doherty PA in [105] were connected to the same RF input, and no optimization process has been reported for this DUT.

5.2.2 Testbench

The effectiveness of the proposed auto-tuning approach-based architecture is validated through experiments that have been carried out using a testbench.

The block diagram and the photo of the testbench are shown in Figure 5.4 and Figure 5.5, respectively.

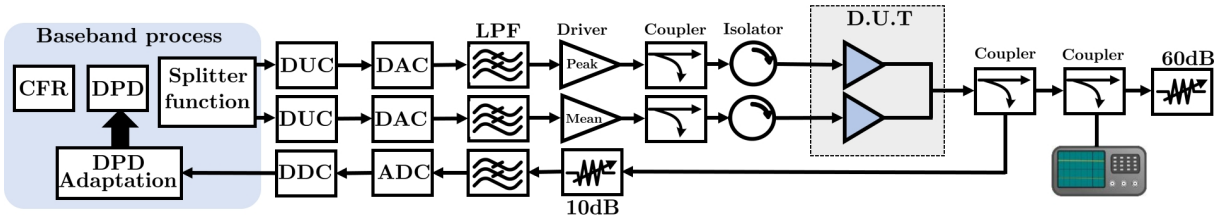


Figure 5.4: Block diagram of testbench of dual-input Doherty PA

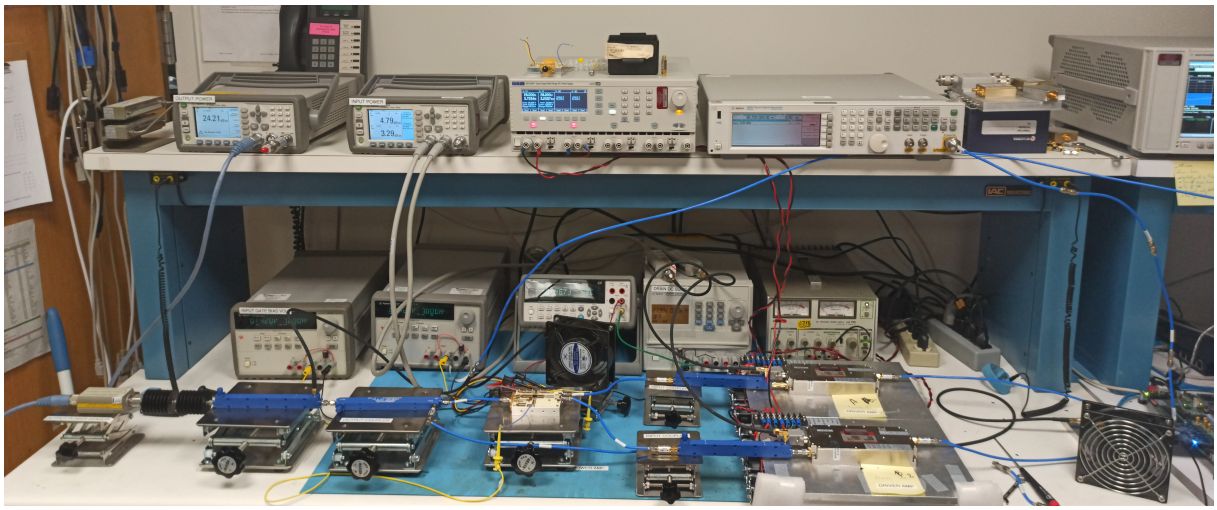


Figure 5.5: Photo of testbench of dual-input Doherty PA

The dual-input Doherty PA is controlled and evaluated using a MATLAB-based linearization and efficiency enhancement technique. It runs on Windows PC. The baseband IQ data are generated and split into two different IQ data inputs sent to the DUT through the AD9371 dual-channel RF transceiver, which is connected to Xilinx FPGA ZC700 through FMC connectors.

A reference clock source is injected into the AD9371 board using the MXG X-Series N5182A vector signal generator at 30.72 MHz with 5 dBm of power. The transceiver AD9371 up-converts the baseband signals to the carrier frequency f_c at 3 GHz using a 245.76 MHz sampling frequency. Two low-pass filters are used for each path (main and peaking) to remove unwanted frequencies.

Each RF signal stimulates the dual-input Doherty PA through a coupler ZUDC30-183 and driver AMP1132. The drivers are used to control DUT's input power level, while the two couplers are used to measure the RF input powers: the peaking power P_p and main power P_m. Two couplers are installed in cascade at the output of the DUT. The first coupler is used to attenuate the RF output signal for the observation path. A Keysight N9010A MXA spectrum analyzer is used to characterize the signal's spectrum at the output of DUT through the second coupler. The RF instruments and DC supplies used in this testbench are connected to the PC workstation by LAN over Ethernet.

During the testbench setup, a calibration process of the RF instruments has been carried out to ensure that the displayed measurements are traceable and reliable.

For the observation path, The RF output signal is down-converted to the baseband by AD9371, which provides the baseband signal to the PC workstation. Around 100000 IQ samples were recorded for the baseband process with a sampling rate of 245.76 MSPS.

After the reception of the IQ data through the observation Rx path, several synchronization processes are applied using Matlab to prepare the baseband input-output signals from the PA for the processing. This acquisition process is organized as follows:

- Synchronization: The input-output baseband signals are normalized and synchronized in the time domain using a cross-correlation operation.
- f_c correction: This process removes the frequency offset between the transmitter and the receiver.
- Fractional time-shift: This method is proposed in [106], which is based on a B-spline filter that produces a time-shift $\Delta \in [0, 1]$, which is an arbitrary fraction of the sampling period.
- IQ imbalance correction: Due to the distortion of the down-converter at the receiver side, which introduces an image interference, that may have an error floor limiting the demodulation performance. Moreover, the IQ imbalance introduced by the local oscillator may be assumed constant over the signal bandwidth and can be characterized by the phase mismatch that can occur between the I and Q components, which becomes not precisely 90 degrees, and an amplitude mismatch due to the difference in gain of the I and Q branch mixers. In this work, a circularity-based blind compensation algorithm proposed in [107] is used as an IQ imbalance compensator.

NMSE between the baseband input and output of DUT, before and after the application of this process, is used as an indicator to assess the effectiveness of these acquisition processes. Once NMSE has improved significantly, the baseband input-output signals are then ready to be trained for the identification algorithm.

5.3 Free-Parameters

To meet the objectives of optimizing the dual-input Doherty PA, which consists of improving efficiency while maintaining a better linearity level, we propose an architecture described in Figure 5.6.

In this work, the DPD is used to linearize the DUT by compensating for the nonlinearities of dual-input Doherty PA. Additionally, CFR is used to reduce the PAPR of the transmitted signal so that dual-input Doherty PA can operate with less BO. Both CFR and DPD are implemented in baseband.

On the other hand, the dual-input Doherty PA requires two separate input signals from the PC workstation. Therefore, the baseband signal to be sent to the PA should

be divided into two input signals, which are different in amplitude and phase, using a splitting function, so-called digital splitter, designed in baseband.

Each block has parameters to be set or controlled, which requires a design of a control engine based on an optimization approach that optimally determines these parameters to ensure an operating point of DUT exhibiting a better trade-off between efficiency and linearity.

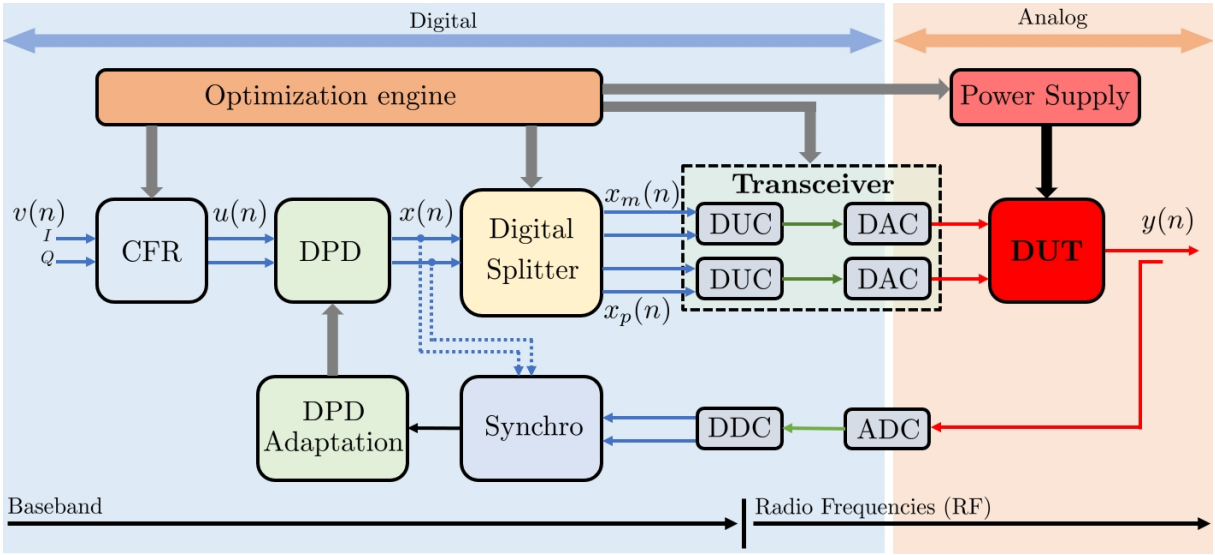


Figure 5.6: Block diagram of the proposed architecture

In this architecture, the DPD block will not be controlled by this control engine since the DPD technique requires linear regression techniques such as the LS method to identify the model coefficients.

The baseband notation of the input and output block is defined as follows. The input signal is denoted by $v(n)$, which is the CFR input. The input of DPD is denoted by $u(n)$, which is the CFR output. $x(n)$ is the output of DPD, which presents the input of the digital splitter. The splitter outputs are denoted by $x_m(n)$ and $x_p(n)$, which are DUT's inputs. $y(n)$ is the baseband output of DUT after the acquisition process.

5.3.1 Free-Parameter of CFR

In this work, the CFR technique used to reduce the PAPR is based on peak cancellation [108] [109].

The peak cancellation is carried out through two operations: clipping and filtering. Its principle is illustrated in Figure 5.7.

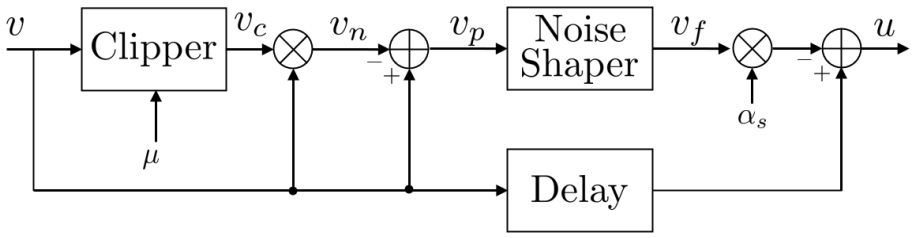


Figure 5.7: Block diagram of peak cancellation technique

A clipped signal $v_n(n)$ is first created by clipping the input $v(n)$ when it exceeds a specified threshold μ. This clipped signal is subtracted from the original input $v(n)$ to

generate a peak signal $v_p(n)$, with

$$v_p(n) = \begin{cases} v(n) - v(n)\frac{\mu}{|v(n)|} & \text{if } |v(n)| > \mu \\ 0 & \text{if } |v(n)| \leq \mu \end{cases} \tag{5.1}$$

The peak signal $v_p(n)$ is then filtered using a noise shaping filter and the final signal $u(n)$ is therefore calculated by subtracting a time-aligned weighted version of $v_f(n)$ from $v(n)$.

In this block, a clipping threshold μ is considered as a free-parameter of CFR.

5.3.2 Free-Parameters of Digital Splitter

The motivation behind using two separate RF inputs is to eliminate analog input splitters, such as the Wilkinson divider, and to allow independent power control to the main and peaking amplifier.

As mentioned before, the baseband signal $x(n)$, which is a complex signal represented by an amplitude X and a phase θ, i.e. $x = Xe^{j\theta}$, is divided into two complex signals x_m and x_p defined as:

$$\begin{aligned} x_m &= \alpha_m X e^{j(\theta)} = \alpha_m x \\ x_p &= \alpha_p X e^{j(\theta-\phi)} = \alpha_p e^{-j\phi} x \end{aligned} \tag{5.2}$$

The three parameters α_m, α_p, and ϕ in (5.2) have to be carefully tuned to ensure the high efficiency of the PA.

We suggest using only two parameters to design the digital splitter: the power ratio between the amplitude of the two inputs, denoted by α, instead of α_m and α_p, and the phase shift ϕ between their phases.

The relationship between α and (α_m,α_p) is defined as follows. The power of x_m and x_p are defined as:

$$\begin{aligned} P_m &= \alpha_m^2 P_{in} \\ P_p &= \alpha_p^2 P_{in} \end{aligned} \tag{5.3}$$

with

$$P_{in} = P_m + P_p \tag{5.4}$$

where P_{in} is the input powers of the baseband signals x.

It gives

$$\alpha_m^2 + \alpha_p^2 = 1 \tag{5.5}$$

So, if we define α as:

$$\alpha_m^2 = \alpha \tag{5.6}$$

We have

$$\alpha_p^2 = 1 - \alpha_m^2 = 1 - \alpha \tag{5.7}$$

Hence, the baseband parameters α_m and α_p can be expressed in terms of α by:

$$\alpha_m = \sqrt{\alpha} \ ; \ \alpha_p = \sqrt{1-\alpha} \tag{5.8}$$

The phase shift ϕ between the main and peaking branches is critical. It affects the PA efficiency, the amplification gain, and the linearity. It can also independently be controlled at different frequencies f_c.

We propose to assign the original phase θ of x to x_m and add a phase shift ϕ to the peaking signal.

$$\begin{aligned} \theta_m &= \theta \\ \theta_p &= \theta - \phi \end{aligned} \tag{5.9}$$

From (5.8) and (5.9), the baseband input signals x_m and x_p can be expressed as:

$$\begin{aligned} x_m &= \sqrt{\alpha}x \\ x_p &= \sqrt{1-\alpha}e^{-j\phi}x \end{aligned} \tag{5.10}$$

The architecture of the digital splitter is described in Figure 5.8. The design of a digital splitter consists of optimizing the power ratio α and phase shift ϕ, which are taken as free parameters in the optimization engine.

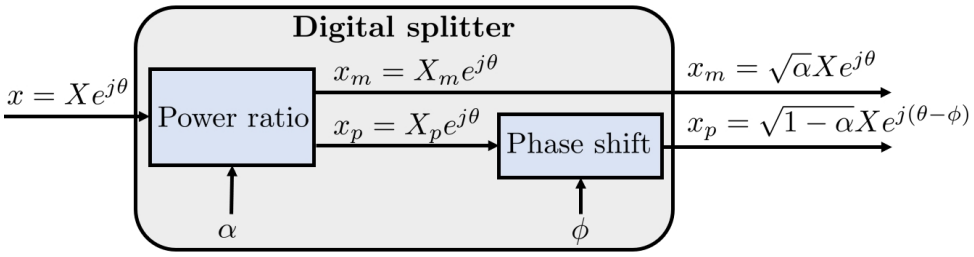


Figure 5.8: Block diagram of digital splitter

5.3.3 Free-Parameters of Transceiver

In the calibration process, it has been shown that two essential operations are needed to be established from the baseband: fixing the DAC resolution, which is integrated into the RF transceiver, and setting the gain attenuation, which controls the power level of the transmitted signal.

For DAC resolution, it is recommended to scale the IQ data to 2^{15} in the baseband to ensure high accuracy of data to be transmitted to minimize loss of information. The gain attenuation directly controls the power level of the signal in the Tx branch based on an internal step attenuator with a step of 0.05 dB.

The block diagram of the transceiver is depicted in Figure 5.9.

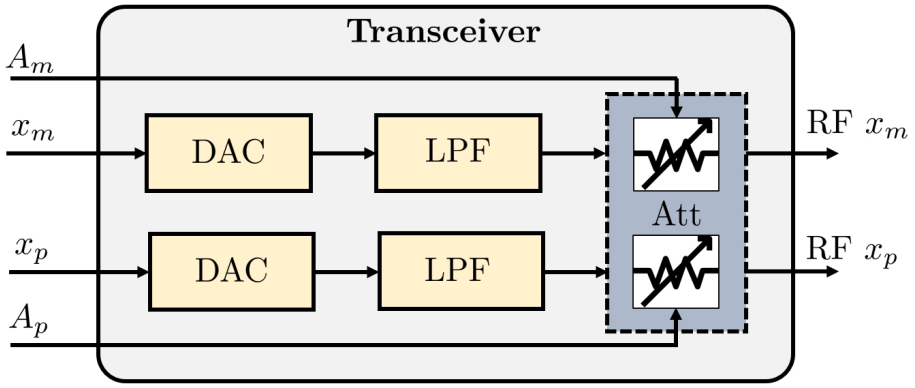


Figure 5.9: Block diagram of transceiver

In the transceiver block, we use two parameters A_m and A_p defined in the baseband to control the attenuation in the branch Tx1 occupied by the main amplifier and in Tx2 occupied by the peaking amplifier.

We have approximately estimated the relationship between A_m (and A_p) and the average power of the main (and peaking) RF signal as:

$$\begin{aligned} P_{m,dBm}(A_m) &= A_m + a\text{PAPR}(x_m) + b\sqrt{\alpha} + c \\ P_{p,dBm}(A_p) &= A_p + a\text{PAPR}(x_p) + b\sqrt{1-\alpha} + c \end{aligned} \tag{5.11}$$

where a, b, and c are parameters defined empirically from some preliminary tests and stored in LUT and indexed in terms of the center frequency f_c, and the signal bandwidth. Indeed, the process behind estimating a, b, and c consists of performing many tests with waveforms of different bandwidths, at different frequencies, and for different power levels. These test results provide a datasets of a, b, and c that will be used subsequently according to the parameters of the scenario at hands.

If A_m and A_p have the same baseband numerical value, and $x_m = x_p$ with $\alpha = 0.5$, the power $P_{m,dBm}$ and $P_{p,dBm}$ are assumed to be the same. However, in practice, we have observed that by assigning the same numerical value to A_m and A_p, the measured powers $P_{m,dBm}$ and $P_{p,dBm}$ through the power sensors are different. This difference between $P_{m,dBm}$ and $P_{p,dBm}$ can be adjusted and compensated in baseband by using a parameter denoted by ψ with

$$P_{m,dBm}(A_m) = P_{p,dBm}(A_p + \psi) \tag{5.12}$$

Finding ψ that satisfies (5.12) can be done in the calibration process. However, we propose to take ψ as a free parameter controlled from baseband, which could be viewed as a hardware parameter since it can adjust the input power distribution over the main and peaking amplifiers.

5.3.4 Free-Parameters of DUT

The main and peaking input of dual-input Doherty PA controls the main and peaking amplifier, biased with $V_{GS,m}$ and $V_{GS,p}$, respectively.

These biased voltages are controlled from baseband and defined within a range of DC voltage. The DC power supply used to manage the gate bias voltages is connected to the PC workstation through an Ethernet connection that enables real-time voltage monitoring from the baseband.

Therefore, $V_{GS,m}$ and $V_{GS,p}$ are taken as a free-parameters.

Since the output impedance matching network is not changing, and since we configure the input side of dual-input Doherty PA, not the output side, it is not necessary to change the drain bias V_{DC} while handling the dual-input Doherty PA.

The free parameters of the proposed architecture to be controlled are summarized in Table 5.1.

Table 5.1: Free-parameters of the proposed architecture

Bloc	Free-parameter	Symbol	Unit
CFR	Threshold of PAPR reduction	μ	dB
Digital splitter	Power ratio	α	%
	Phase shift	ϕ	Degree
Transceiver	Attenuation difference	ψ	dB
DUT	Main bias voltage	$V_{GS,m}$	Volt
	Peaking bias voltage	$V_{GS,p}$	Volt

5.4 Linearization and Efficiency Enhancement of Dual-Input Doherty Power Amplifier

Finding the optimal configuration of each block in Table 5.1 can be seen as an optimization problem to be solved.

The brute-force search can help find the optimal free parameters by exploring all possible combinations in the searching space. However, the brute-force search is not a practical solution to be implemented in real-time applications. Consequently, an auto-tuning approach based on an optimization algorithm is proposed to meet this need.

5.4.1 State-of-the-Art

In the literature, the design and study of dual-input Doherty PA with enhanced efficiency have been reported in many research works. Few of them deal with the joint optimization of PA parameters and their linearization technique.

The first work has been reported in [110], in which the authors have confirmed the interest of using a combination of splitting the input signal and vector-switched digital predistorter, in which the linearity is improved while maintaining high efficiency. The splitter's design has been achieved by performing several combinations of α and ϕ in a simulation environment, which does not accurately present the behavior of dual-input Doherty PA.

In [111], the authors presented a hardware implementation of a real-time DPD and digital splitter for multi-input PA. The static splitter was achieved by a nonlinear static LUT function, where the signals x_m and x_p at the output of the splitter are expressed as:

$$\begin{cases} x_m(n) &= x(n) \times \mathrm{LUT}_1(|x(n)|) \\ x_p(n) &= x(n) \times \mathrm{LUT}_2(|x(n)|) \end{cases} \tag{5.13}$$

where LUT_1 and LUT_2 are nonlinear complex function, which are created by performing an exhaustive search. However, the voltages $V_{GS,m}$ and $V_{GS,p}$ of the Doherty PA used in measurement tests are biased identically, which raises concerns about its efficiency.

The first work related to the online learning-based optimization of dual-input Doherty PA is proposed in [112]. The authors proposed an adaptive technique based on a simultaneously perturbed stochastic approximation (SPSA) algorithm to tune the free parameters α, ϕ, $V_{GS,m}$ and $V_{GS,p}$. This algorithm has been performed according to a cost function J defined as:

$$J(\Theta) = \omega \times G(\Theta) + (1 - \omega) \times \mathrm{PAE}(\Theta) \tag{5.14}$$

where $\Theta = [\alpha \ \phi \ V_{GS,m} \ V_{GS,p}]$ is the vector of free parameters to be optimized, ω is a weighting coefficient, and G is the gain of PA. In [112], the authors showed a significant improvement in G and PAE. However, the linearity requirement has not been met since SPSA has focused only on efficiency enhancement.

In [113], a machine learning-based optimization is proposed to optimize the free-parameters used in [112]. Practically, [113] is an extension of the work reported in [112] where the cost function is updated from its previous version, which includes, in addition to PAE and gain, the output power P_{out} and ACPR as metric referring to the linearity.

The optimization process used in [113] consists of two phases:

- Phase I: A simulated annealing-based global optimization algorithm runs until the cost function achieves its optimal value or stops after a given number of iterations.
- Phase II: The aim is to fine-tune the optimized free parameters using an approach of learning-based control based on the extremum-seeking algorithm [114].

This auto-tuning process was validated in terms of the free-parameters convergence towards optimal values, which corresponds to a maximization of the cost function according to its design. It is important to emphasize that no DPD or CFR process was introduced in [113].

On the other hand, an adaptive signal separation is proposed in [115]. The splitting function is constructed using a dictionary (LUT) derived from the static measured results of the dual-input Doherty PA. The results in [115] once again confirmed the interest in optimizing the separation of the input signals to be transmitted to the dual-input Doherty PA, which can improve efficiency.

In another work reported in [116], the authors only focused on optimizing the phase shift between the two RF inputs using an exhaustive search for dual-input load modulated balanced PA. The phase shift is determined according to NMSE and ACPR by sweeping the phase over a determined interval [0° 350°] with a resolution of 10°, which makes it a simple solution to be implemented but with difficulty in knowing whether the phase shift is optimal or not, since the PA performances are very sensitive to the phase shift between its inputs. Besides, sweeping the phase over a large interval can be critical, especially at intervals where there will be no output power, leading to heat dissipation in the device, which can damage it.

5.4.2 Proposed Auto-Tuning Approach

5.4.2.1 General Optimization Algorithms

In this work, the proposed auto-tuning approach to optimize the free-parameters is based on an efficient hybrid heuristic search control (HHSC) based on two types of model-free optimization methods: simulated annealing (SA) as a global optimization search and extremum-seeking control (ESC) as an adaptive control to fine-tune the optimized results.

In this study, the choice of SA and ESC is in line with the state-of-the-art elaborated on the dual-input Doherty PA manufactured by MERL and mounted on their testbench. The first work related to this axis of research is reported in [112] and [113], against which the main contribution of our study introduces DPD linearization and CFR technique by including the clipping threshold μ in the optimization process as a free-parameter.

The vector of free parameters to be optimized is denoted by Θ. The cost function corresponding to Θ is denoted by $J(\Theta)$ or J for simplicity.

5.4.2.1.1 Simulated Annealing

One of the best-known heuristic search methods for addressing the complex black-box global optimization problems is the SA algorithm proposed in [117].

Physical annealing in the metallurgy domain inspires the principle of the SA algorithm. Physical annealing is the process of heating a material until it reaches an annealing temperature. Then it will be cooled down slowly to increase the size of its crystals and reduce their defects. When the material is hot, the molecular structure is weaker and is more likely to change. When the material cools down, the molecular structure is more rigid and is less responsive to change.

Following the analogy with metallurgy, the slow cooling in simulated annealing depends on the slight decrease in the probability of accepting a worse solution as the solution space is explored. The algorithm should perform an extensive search to find the global optimum solution, so accepting worse solutions is fundamental.

SA algorithm is an almost straightforward stochastic search based on the Metropolis Monte Carlo method [118], the concept of which is to accepts not only the solutions that

improve J, but also some solutions that worsen it with a probability p known as the Metropolis criterion and defined as:

$$p(\Delta E) = e^{-\frac{\Delta E}{k_{bolt}\mathcal{T}}} \tag{5.15}$$

where ΔE is the change in cost function, k_{bolt} is Boltzmann's constant, and $\mathcal{T}$ is the control parameter analogous to the temperature of the annealing process.

During the search, the temperature is gradually decreased until reaching zero value in the perfect case.

SA algorithm is massively used in real-life applications to solve a global optimization problem, mainly when the problem model is not explicitly given and cannot be evaluated by computer simulation. Furthermore, the definition of the neighborhood is an essential aspect of the SA algorithm, particularly when a large neighborhood of SA leads to a greater probability of arriving at a global optimum.

In case of a global optimization problem, the standard procedure of SA is given by the following steps:

- Generate initial solution by choosing a random vector Θ_0 composed of the free-parameters to be optimized, and evaluate $J(\Theta_0)$.
- Initialize the temperature $\mathcal{T}$. If $\mathcal{T}$ is too high, the computation time is important. If $\mathcal{T}$ is too low, there is a risk of not sufficiently exploring the parameter space and remaining in a local minimum.
- Select a new solution in the neighborhood space Ψ of the current solution Θ_i where a solution $\Theta_i^{'}$ is accepted as a new solution depending on $\mathcal{T}$. $J(\Theta_i)$ and $J(\Theta_i^{'})$ are evaluated and compared. If $J(\Theta_i^{'})$ is better (maximized or minimized depends on the design of J) than $J(\Theta_i)$, then $\Theta_i^{'}$ is selected as a best solution at the current iteration. Otherwise, $\Theta_i^{'}$ can also be accepted depending on the new probability in (5.15). The cost function J will be defined in the next section.
- Decrease the temperature $\mathcal{T}$ by a temperature reduction rate $\mathcal{C}$ throughout the process search by $\mathcal{T} = \mathcal{C} \times \mathcal{T}$.
- Repeat the process while $\mathcal{T} > \mathcal{T}_f$.

The free-parameters optimized using SA are denoted by $\Theta_{\text{opt,SA}}$. The cost function corresponding to $\Theta_{\text{opt,SA}}$ is denoted by $J_{\text{opt,SA}}$.

The algorithm of SA is described in Algorithm (12).

Algorithm 12: Algorithm of SA

Initialization: $\mathcal{T}_0$, $\mathcal{T}_f$, $\mathcal{C}$, k_{bolt}, $iter = 0$
Define number of SA iterations SA_{max}
Random initial solution Θ_0
$\Theta = \Theta_0$
$\Theta_{\text{opt,SA}} = \Theta$
Evaluate $J(\Theta)$
$\mathcal{T} = \mathcal{T}_0$
while $\mathcal{T} > \mathcal{T}_f$ **do**
  while $iter < SA_{max}$ **do**
    $iter = iter + 1$
    Select a random neighbor $\Theta' \in \Psi(\Theta)$
    $\Delta E = J(\Theta') - J(\Theta)$
    if $\Delta E > 0$ **then**
      $\Theta = \Theta'$
      if $J(\Theta')$ *is better than* $J(\Theta_{opt,SA})$ **then**
        $\Theta_{\text{opt,SA}} = \Theta'$
        $J_{\text{opt,SA}} = J(\Theta')$
      end
    else
      Generate uniformly distributed random $r \in [0\ 1]$
      if $r < e^{\frac{-\Delta E}{k_{bolt}\mathcal{T}}}$ **then**
        $\Theta = \Theta'$
      end
    end
  end
  $\mathcal{T} = \mathcal{C} \times \mathcal{T}$
  $iter = 0$
end
Return $\Theta_{\text{opt,SA}}$

5.4.2.1.2 Extremum-Seeking Control

Once SA algorithm has reached $J_{\text{opt,SA}}$, the optimization procedure switches to ESC to fine-tune $\Theta_{\text{opt,SA}}$.

Generally, the adaptive control process is used to regulate linear and nonlinear systems [119]. However, in some applications, a control process may be used to optimize an objective function of unknown parameters or keep its cost function at its extremum, as is shown in [113]. ESC can solve optimization problems and optimize free parameters according to an unknown or uncertain cost function.

ESC is an online model-free optimization method that relies on feedback from output measurements. It is used to determine and track the optimal operating point of a given system when no model information is available.

The idea of ESC dates back to the first decades of the previous century [120] [121]. Various ESC techniques and applications have been developed and reported in the literature in the mid-twentieth century. The stability of ESC was rigorously confirmed in [122] where the authors proved the local stability of a near-optimal solution for a broad set of dynamical systems. The behavior of ESC is practically oriented by the local gradient of the equilibrium system. A feedback loop is then applied to drive the system to an operating point where the gradient is zero.

One of the most popular ESC methods is the perturbation-based ESC, which is proposed in [123]. This method efficiently probes the system using a sinusoidal perturbation (or a dither signal) to obtain a gradient estimate.

The concept of perturbation-based ESC is fundamentally depicted by the block diagram in Figure 5.10, where it consists of a target system, the output of which is the cost function J, a perturbation signal $a\sin(\omega t)$, a gain $\mathcal{K}$, and an integrator.

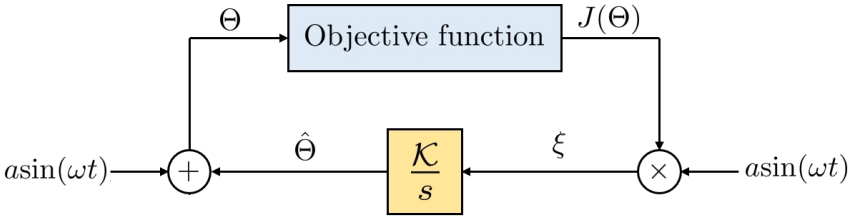


Figure 5.10: ESC scheme

According to Figure 5.10, the system is perturbed by a dither signal $a\sin(\omega t)$, which results in an output of the cost function $J(\Theta)$. This output is then multiplied by the dither signal and denoted by ξ. After multiplying ξ by a gain $\mathcal{K}$ and passed through an integrator $\frac{1}{s}$, the resulting signal $\hat{\Theta}$ estimates the gradient of the cost function with respect to Θ.

The loop of ESC can be written as the following dynamical system

$$\begin{aligned} \xi &= J(\Theta) \times a\sin(\omega t) \\ \frac{d\hat{\Theta}}{dt} &= K \times \xi \\ \Theta &= \hat{\Theta} + a\sin(\omega t) \end{aligned} \tag{5.16}$$

The integrator's role is to optimally adjust the operating point towards the optimum according to the gradient information of the output. The choice of the gain $\mathcal{K}$ may influence the whole control system. Besides, it decides the controller bandwidth, which indicates the speed of the controller. The bandwidth of the controller should generally be less than that of the gradient estimator. Otherwise, the controller cannot get the updated gradient information, which makes the system unstable.

In our approach, the perturbation-based ESC is used as an off-line process, which is placed downstream of SA to fine-tune $\Theta_{\text{opt,SA}}$, The input of ESC is $\Theta_{\text{opt,SA}}$. The optimized configuration by ESC is denoted by $\Theta_{\text{opt,HHSC}}$. The cost function corresponding to $\Theta_{\text{opt,HHSC}}$ is denoted by $J_{\text{opt,HHSC}}$.

In this study, the values of the ESC parameters ω, $\mathcal{K}$, and a are initialized by the study reported in [113].

5.4.2.1.3 Algorithm of HHSC

HHSC is the core of the proposed auto-tuning approach to optimize the free parameters of the dual-input Doherty PA. The principle of HHSC consists of combining SA and ESC in cascade, as illustrated in Figure 5.11.

The interest in using two algorithms of different types reinforces the convergence of the auto-tuning approach towards the optimal solution.

Starting from an initial solution Θ_0, SA optimizes the free parameters according to a designed cost function J. The optimized solution $\Theta_{\text{opt,SA}}$ returned by SA will be the initial solution for the ESC process.

The algorithm of HHSC is described in Algorithm (13).

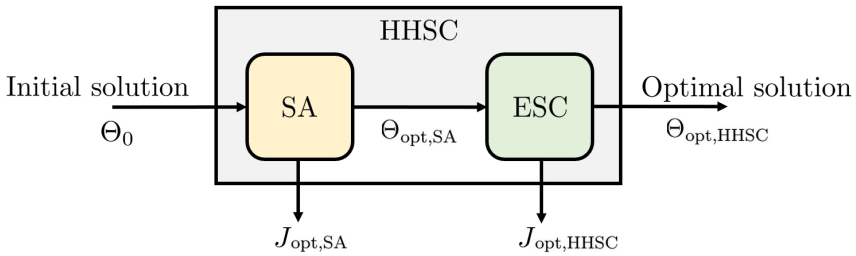


Figure 5.11: Block diagram of HHSC

Algorithm 13: Algorithm of HHSC

Initialization: $\mathcal{T}_0$, $\mathcal{T}_f$, $iter = 0$, $\mathcal{K}$, a, ω
Run Algorithm of SA described in (12)
$\Theta_{0,\text{ESC}} = \Theta_{\text{opt,SA}}$
$\Theta = \Theta_{0,\text{ESC}}$
$J_{old} = J_{\text{opt,SA}}$
while *1* **do**
  $\xi = J(\Theta) \times a\sin(\omega t)$
  $\hat{\Theta} = K \times \int \xi$
  $\Theta = \hat{\Theta} + a\sin(\omega t)$
  Evaluate $J(\Theta)$
  if $J(\Theta) > J_{old}$ **then**
    $J_{old} = J(\Theta)$
  else
    $\Theta_{\text{opt,HHSC}} = \Theta$
    $J_{\text{opt,HHSC}} = J(\Theta_{\text{opt,HHSC}})$
    end while loop
  end
end
Return $\Theta_{\text{opt,HHSC}}$

5.4.3 Principle of the Proposed Auto-Tuning Approach

The principle of the auto-tuning approach proposed in this work to optimize and linearize the dual-input Doherty PA is summarized in the flowchart in Figure 5.12.

The process of the proposed approach is mainly composed of five sub-processes:

- Design of the cost function to control the convergence of HHSC.
- Joint optimization of CFR, digital splitter, transceiver, and DUT by HHSC using the cost function designed.
- DPD linearization based on ILA.
- Update the cost function designed in the first sub-process.
- Optimal pruning of free parameters in HHSC.

Each sub-process of the flowchart in Figure 5.12 will be detailed in the following sections.

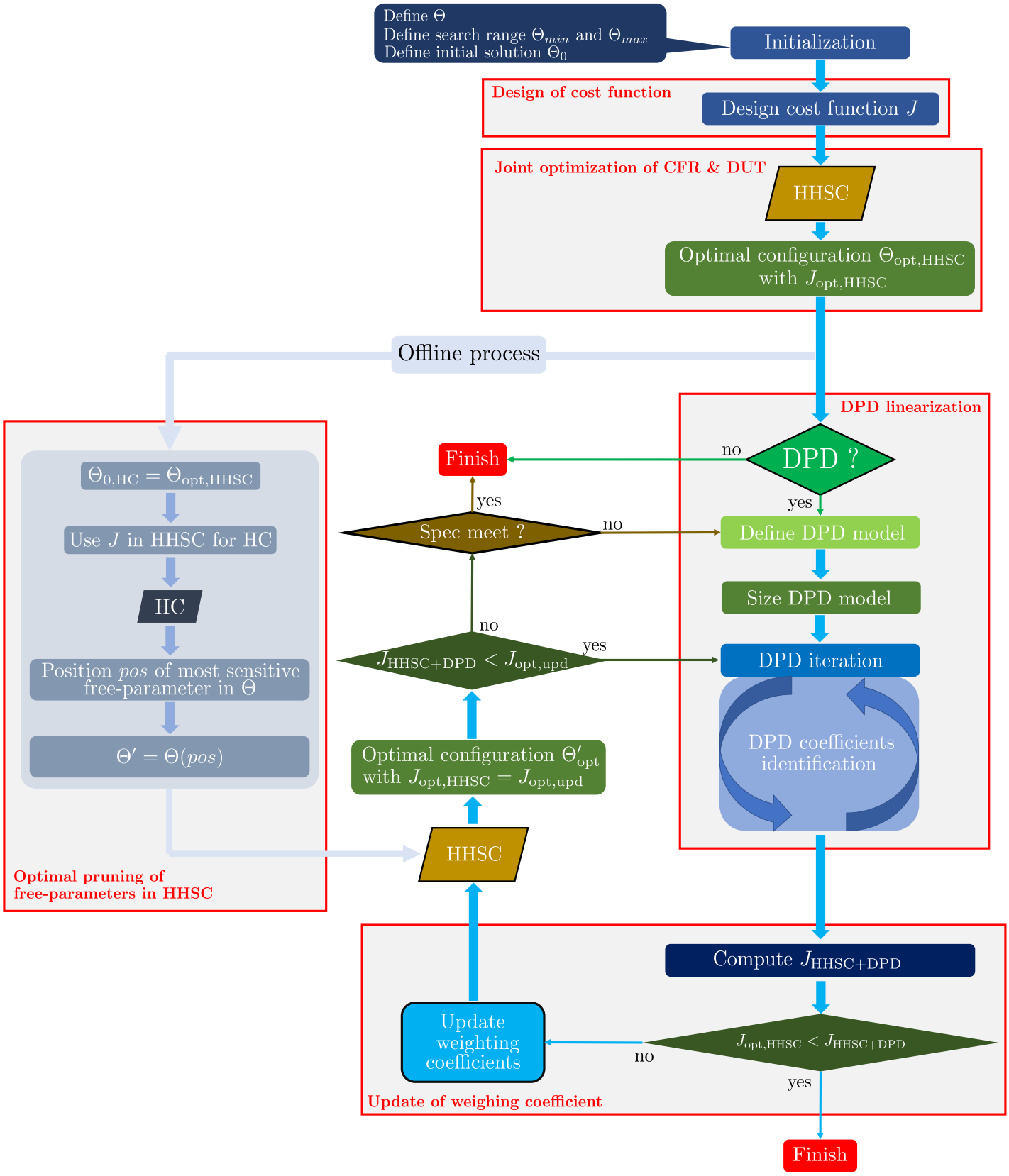


Figure 5.12: Flowchart of the proposed auto-tuning approach

5.4.3.1 Design of Cost Function

The cost function is an important aspect that defines the optimality criterion.

In this study, the cost function will be designed to ensure a good trade-off between linearity and efficiency. The linearity requirement is represented in terms of two figures of merit (FOMs): EVM and ACPR, while the efficiency and output power level requirement are represented by the power added efficiency PAE and the output power P_{out}.

Note that in this study, P_{out} represents the peak power of the signal at the output of DUT $y(n)$, which is given by:

$$P_{out} = P_{avg} + \text{PAPR}(y) \tag{5.17}$$

where P_{avg} is the average power of the signal, given by the power sensor from testbench.

These FOMs are accordingly weighted to their importance in the cost function. Additionally, some FOM thresholds can also be defined to penalize further not meeting the targeted specifications.

The optimization problem is a multi-objective optimization (MOO) since more than one objective function are optimized simultaneously.

The MOO applications have been reported in the literature for many fields such as engineering, economics, finance, etc. The various types of MOO problems are tracked by several methods as well [124]. The MOO consists of finding the optimal solution of more than one desired target. Making decisions in MOO is a significant problem that practically leads to a proper trade-off on some contradictory issues.

The MOO problem can be written as follows:

$$\begin{aligned} \text{min or max} \quad & f_1(x), f_2(x), \ldots, f_n(x) \\ \text{subject to} \quad & x \in \mathcal{U} \end{aligned} \tag{5.18}$$

where x is the optimal solution, n is the number of objective functions, $\mathcal{U}$ is variable space which embodies all different solution and $f_n(x)$ is the n^{th} objective function.

Generally, solving the problems of MOO can be classified into two methods: the Pareto method and the scalarization method.

The Pareto method is based on separating the optimal solution of each objective function during the optimization process. Dominance is an essential concept of the Pareto method based on differentiating the dominated and non-dominated solutions. For the MOO problem, a solution is reached when one objective function cannot increase without impacting the others. This condition is called Pareto optimality or Pareto efficiency.

The scalarization method allows converting MOO problem into single-objective optimization (SOO) problem such that optimal solutions to SOO problem are Pareto optimal solutions to MOO problem [125]. In [126], the scalarization method incorporates multi-objective functions into scalar fitness function as in the following equation:

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x) \tag{5.19}$$

where $w_1, w_2, \ldots, w_n$ are the parameters of the scalarization called weighting coefficients of the cost function F.

A weighting coefficient w_i is the weight of its objective function f_i, which refers to the relative importance of that objective function to the cost function F. This method is known as the weighted sum method. This method's challenge is to correctly set each objective function's weight to obtain the Pareto optimal solution in the desired region in the objective space $\mathcal{U}$.

Another famous method proposed in [127] is called the ϵ-constraint method, which is based on keeping just one of the objectives and restricting the others within user-specific values. It can be defined as:

$$\begin{aligned} \text{min or max} \quad & f_j(x) \\ \text{subject to} \quad & x \in \mathcal{U} \\ & f_i(x) \leq \epsilon_i \text{ for } i \in \{1, \ldots, k\} \text{ and } i \neq j \end{aligned} \tag{5.20}$$

In this work, we propose to design the cost function J for HHSC based on the weighted sum method but with constraints. It is defined as:

$$J = w_1 J_{\text{EVM}} + w_2 J_{\text{ACPR}} + w_3 J_{\text{PAE}} + w_4 J_{\text{P}_{out}} \tag{5.21}$$

with

$$\begin{cases} J_{\text{EVM}} = w_{1,1} \left| \frac{\text{EVM}_t}{\text{EVM}_{\text{C}}} \right| + w_{1,2} \left| \frac{\text{EVM}_t}{\text{EVM}_{\text{DD}}} \right| + w_{1,3} \left| \frac{\text{EVM}_t}{\text{EVM}_{\text{CDD}}} \right| \\ J_{\text{ACPR}} = \left| \frac{\text{ACPR}}{\text{ACPR}_t} \right| \\ J_{\text{PAE}} = \left| \frac{\text{PAE}}{\text{PAE}_t} \right| \\ J_{\text{P}_{out}} = \left| \frac{\text{P}_{out}}{\text{P}_{out,t}} \right| \end{cases} \tag{5.22}$$

where EVM_t, ACPR_t, PAE_t, and $\text{P}_{out,t}$ are EVM target, ACPR target, output power target, and efficiency target, respectively, that the user attempts to reach.

Since CFR as a nonlinear process deteriorates EVM dramatically, we propose to use in J_{EVM}, EVM of CFR denoted by EVM_{C}, EVM of DPD and DUT denoted by EVM_{DD}, and EVM of the whole system including CFR, DPD, and DUT denoted by EVM_{CDD}.

There is a way to present the three EVMs in one feature, denoted by EVM_{ms} by using the mean square of EVM_{CDD}, EVM_{C}, and EVM_{DD}, which can be defined as:

$$\text{EVM}_{ms} = \sqrt{\frac{\text{EVM}_{\text{C}}^2 + \text{EVM}_{\text{DD}}^2 + \text{EVM}_{\text{CDD}}^2}{3}} \tag{5.23}$$

However, in this work, EVM_{CDD}, EVM_{C}, and EVM_{DD} will be presented separately.

The constraints of the cost function designed in (5.21) are defined as:

$$\begin{cases} w_1 = w_{1,1} + w_{1,2} + w_{1,3} \\ \sum_{i=1}^{4} w_i = 1 \\ \max J_{\text{EVM}} = \max J_{\text{ACPR}} = \max J_{\text{PAE}} = \max J_{\text{P}_{out}} = 1 \end{cases} \tag{5.24}$$

The computation of ACPR is defined in terms of ACPR_{L1} and ACPR_{U1} as:

$$\text{ACPR} = 10 \log_{10} \left(\frac{10^{\frac{\text{ACPR}_{L1}}{10}} + 10^{\frac{\text{ACPR}_{U1}}{10}}}{2} \right) \tag{5.25}$$

In (5.21), each objective function corresponding to each FOM is normalized by its target value, which is defined as a user specification.

The cost function design is carried out such that HHSC attempts to maximize J to 1, indicating that the user's specifications are met. When J=1, all FOMs have reached their targets, which presents the perfect case, i.e.,

$$J = w_1 + w_2 + w_3 + w_4 = 1$$

In this way, we give the cost function a logical interpretation since the different FOMs do not have the same meaning or same unit.

5.4.3.2 Joint Optimization of CFR and DUT

In the second block of our approach, we propose to optimize CFR, digital splitter, transceiver, and DUT jointly with HHSC by using the cost function J designed in the previous section. In the following, we use the acronym DUT to refer to the digital splitter, the transceiver, and DUT together. The DPD is not integrated at this level.

The free-parameters to be optimized are defined as:

$$\Theta = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}] \tag{5.26}$$

Their searching spaces are summarized in Table 5.2.

Table 5.2: Free-parameters and their searching range

Block	Free-parameter	Symbol	Searching range	Unit
CFR	Threshold of PAPR reduction	μ	$[0\ \mu_{max}]$	dB
Digital splitter	Power ratio	α	[0 1]	-
	Phase shift	ϕ	$[\phi_{min}\ \phi_{max}]$	Degree
Transceiver	Attenuation difference	ψ	$[\psi_{min}\ \psi_{max}]$	dBm
DUT	Main bias voltage	$V_{GS,m}$	$[V_{GS,min}\ V_{GS,max}]$	Volt
	Peaking bias voltage	$V_{GS,p}$	$[V_{GS,min}\ V_{GS,max}]$	Volt

The boundaries Θ_{min} and Θ_{max} are defined as well. The interval $[\Theta_{min}\ \Theta_{max}]$ presents the searching range of each free parameter.

Some preliminary tests, or information about the system, especially DUT from previous works, are necessary to determine the optimization interval range and avoid some critical searching range, especially for the phase.

HHSC is then ready to optimize the vector Θ starting from a given initial solution $\Theta_0 = [\mu_0\ \alpha_0\ \phi_0\ \psi_0\ V_{GS,m_0}\ V_{GS,p_0}]$, which is arbitrarily defined.

The optimized configuration is $\Theta_{\text{opt,HHSC}}$ with its corresponding cost function $J_{\text{opt,HHSC}}$.

5.4.3.3 DPD Linearization

When HHSC optimizes the free-parameters, it will be decided whether to include DPD in the optimization process or not. If not included, the cost function will be re-designed only based on the efficiency requirement as long as the low linearity specifications can be targeted, assuming that a subsequent application of DPD will meet the system requirements. For the wideband signal scenarios, the linearity specifications will be more challenging to meet. Therefore, it is preferred to include DPD in the auto-tuning approach.

In this study, the DVR model in (3.3) has been chosen to serve as a DPD model as it demonstrates its ability to linearize strong nonlinear behavior with memory. The DVR model has been sized to find its optimal structure using the HC algorithm presented in Chapter 3 but with uniform segmentation.

The ILA is used to identify the DPD coefficients, which are iteratively estimated by finding the LS solution that minimizes the LS criterion, presented in (2.18). The DPD coefficients to be extracted are not taken as a free-parameters for this study. One of the most impressive properties of using DPD adaptively is that DPD models are linear with respect to their coefficients, allowing the use of a linear regression technique such as

LS. However, the integration of the estimation process of the DPD coefficients into the optimization process involves nonlinear estimation techniques, which will considerably increase the complexity of the search.

On the other hand, one of the critical issues that arise when using DPD is the normalization gain selection. In [128], a normalization gain G is proposed to eases the DPD identification, which is defined as:

$$G = \frac{max(|y(n)|)}{max(|x(n)|)} \tag{5.27}$$

where G is chosen as the quotient between the maximum envelope values of the PA input $x(n)$ and output $y(n)$.

In this work, we use the definition of G in (5.27) in ILA since this solution can simplify the DPD implementation because the original input and the predistorted signal can be normalized by the same scaling factor, which facilitates the power control. Figure 5.13 shows the integration of DPD using ILA to linearize the dual-input Doherty PA.

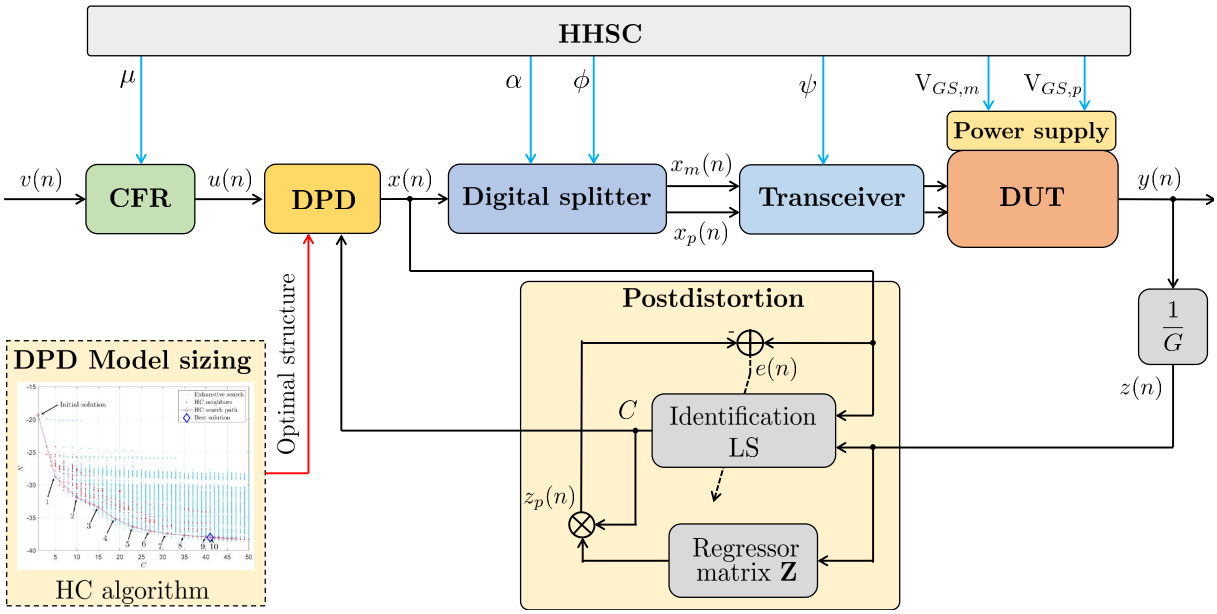


Figure 5.13: Integration of DPD using ILA to the entire system

5.4.3.4 Update of weighting Coefficient

The DPD may apply a back-off to the operating point of dual-input Doherty PA, which dramatically reduces the efficiency requirement. Therefore, the cost function J with DPD, which is denoted by $J_{\text{HHSC+DPD}}$, with updated FOMs and the same initial weighting coefficients will probably have deteriorated, and some free-parameters in $\Theta_{\text{opt,HHSC}}$ will no longer be optimal.

To maximize the cost function again, we propose to rely on the design of the cost function J by adapting its weighting coefficients $w = [w_1\ w_2\ w_3\ w_4]$ according to the change effected by the DPD.

An intuitive approach can be used by attributing an equal weight to each FOM according to the following equation:

$$w_i = \frac{1}{n} \tag{5.28}$$

where $i = 1, 2, \ldots, n$ and n is the number of objective functions that present FOMs.

In our context, the cost function is designed by combining efficiency and linearity, as is shown in (5.21). On the other hand, DPD with an efficient optimal DPD model can significantly improve linearity, which leads to reducing the weight of linearity FOMs in (5.21). Hence, we propose to design an adaptive cost function, in which the weight coefficients w are adaptive according to the improvement of linearity and efficiency over each block from the flowchart in 5.12. The weighting coefficients w are updated with respect to how much DPD improves linearity FOMs, e.g., EVM and ACPR, compared to before applying it.

Starting from initial weighting coefficients w, we apply HHSC to optimize the free-parameters Θ with $J_{\text{opt,HHSC}}$, then DPD to linearize the DUT under the optimized free-parameters.

It is required to re-compute the cost function $J_{\text{HHSC+DPD}}$ once the DPD is performed, and compare it to $J_{\text{opt,HHSC}}$. if $J_{\text{HHSC+DPD}} < J_{\text{opt,HHSC}}$, we propose to update the weighting coefficients $w_{1,2}$, $w_{1,3}$, w_2, w_3, and w_4. $w_{1,1}$ is not concerned since it depends on the CFR operation.

In this process, we are only focusing on $w_{1,2}$, $w_{1,3}$, and w_2 that refers to linearity FOMs improved by DPD. We calculate the ratio n_i of the difference before and after DPD improvement for each linearity FOM. The ratio n_i is expressed as:

$$\begin{aligned}
\mathbf{n_1} &= \left| \frac{\text{EVM}_{\text{HHSC,DD}} - \text{EVM}_{\text{HHSC+DPD,DD}}}{\text{EVM}_t} \right| \\
\mathbf{n_2} &= \left| \frac{\text{EVM}_{\text{HHSC,CDD}} - \text{EVM}_{\text{HHSC+DPD,CDD}}}{\text{EVM}_t} \right| \\
\mathbf{n_3} &= \left| \frac{\text{ACPR}_{\text{HHSC}} - \text{ACPR}_{\text{HHSC+DPD}}}{\text{ACPR}_t} \right|
\end{aligned} \tag{5.29}$$

Next, we update the weighting coefficients of FOM linearity as:

$$\begin{aligned}
w_{1,2} &= w_{1,2} \times (1 - \frac{10}{n_1}) \\
w_{1,3} &= w_{1,3} \times (1 - \frac{10}{n_2}) \\
w_1 &= w_{1,1} + w_{1,2} + w_{1,3} \\
w_2 &= w_2 \times (1 - n_3)
\end{aligned} \tag{5.30}$$

Once $w_{1,2}$, $w_{1,3}$, and w_2 are updated, we propose to assign an equal weight between w_3 and w_4 according to:

$$\begin{aligned}
w_3 + w_4 &= 1 - w_{1,1} - w_{1,2} - w_{1,3} - w_2 \\
w_4 &= w_3
\end{aligned} \tag{5.31}$$

The algorithm for updating the weighting coefficients w is described in Algorithm (14).

Algorithm 14: The update of weighting coefficient w

Initialize weighting coefficients $w = [w_{1,1}\ w_{1,2}\ w_{1,3}\ w_2\ w_3\ w_4]$
Apply HHSC
Get $J_{\text{opt,HHSC}}$
Measure EVM_{C}, EVM_{CD}, EVM_{CDD}, ACPR_{L1}, and ACPR_{U1} before DPD
Compute ACPR=f(ACPR_{L1},ACPR_{U1}) as in (5.25)
$\text{EVM}_{\text{HHSC,DD}}$=$\text{EVM}_{\text{DD}}$
$\text{EVM}_{\text{HHSC,CDD}}$=$\text{EVM}_{\text{CDD}}$
$\text{ACPR}_{\text{HHSC}}$=ACPR
Apply DPD using ILA
Measure EVM_{C}, EVM_{CD}, EVM_{CDD}, ACPR_{L1}, and ACPR_{U1} with DPD
Compute ACPR=f(ACPR_{L1},ACPR_{U1})
$\text{EVM}_{\text{HHSC+DPD,DD}}$=$\text{EVM}_{\text{DD}}$
$\text{EVM}_{\text{HHSC+DPD,CDD}}$=$\text{EVM}_{\text{CDD}}$
$\text{ACPR}_{\text{HHSC+DPD}}$=ACPR
Compute $J_{\text{HHSC+DPD}}$
if $J_{HHSC+DPD} < J_{opt,HHSC}$ **then**
  $\mathbf{n_1} = \left| \frac{\text{EVM}_{\text{HHSC,DD}} - \text{EVM}_{\text{HHSC+DPD,DD}}}{\text{EVM}_t} \right|$
  $\mathbf{n_2} = \left| \frac{\text{EVM}_{\text{HHSC,CDD}} - \text{EVM}_{\text{HHSC+DPD,CDD}}}{\text{EVM}_t} \right|$
  $\mathbf{n_3} = \left| \frac{\text{ACPR}_{\text{HHSC}} - \text{ACPR}_{\text{HHSC+DPD}}}{\text{ACPR}_t} \right|$
  $w_{1,2} = w_{1,2} \times (1 - \frac{10}{n_1})$
  $w_{1,3} = w_{1,3} \times (1 - \frac{10}{n_2})$
  $w_1 = w_{1,1} + w_{1,2} + w_{1,3}$
  $w_2 = w_2 \times (1 - n_3)$
  $w_3 = \frac{1 - w_1 - w_2}{2}$
  $w_4 = w_3$
else
  Finish
end

5.4.3.5 Optimal Pruning of free-parameters in HHSC

Once w is updated to the DPD contribution, the cost function's design J is changed.

According to the flowchart in 5.12, we run the HHSC again, but only on reduced free parameters in Θ. The HHSC will only be performed on one free parameter that is the most sensitive one in Θ and has the most significant impact on the behavior of J. The pruning process is an off-line procedure, as shown in the flowchart, which aims to reduce the complexity of the HHSC when the weighting coefficients are updated.

Pruning the free parameters in HHSC is optimally achieved using the HC algorithm. The motivation behind using the HC algorithm is that it is not a black-box optimization process. The neighborhood property in the HC algorithm makes it possible to follow the algorithm's evolution at each iteration.

Here, the cost function is used for the joint optimization of CFR and DUT in the second block from the flowchart.

The HC algorithm starts from a given initial element $\Theta_{1_{(0),\mathrm{HC}}}$ at the first iteration and continually moves in the direction of the element with the best cost function value among its neighbors.

In the following, we denote $\Theta_{q_{(i)},\mathrm{HC}}$ by $\Theta_{q_{(i)}}$, and $J_{q_{(i)},\mathrm{HC}}$ by $J_{q_{(i)}}$.

At the q^{th} iteration, the search procedure starts from $\Theta_{q_{(0)}}$ and test its neighbors $\Theta_{q_{(1)}}, \Theta_{q_{(2)}}, \ldots, \Theta_{q_{(M)}}$, where M is the number of neighbors of $\Theta_{q_{(0)}}$.

In this study, the neighborhood definition is inspired by the proposed one for the DVR model in Chapter 3. The vector Θ consists of 6 free-parameters: $[\mu\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}]$. As these free-parameters can have their values changed independently, they compose a 6-dimension space. The neighbor of element Θ is defined as an 6-tuple

$$\{\mu\times(1+\delta_\mu)\ ;\ \alpha\times(1+\delta_\alpha)\ ;\ \phi\times(1+\delta_\phi)\ ;\ \psi\times(1+\delta_\psi)\ ;\ V_{GS,m}\times(1+\delta_{V_{GS,m}})\ ;\ V_{GS,p}\times(1+\delta_{V_{GS,p}})\}$$

where $\delta_{\Theta(i)} \in [-1,1] \times \frac{\Theta(i)}{10}$ with $i = 1,..,6$.

The main property of this neighborhood definition is to apply the operation of $\delta_{\Theta(i)}$ to each free parameter $\Theta(i)$ individually.

According to this definition, the element $\Theta_{q_{(0)}} = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}]$ at the q^{th} iteration has 12 neighbors, which are:

$$\begin{cases}
\Theta_{q_{(1)}} = [\mu \times (1 + \frac{\mu}{10})\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(2)}} = [\mu \times (1 - \frac{\mu}{10})\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(3)}} = [\mu\ \alpha \times (1 + \frac{\alpha}{10})\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(4)}} = [\mu\ \alpha \times (1 - \frac{\alpha}{10})\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(5)}} = [\mu\ \alpha\ \phi \times (1 + \frac{\phi}{10})\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(6)}} = [\mu\ \alpha\ \phi \times (1 - \frac{\phi}{10})\ \psi\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(7)}} = [\mu\ \alpha\ \phi\ \psi \times (1 + \frac{\psi}{10})\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(8)}} = [\mu\ \alpha\ \phi\ \psi \times (1 - \frac{\psi}{10})\ V_{GS,m}\ V_{GS,p}] \\
\Theta_{q_{(9)}} = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m} \times (1 + \frac{V_{GS,m}}{10})\ V_{GS,p}] \\
\Theta_{q_{(10)}} = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m} \times (1 - \frac{V_{GS,m}}{10})\ V_{GS,p}] \\
\Theta_{q_{(11)}} = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p} \times (1 + \frac{V_{GS,p}}{10})] \\
\Theta_{q_{(12)}} = [\mu\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p} \times (1 - \frac{V_{GS,p}}{10})]
\end{cases}$$

The element $\Theta_{q_{(i)}}$ with the maximized cost function $J_{q_{(i)}}$ is the solution denoted by $\Theta_{q_{(s)}}$. With the neighborhood definition, the best solution $\Theta_{q_{(s)}}$ can be compared with the initial solution $\Theta_{q_{(0)}}$ since only one free-parameter is changed. An efficient way to do the comparison is to subtract $\Theta_{q_{(s)}}$ from $\Theta_{q_{(0)}}$, which make it easy to locate the position of the nonzero element in the vector $V = \Theta_{q_{(s)}} - \Theta_{q_{(s)}}$. The index of the nonzero element, denoted by idx, will then be stored in LUT, and the HC algorithm moves to the next iteration as long as $J_{q_{(s)}}$ is better than $J_{q-1_{(s)}}$. Otherwise, the HC algorithm stops.

The size of LUT is 6×2, where 6 refers to the number of free parameters in Θ (6 inputs). The cell corresponding to each free parameter, noted by *pos*, is incremented when the HC algorithm finds idx. Once the HC algorithm is finished, the free parameter to be used in HHSC, noted by Θ', is determined by the maximum incremented variable in the second column of LUT. If many free parameters have the same number of occurrences in LUT, the algorithm will take them as Θ'.

The optimal pruning of free parameters in HHSC is described in Algorithm 15.

Algorithm 15: Optimal pruning of free-parameters in HHSC

Define cost function J
Choose the initial element $\Theta_0 = \Theta_{\text{opt,HHSC}}$
Set $q = 1$
$\Theta_{q_{(0)},\text{HC}} = \Theta_{0,\text{HC}}$
$J_{q_{(0)}} = J_{0,\text{HC}}$
while *(1)* **do**
  Define number of neighbors M of $\Theta_{q_{(0)}}$
  for $i \leftarrow M$ **do**
    Determine the neighbor $\Theta_{q_{(i)}}$
    Evaluate $J_{q_{(i)}}$
  end
  $\Theta_{q_{(s)}} = \text{argmin}_{\Theta_{q_{(i)}}} \left(J_{q_{(i)}} \right)$
  if $J_{q-1_{(s)}} < J_{q_{(s)}}$ **then**
    $V = \Theta_{q_{(s)}} - \Theta_{q_{(0)}}$
    Find index idx of nonzero element in V
    $\text{LUT}(q) = idx$
    $q = q + 1$
    $\Theta_{q_{(0)}} = \Theta_{q-1_{(s)}}$
  else
    end while loop
  end
end
pos = Most frequent values in LUT
$\Theta' = \Theta(\max(pos))$
Apply HHSC to optimize $J(\Theta')$ using J with updated w

The cost function that corresponds to the optimized free-parameter Θ'_{opt}, denoted by $J_{\text{opt,upd}}$, will be compared to $J_{\text{HHSC+DPD}}$.

If $J_{\text{opt,upd}}$ is better than $J_{\text{HHSC+DPD}}$, HHSC has improved FOMs compared to those from the previous optimal configuration, and a DPD is required to linearize the DUT. Otherwise, the user must manually check whether the linearity-efficiency specifications are met or not.

5.5 Experimental Results

5.5.1 General Consideration

The proposed auto-tuning approach for dual-input Doherty PA is tested using a 64-QAM modulated 20 MHz bandwidth LTE signal with a roll-off factor of 0.6 at f_c=3 GHz with 7.5 dB of PAPR.

Following the flowchart in Figure 5.12, we define the search range of the free parameters in Table 5.1 by setting the upper and lower bounds for each free parameter. The search range is determined empirically according to some preliminary tests:

- Threshold of PAPR reduction μ: The CFR applies a nonlinear process by clipping the input signal $v(n)$ according to a clipping threshold μ, which causes an EVM_C degradation. Figure 5.14 shows the behavior of EVM_C and PAPR of the output

$u(n)$ of the CFR block according to the variation of μ. PAPR of $v(n)$ decreases with respect to μ. The EVM of $v(n)$ degrades exponentially with increasing μ. As we target an EVM_t around 3%, we define the search interval $[0\ \mu_{max}] = [0\ 1.2]$, for which when $\mu = 1.2$, the degradation of the EVM by CFR is almost 3%.

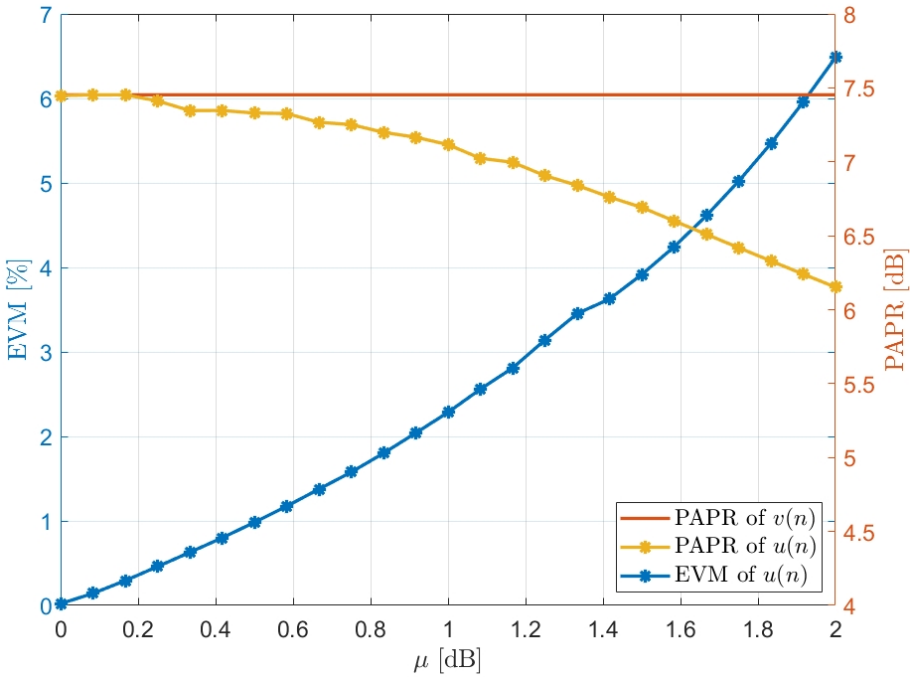


Figure 5.14: Behavior of EVM_C and PAPR of $v(n)$ versus μ

- Power ratio α: Since α presents the power distribution between the main and peaking branches, we set its search range to [0 1].
- Phase shift ϕ: The search range [-180° 180°] is the standard interval to be set. However, based on some preliminary tests, it was shown that there is some critical search range to be avoided. Therefore, [-10° 200°] is taken as a search range for the free parameter ϕ.
- Attenuation difference ψ: As ψ is categorized as a hardware free-parameter that refers to the physical power difference between the main and peaking amplifier, we set its search range to $[\psi_{min}\ \psi_{max}] = [-2\ 2]$ dB.
- Bias voltage V_{GS}: We set the bias voltage search range for both V_{GSm} and $V_{GS,p}$ to $[V_{GS,min}\ V_{GS,max}] = [-4\ \ -1]$ V in order to provide a flexible variation between deep-class C condition that should enhance efficiency and a near-class B bias where linearity should be improved. The drain bias V_{DC} is 28 V.

Before starting the auto-tuning approach, we propose to investigate the behavior of the free parameters over their search range. For this, we propose sweeping each of ϕ and α, which are taken as an example, over their search range, separately, while keeping the other free parameters on arbitrary values. This evaluation has been performed according to the linearity and efficiency FOMs used in the cost function in (5.21).

5.5.1.1 Sweeping ϕ

In this test, the phase shift ϕ is swept over its determined search range [-10° 200°] with a resolution of 2°, while keeping the other free-parameters to $\mu = 0$, $\alpha = 0.2$, $\psi = 0.1$, $V_{GS,m} = -1.5V$, $V_{GS,p} = -2.5V$.

Figure 5.15 presents the FOMs according to the variation of ϕ. As can be seen, FOMs exhibit important sensitivity to the variation of ϕ. All of them tend to have unimodal behavior, in which there is an optimal ϕ associated with each extremum of FOM. For the EVM extremum (here EVM_{DD}=EVM_{CDD} since $\mu = 0$), the optimal phase is located nearly to 80°, while for the other FOMs extremum (PAE, P_{out}, and ACPR), the optimal ϕ is located near to 140°. Here, ACPR is computed in terms of ACPR_{U1} and ACPR_{L1} as is described in (5.25).

This example again proves the benefit of optimizing the free parameters since the efficiency and linearity could be worsened or improved, depending on how the free parameters are configured.

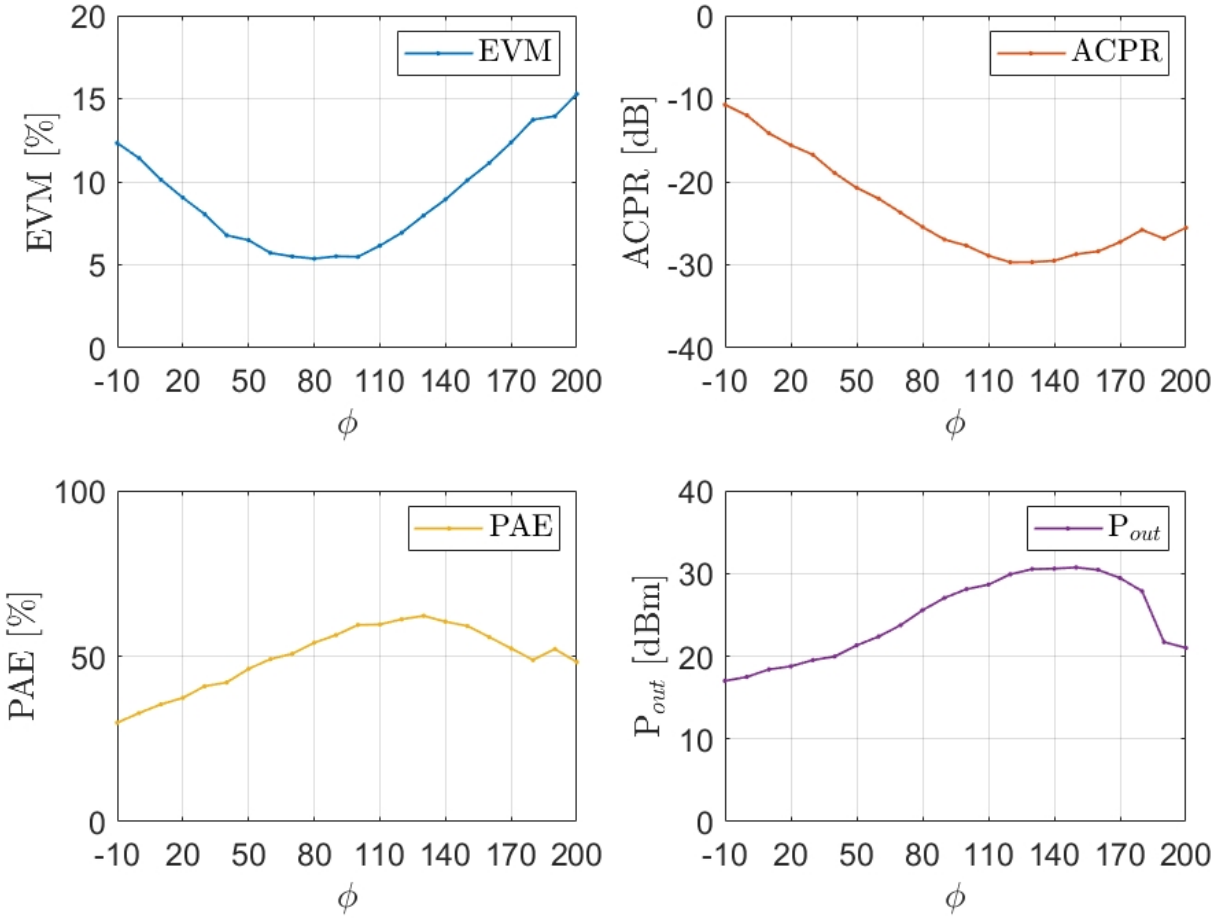


Figure 5.15: Evolution of FOMs when sweeping ϕ

5.5.1.2 Sweeping α

Sweeping α over its determined search range [0 1] has been performed with a resolution of 0.1, while keeping the other free-parameters to $\mu = 0$, $\phi = 110$, $\psi = 0.1$, $V_{GS,m} = -1.5V$, $V_{GS,p} = -2.5V$.

Figure 5.16 presents the FOMs versus the variation of α. As can be seen, the linearity FOMs exhibit a unimodal behavior where the EVM and ACPR extrema are associated with an optimal α, located near 0.25, which presents 44% of the input power to the main amplifier versus 56% to the peaking amplifier.

On the other hand, we can see that PAE and P_{out} are decreasing with α. This could be interpreted so that the more input power is distributed at the peaking amplifier, the more the efficiency FOMs deteriorates.

However, this example reveals the effect of tuning α, but it is not an optimization process since the other free parameters were set to an arbitrary configuration and were not involved in this optimization process.

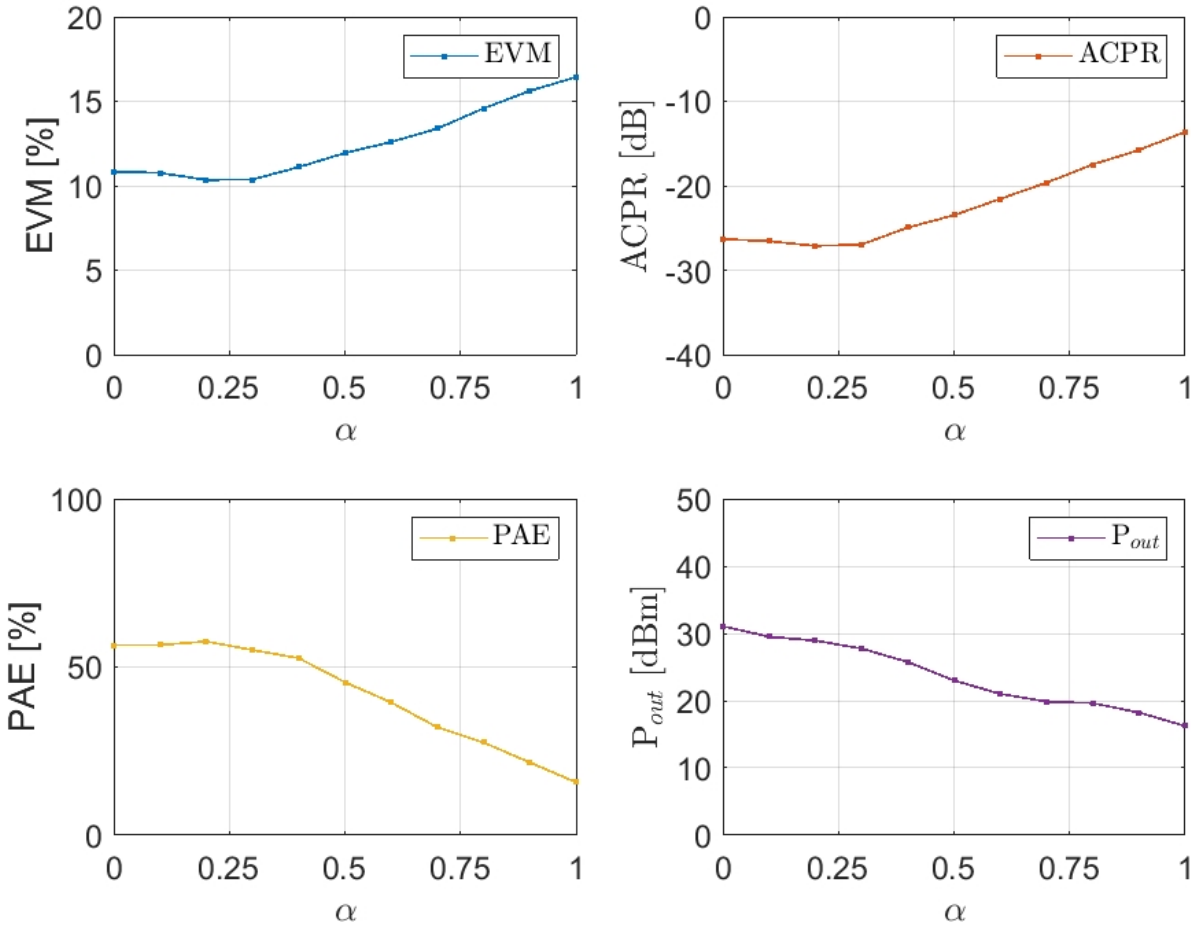


Figure 5.16: Evaluation of FOMs when sweeping α

5.5.2 Cost function

At this point, it is necessary to determine the initial weighting coefficients w and the target FOMs.

We initialize the weighting coefficients $w = [w_1\ w_2\ w_3\ w_4]$ by:

$$\begin{cases} w_{1,1} & = \frac{0.1}{3}\ ;\ w_{1,2} = \frac{0.1}{3}\ ;\ w_{1,3} = \frac{0.1}{3} \\ w_2 & = 0.1 \\ w_3 & = 0.4 \\ w_4 & = 0.4 \end{cases}$$

We attribute more weights to the efficiency since DPD will be included in this test to linearize the DUT. This refers to the fact that the linearity requirements are more relaxed, as it is easier to meet with DPD, unlike efficiency.

Regarding target FOMs, they are defined as:

$$\begin{cases} \text{EVM}_t & = 3\% \\ \text{ACPR}_t & = -50 \text{ dB} \\ \text{PAE}_t & = 60\% \\ \text{P}_{out,t} & = 40 \text{ dBm} \end{cases}$$

5.5.3 Joint Optimization of CFR and DUT

The initial solution for HHSC is defined as:

$$\Theta_0 = [\mu_0 \ \alpha_0 \ \phi_0 \ \psi_0 \ V_{GS,m_0} \ V_{GS,p_0}] = [0 \ 0.5 \ 40° \ 0 \ -1.5 \ -2.5]$$

In this step, HHSC is performed according to Algorithm (13). For the SA algorithm, we set $\mathcal{T}_0 = 1$, $\mathcal{T}_f = 0.01$, $\mathcal{C} = 0.96$, $k_{bolt} = 1$, and $SA_{max} = 60$.

The convergence of J is shown in Figure 5.17.

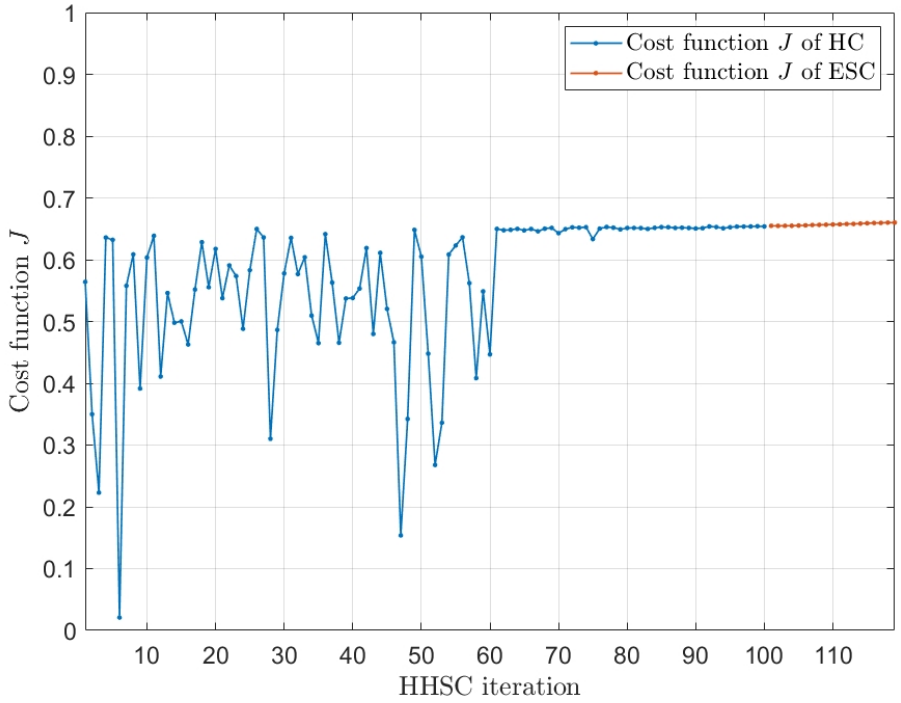


Figure 5.17: Evolution of cost function J over HHSC iterations

In HHSC, we set empirically 100 iterations for SA to converge, while ESC requires 20 iterations. As SA is defined as a stochastic optimization method, we can see from Figure 5.17 that the stochastic behavior has been exhibited in the first 60 iterations where J evolves randomly.

After 61 iterations, the SA algorithm returns the optimal solution. After 120 iterations in total, HHSC returns the optimal configuration $\Theta_{\text{opt,HHSC}}$ summarized in Table 5.3.

Table 5.3: Optimal free-parameters

	μ	α	ϕ	ψ	$V_{GS,m}$	$V_{GS,p}$	J
Θ_0	0	0.5	40°	0	-1.5	-2.5	0.57
$\Theta_{\text{opt,SA}}$	0.62	0.14	164°	1.54	-1.37	-1.95	0.65
$\Theta_{\text{opt,HHSC}}$	0.65	0.15	164°	1.53	-1.39	-1.95	0.66

The cost function $J_{\text{opt,HHSC}} = 0.66$.

The evolution of free parameters over HHSC is illustrated in Figure 5.18, in which the free parameters have similar behavior to J according to the HHSC iterations.

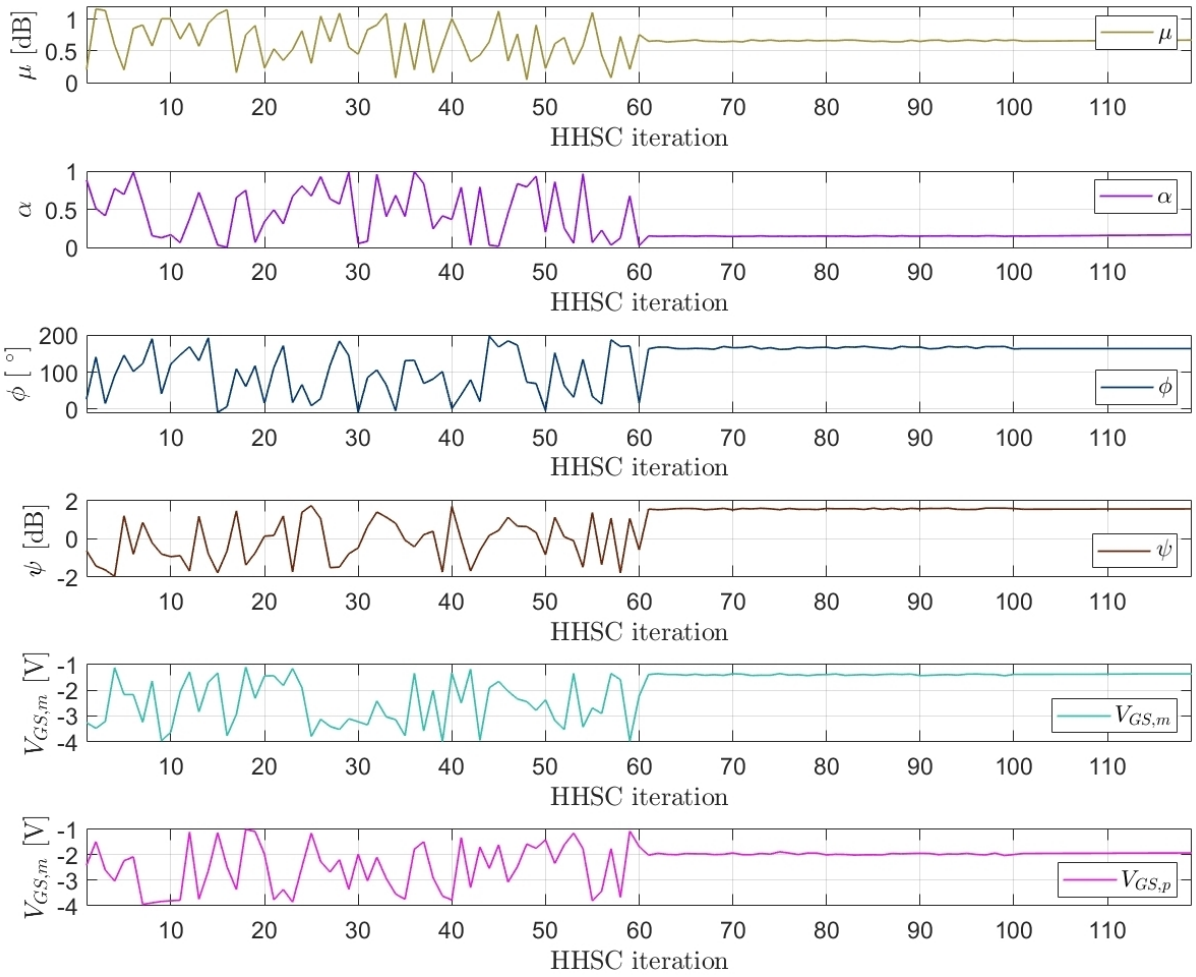


Figure 5.18: Evolution of free-parameters over HHSC iterations

The FOMs corresponding to $J_{\text{opt,HHSC}}$ are shown in Table 5.4.

Table 5.4: FOMs according to the optimal free-parameters $\Theta_{\text{opt,HHSC}}$ with P_{in}=11.36 dBm

EVM_C [%]	EVM_{DD} [%]	EVM_{CDD} [%]	$ACPR_{U1}$ [dB]	$ACPR_{L1}$ [dB]	PAE [%]	P_{out} [dBm]	$J_{\text{opt,HHSC}}$
1.48	2.34	2.72	-42.32	-42.39	16.21	32.92	0.66

It should be noted that HHSC shown in Figure 5.18 was performed on the linear region of the dual-input Doherty PA. Indeed, since SA generates random solutions, it was decided to reduce the operating point of DUT while HHSC is running in order to set up a security measure of the DUT.

Once the free parameters are optimized, we raise the back-off by increasing the input power from 11.36 dBm to 21.16 dBm so that the system performs nearly at $P_{out,t}$. The FOMs after increasing the input power are summarized in Table 5.5

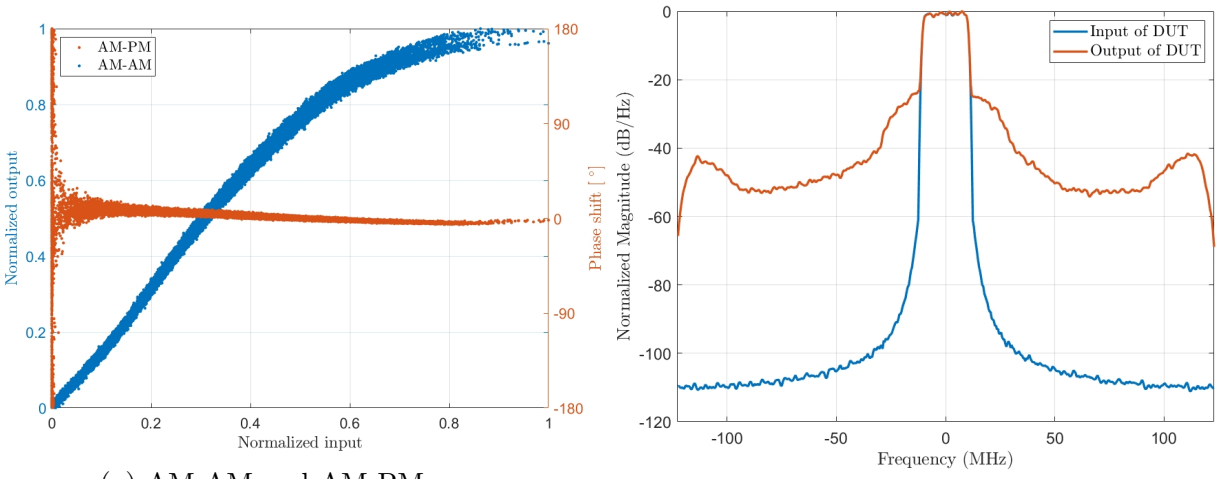
Table 5.5: FOMs according to the optimal free-parameters $\Theta_{\text{opt,HHSC}}$ with P_{in}=21.16 dBm

EVM_C [%]	EVM_{DD} [%]	EVM_{CDD} [%]	$ACPR_{U1}$ [dB]	$ACPR_{L1}$ [dB]	PAE [%]	P_{out} [dBm]	$J_{\text{opt,HHSC}}$
1.48	7.26	8	-27.44	-27.15	46.58	35.63	0.81

According to the results from Table 5.18, the efficiency FOMs are enhanced by sacrificing the linearity FOMs. This confirms the choice of the initial weighting coefficients where w_3 and w_4 have more influence than w_1 and w_2.

By assigning the free-parameters to their optimal configuration $\Theta_{\text{opt,HHSC}}$ in Table 5.3, the DUT, including the digital splitter and transceiver, can eventually be seen as a single-input single-output system where the input is $x(n)$ and the output is denoted $y(n)$.

The AM-AM and AM-PM characteristics of the dual-input Doherty PA are shown in Figure 5.19a, where we can see a saturation at high power, which leads to strong nonlinearities. Besides, the memory effects are exhibited as well. The spectra of the input-output signals of the dual-input Doherty PA is illustrated in Figure 5.19b



(a) AM-AM and AM-PM curves

(b) Spectra of input-output signals

Figure 5.19: Dual-input Doherty PA characteristics with 20MHz LTE signal

The AM-AM characteristic of the RF input-output signal is shown in Figure 5.20, where PAPR of x and $y(n)$ are depicted.

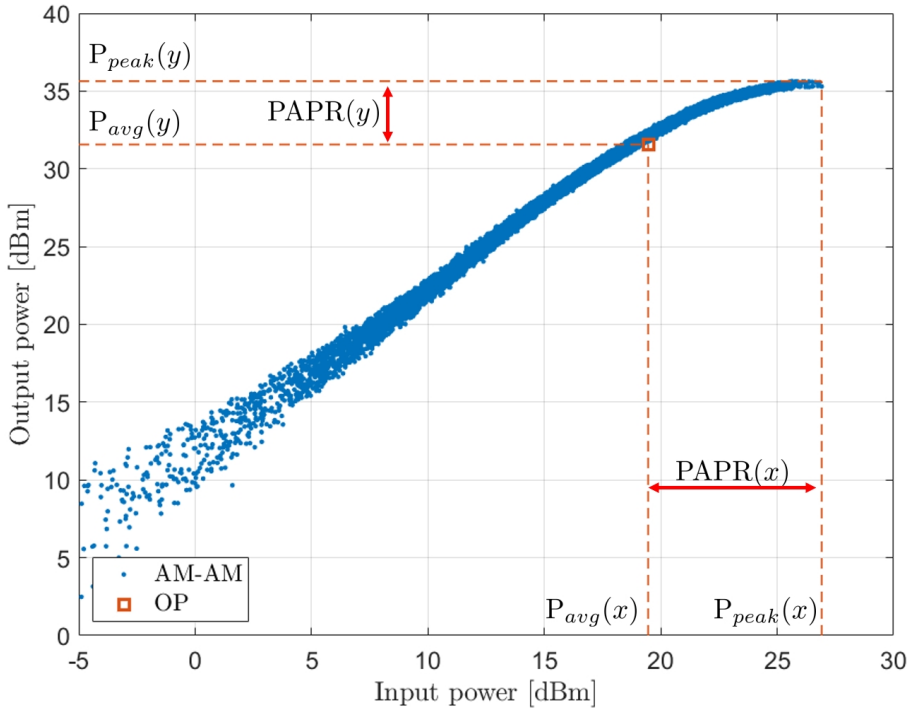


Figure 5.20: AM-AM characteristic using RF input-output signals

5.5.4 DPD Linearization

The DPD is carried out in two steps:

- Determination of the optimal DVR model.
- Convergence of linearity FOMs using ILA.

5.5.4.1 Determination of Optimal DVR Model

According to the HC algorithm presented in Chapter 3, the structure of the DVR model is optimally sized.

In this study, the cost function, denoted by Y, is defined as a search criterion to ensure a good trade-off between three features: modeling accuracy presented by NMSE, denoted by N, model complexity presented by the number of coefficients C, and computational complexity, which is given by the condition number of the regressor matrix $\mathbf{Z^H Z}$ in (2.18), denoted by $Cond$.

The cost function is used to control the convergence of the HC algorithm to return an optimal DVR model with a considerable ability to be implemented in hardware. We define the cost function as:

$$Y = 0.5N + 0.25C + 0.25\log_{10}(Cond) \tag{5.32}$$

As discussed in the previous chapter, the hardware implementation relies on the numerical properties of the DPD model, which in this case, is presented by C and $Cond$. The design of the cost function in (5.32) may deteriorate C, and N since the objective function of $Cond$ is sized in such a way to override the influence of C and N slightly.

However, since we aim to implement DPD on FPGA, which is a Zynq in our case, the deterioration of N can be overcome by the DPD convergence towards the solution that presents a better trade-off between linearization performance, complexity, and numerical properties.

Figure 5.21 illustrates the HC algorithm's evolution in terms of C, $Cond$, and N, where there are a total of 1659 DVR model structures tested through 9 HC iterations. The blue dots present the neighbors tested by the HC algorithm. The red dots indicate the search path taken by the HC algorithm. The green diamond highlights the best solution.

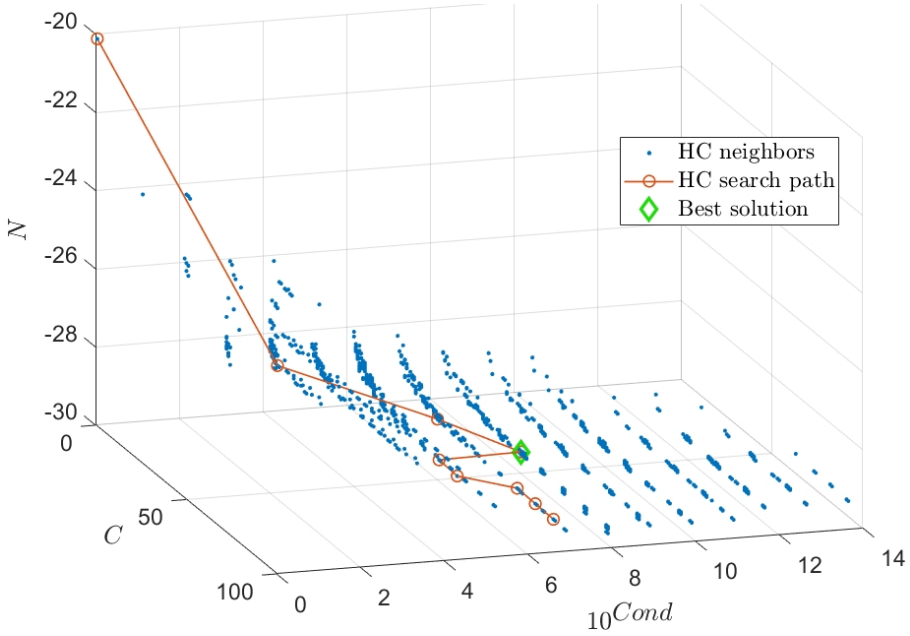


Figure 5.21: Evolution of the HC algorithm in 3D in terms of N, C and $Cond$ for sizing the DVR model

The parameters of the optimal DVR model structure are:

$$
\begin{aligned}
&K = 4 \\
&\beta = [0.25\ 0.5\ 0.75] \\
&M_{lin} = 1\ ;\ M = 3 \\
&T_S = [T_{1,0}\ T_{1,2}\ T_{1,4}]
\end{aligned}
$$

with $C = 50$ coefficients, $10^{Cond} = 10^8$, and $N = -29.45$ dB.

Replacing the parameters of the optimal structure in (3.3), the optimal DVR model used in DPD is expressed by:

$$
\begin{aligned}
x(n) = &\sum_{i=0}^{1} a_i u(n-i) \\
&+ \sum_{k=1}^{4}\sum_{i=0}^{3} c_{ki,1}||u(n-i)| - \beta_k|e^{j\theta(n-i)} \\
&+ \sum_{k=1}^{4}\sum_{i=0}^{3} c_{ki,2}||u(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |u(n)| \\
&+ \sum_{k=1}^{4}\sum_{i=0}^{3} c_{ki,3}||u(n-i)| - \beta_k|e^{j\theta(n-i)} \cdot |u(n)|^4
\end{aligned}
$$

5.5.4.2 DPD using ILA

ILA needs multiple iterations to converge. In this study, we set the number of DPD iterations to 10.

As NMSE being considered as a strong indicator of the ILA convergence, Figure 5.22 presents the evolution of the NMSE according to the DPD iterations. As can be seen, the convergence of NMSE is rapid, in which NMSE is improved significantly from the first DPD iteration.

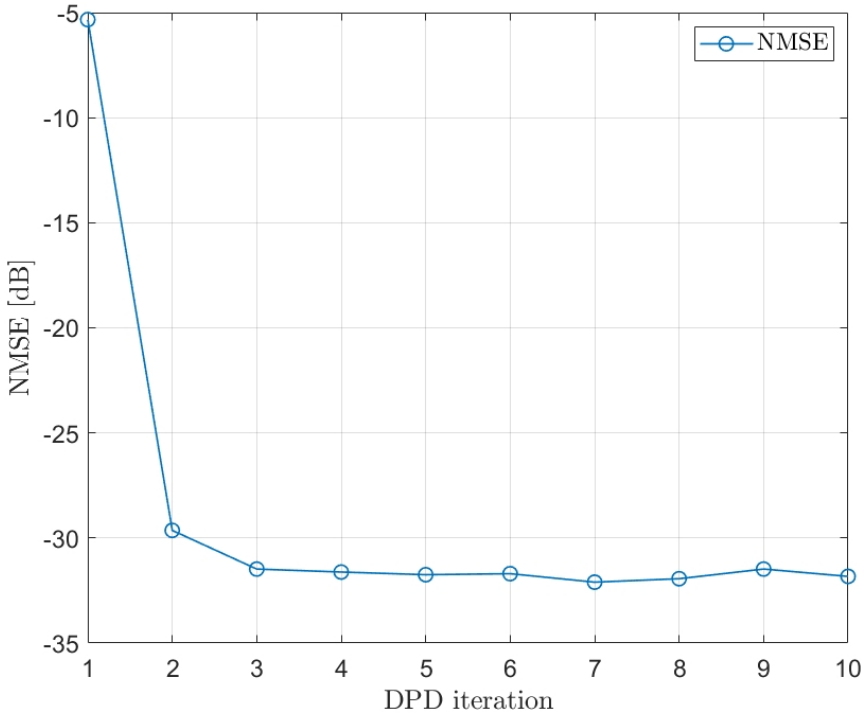


Figure 5.22: Evolution of NMSE according to DPD iterations

The NMSE is computed between $x(n)$ and $z_p(n)$. At the first DPD iteration, the coefficients of the DPD model are initialized by $c = [1, 0, \ldots, 0, 0]$, which make it a transparent block, by which $x(n) = u(n)$, and $z_p(n) = z(n)$.

DPD has significantly improved the linearity FOMs (EVM and ACPR). Figure 5.23 shows the improvement of $\mathrm{EVM_{DD}}$ and $\mathrm{EVM_{CDD}}$ according to the DPD iterations. $\mathrm{EVM_C}$ is presented as well.

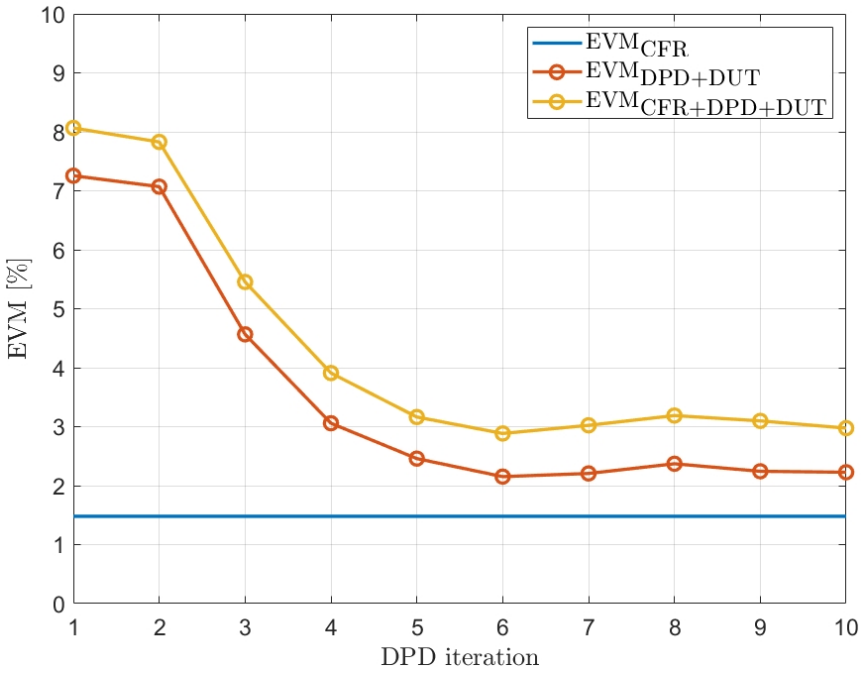
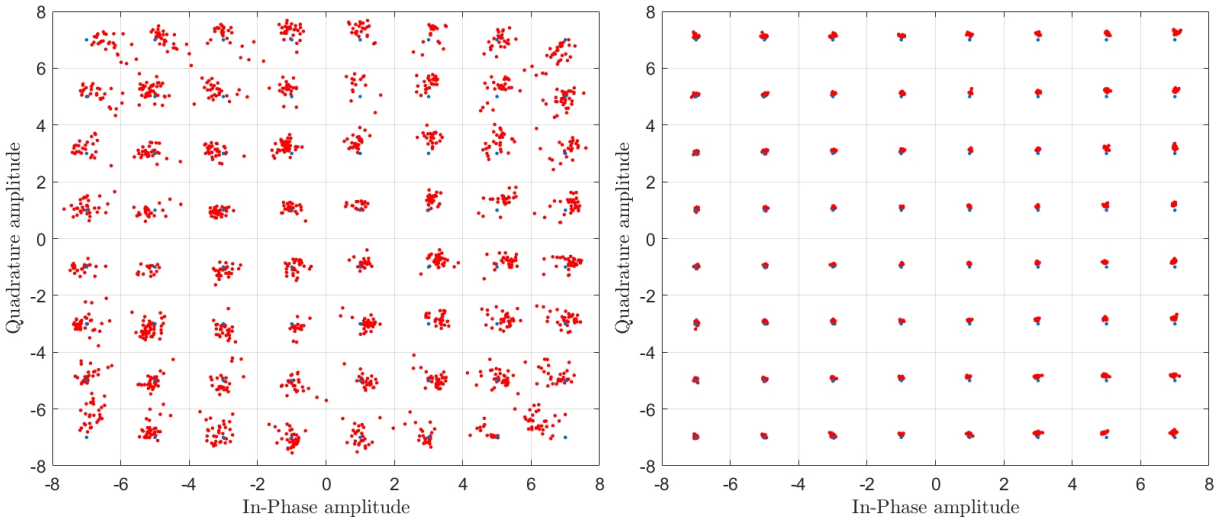


Figure 5.23: EVM of CFR, DPD+DUT, and CFR+DPD+DUT versus DPD iterations

We can see that EVM_{CDD} (EVM_{DD} as well) is improved from value 8% to 2.97%. This can be confirmed in Figure 5.24, where the red dots present the IQ constellation of $y(n)$, and the blue dots are the reference IQ constellation of $v(n)$.



(a) IQ constellation without DPD - EVM=8% (b) IQ constellation with DPD - EVM=2.97%

Figure 5.24: IQ constellation to compute EVM_{CDD} without and with DPD

Regarding ACPR, Figure 5.25 presents the improvement of $ACPR_{L1}$ and $ACPR_{U1}$ according to the DPD iterations. Starting from -27.25 dB before DPD, ACPR has significantly been improved by over 20 dB.

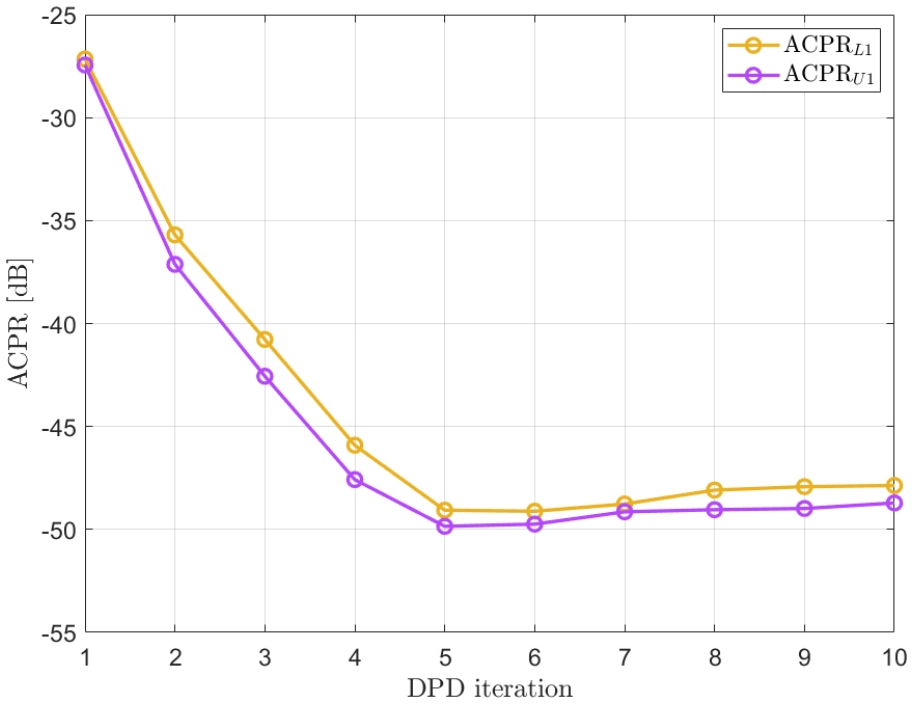


Figure 5.25: ACPR$_{L1}$ and ACPR$_{U1}$ versus DPD iterations

The ACPR improvement can be confirmed in Figure 5.26, where the output signal of dual-input Doherty PA without DPD is shown in the red plot and with DPD in the green plot.

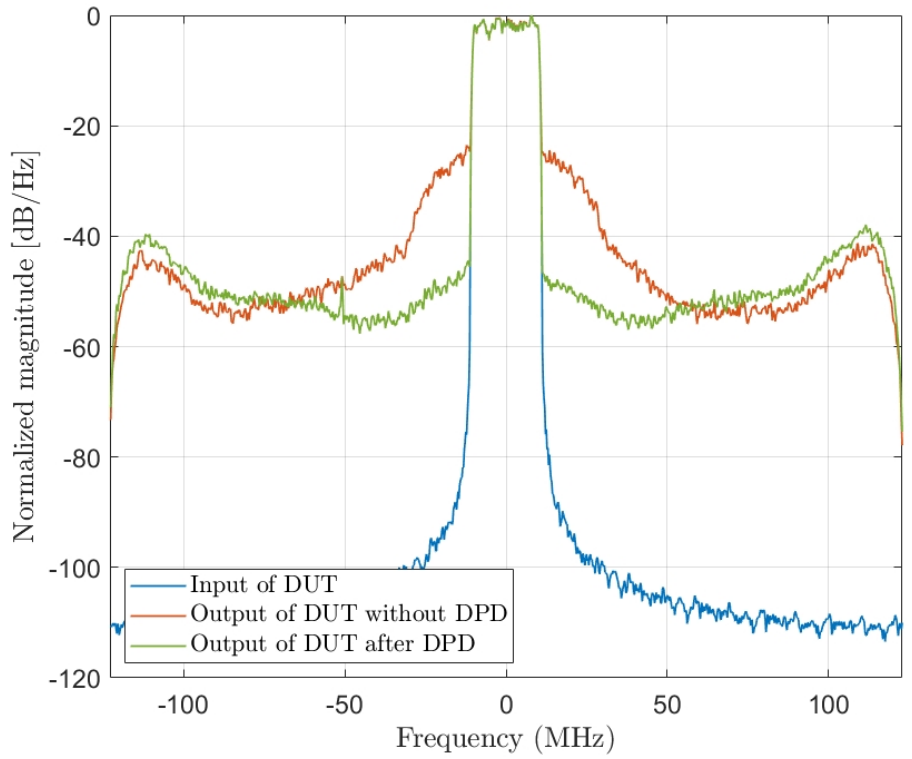
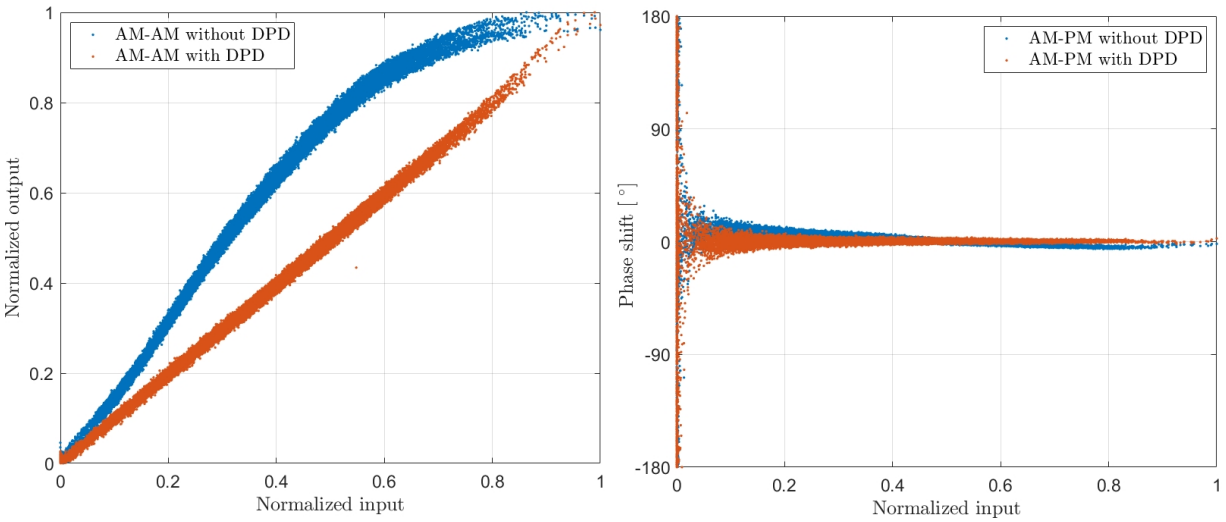


Figure 5.26: Spectra of the output signal of DUT without and with DPD

Figure 5.27 presents AM-AM and AM-PM characteristics of DUT without and with DPD, where Figure 5.27a shows that DPD makes AM-AM linear, while the phase difference in Figure 5.27b is reduced to within almost 3 degrees.



(a) AM-AM without and with DPD (b) AM-PM without and with DPD

Figure 5.27: Characteristics of dual-input Doherty PA without and with DPD

Since the DVR model is optimally sized to ensure a good trade-off between performance linearization, model complexity, and numerical stability of the identification process. The numerical properties of the optimal DVR model at the final DPD iteration are:

$$C = 50$$
$$Cond = 10^8$$
$$\epsilon = 4$$

where ϵ is the dynamic range of the model coefficients defined in 2.23.

The impressive numerical properties of the DVR model used as a predistorter in this study can be confirmed in Figure 5.28, where the real and imaginary parts of the model coefficients are depicted.

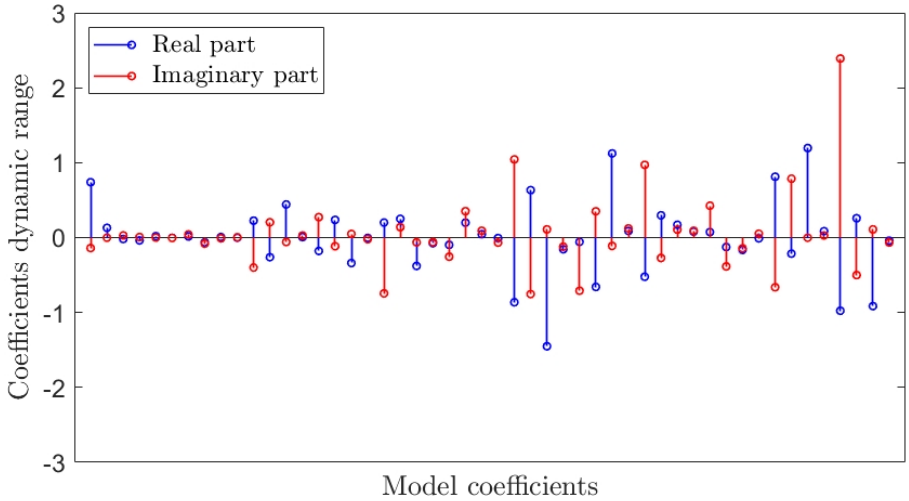


Figure 5.28: Real and imaginary parts of the model coefficients

Linearity FOMs are greatly improved by DPD, which closely meets the targeted linearity FOMs under the optimal configuration $\Theta_{\text{opt,HHSC}}$.

However, DPD applies a BO to the dual-input Doherty PA, which may deteriorate the efficiency. At the final DPD iteration, the efficiency FOMs are:

$$\begin{cases} \text{PAE} & = 30.75\% \\ \text{P}_{out} & = 34.09 \text{ dBm} \end{cases}$$

that does not meet the target FOMs ($P_{out,t}$ and PAE_t) imposed in the user specifications.

Table 5.6 summarizes the FOMs without and with DPD. The cost function is computed using FOMs without and with DPD.

Table 5.6: FOMs according to the optimal free-parameters

	EVM_C [%]	EVM_{DD} [%]	EVM_{CDD} [%]	$ACPR_{U1}$ [dB]	$ACPR_{L1}$ [dB]	PAE [%]	P_{out} [dBm]	$J_{opt,HHSC}$
w/o DPD	1.48	7.26	8	-27.44	-27.15	46.58	35.63	0.81
w DPD	1.48	2.22	2.97	-48.71	-47.86	30.75	34.09	0.78

From Table 5.6, the cost function is decreased after the DPD application. As DPD partially modifies the system conditions, the configuration $\Theta_{opt,HHSC}$ may not always be the optimal solution, which stimulus the motivation to update the weighting coefficients and re-launch a new HHSC.

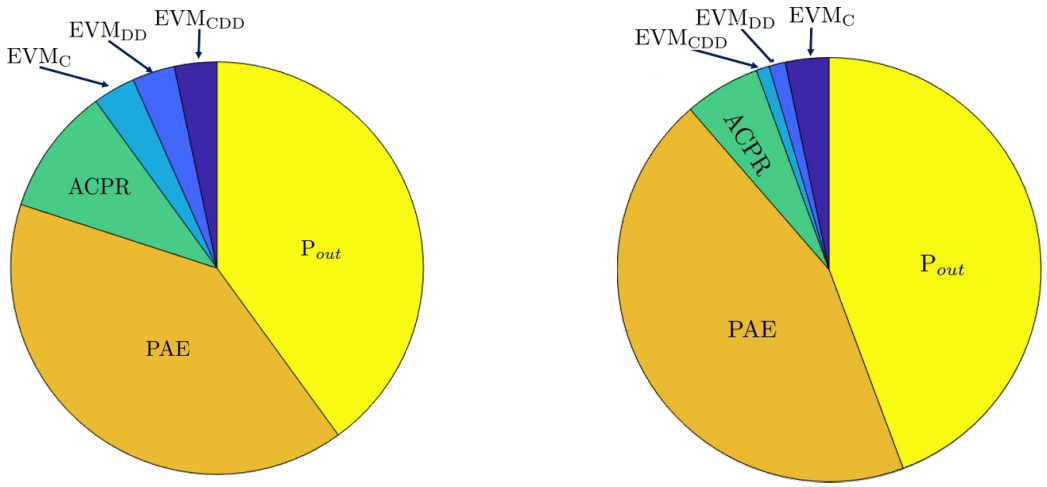
5.5.5 Update Weighting Coefficients

Following the flowchart in Figure 5.12, the weighting coefficients w should be updated to take into account the linearity improvement by DPD to the DUT in which the efficiency is influenced.

According to Algorithm (14), the weighting coefficients w are updated as follows:

$$\begin{cases} w_{1,1} & = 0.05 \ ; \ w_{1,2} = 0.01 \ ; \ w_{1,3} = 0.01 \\ w_2 & = 0.05 \\ w_3 & = 0.44 \\ w_4 & = 0.44 \end{cases}$$

Being the weighting coefficients w reflect each FOM's impact in the value of the cost function J, Figure 5.29 presents the weighting contribution of each FOM, of which Figure 5.29a illustrates the contribution with the initial weighting coefficients, and Figure 5.29b illustrates the contribution after updating w.



(a) With initial w (b) With updated w

Figure 5.29: Impact distribution of FOMs in the design of J

5.5.6 HHSC Optimal Pruning

5.5.6.1 HC Algorithm

Optimal pruning of free parameters in HHSC is performed with the presence of the DPD block where the predistorter is filled by the model coefficients identified at the final DPD iteration and presented in Figure 5.28.

Starting from the initial solution $\Theta_{0,\mathrm{HC}} = [0.65\ 0.15\ 164^\circ\ 1.53\ -1.39\ -1.95]$, the evolution of the HC algorithm to prune the free-parameters for HHSC is illustrated in Figure 5.30. The red dots present the solution at the HC iteration. The blue dots present the neighbors.

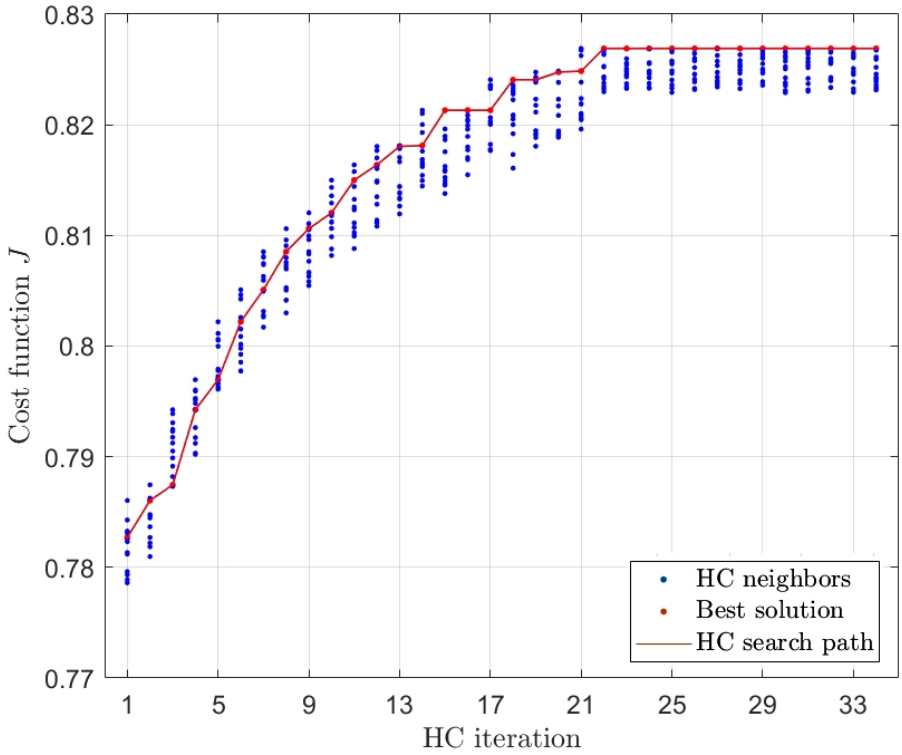


Figure 5.30: Optimal pruning of free-parameters based on HC algorithm

In order to investigate the behavior of the HC algorithm, Figure 5.31 presents which free-parameter has been stored in the dictionary through the HC iterations.

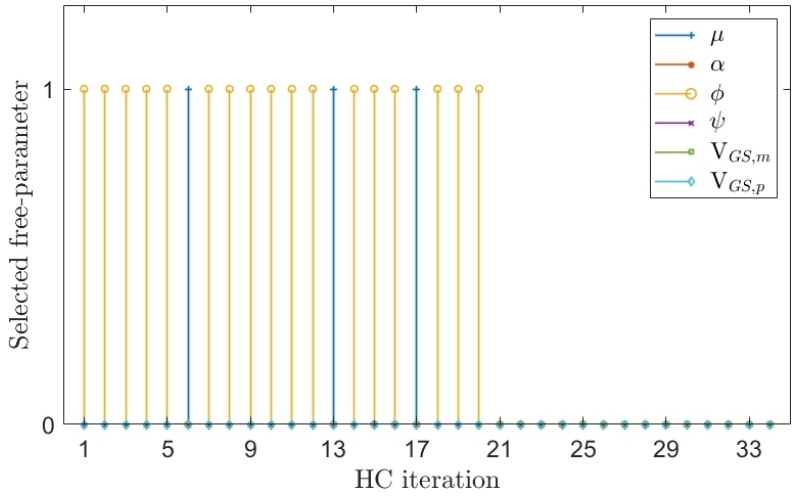


Figure 5.31: Selected Free-parameter over HC iterations

As can be seen, the free parameter ϕ has often been repeated, which means that the optimization behavior of HHSC is seen as most sensitive to ϕ.

On the other hand, the free-parameter μ has been stored three times, where the configuration $[\mu \times (1 \pm \frac{\mu}{10})\ \alpha\ \phi\ \psi\ V_{GS,m}\ V_{GS,p}]$ has the best cost function J at the 6^{th}, 13^{th}, and 17^{th} HC iteration.

After the 21^{th} HC iteration, no configuration was found with a better J, which triggers the HC algorithm to stop.

5.5.6.2 HHSC with Pruned Free-Parameters

As discussed before, the goal of the HC algorithm here is to prune optimally the free parameters of HHSC, which aims to find the most sensitive free parameter to CFR and DUT with DPD.

Hence, $\Theta' = [\phi]$ is the most sensitive free parameter, which will be used in HHSC according to the cost function J designed with the updated weighting coefficients w.

Starting from the initial solution $\Theta'_0 = [164°]$, which presents the optimal ϕ from the previous HHSC, Figure 5.32 present the evolution of the cost function J and the free-parameter ϕ according to the HHSC iterations.

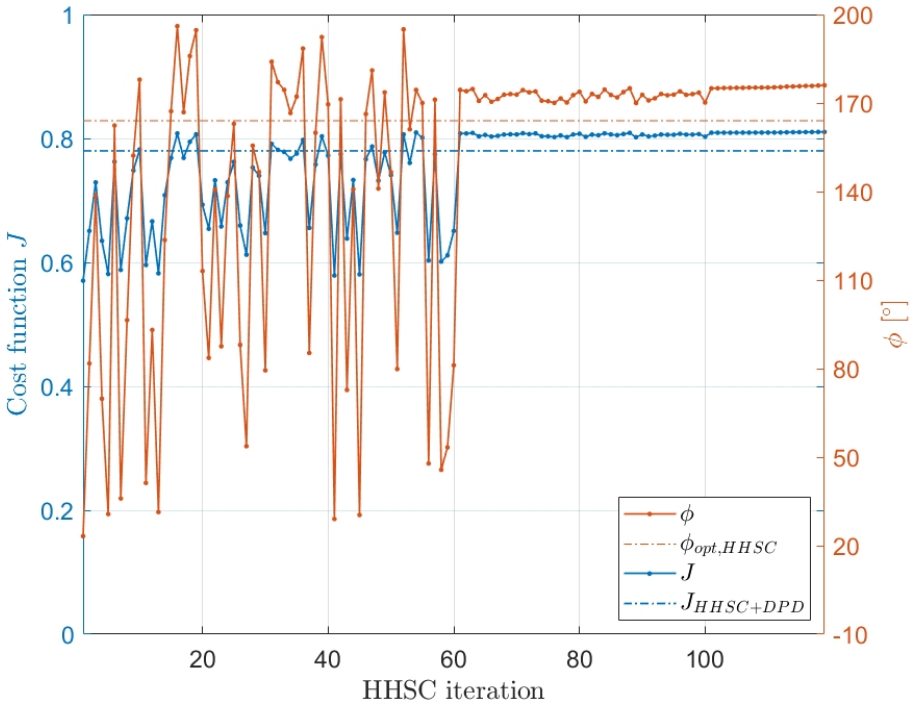


Figure 5.32: Evolution of cost function J and Θ' over HHSC iterations

According to Figure 5.32, the free-parameter ϕ has been re-optimized, where its optimal value becomes $\Theta'_{\text{opt}} = 175.9°$ that corresponds to the optimal cost function $J_{\text{opt,upd}} = 0.81$. By comparing $J_{\text{opt,upd}}$ and $J_{\text{HHSC+DPD}}$, the DPD coefficients are required to be updated using ILA since the optimal configuration of DUT is changed.

Table 5.7 summarizes the results, where FOMs are presented before and after applying DPD. At this level, the application of DPD consists only of updating the existing DPD coefficients in the predistorter since the optimal pruning of free parameters HHSC was performed with DPD.

With the new optimal configuration, DPD has improved the linearity FOMs. On the other hand, the efficiency FOMs are improved. This can be confirmed in Table 5.7, where the PAE is improved by almost 5.5%. By comparing the cost function, it can be seen

Table 5.7: FOMs according to the optimal free-parameters after the optimal pruning of free-parameters in HHSC

	EVM_C [%]	EVM_{DD} [%]	EVM_{CDD} [%]	$ACPR_{U1}$ [dB]	$ACPR_{L1}$ [dB]	PAE [%]	P_{out} [dBm]	$J_{HHSC+DPD}$
Optimal pruning of HHSC	1.48	2.22	2.97	-48.71	-47.86	30.75	34.09	0.81
Update DPD coeff.	1.48	2.04	2.46	-49.21	-50.10	36.11	39.11	0.85

that $J_{\text{HHSC+DPD}} = 0.85$ becomes better than $J_{\text{opt,HHSC}} = 0.82$, which leads to finishing the auto-tuning approach.

Figure 5.33 illustrates the final architecture of the dual-input Doherty PA with CFR and DPD. The final optimal configuration is also highlighted. The AM-AM and AM-PM characteristics of the whole system (CFR+DPD+DUT) are shown in Figure 5.33, along with the spectra of the input-output signals and the IQ constellation of $u(n)$ plotted in blue and $y(n)$ plotted in red.

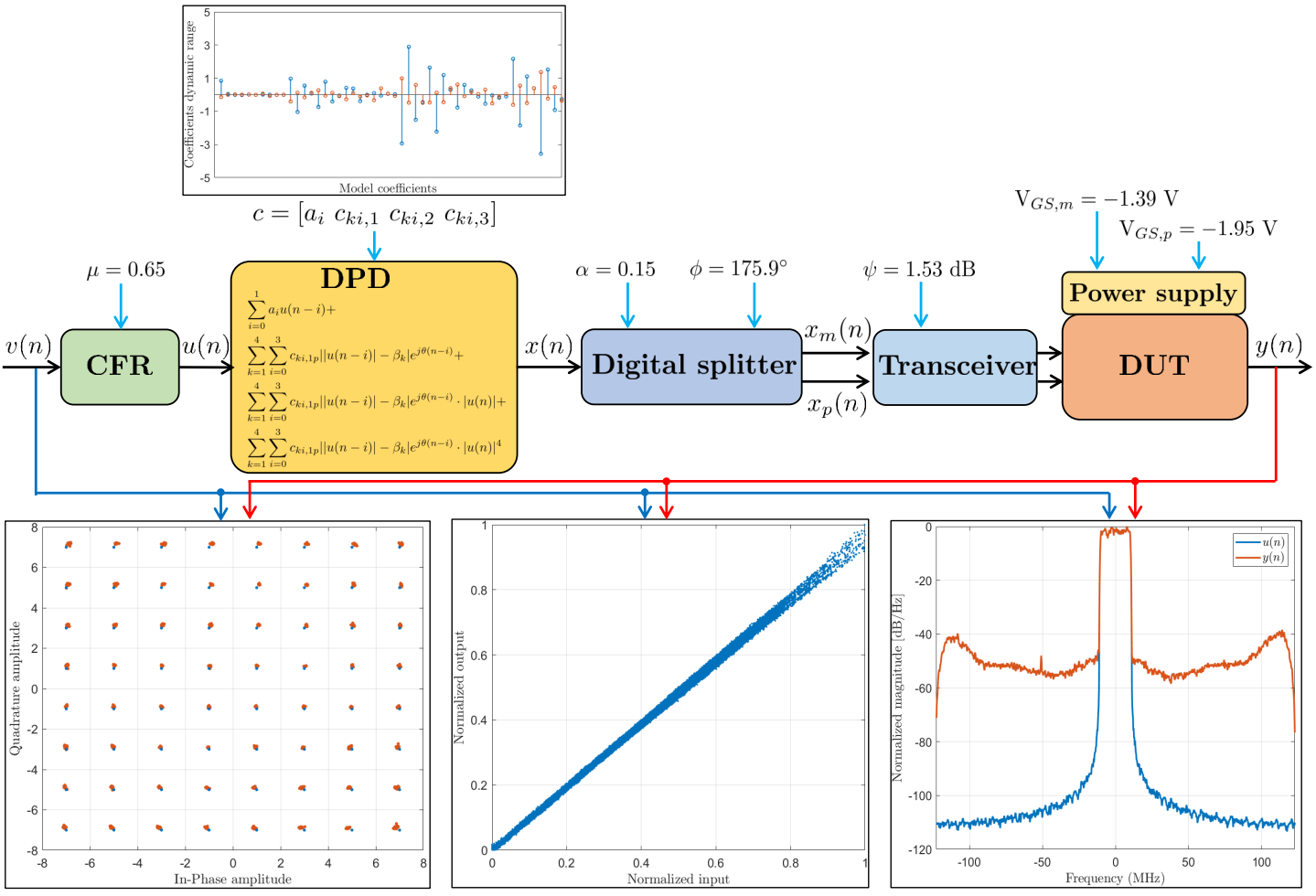


Figure 5.33: Final optimal configuration with DPD

5.6 Conclusion

In this chapter, we proposed an auto-tuning approach to exploit at best dual-input Doherty PA to maximize power efficiency while being compliant with the linearity specifica-

tions.

The proposed auto-tuning approach relies on conducting a global optimization combined with a control process to find the optimal configuration of a set of the crucial circuit and system-level parameters that are appropriately merged with the DPD linearization and the CFR technique.

This proposed approach has been performed according to an adaptive designed cost function, representing the trade-off between efficiency and linearity.

In order to sharpen the optimal configuration, we propose a new approach based on the HC algorithm to prune the free parameters optimally when DPD linearization is applied.

The proposed approach has been validated through experimental results, in which we use a 20 MHz LTE signal scenario.

The proposed approach to optimizing the dual-input Doherty PA has been well validated by presenting a good trade-off between linearity, computational complexity, and efficiency. Besides, the DPD model used, which is optimally sized, has very good numerical properties, making it a perfect candidate for its implementation on hardware such as FPGA.

Conclusion and Perspectives

Contributions

In this dissertation, we have focused on digital predistortion, which is a linearization technique of power amplifiers to eliminate the distortions due to the nonlinearities and memory effects. This dissertation mainly focuses on three aspects, which have made our contribution: study and comparison of DPD models, design of an optimal DPD model with its implementation strategy, and optimization and linearization of dual-input Doherty PA with efficiency enhancement.

1. A comparative study is made between DPD models, which are categorized in this study into two families: global models derived from the Volterra series and models with segmentation. The analysis comparison has been made according to three aspects: modeling accuracy presented by NMSE and ACPR, model complexity presented by the number of model coefficients, and numerical properties presented by the dynamic range of coefficients and the condition number of regressor matrix for the identification process of predistorter. The validation of this study has been performed using measurements from two different PAs (LDMOS 50 W PA and Doherty three-way 500 W PA) using different scenarios in terms of signal bandwidth and output power.

2. For models with segmentation, particularly the DVR model, two major studies were developed: optimizing the thresholds and determining the optimal structure. For the thresholds optimization, a new approach is proposed to set the thresholds optimally using an iterative approach based on the decomposition of the global optimization problem into a set of unimodal sub-problems so that a unidirectional minimization, such as the golden section search, can be used to optimize the positions of thresholds. For the DVR model sizing, the hill-climbing algorithm is proposed to search for the optimal model structure. The convergence of hill-climbing is confirmed in terms of modeling accuracy of the optimal solution, algorithm complexity, and execution time. Two major contributions were highlighted: search criterion representing a trade-off between modeling accuracy and model complexity. A neighborhood definition proposed to control the algorithm toward the optimal solution.

3. A new proposed approach to design an optimal DVR model is based on the combination of threshold optimization and model sizing. The proposed approach is achieved through three different algorithm versions. The first algorithm version relies only on optimizing the thresholds in model structures located on the envelope of model sizing. In contrast, the second version consists in integrating the thresholds optimization process into the model sizing but assuming that the memory depth and model terms are not sensitive to the optimal thresholds. So, the thresholds are optimized only if the number of segments of a model structure changes. The third version is an update of the second version, in which the terms of the model are viewed as a sensitive parameter leading to optimizing the thresholds when the

number of segments and the model terms are changed. On the other hand, in this framework, we have investigated the impact of nonlinearities order in the DVR model by which we highlight the interest of using models with segmentation with reduced orders of nonlinearities, unlike model from Volterra series, such as GMP, where the nonlinearities order is highly required to achieve the same performances.

4. An auto-tuning approach is proposed to find the optimal configuration of a set of essential circuits and system-level parameters of dual-input Doherty PA. This approach is achieved by combining two optimization methods: simulated annealing as a global optimization algorithm and extremum-seeking as an adaptive control. The proposed approach's convergence is constrained by a proposed adaptive cost function designed to purposely represent a trade-off between linearity and efficiency. CFR technique and DPD linearization are integrated optimally to the proposed architecture of dual-input Doherty PA to meet the user specifications by enhancing efficiency while maintaining a conforming linearity level. An optimal pruning search based on the hill-climbing algorithm is proposed to reinforce and control the approach's convergence toward the specified user targets.

Perspectives

To extend this dissertation, some works could be developed:

1. Regarding the comparison study of DPD models, the DVR model is the only model with segmentation used in this study. It will be interesting to investigate other models, such as B-spline models and vector switched models, in the analysis comparison.

2. Generally, the PA characteristics and system performance depend on many factors such as the average power, the probability density function of the signal, the signal bandwidth, the circuit temperature, etc. The approach to design the optimal DVR model needs to be redesigned to fit the PA characteristics if changed, leading to an optimal reconfigurable DVR model. For that, it will be necessary to add the PA characteristics features (average power for example) into the design process of the DVR model. Besides, the runtime of the design algorithm should be decreased so that the real-time implementation can be viable.

3. Apply a stochastic approach based on simulated annealing algorithm to design the DVR model.

4. Implement the optimal DVR model on digital hardware, e.g., FPGA.

5. Reduce the complexity of the proposed auto-tuning approach to optimize the dual-input Doherty PA by combining CFR, DPD, and digital splitter into one block, in which a nonlinear LS method, such as Newton-Raphson algorithm, will be involved to estimate the DPD coefficients, as well as the optimal configuration.

6. Since the optimization of dual-input Doherty PA is conducted under specified conditions, especially in terms of center frequency, output power level, signal bandwidth, it will be exciting to perform the auto-tuning approach according to a vast data set of center frequencies, output power levels, and signal bandwidths, in order to apply some machine-learning models to fit this data and find an optimal global configuration.

Further research works are currently in progress for submission, which are listed along with the abstract:

- Chouaib Kantana, Olivier Venard and Genevieve Baudoin, "A System Approach to Design an Optimal Digital Predistortion for Power Amplifier Linearization".

  – Abstract: *The segmentation approach has proven to be an efficient approach to replace the Volterra series for behavioral modeling of power amplifier (PA) or serve as a digital predistortion (DPD) model. The Decomposed Vector Rotation (DVR) is one of the most common model based on the segmentation approach that represents nonlinear dynamical systems with memory effects. Crucial, the study of the DVR model involves two principal aspects: the optimization of the thresholds that bound the segments and the determination of the optimal model structure, i.e., determining the optimal number of segments, the optimal memory depth, and the optimal set of model terms. In this paper, we propose an efficient system approach to design an optimal DVR model by jointly optimizing the model structure and threshold segmentation. Convergence towards the optimal model is ensured by a cost function that provides the best trade-off between modeling accuracy and model complexity. Experimental results on different amplifiers confirmed the system approach's effectiveness in designing an optimal DVR model with good linearization performance, reduced number of coefficients, and remarkable numerical properties, making it friendly to hardware implementation, for which an implementation strategy is investigated.*

- Chouaib Kantana, Rui Ma, Mouhacine Benosman, Olivier Venard and Genevieve Baudoin, "On the Low-Complex Optimal model of Digital Predistortion: A Stochastic Approach".

  – Abstract: *This paper investigates the application of a stochastic approach based on simulated annealing (SA) algorithm to design an optimal decomposed vector rotation (DVR) model for digital predistortion (DPD). The design of DVR is based on the optimization of thresholds β that bounds the segments and the model sizing that find the optimal model structure. The combination of threshold optimization and model sizing is viewed as a nonlinear regression problem requiring nonlinear estimation techniques. The SA algorithm is converged to the optimal solution according to a cost function that presents a trade-off between linearization performance and model complexity. This study reveals that the SA algorithm can be used as a nonlinear estimation technique. Experimental results using measurements from dual-input Doherty power amplifier (DIDPA) are provided, demonstrating the efficiency of SA in designing the optimal DVR model. A comparison is made between the results obtained by the hill-climbing heuristic to size the model and the golden search (GS) technique to optimize β.*

- Chouaib Kantana, Rui Ma, Mouhacine Benosmane, "A System Approach for the Linearization and Efficiency Enhancement of Dual-Input Doherty Power Amplifier".

  – Abstract: *In this paper, we propose an efficient system approach to improve the power efficiency of dual-input Doherty power amplifier (DIDPA) by maintaining a reliable level of linearity following the communication standards. The system approach handles DIDPA according to a specific workflow. Firstly, an auto-tuning process based on a hybrid heuristic search control (HHSC) is applied to optimally define DIDPA configuration by optimizing its free-parameters, including peak-to-average power (PAPR) reduction threshold. The HHSC is driven by a cost function designed to moderate the inherent trade-off between linearity and power efficiency. The digital predistortion (DPD) is then integrated to linearize DIDPA using an optimal reduced-complexity model based on the segmentation approach. The cost function coefficients are updated optimally based on the linearity improvement by DPD. An*

optimal pruning process of the free-parameters, based on hill-climbing (HC) heuristics, is proposed to reduce the HHSC complexity in order to refine the optimal DIDPA configuration with the updated cost function. The system approach has been approved by experimental results using an LTE 20−MHz signal with a PAPR of 8−dB PAPR. In the first step where HHSC is applied, DIDPA exhibited a drain efficiency of 61%. DPD linearization improved linearity using a low-complex model with only 30 coefficients, which exhibited an error vector magnitude (EVM) of 2.5% and an adjacent channel power ratio (ACPR) of -50−dB at an averaged output power of 34−*dBm. By updating the cost function coefficients and pruning the free-parameters, DIDPA exhibited an EVM of 3%, an ACPR of -50−dB, and a drain efficiency of 47% at an average output power of 39−dBm.*

Résumé détaillé de la thèse en français

Approche système d'optimisation d'un prédistorteur numérique avec une complexité réduite pour la linéarisation des amplificateurs de puissance RF

Au cours des dernières décennies, les systèmes de communication sans fil ont connu une évolution spectaculaire dans la manière de fournir des services de transmissions à des débits de plus en plus élevés pour des applications telles que la communication mobile, les réseaux Wi-Fi, la diffusion multimédia, etc. Ces progrès ont conduit à une prolifération de nouveaux services radio. Or, les exigences croissantes des utilisateurs en termes de service, de qualité, et de connectivité mettent massivement à l'épreuve les ressources énergétiques et fréquentielles. Dans les systèmes de radiofréquence (en anglais : Radio Frequency - RF), l'amplificateur de puissance (en anglais : Power Amplifier - PA) est un élément crucial de la chaîne dont le rôle est d'amplifier le signal transmis afin d'élever le niveau de puissance de la transmission. Le PA demeure l'un des composants les plus importants en termes de coût, d'autonomie, et de rendement énergétique. Le rendement énergétique maximal du PA est atteint lorsque le point de fonctionnement est proche de la zone de saturation, malheureusement dans cette zone, les non-linéarités de l'amplificateur distordent le signal transmis. Ces distorsions peuvent devenir particulièrement fortes pour les signaux à large bande.

Sur le plan spectral, des formats de transmission tels que l'accès multiple par répartition en code (en anglais : Code Division Multiple Access - CDMA) et le multiplexage par répartition orthogonale de la fréquence (en anglais : Orthogonal Frequency-Division Multiplexing - OFDM) sont de plus en plus utilisés pour acheminer des flux d'informations élevés sur un espace fréquentiel minimal, dont l'objectif est de surmonter la limitation spectrale des ressources et de faire preuve d'une efficacité spectrale élevée. Ces formats de transmission génèrent une enveloppe de signal transmis non-constante, ce qui implique un rapport puissance crête sur la puissance moyenne (en anglais : Peak-to-Average Power Ratio - PAPR) élevé. Par conséquent, ces signaux présentent une sensibilité aux non-linéarités du canal de transmission, et en particulier à celle du PA.

L'étude de PA repose enfin sur l'assurance d'un véritable compromis entre linéarité et efficacité énergétique. L'efficacité énergétique représente une caractéristique fondamentale quantifiée principalement par deux définitions : le rendement de drain η_{DC} et le rendement en puissance ajoutée η_{PAE} (en anglais : Power Added Efficiency - PAE). Le rendement η_{DC} est défini par le rapport entre la puissance de sortie et la puissance continue consommée. Le rendement η_{PAE} est le rapport entre la puissance RF fournie en sortie du PA diminuée de la puissance RF à son entrée et la puissance consommée sur l'alimentation. Le rendement énergétique est un paramètre crucial qui dépend entre

autres du choix du point de polarisation des transistors. Le PA doit être optimisé lors de la phase de conception afin de maximiser son rendement énergétique.

Le comportement non-linéaire du PA est généralement analysé en l'excitant par des signaux de natures diverses (signal mono-porteuse, signal à deux tons, signal modulé), et déterminant le gain, la puissance de saturation, le point de compression à 1 dB et les distorsions d'intermodulation. Dans le cas où le PA serait excité par un signal modulé, la non-linéarité est généralement quantifiée par 2 paramètres :

- ACPR (en anglais : Adjacent Channel Power Ratio) pour quantifier les remontées spectrales dans les bandes fréquentielles adjacentes au canal principal. Il est défini par le rapport des puissances entre celle du canal principal et celle dans les canaux adjacents, sous la forme suivante :

$$ACPR_{(dB)} = 10\log_{10}\Big(\frac{2\int_{BW_c} P(f)df}{\int_{BW_g} P(f)df + \int_{BW_d} P(f)df}\Big)$$

  où $P(f)$ est la densité spectrale de la puissance du signal en sortie du PA.

- EVM (en anglais : Error Vector Magnitude) pour mesurer l'écart de position entre celles des symboles reçus et les positions idéales attendues. La mesure EVM est effectuée sur les données I/Q en bande de base, ce qui nécessite une démodulation du signal RF amplifié. Il est défini par :

$$EVM_{(\%)} = 100\sqrt{\frac{\sum_{k=0}^{N} |s_{k,ideal} - s_{k,reel}|^2}{\sum_{k=0}^{N} |s_{k,ideal}|^2}}$$

Dans les standards de communication, l'EVM et l'ACPR sont donnés de manière complémentaire. L'ACPR évalue l'effet de la non-linéarité de l'amplificateur sur les bandes adjacentes, tandis que l'EVM quantifie les distorsions dans la bande utile.

Sous certaines conditions initiales, le PA est considéré comme un système à fonctionnement statique, ce qui signifie que la sortie de PA à l'instant t ne dépend que de l'entrée au même instant. En réalité, le PA présente souvent des phénomènes dispersifs non-linéaires et dynamiques appelés effets de mémoire. Cela signifie que la sortie du PA à tout instant dépend non seulement de l'entrée instantanée correspondante, mais aussi des entrées à d'autres instants précédents, ce qui impacte ses caractéristiques fréquentielles.

La caractérisation du PA dans la figure 1 montre clairement le compromis entre linéarité et rendement énergétique. Afin de répondre aux spécifications de linéarité imposées par les normes de communication (3GPP par exemple), la puissance moyenne du signal d'entrée doit être abaissée en appliquant un recul (en anglais : back-off) supérieur au PAPR du signal à amplifier, tel que montré dans la figure 1. Grâce au recul, le PA fonctionne dans sa zone linéaire. Mais dans cette zone le rendement est généralement très faible. Afin d'ajuster le point de fonctionnement qui répond au compromis linéarité-rendement, le back-off doit être ajusté de manière optimale. Les techniques de linéarisation et d'amélioration du rendement ont pour but de faire fonctionner le PA près de sa zone de saturation où le PAE est maximum, tout en assurant d'excellentes performances de linéarité.

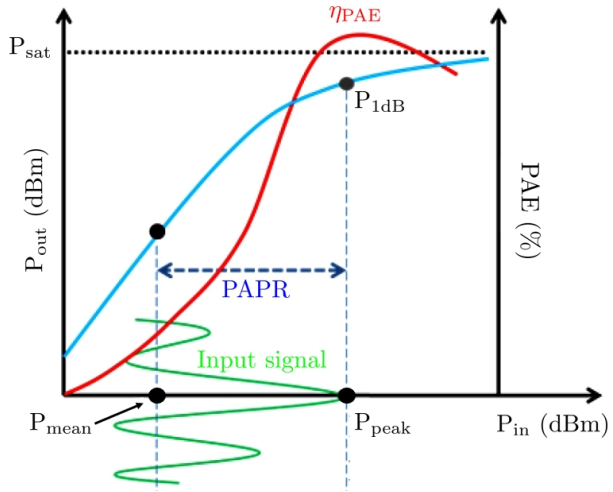


Figure 1 : Compromis linéarité-rendement

Une des techniques pour améliorer le rendement énergétique consiste à réduire le PAPR du signal transmis, nous parlons de réduction du facteur de crête (en anglais : Crest Factor Reduction - CFR). Un exemple typique de CFR est la méthode d'écrêtage, qui consiste à écrêter l'amplitude du signal émis selon un seuil d'écrêtage, ce qui permet de réduire la dynamique d'amplitude, et donc le PAPR. Par contre, l'écrêtage d'amplitude de signal par le CFR dégrade le signal de façon non-linéaire. L'enjeu majeur des techniques de CFR est de réduire le PAPR en dégradant le signal le moins possible tout en préservant le débit de transmission le débit de transmission, et sans ajouter de traitement supplémentaire à la partie réception.

Quant à la linéarité, des techniques ont été développées dans le but de faire fonctionner le PA dans sa zone non-linéaire et de l'associer à un dispositif de compensation des non-linéarités sans sacrifier sa puissance de sortie. Plusieurs techniques de linéarisation ont été proposées telles que les techniques de contre-réaction ou de feedforward et la technique de la prédistorsion analogique ou numérique (en anglais : Digital Predistortion - DPD) qui est actuellement la plus répandue et constitue un axe de recherche fondamental de cette thèse. Le principe du DPD consiste à introduire en amont du PA un module appelé prédistorteur qui applique une déformation (distorsion) au signal d'entrée pour que le système issu de la cascade DPD + PA soit une amplification linéaire. Ce principe est illustré dans la figure 2.

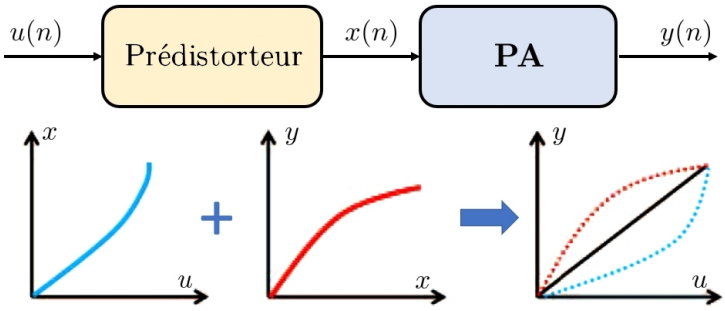


Figure 2 : Principe de la DPD

La DPD est généralement réalisée à l'aide de modèles dynamiques non-linéaires similaires aux modèles comportementaux de PA implémentés en bande de base, dont la caractéristique est l'inverse de celle de PA. Les paramètres de la DPD sont identifiés à partir du signal original à transmettre et des signaux d'entrée et de sortie du PA et selon l'architecture d'identification de la DPD. Deux approches architecturales sont principalement utilisées : l'architecture d'apprentissage indirect (en anglais : Indirect Learning

Architecture - ILA), et l'architecture d'apprentissage direct (en anglais : Direct Learning Architecture - DLA). Pour compenser les non-linéarités du PA et les effets mémoire, il est nécessaire d'utiliser un modèle non-linéaire et à mémoire.

A travers la littérature, nous distinguons les modèles non-linéaires et à mémoire en deux catégories : les modèles dérivés de la série de Volterra et les modèles basés sur les réseaux de neurones. D'autre part, les modèles de la DPD reposent généralement sur deux approches. L'approche globale qui consiste à utiliser des modèles avec un ordre de non-linéarité élevé et une profondeur de mémoire pour modéliser le comportement non-linéaire du système. L'approche de segmentation qui consiste à diviser le signal d'entrée en différentes régions ou segments et à appliquer un modèle d'ordre de non-linéarité très faible sur chaque segment. La comparaison des modèles issus des deux approches constitue le premier objectif de cette thèse qui consiste à analyser les performances de chaque modèle selon des critères résultant du compromis entre les performances de linéarisation, la complexité et les propriétés numériques pour l'implémentation matérielle.

Le deuxième objectif de cette thèse est de concevoir un modèle optimal de DPD basé sur l'approche de segmentation. Dans cette étape, une analyse détaillée du modèle a été réalisée, afin d'étudier ses caractéristiques de linéarisation. Deux aspects sont traités en relation avec l'étude de modèles basés sur l'approche segmentation : l'optimisation des seuils qui séparent les segments et le dimensionnement optimal de sa structure. Le premier aspect consiste à positionner de manière optimale les seuils qui séparent les segments. L'avantage d'optimiser la segmentation permet d'améliorer les performances de linéarisation, contrairement à une segmentation uniforme où les seuils sont répartis uniformément sur la dynamique du signal d'entrée. Le deuxième aspect concerne le dimensionnement du modèle où un algorithme heuristique de type hill-climbing (HC) a été utilisé pour déterminer la structure optimale en fonction du nombre de segments, de la profondeur mémoire et des fonctions utilisées dans le modèle. Ces deux aspects ont été combinés dans une étude où une approche système a été proposée pour concevoir un modèle optimal global.

Le modèle optimal est par la suite utilisé pour linéariser un PA d'une architecture avancée de type Doherty à double entrée (en anglais : Dual-Input Doherty Power Amplifier - DIDPA), ce qui constitue notre troisième objectif de la thèse. Le DIDPA a d'abord été optimisé par une proposition d'approche reconfigurable dont le principe consiste à optimiser les paramètres du système selon une fonction de coût adaptative. Le CFR et la DPD sont insérés en amont du DIDPA afin de maximiser ses performances tout en conservant une linéarité respectant les standards de communication.

Cette thèse s'inscrivait dans le cadre du projet FUI22 APOGEES (AmPlification reOnfiGurablE multimodes). L'objectif du projet APOGEES était de résoudre l'ensemble des problèmes de reconfigurabilité des systèmes d'amplification pour répondre aux nombreux défis générés par les usages inédits du spectre et les évolutions de la société pour le trafic aérien, les communications tactiques et l'utilisation opportuniste du spectre pour la télévision. Le projet était porté par Arelis et labellisé par les pôles Aerospace Valley (pôle principal), Images & Réseaux et Elopsys. Les partenaires impliqués dans ce projet étaient : Centrale Supelec Rennes, ESIEE - Paris, Telerad, Thales Communications & Sécurité, Université Bordeaux IMS, Université Nantes IETR, Université Poitiers XLIM. Au cours de cette thèse, j'ai obtenu l'opportunité d'effectuer un séjour scientifique en tant que chercheur visiteur à MERL (Mitsubishi Electric Research Laboratories) à Cambridge aux États-Unis. Cette mobilité internationale a été encadrée par le Dr Rui MA, chercheur senior au MERL. Le sujet est intitulé : "Prédistorsion numérique par l'apprentissage automatique pour linéariser le PA Doherty à double entrée".

Analyse et étude comparative des modèles de DPD

La fonction de prédistorsion est souvent représentée par des modèles mathématiques similaires aux modèles comportementaux des amplificateurs de puissance. La plupart de ces modèles sont dérivés de la série de Volterra. A temps continu, la relation générale entre un signal d'entrée $x(t)$ et la sortie $y(t)$ est donnée par :

$$y(t) = \sum_{k=1}^{+\infty} \int_0^{+\infty} \cdots \int_0^{+\infty} h_k(\tau_1, .., \tau_k) \prod_{j=1}^{k} x(t - \tau_j) d\tau_j$$

où k représente l'ordre de la non-linéarité et $h_k(\tau_1, .., \tau_k)$ représente le noyau de la série de Volterra à l'ordre k. A temps discret, la représentation de la série de Volterra doublement tronquée entre un signal d'entrée $x(n)$ et une sortie $y(n)$ est donnée par :

$$y(n) = \sum_{k=1}^{K} \sum_{i_1=0}^{M} ... \sum_{i_k=0}^{M} h_k(i_1, ..., i_k) \prod_{j=1}^{k} x(n - i_j)$$

où K représente l'ordre de non-linéarité et M est la profondeur mémoire. La série de Volterra est largement connue pour sa robustesse dans la modélisation de systèmes dynamiques non-linéaires. Son inconvénient est le fait que les coefficients de la série augmentent de façon exponentielle avec ses paramètres. Cet inconvénient rend l'utilisation de la série de Volterra trop limitée pour les applications de DPD en temps réel. Ceci motive à mettre en place des modèles basés sur la série de Volterra, mais ne retenant que quelques termes de la série. Ces modèles sont communément appelés modèles réduits ou dérivés de la série de Volterra. Les plus connus sont les polynômes à mémoire (en anglais : Memory Polynomial - MP), les polynômes à mémoire généralisés (en anglais : Generalized Memory Polynomial - GMP), les modèles dérivés de la série de Volterra dynamique, comme le DDR (en anglais : Dynamic Deviation Reduction), MDDRV (en anglais : Modified Dynamic Deviation Reduction-based Volterra - MDDRV).

La modélisation peut être effectuée globalement par un modèle unique ou bien par morceau en s'appuyant sur une segmentation de l'amplitude en différentes régions ou segments. Les modèles basés sur l'approche segmentale sont capables de représenter de fortes non-linéarités et moins sensibles aux problèmes numériques contrairement aux modèles globaux. En effet, les segments peuvent être modélisés par des modèles d'ordre de non-linéarité plus faible que pour le modèle global. *Chua* dans las années 70 a proposé un modèle par morceau appelé CPWL (en anglais : Canonical PieceWise Linear - CPWL) représentant les segments par un filtre linéaire et une valeur absolue. La relation générale d'un modèle CPWL entre une entrée $x(n)$ et une sortie $y(n)$ en bande de base est donnée par :

$$y(n) = \sum_{i=0}^{M} a_i x(n - i) + \sum_{k=1}^{K} c_{k,i} \left| \sum_{i=0}^{M} a_{k,i} x(n - i) - \beta_k \right|$$

où K est le nombre de segments, β_K est le vecteur des seuils qui délimitent les segments et M est la profondeur mémoire. Le modèle CPWL S'applique à des signaux réels. Afin que le modèle CPWL soit utilisé pour des signaux complexes comme un prédistorteur dont les coefficients sont linéaires avec le modèle, une modification et une simplification ont été apportées par *Anding Zhu*, aboutissant à un modèle dérivé d'une forme simplifiée du CPWL, appelé DVR (en anglais : Decomposed Vector Rotation).

Notre étude comparative consiste à analyser et comparer les modèles DPD qui appartiennent à deux familles différentes : l'approche globale avec un modèle dérivé des séries de Volterra et l'approche par segmentation. Ces modèles sont présentés dans le tableau 1. Le nombre de coefficients en fonction des paramètres (tels que profondeur mémoire, ordre de non-linéarité) est également donné.

Tableau 1 : Comparaison des modèles DPD étudiées

Catégorie	Modèles DPD	Relation entrée-sortie	Paramètres	Nombre de coefficients
Approche globale	MP	$y(n) = \sum_{k=1}^{K} \sum_{i=0}^{M} a_{ki} x(n-i)\|x(n-i)\|^{k-1}$	K: ordre de non-linéarité M: profondeur de mémoire	$K \times (M+1)$
	GMP	$y(n) = \sum_{k=0}^{K_a-1} \sum_{l=0}^{L_a-1} a_{kl} x(n-l)\|x(n-l)\|^k$ $+\sum_{k=1}^{K_b} \sum_{l=0}^{L_b-1} \sum_{m=1}^{M_b} b_{klm} x(n-l)\|x(n-l-m)\|^k$ $+\sum_{k=1}^{K_c} \sum_{l=0}^{L_c-1} \sum_{m=1}^{M_c} c_{klm} x(n-l)\|x(n-l+m)\|^k$	K_a, K_b, K_c: ordre de non-linéarité L_a, L_c, L_c: profondeur de mémoire M_b, M_c: longueur de retard et d'avance	$K_a \times L_a$ $+K_b \times L_b \times M_b$ $+K_c \times L_c \times M_c$
	DDR-1	$y(n) = \sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} b_{2k+1,1} \|x(n)\|^{2k} x(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,2} \|x(n)\|^{2(k-1)} x^2(n) x^*(n-i)$	K: ordre de non-linéarité M: profondeur de mémoire	$\frac{K+1}{2} + K \times M$
	DDR-2	$y(n) = \sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} b_{2k+1,1} \|x(n)\|^{2k} x(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,2} \|x(n)\|^{2(k-1)} x^2(n) x^*(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,3} \|x(n)\|^{2(k-1)} x(n) \|x(n-i)\|^2$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1,4} \|x(n)\|^{2(k-1)} x^*(n) x^2(n-i)$	K: ordre de non-linéarité M: profondeur de mémoire	$\frac{K+1}{2} \times (M+1)$ $+3M\frac{K-1}{2}$
	MDDRV-1	$y(n) = \sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} a_{2k+1,1} \|x(n)\|^{2k} x(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,2} \|x(n)\|^{2(k-1)} x^2(n) x^*(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1} \|x(n-i)\|^{2k} x(n-i)$	K: ordre de non-linéarité M: profondeur de mémoire	$\frac{K+1}{2} + (\frac{3K-1}{2}) \times M$
	MDDRV-2	$y(n) = \sum_{k=0}^{\frac{K-1}{2}} \sum_{i=0}^{M} a_{2k+1,1} \|x(n)\|^{2k} x(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,2} \|x(n)\|^{2(k-1)} x^2(n) x^*(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,3} \|x(n)\|^{2(k-1)} x(n) \|x(n-i)\|^2$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} a_{2k+1,4} \|x(n)\|^{2(k-1)} x^*(n) x^2(n-i)$ $+\sum_{k=1}^{\frac{K-1}{2}} \sum_{i=1}^{M} b_{2k+1} \|x(n-i)\|^{2k} x(n-i)$	K: ordre de non-linéarité M: profondeur de mémoire	$\frac{K+1}{2} \times (M+1)$ $+2 \times (K-1) \times M$
Approche de segmentation	DVR-2	$y(n) = \sum_{i=0}^{M} a_i x(n-i)$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1} \|\|x(n-i)\| - \beta_k\| e^{j\theta(n-i)}$	K: nombre de segments M: profondeur de mémoire	$(K+1) \times (M+1)$
	DVR-6	$y(n) = \sum_{i=0}^{M} a_i x(n-i)$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1} \|\|x(n-i)\| - \beta_k\| e^{j\theta(n-i)}$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,21} \|\|x(n-i)\| - \beta_k\| e^{j\theta(n-i)} . \|x(n)\|$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,22} \|\|x(n-i)\| - \beta_k\| . x(n)$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,23} \|\|x(n-i)\| - \beta_k\| . x(n-i)$ $+\sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,24} \|\|x(n)\| - \beta_k\| . x(n-i)$	K: nombre de segments M: profondeur de mémoire	$(1+M) \times (1+5 \times K)$

Les modèles du tableau 1 sont évalués selon des critères qui se répartissent en deux catégories. La première catégorie comprend des paramètres pour évaluer les performances de linéarisation. En plus de l'ACPR et de l'EVM, le NMSE est également utilisé comme paramètre pour refléter la convergence de l'architecture DPD dans laquelle les coefficients de chaque modèle sont identifiés. Le NMSE entre une entrée $x(n)$ et la sortie $y(n)$ est donné par :

$$\mathrm{NMSE}_{(dB)} = 10\log_{10}\frac{\sum_{n=1}^{N}|x(n)-z_p(n)|^2}{\sum_{n=1}^{N}|x(n)|^2}$$

La deuxième catégorie de comparaison met en évidence la complexité de l'implémentation matérielle du modèle DPD. Il est présenté par le nombre de coefficients du modèle et les propriétés numériques qui se réfèrent à la stabilité numérique d'identification des coefficients ainsi qu'à la dynamique des coefficients.

Les modèles de DPD sont évalués sur un PA de type Doherty avec une puissance de sortie crête de 57 dBm (500 W) et un gain linéaire de 16 dB. Ce PA est excité par un signal LTE de 20 MHz et un PAPR de 8 dB à une fréquence centrale de 2.14 GHz, et par une fréquence d'échantillonnage de 200 Ms/s.

Les paramètres du modèle GMP sont déterminés de manière optimale par l'algorithme HC [84]. Pour les autres modèles de l'approche globale, nous avons choisi de limiter l'ordre maximum de non-linéarité et la profondeur mémoire au même maximum que les valeurs du modèle GMP, soit : $K = 11$ et $M = 4$. Concernant les modèles de segmentation, nous fixons le nombre de segments K à 10 et $M = 4$ pour le modèle DVR-2, et $K = 3$ et $M = 4$ pour le modèle DVR-6.

Le tableau 2 présente une comparaison générale des modèles du tableau 1 en fonction des performances de linéarisation, obtenues par une architecture postdistorsion.

Tableau 2 : Comparaison des performances de linéarisation

Modèles de DPD	NMSE (dB)	ACPR_L (dB)	ACPR_R (dB)
Sans DPD	**-10.29**	**-23.32**	**-22.46**
MP	-32.65	-41.23	-38.90
GMP	-35.39	-43.02	-42.46
DDR-1	-29.39	-38.37	-36.26
DDR-2	-31.81	-39.81	-38.32
MDDRV-1	-29.39	-38.37	-36.26
MDDRV-2	-31.82	-39.81	-38.33
DVR-2	-32.84	-41.60	-38.95
DVR-6	-34.94	-41.88	-41.81

Le tableau 3 présente le nombre de coefficients de chaque modèle de DPD utilisés pour les résultats de linéarisation du tableau 2, leur dynamique et le conditionnement de la matrice qui intervient dans le calcul de la procédure d'identification.

Tableau 3 : Comparaison des propriétés numériques

Modèles DPD	Nombre de coefficients	Conditionnement	dynamique des coefficients
MP	55	$1.12e^{27}$	$2.18e^{11}$
GMP	43	$5.24e^{23}$	$1.75e^{10}$
DDR-1	50	$7.52e^{22}$	$4.59e^{8}$
DDR-2	90	$3.16e^{24}$	$2.53e^{9}$
MDDRV-1	70	$7.54e^{22}$	$4.83e^{8}$
MDDRV-2	110	$3.72e^{24}$	$2.53e^{9}$
DVR-2	55	$5.40e^{7}$	1
DVR-6	71	$2.70e^{9}$	11.65

Selon les tableaux 2 et 3, le modèle GMP atteint les meilleures performances de linéarisation à 43 coefficients. Le modèle DVR-6 avec 3 segments obtient des performances similaires, mais avec presque le double du nombre de coefficients. Cependant, le modèle DVR-6 présente des propriétés numériques impressionnantes qui en font un candidat très intéressant pour l'implémentation matérielle.

Afin de valider l'efficacité de cette étude comparative, les modèles GMP et DVR-6 seront spécifiquement comparés dans différents scénarios en utilisant un PA de technologie LDMOS 50 W avec une fréquence centrale de 1.78 GHz et une fréquence d'échantillonnage de 200 Ms/s.

Les tests sont effectués avec des signaux LTE avec une largeur de bande de 5, 10 et 20 MHz et un PAPR maximum de 12 dB, et à 4 niveaux de puissance de sortie : 34, 38, 42 et 44 dBm. Le tableau 4 présente les performances de linéarisation des modèles GMP et DVR-6 pour le même nombre de coefficients (39 coefficients).

Tableau 4 : Performances de linéarisation des modèles GMP et DVR-6

Largeur de bande	NMSE (dB)		$ACPR_L$ (dB)		$ACPR_R$ (dB)	
	DVR-6	GMP	DVR-6	GMP	DVR-6	GMP
5 MHz	-39.66	-39.75	-42.43	-42.53	-40.81	-40.84
10 MHz	-40.31	-40.76	-44.78	-44.48	-43.43	-43.57
20 MHz	-38.82	-39.41	-49.38	-49.76	-48.80	-49.94

Le tableau 5 présente les propriétés numériques utilisées par les deux modèles pour obtenir les résultats du tableau 4. Les propriétés numériques sont présentées par la dynamique des coefficients et le conditionnement de la matrice impliquée dans la procédure d'identification.

Tableau 5 : Propriétés numériques des modèles GMP et DVR-6

Bandwidth	$\log_{10}$(Conditionnement)		Dynamique des coefficients	
	DVR-6	GMP	DVR-6	GMP
5 MHz	13	17	109	34887
10 MHz	14	18	327	19700
20 MHz	13	18	307	16278

En comparant les performances des deux modèles pour chaque largeur de bande, les résultats du tableau 4 confirment que le modèle GMP offre une capacité de modélisation légèrement meilleure que le modèle DVR-6 pour les largeurs de 5 et 20 MHz.

Pour 10 MHz, le modèle DVR-6 a de meilleures performances de NMSE, mais moins bonnes en termes d'ACPR. Selon le tableau 5, le conditionnement de la matrice d'identification

du DVR-6 est plus petit que celui du GMP. La dynamique des coefficients du modèle DVR-6 est bien inférieure à celle du GMP. Par conséquent, le modèle DVR-6 présente un bon compromis entre les performances de linéarisation et les propriétés numériques, ce qui en fait un bon candidat pour une implémentation matérielle de la DPD.

Le tableau 6 montre les performances de linéarisation et les propriétés numériques pour chaque niveau de puissance pour une largeur de bande de 20 MHz.

Tableau 6 : Comparaison des modèles GMP et DVR-6 pour chaque puissance de sortie

Puissance de sortie	34 dBm		38 dBm		42 dBm		44 dBm	
	GMP	DVR-6	GMP	DVR-6	GMP	DVR-6	GMP	DVR-6
NMSE (dB)	-32.58	-32.34	-36.28	-35.45	-36.34	-35.36	-39.25	-37.73
ACPR_L (dB)	-50.41	-49.93	-50.44	-49.43	-46.77	-45.70	-48.99	-47.46
ACPR_R (dB)	-49.38	-49.27	-49.85	-49.11	-46.03	-45.52	-48.35	-47.86
Conditionnement	$9.38e^{9}$	$5.15e^{11}$	$7.91e^{11}$	$8.63e^{10}$	$2.54e^{13}$	$1.94e^{10}$	$5.86e^{17}$	$8.47e^{10}$
Dynamique des coeff.	8	42	75	31	474	8	$1.18e^{5}$	104

Le modèle GMP fonctionne mieux que le DVR-6 lorsque le PA opère dans des zones de non-linéarités faibles. D'autre part, le modèle DVR-6 confirme ses propriétés numériques pour différentes largeurs de bande et différents niveaux de puissances de sortie.

En conclusion de cette étude comparative, le modèle GMP offre un bon compromis entre les performances de précision et la complexité. Le modèle DVR peut atteindre des performances similaires, mais avec une augmentation du nombre de coefficients. Cependant, le modèle DVR possède des propriétés numériques impressionnantes qui en font un bon choix pour l'implémentation matérielle. Pour cela, la contribution suivante de cette thèse se concentrera autour de la conception et de l'optimisation du modèle DVR, en déterminant sa structure et en proposant une architecture d'implémentation.

Optimisation du modèle DVR

Les modèles dérivés de la série de Volterra tels que MP, GMP ou DDR ont prouvé leur efficacité comme modèle de prédistorsion. Cependant, pour un ordre élevé de non-linéarité, la matrice de régression intervenant dans l'identification des coefficients est très mal conditionnée ce qui peut générer des instabilités numériques si nous n'utilisons pas de méthodes de régularisation et la dynamique des coefficients peut être élevée.

L'évolution des architectures de PA avec une efficacité énergétique élevée, telle que Doherty et suiveur d'enveloppe (en anglais : Envolope Traking - ET), conduit à des PA souvent peu linéaires. En outre, les besoins de débits de données plus élevés conduisent à utiliser des signaux modulés avec un PAPR élevé, ce qui stimule les non-linéarités de PA. De plus, les systèmes 5G et MIMO permettent un débit de données excessivement élevé, ce qui est un véritable défi pour la DPD en termes de largeur de bande, de non-linéarités et de comportements dynamiques. Compte tenu de tous ces faits, il devient plus difficile de concevoir un système de DPD efficace pour obtenir une caractéristique inverse précise avec une bonne efficacité et une faible complexité d'identification. Cela conduit à la recherche d'une approche alternative pour la DPD. L'une de ces approches est l'approche de segmentation, où le signal d'entrée est divisé en plusieurs segments séparés par des seuils. L'un de ces modèles basés sur l'approche de segmentation est le modèle DVR qui a fait le sujet de l'étude comparative. L'étude du modèle DVR repose généralement sur deux aspects : la manière de positionner les seuils et la détermination de la structure optimale.

Comme indiqué précédemment, le modèle DVR proposé initialement par *Anding Zhu*

est défini par relation entrée-sortie :

$$y(n) = \sum_{i=0}^{M_{lin}} a_i x(n-i) + \sum_{T_t \in S} T_t$$

où $x(n)$ et $y(n)$ représentent l'entrée et la sortie, respectivement, M_{lin} est la profondeur de mémoire du terme linéaire, a_i est le vecteur des coefficients et S est l'ensemble de termes T_t utilisées dans le modèle DVR avec $S \subset \mathbf{T} = [T_{1,[0,...,P]}, T_2, T_3, T_4, T_5, T_6, T_7]$.

$$\begin{aligned}
T_{1,p} &= \sum_{k=1}^{K} \sum_{i=0}^{M} c_{ki,1p} ||x(n-i)| - \beta_k| e^{j\theta(n-i)} \cdot |x(n)|^p \\
T_2 &= \sum_{k=1}^{K} \sum_{i=i_2}^{M} c_{ki,2} ||x(n-i)| - \beta_k| \cdot x(n) \\
T_3 &= \sum_{k=1}^{K} \sum_{i=i_3}^{M} c_{ki,3} ||x(n-i)| - \beta_k| \cdot x(n-i) \\
T_4 &= \sum_{k=1}^{K} \sum_{i=i_4}^{M} c_{ki,4} ||x(n)| - \beta_k| \cdot x(n-i) \\
T_5 &= \sum_{k=1}^{K} \sum_{i=i_5}^{M} c_{ki,5} ||x(n-i)| - \beta_k| \cdot x^2(n) \cdot x^*(n-i) \\
T_6 &= \sum_{k=1}^{K} \sum_{i=i_6}^{M} c_{ki,6} ||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \\
T_7 &= \sum_{k=1}^{K} \sum_{i=i_7}^{M} c_{ki,7} ||x(n-i)| - \beta_k| \cdot x^*(n) \cdot x^2(n-i)
\end{aligned}$$

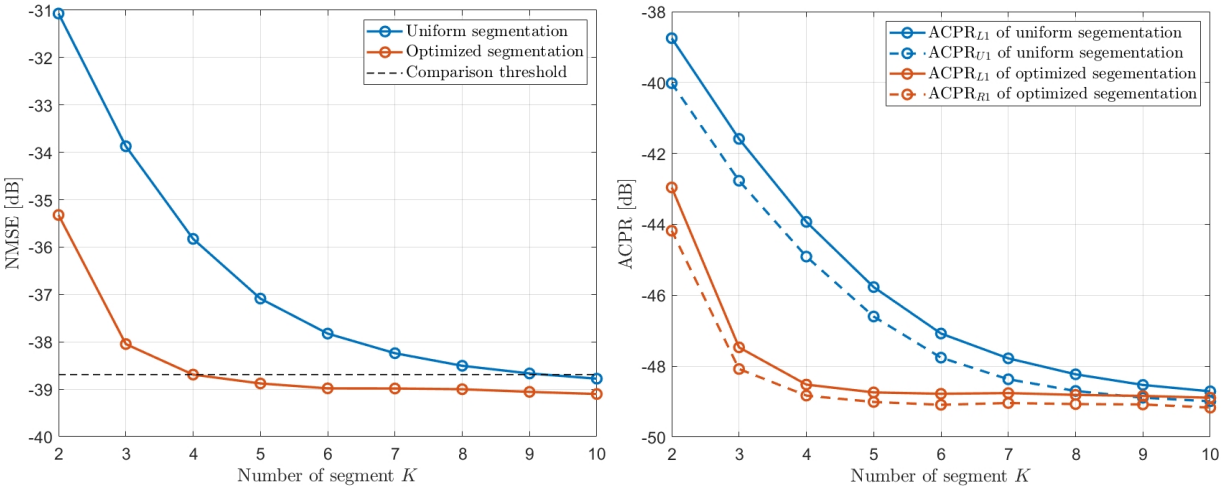
où K est le nombre de segments, M est la profondeur de mémoire, β est le vecteur des seuils séparant les segments et c_{ki} est le vecteur des coefficients du modèle DVR. Le cardinal de $\mathbf{T}$ est $P+7$. Les indices $i_2, i_3, ..., i_7$ sont égaux à 0 ou 1 selon l'activation du terme T_i.

L'étude du modèle DVR est validée par des acquisitions expérimentales des signaux entrée-sortie sur un PA de type LDMOS 50 W, stimulé par un signal LTE de largeur de bande de 20 MHz avec un PAPR de 12 dB à une puissance de sortie de 44 dBm. 64000 échantillons de donnée I/Q sont utilisés pour l'algorithme d'identification de DPD.

Le vecteur des seuils $\boldsymbol{\beta}_K$ dans la représentation du modèle DVR définit le positionnement des segments. Les seuils peuvent être sélectionnés d'une façon uniforme $\beta_k = k/K$ pour $k = 1; 2; ..; K-1$ ou optimale. Dans la littérature, beaucoup de travaux considèrent des modèles à segmentation uniforme, peu d'entre eux traitent l'optimisation de la segmentation. Le premier travail sur l'espacement optimal a été mené pour la DPD en LUT (en anglais : LookUp Table) sans mémoire [67]. Dans [68], une procédure itérative est proposée pour optimiser l'espacement non-uniforme de la LUT. Pour les modèles de segmentation, l'avantage d'une segmentation optimale par rapport à une segmentation uniforme a été établi dans [69] où les auteurs suggèrent de réduire la complexité de l'algorithme en considérant la version sans mémoire du modèle DVR. Les auteurs ont utilisé des méthodes basées sur Gauss-Newton telles que Levenberg-Marquardt [70] pour optimiser conjointement les seuils et identifier les coefficients du modèle. Dans [71], les auteurs ont utilisé une approche d'optimisation globale basée sur l'algorithme génétique (en anglais : Genetic Algorithm - GA) pour optimiser les seuils du modèle CPWL pour la radio sur fibre (en anglais : Radio-over-Fiber - ROF). L'ACPR est utilisé comme facteur de mérite pour le GA.

Dans cette étude d'optimisation du modèle DVR, une nouvelle approche a été proposée pour optimiser les seuils β. Le principe de cette approche consiste à décomposer le problème d'optimisation global en sous-problèmes de minimisation unidirectionnelle basée sur la section dorée (en anglais : Golden Section - GS). Pour une fonction unimodale donnée $f(x)$, le principe de GS consiste à trouver l'optimum x_{opt} qui correspond à la valeur minimale de $f(x)$, en rétrécissant itérativement l'intervalle de recherche $[a\ b]$ contenant cet optimum jusqu'à ce qu'une précision spécifiée soit atteinte. La mise à jour de cet intervalle se fait itérativement selon un facteur appelé le nombre d'or $\tau = \frac{1+\sqrt{5}}{2}$ qui vérifie $\tau - 1 = \frac{1}{\tau}$.

Le principe de l'approche proposée est basé sur la recherche GS. Pour un nombre de segments K donné, le nombre de seuils $\boldsymbol{\beta}_K$ est $K-1$: $\boldsymbol{\beta} = \beta_1, \beta_2, ..., \beta_{K-1}$. L'approche proposée a été validée sur le modèle DVR avec $T_S = [T_{1,0}\ T_{1,1}\ T_2\ T_3\ T_4]$, $M_{lin} = 2$, et $M = 1$. La figure 3 présente les avantages de la segmentation optimisée en comparant avec la segmentation uniforme en termes de performance de linéarisation (NMSE et ACPR) pour différents nombres de segments K.



(a) NMSE en fonction de K

(b) ACPR en fonction de K

Figure 3 : Comparaison de la segmentation uniforme et la segmentation optimisée en fonction des performances de linearisation

D'après la figure 3.a, les performances de linéarisation en termes de NMSE pour 8 segments utilisant une segmentation uniforme pourraient être atteintes avec seulement 3 segments pour une segmentation optimale. Ceci est confirmé dans la figure 3.b en termes d'ACPR, où la segmentation optimale ne nécessite que 4 segments pour obtenir les mêmes performances de linéarisation que la segmentation uniforme avec 10 segments. Le NMSE et l'ACPR du modèle DVR avec une segmentation optimale convergent vers une asymptote à partir de 4 segments, tandis qu'une segmentation uniforme nécessite plus de 10 segments pour obtenir les meilleures performances de linéarisation. Ainsi, la segmentation optimale réduit considérablement la complexité puisque le nombre de coefficients à estimer est réduit. Pour le scénario considéré, seuls 43 coefficients sont nécessaires avec la segmentation optimisée pour obtenir les mêmes performances de linéarisation qu'une segmentation uniforme avec 93 coefficients. Ceci confirme l'intérêt d'optimiser les seuils du modèle DVR.

L'approche proposée est comparée avec le GA et une recherche exhaustive de type force brute. Les résultats de l'approche proposée sont évalués avec K à 4, $M_{lin} = 2$ et $M = 1$ et comparés avec la segmentation uniforme et la segmentation optimale obtenue par force brute et le GA. Les résultats sont résumés dans le tableau 7 et discutés en termes de NMSE et ACPR. La complexité est également discutée et évaluée par le nombre de

calculs de NMSE.

Tableau 7 : Comparaison des performances de linéarisation, des solutions optimales et de la complexité

		Segmentation uniforme	Force brute	GA	Approche proposée
Segmentation		[0.25 0.5 0.75]	[0.58 0.81 0.92]	[0.59 0.81 0.92]	[0.59 0.81 0.92]
NMSE (dB)		-35.82	-38.69	-38.69	-38.69
ACPR(dB)	L1	-43.93	-48.52	-48.51	-48.49
	U1	-44.91	-48.83	-48.83	-48.82
Nombre de calculs de NMSE		1	161700	2404	341

Dans ce scénario, l'approche proposée a convergé vers la même segmentation optimale que la force brute et la recherche GA, c'est-à-dire $\boldsymbol{\beta}_{opt} = [0.59\ 0.81\ 0.92]$. Selon le tableau 7, seules 341 évaluations de fonctions objectif sont effectuées pour optimiser les seuils, tandis que GA nécessite 2404 évaluations et la force brute nécessite 161700 évaluations pour trouver les mêmes résultats optimaux. Quant à la segmentation uniforme, le NMSE de l'approche proposée a été considérablement amélioré de près de 4 dB. L'ACPR est également amélioré de près de 5 dB. Ceci est confirmé dans la figure 3 pour $K = 4$.

Dans ce cadre, nous avons montré que la segmentation optimale améliore les performances de linéarisation par rapport à la segmentation uniforme pour le modèle DVR avec le même nombre de coefficients. Nous avons proposé une approche de détermination des seuils optimaux avec une complexité réduite par rapport à l'approche GA et à l'approche force brute. La détermination des seuils optimaux a été effectué pour un nombre segments K fixé à priori. Une autre question porte sur la détermination de la valeur optimale de K et des autres paramètres de structure du modèle DVR. Cette question fait l'objet de la section suivante.

Dimensionnement du modèle DVR

Le dimensionnement du modèle DVR consiste à trouver une structure de modèle optimal qui conduit au meilleur compromis entre précision de modélisation et complexité réduite. La structure du modèle DVR est définie par le nombre de segments K, la profondeur de mémoire du terme linéaire M_{lin}, la profondeur de la mémoire M, l'ordre maximum de la non-linéarité P, et le nombre de termes T_t. La structure du modèle DVR peut être déterminée à l'aide de recherches exhaustives en explorant toutes les combinaisons de valeurs de paramètres possibles. En fixant l'ordre maximum de K et M à K_{max} et M_{max}, respectivement, la recherche exhaustive nécessite $K_{\max} \cdot M_{\max} \cdot L$ tests, où $L = \sum_{k=1}^{P+7} \binom{P+7}{k} = 2^{P+7} - 1$ présente le nombre total de combinaisons possibles de termes du modèle. Limiter le nombre maximum de coefficients à tester pourrait réduire considérablement l'espace de recherche. Cependant, cette approche de la recherche exhaustive reste très lourde en calcul.

Dans la littérature, un des premiers travaux dédié au dimensionnement de la structure optimale pour les modèles de DPD a été fait pour le modèle GMP en utilisant un algorithme de type GA [54]. Dans cet algorithme, la fonction de coût représente une combinaison linéaire du NMSE et du nombre de coefficients. Dans [57], les auteurs ont proposé un algorithme heuristique HC pour trouver la structure optimale du modèle GMP avec une complexité de calcul réduite. Dans [81], une étude comparative de deux algorithmes d'optimisation, l'algorithme HC et le GA, a été réalisée en termes de vitesse de convergence et de performance de linéarisation de la solution obtenue. Les deux algorithmes

ont été comparés sur le même critère de recherche et obtiennent des performances très proches avec une complexité de calcul très inférieure à une recherche exhaustive.

Dans cette étude, l'approche proposée dans [84] est étendue pour le modèle DVR pour trouver une structure optimale avec une complexité réduite et de meilleures performances de linéarisation. Le HC est un algorithme d'optimisation heuristique qui part d'une solution initiale et essaie par la suite de trouver une meilleure solution en comparant la solution actuelle avec ses voisines. S'il existe une meilleure solution parmi les voisins, elle est considérée comme la nouvelle solution. La procédure de recherche est répétée jusqu'à ce qu'aucune meilleure solution ne puisse être trouvée. La définition du voisinage est un processus nécessaire pour pouvoir mettre en oeuvre l'algorithme HC.

La fonction de coût est un aspect essentiel de l'algorithme HC qui conduit à un compromis entre les performances de linéarisation et la complexité du modèle. Le NMSE, notée par N, est utilisé pour évaluer les performances de linéarisation. La complexité du modèle est représentée par le nombre de coefficients, noté par C. La fonction de coût, notée par f, est définie en combinant N et C en un seul critère par une combinaison additive

$$f = N + \alpha.C$$

où α est un paramètre positif appelé le coefficient de pondération.

Une nouvelle définition du voisinage est proposée pour le modèle DVR. Pour le modèle DVR, il y a quatre paramètres à dimensionner : le nombre de segments K, la profondeur de mémoire M_{lin} pour le terme linéaire, la profondeur de mémoire M pour l'ensemble T_S, et les termes qui seront choisis dans l'ensemble $\mathbf{T}$.

Dans un espace discret U, l'élément x_i qui représente une structure de modèle DVR est caractérisé par 3 entiers : K_i, $M_{lin,i}$, M_i et par T_{S_i} qui est une k combinaison de l'ensemble $\mathbf{T}$ avec $k = \{1, ..., P+7\}$. Comme ces paramètres peuvent être modifiés indépendamment, ils composent un espace discret à 4 dimensions. Le voisin de l'élément x_i est défini comme un nœud $(K_i + \delta_1,\ M_{lin,i} + \delta_2,\ M_i + \delta_3,\ \mathcal{T}_{U,i} + \delta_4)$, où $\delta_{1,2,3} \in [0, \pm 1]$. Le processus impliqué derrière δ_4 correspond soit à ajouter à T_{S_i} un terme de son complément relatif dans $\mathbf{T}$, soit à retirer un terme de T_{S_i}, soit à laisser inchangé T_{S_i}.

La complexité de l'algorithme HC est caractérisé par quatre facteurs :

- Le nombre total d'éléments (voisins) évalués par l'algorithme HC, qui correspond automatiquement au nombre d'évaluations f.
- Le nombre total d'itérations nécessaires pour exécuter l'algorithme HC, où une itération HC est effectuée lorsque les fonctions de coût des voisins d'une solution initiale sont calculées et la solution optimale est déterminée.
- Le temps d'exécution de l'algorithme HC est également pris en compte comme indicateur relatif de la complexité de l'algorithme HC.
- Le nombre de coefficients du modèle de la structure de chaque voisin est introduit, car il reflète la complexité calculatoire du processus d'identification. Cela fait également référence à la complexité du modèle, qui est un aspect critique pour le dimensionnement du modèle DVR.

Ces critères sont des indicateurs pour évaluer la complexité de l'algorithme HC et les propriétés numériques des modèles évalués au cours de l'algorithme HC. La complexité de l'algorithme HC peut être réduite en élaguant les voisinages en ajoutant des contraintes sur le nombre de coefficients.

L'algorithme HC a été validé par des mesures expérimentales de PA avec un signal LTE de largeur de bande de 20 MHz. La figure 5 présente les résultats de l'algorithme HC. Les résultats de la recherche exhaustive obtenus en explorant toutes les structures possibles

sont tracées en point bleu en fonction de N et C. Les points en violet représentent les structures évaluées par l'algorithme HC. Les points rouges représentent les meilleures structures trouvées à chaque itération. Le nombre maximum de coefficients que l'algorithme HC ne doit pas dépasser est fixé à 50 coefficients.

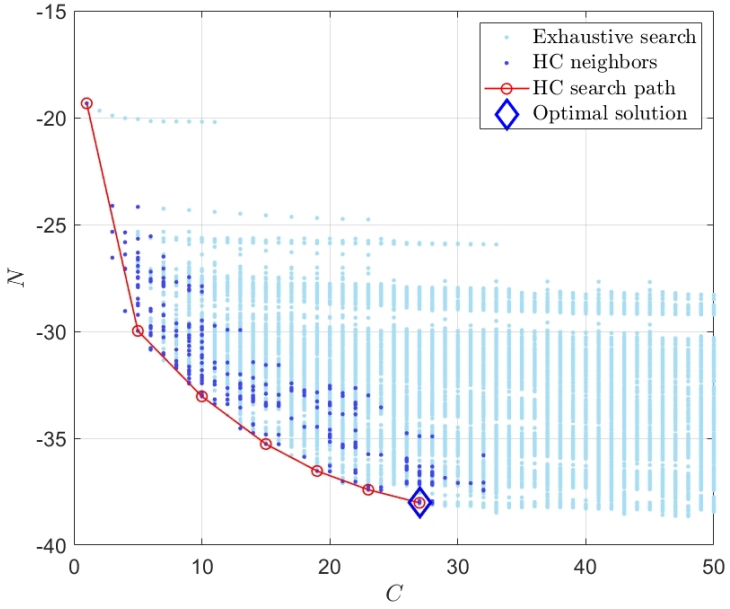


Figure 5 : Algorithme HC avec $\alpha = 0.05$

Dans ce scénario, l'algorithme HC évalue 399 éléments (structures) en 1.18 minute et sur 6 itérations, tandis que la recherche exhaustive a évalué 24357 éléments en 19 heures. Les paramètres de la structure optimale sont : $K = 4$, $M_{lin} = 2$, $M = 1$, et $T_S = [T_{1,0}\ T_3\ T_6]$.

Le modèle DVR optimal à 27 coefficients avec un NMSE de -38.06 dB. L'algorithme HC a été mis en œuvre ici avec une segmentation uniforme, e.g., $\beta = [0.25\ 0.5\ 0.75]$. En remplaçant ces paramètres dans la relation générale du modèle DVR, le modèle optimal s'exprime par :

$$\begin{aligned} y(n) = &\sum_{i=0}^{2} a_i x(n-i) \\ &+ \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,10}||x(n-i)| - \beta_k| e^{j\theta(n-i)} \\ &+ \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,3}||x(n-i)| - \beta_k| \cdot x(n-i) \\ &+ \sum_{k=1}^{4}\sum_{i=0}^{1} c_{ki,6}||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \end{aligned}$$

Dans la figure 5, les résultats sont obtenus avec un ordre maximal de non-linéarité P configuré à 1. La présence de ce terme dans le modèle DVR est importante afin de représenter de fortes non-linéarités. Pourtant, l'ordre de non-linéarité dans le modèle DVR impacte ses performances, notamment en terme des propriétés numériques.

Le dimensionnement du modèle DVR est une des contributions majeures de cette thèse où l'algorithme HC proposé pour le modèle GMP a été étendu au modèle DVR pour trouver la structure optimale du modèle DVR avec un temps d'exécution réduit. Deux aspects essentiels sont mis en évidence dans ce cadre : la nouvelle définition du voisinage que nous proposons pour l'algorithme HC au modèle DVR et l'utilisation d'une fonction

de coût qui permet d'obtenir le meilleur compromis entre précision de modélisation et complexité.

Approche système pour la conception d'un modèle DVR optimal

La conception d'un modèle DVR optimal consiste à déterminer sa structure optimale avec une segmentation optimale selon les mêmes critères que ceux utilisés précédemment. Fondamentalement, ce processus présente une combinaison de l'algorithme HC et de l'approche proposée pour optimiser les seuils. Dans cette étude, nous proposons des approches système (SysApp) pour effectuer cette optimisation conjointe afin de concevoir le modèle DVR optimal. Ces SysApps sont développées, discutées et comparées à l'algorithme HC en terme de solutions optimales, de performances de linéarisation et de complexité.

Les SysApp pour concevoir le modèle DVR optimal sont développées à travers les différentes versions suivantes :

- Approche directe qui peut se voir comme l'approche la plus intuitive pour concevoir le modèle DVR optimal. Cette approche est basée sur la combinaison de l'approche proposée basée sur GS pour optimiser les seuils $\boldsymbol{\beta}$ et l'algorithme HC pour dimensionner le modèle DVR. En effet, l'optimisation de $\boldsymbol{\beta}_k$ est effectuée pour chaque élément (structure) évalué par l'algorithme HC. La fonction de coût f est calculée en fonction de C et N de chaque modèle en tenant compte de sa segmentation optimisée $\boldsymbol{\beta}_{opt}$. Cette approche est indiquée par D-SysApp (en anglais : Direct System Approach). À première vue, D-SysApp requiert une complexité de calcul importante puisqu'elle sera principalement évaluée par le nombre total de calculs de NMSE nécessaire par l'algorithme HC et la recherche GS.

- D-SysApp peut être simplifiée en terme de complexité en optimisant les seuils uniquement pour des structures de modèle DVR spécifiques. Ces structures sont déterminées selon 2 approches :

  - Les structures qui se trouvent sur l'enveloppe de l'algorithme HC. Cette approche est indiquée par Senv-SysApp (en anglais : Simplified approach on Envelope results). Son principe consiste à appliquer le processus d'optimisation des seuils par le GS une fois que l'algorithme HC est terminé. La complexité de Senv-SysApp est principalement évaluée par la complexité de l'algorithme HC, avec la complexité de la recherche GS multipliée par le nombre de structures trouvées sur l'enveloppe.
  - Les structures qui sont les meilleures structures (solutions) à la fin de chaque iteration de l'algorithme HC. Cette approche est indiquée par Sitr-SysApp (en anglais : Simplified approach on Iteration results). Sitr-SysApp consiste à insérer la recherche GS dans l'algorithme HC, qui sera activé une fois l'itération HC terminée et appliquée au meilleur voisin de cette itération HC. Cela permet de réduire la complexité qui sera principalement évaluée par celle de l'algorithme de HC.

- La motivation derrière l'approche conjointe optimale, indiquée par O-SysApp (en anglais : Optimal System Approach), est d'effectuer l'algorithme HC pour dimensionner le modèle DVR sur un ensemble de voisins avec une segmentation optimale, c'est-à-dire que la segmentation optimale des modèles évalués par le HC doit être

disponible et stockée dans une LUT. La taille de la LUT dépend du nombre de fois que le processus d'optimisation de $\boldsymbol{\beta}$ a été exécuté. Le processus impliqué derrière la LUT pour stocker les seuils optimaux est effectué en conjonction avec l'algorithme HC, ce qui en fait une approche conjointe. Bien que le cas parfait soit celui où la segmentation optimale de toutes les structures est disponible et stockée, cela ne peut pas être fait puisqu'il nécessite une recherche exhaustive, ce qui conduit à D-SysApp. O-SysApp est développé à travers deux versions :

- Une première version d'approche système appelée Ored-SysApp (en anglais : Optimal approach on Reduced parameters) qui considère que la segmentation optimale n'est pas sensible à la variation des paramètres M et T_S. Autrement dit, les seuils $\boldsymbol{\beta}$ ne sont optimisés que lorsque le nombre de segments K change lors de l'algorithme HC. La segmentation optimale correspondant à chaque K est enregistrée dans la LUT. La segmentation optimale pour chaque K est stockée dans la LUT. La taille de la LUT dépend du nombre maximum de segments explorés par l'algorithme HC. Une segmentation $\boldsymbol{\beta}_{opt}^{(K)}$ optimale pour un K donné couvre toutes les structures qui ont le même K, quels que soient les autres paramètres (M, T_S).
- Une deuxième version appelée Ogen-SysApp (en anglais : Optimal approach on Generalized parameters) dans laquelle les seuils $\boldsymbol{\beta}$ sont optimisés lorsque le nombre de segments K ou les termes du modèle T_S changent par l'algorithme HC. Ogen-SysApp représente une version améliorée d'Ored-SysApp, où l'algorithme est similaire, sauf que lorsque les termes du modèle changent, le processus d'optimisation des seuils est appelé.

L'efficacité des SysApp proposées est validée en utilisant les mêmes conditions expérimentales que celles rapportées pour le dimensionnement et l'optimisation des seuils.

La figure 6 compare le comportement de chaque SysApp avec l'algorithme HC avec une segmentation uniforme, où Ored-SysApp est nettement sous-optimal de 10 à 30 coefficients, tandis que Senv-SysApp, Sitr-SysApp et Ogen-SysApp sont presque optimaux à partir de 15 coefficients. En revanche, l'approche directe (D-SysApp) est optimale à partir de 10 coefficients. Les résultats de la figure 6 sont obtenus avec $\alpha = 0.05$.

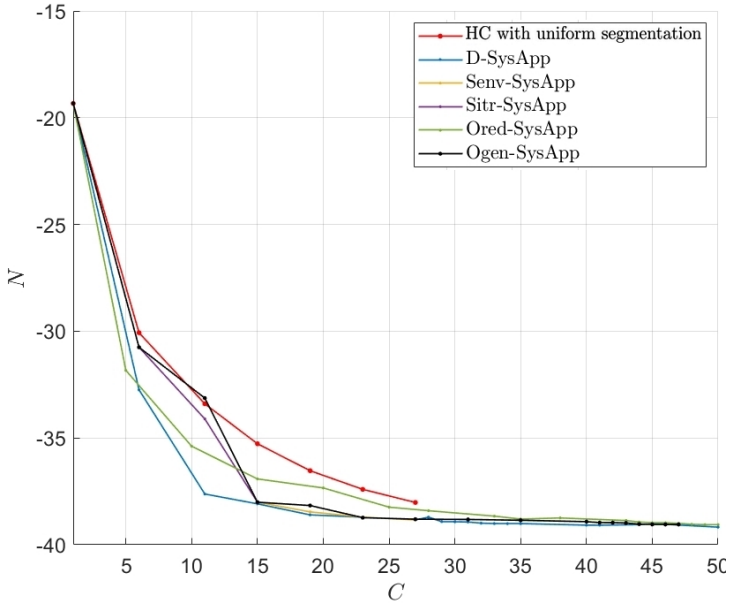


Figure 6 : Comparaison du comportement de différentes SysApps

Le tableau 8 résume les solutions optimales obtenues par les différentes SysApp pour concevoir le modèle DVR.

Tableau 8 : Comparaison des solutions optimales des SysAPp

		Algorithme HC	D-SysApp	Senv-SysApp	Sitr-SysApp	Ored-SysApp	Ogen-SysApp
Structure de modèle	K	4	4	4	3	4	5
	β_{opt}	[0.55 0.78 0.92]	[0.59 0.82 0.93]	[0.55 0.78 0.92]	[0.26 0.88]	[0.66 0.83 0.93]	[0.11 0.67 0.84 0.94]
	M_{lin}	2	2	2	2	2	2
	M	1	1	1	1	1	1
	T_S	$T_{1,0}$ T_3 T_6	T_2 T_6	$T_{1,0}$ T_3 T_6	T_3 T_6	$T_{1,0}$ $T_{1,1}$ T_4 T_5	T_3 T_6
C		27	19	27	15	35	23
N		-38.02	-38.6	-38.85	-38.01	-38.80	-38.93
f		-36.32	-37.65	-37.5	-37.26	-37.05	-37.78

Selon le tableau 8, les solutions optimales ont la même structure de modèle en termes de profondeurs de mémoires (M_{lin} et M), tandis que le nombre de segments K et les termes du modèle T_S sont différents d'une solution à l'autre. Par ailleurs, les solutions optimales disposent du même niveau de performance de linéarisation, mais avec une complexité différente. En calculant la fonction de coût avec $\alpha = 0.05$, la meilleure solution optimale est celle renvoyée par Ogen-SysApp suivi par D-SysApp, ce qui est tout à fait raisonnable puisqu'elle est basée sur une intégration totale de la recherche GS au sein de l'algorithme HC .

La complexité des SysApp proposée pour concevoir le modèle DVR est évaluée de la même manière que la réduction de la complexité de l'algorithme HC décrite dans la section 3.5.5. Le tableau 9 présente la complexité de chaque SysApp, qui est évaluée par le nombre total de modèles évalués, les itérations HC, le nombre total de calculs de NMSE incluant ceux du processus d'optimisation des seuils, le temps d'exécution et la somme des nombres de coefficients de tous les modèles identifiés

Tableau 9 : Comparaison de la complexité des approches système

	D-SysApp	Senv-SysApp	Sitr-SysApp	Ored-SysApp	Ogen-SysApp
Modèles DVR évalués	892	399	256	750	506
Itérations de HC	21	6	3	17	16
Nombre de calculs de NMSE	51979	910	356	1100	3866
Temps d'exécution (min)	294	2.1	0.6	4.9	8.54
Nombre de coefficients	74511	5952	2575	19185	28537

Selon le tableau 9, le comportement de chaque SysApp impacte la complexité de sa convergence vers la solution optimale. Le temps d'exécution de D-SysApp est immense avec un nombre élevé de calculs NMSE, par conséquent, il ne peut pas être un candidat réaliste de SysApp pour concevoir le modèle DVR. Senv-SysApp et Sitr-SysApp restent des candidats acceptables en termes de temps d'exécution et de nombre de calculs de NMSE. Ored-SysApp et Ogen-SysApp ont une complexité plus grande que Sens-SysApp et Sitr-SysApp.

D'après les résultats du tableau 8 et 9, et puisque D-SysApp est exclu de notre choix pour sa complexité de calcul, nous pouvons conclure que Ogen-SysApp représente un excellent candidat pour concevoir le modèle DVR, car il présente un bon compromis entre les performances de linéarisation et la complexité de calcul. Il faut aussi noter que Sitr-SysApp présente des performances de linéarisation similaires à Ogen-SysApp avec une complexité réduite (15 coefficients de modèle) et avec les mêmes termes de modèle.

Stratégie d'implémentation matérielle du modèle DVR Optimal

Une stratégie d'implémentation matérielle sur un FPGA est discutée et investiguée dans cette partie, où les propriétés numériques du modèle DVR optimal sont soulignées. Selon la littérature sur cet axe de recherche, de nombreux travaux ont été rapportés pour implémenter les modèles de DPD. Dans [89], un design FPGA est proposé pour implémenter un modèle de DPD avec mémoire. Les auteurs ont proposé dans [90] une méthodologie avec une synthèse de haut niveau pour les dispositifs FPGA afin d'implémenter des modèles basés sur la série de Volterra. Une combinaison de processeurs FPGA et ARM est proposée dans [92] pour implémenter le modèle MP, où les auteurs ont utilisé le processeur ARM pour son avantage dans les calculs en virgule flottante. Concernant le modèle DVR, peu de travaux de recherche ont été rapportés dans la littérature consacrée à l'implémentation matérielle. Dans [66], les auteurs ont proposé une structure matérielle en décomposant le modèle DVR en un format sous-composé pour réduire la complexité de mise en œuvre. Cependant, les paramètres du modèle DVR ont été fixés arbitrairement sans évoquer aucune préoccupation quant à la réduction de la complexité du modèle. Dans [94], un modèle dérivé du modèle DVR est proposé, où l'implémentation matérielle a été soulignée.

Il y a deux points critiques à considérer lors de la conversion de logiciel en matériel : la précision et la vitesse. Dans la mise en œuvre matérielle de DPD, il existe principalement deux processus :

- L'application de DPD au signal d'entrée qui doit être échantillonnée à une fréquence d'échantillonnage suffisante pour répondre à la demande croissante de signaux à large bande dans les systèmes de télécommunication actuels tels que la 5G.
- Identifier et mettre à jour les coefficients de DPD, ce qui peut être fait en utilisant des approches bloc par bloc ou échantillon par échantillon.

Généralement, il existe deux structures pour implémenter le modèle de DPD : la méthode par LUT et la structure directe avec multiplieurs et additionneurs. La méthode par LUT stocke la valeur pré-calculée de la fonction de DPD, qui est indexée par le module du signal d'entrée quantifié. D'autre part, la structure directe avec multiplieurs et additionneurs, appelée méthode directe, nécessite des ressources logiques de FPGA pour effectuer l'implémentation d'une fonction de DPD. L'objectif de cette étude est de proposer une stratégie matérielle pour implémenter le modèle DVR optimal conçu par l'approche conjointe.

Dans cette étude, nous adoptons le modèle DVR optimal retourné par Ogen-SysApp qui s'exprime selon le tableau 8 par :

$$\begin{aligned} y(n) = & \sum_{i=0}^{2} a_i x(n-i) \\ & + \sum_{k=1}^{5} \sum_{i=0}^{1} c_{ki,3} ||x(n-i)| - \beta_k| \cdot x(n-i) \\ & + \sum_{k=1}^{5} \sum_{i=0}^{1} c_{ki,6} ||x(n-i)| - \beta_k| \cdot x(n) \cdot |x(n-i)|^2 \end{aligned}$$

Selon la méthode directe, l'implémentation de ce modèle optimal nécessite 42 additionneurs et 33 multiplieurs.

Une implémentation par LUT peut être réalisée en utilisant uniquement 7 additionneurs, 7 multiplieurs, et 2 LUT de 4096 cases et de 12 bits. La figure 7 présente l'implantation du modèle DVR optimal par LUT.

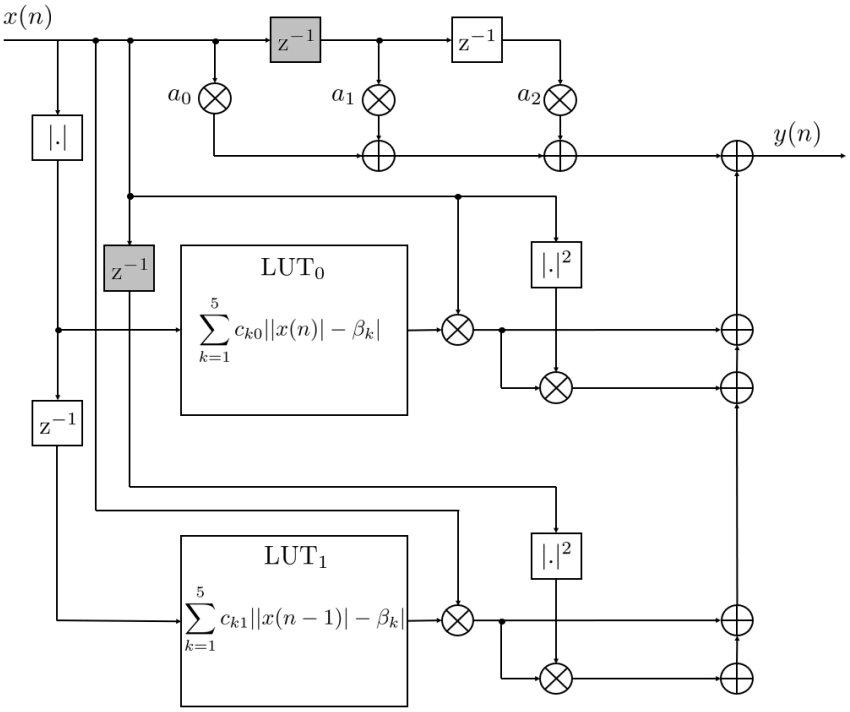


Figure 7 : Implementation du modele DVR optimal par LUT

Optimisation et linéarisation de l'amplificateur de puissance Doherty à double-entrée

Avec l'introduction des signaux modulés avec une amplitude non-constante et un PAPR élevé dans le contexte 4G LTE et 5G, il devient de plus en plus délicat d'améliorer le rendement de PA tout en conservant un niveau de linéarité adéquat. Pour cela, des architectures de PA avancées basées sur une charge dynamique ou une modulation d'alimentation ont été proposées dans la littérature pour éviter de gaspiller des ressources énergétiques excessives, telle que l'architecture Doherty, suiveur d'enveloppe (en anglais : Envelope Tracking - ET), Outphasing. Bien que ces architectures soient conçues majoritairement avec une seule entrée RF, plusieurs études ont été menées dans la littérature pour souligner les avantages de garder des entrées séparées [99].

Le principe de la séparation des entrées RF permet d'obtenir des degrés de liberté supplémentaires, dont des paramètres appelés paramètres libres, sont offerts par les entrées séparées, et peuvent être réglés pour améliorer les performances de PA [103]. la recherche des paramètres libres optimaux peut être considérée comme un problème d'optimisation globale. Nous nous sommes intéressés à l'architecture de Doherty, notamment, le PA Doherty à double-entrée (en anglais : Dual-Input Doherty PA - DIDPA), son architecture est illustrée dans la figure 8.

Le DIDPA se caractérise par deux entrées RF, une alimentation de drain V_{DC} et deux tensions V_{GS} pour polariser les amplificateurs principal et de crête indépendamment. L'amplitude et la phase instantanées de chaque signal d'entrée, ainsi que les tensions V_{GS}, peuvent être contrôlées et ajustées séparément en bande de base, permettant d'améliorer les performances du DIDPA, en particulier son rendement énergétique.

Le DIDPA est intégré dans un système comprenant en plus de la DPD comme technique de linéarisation et le CFR pour réduire le PAPR. Les paramètres libres couvrent les paramètres du DIDPA telles que les tensions de polarisation du DIDPA, le diviseur de puissance entre les 2 entrées RF, et d'autres paramètres du système tel que le facteur de la réduction de PAPR.

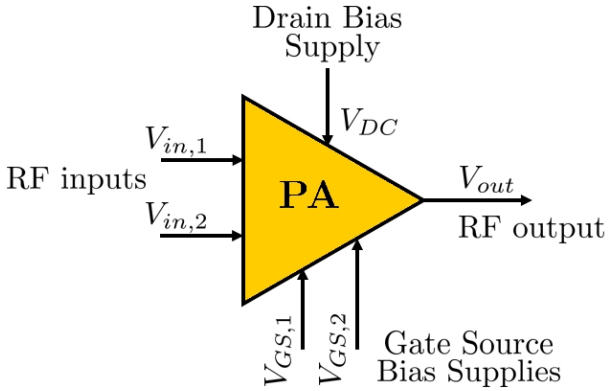


Figure 8 : Block diagram of dual-input PA

La conception et l'étude du DIDPA ont été rapportées dans de nombreux articles. Peu d'entre eux traitent l'optimisation des paramètres libres, la réduction du PAPR et sa linéarisation dans une approche conjointe. Le premier travail connexe est proposé dans [112], où les auteurs ont utilisé un algorithme d'approximation stochastique perturbée (en anglais : Simultaneous Perturbation Stochastic Approximation - SPSA) pour optimiser les paramètres libres de DIDPA. Dans [113], les auteurs ont proposé une optimisation basée sur un recuit simulé (en anglais : Simulated Annealing - SA) pour optimiser le DIDPA avec une fonction de coût formée à partir des valeurs du PAE, le gain du DIDPA, la puissance de sortie P_{out} et l'ACPR comme métrique faisant référence à la linéarité.

Dans cet axe de recherche, nous avons proposé une nouvelle approche de réglage automatique pour améliorer l'efficacité énergétique du DIDPA tout en répondant à l'exigence de linéarité. Cette approche consiste à optimiser les paramètres libres par un contrôle de recherche heuristique hybride (en anglais : Hybrid Heuristic Search Control - HHSC) selon une architecture présentée dans la figure 9.

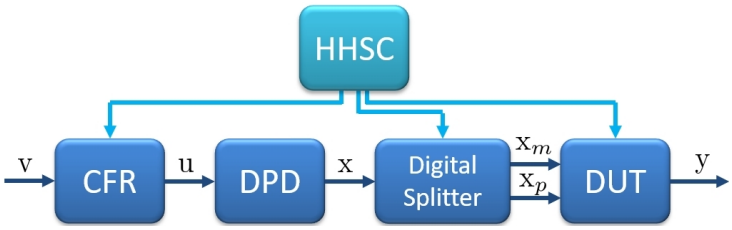


Figure 9 : Bloc diagrame de l'architetcure proposée

Cette architecture est composée de 4 blocs :

- CFR est utilisé pour réduire le PAPR du signal $v(n)$.
- DPD linéarise le DIDPA pour compenser les distorsions.
- Diviseur de puissance est conçu de manière optimale pour diviser le signal $x(n)$ en deux signaux différents $x_m(n)$ et $x_p(n)$
- DUT (en anglais : Device Under Test) présente l'étage puissance de l'architecture y compris le DIDPA avec son bloc d'alimentation.

Chaque bloc a des paramètres libres à définir ou à contrôler par l'algorithme d'optimisation HHSC :

- Le principe du CFR pour réduire le PAPR se base sur la technique d'annulation de pics (en anglais : Peak Cancellation - PC), et se réalise à l'aide d'opérations d'écrêtage brutal et de filtrage.

Le signal d'entrée $v(n)$ est écrêté selon un seuil μ. Un signal de pics $v_p(n)$ est construit en soustrayant le signal clippé du signal original :

$$v_p(n) = \begin{cases} v(n) \times \left(1 - \frac{\mu}{|v(n)|}\right) & \text{si } |v(n)| \geq \mu \\ 0 & \text{si } |v(n)| < \mu \end{cases}$$

Ce signal de pics est filtré avant d'être soustrait du signal original. Nous utilisons le seuil d'écrêtage μ comme paramètre libre du bloc CFR.

- La motivation derrière l'utilisation de deux entrées RF séparées est d'éliminer les séparateurs d'entrée analogiques, tels que le diviseur de Wilkinson, et de permettre un contrôle de puissance indépendant vers l'amplificateur principal et de crête de Doherty. Le diviseur de puissance divise le signal d'entrée $x = Xe^{j\theta}$ en deux signaux complexes x_m et x_p définis comme :

$$x_m = \alpha_m x \ ; \ x_p = \alpha_p e^{-j\phi} x$$

où ϕ représente la différence de phase entre les deux signaux et α représente le rapport de puissance avec la condition

$$\alpha_m = \sqrt{\alpha} \ ; \ \alpha_p = \sqrt{1-\alpha}$$

Dans cette étude, α et ϕ sont considérés comme des paramètres libres.

- Les tensions de polarisation de l'amplificateur principal et crête de DIDPA sont données respectivement par $V_{GS,m}$ et $V_{GS,p}$, et considérées comme des paramètres libres à optimiser par HHSC.

- Dans cette architecture, les coefficients de la DPD sont identifiés par l'architecture ILA à l'aide un processus de régression linéaire telle que la méthode des moindres carrés (en anglais : Least Square - LS). Par conséquent, la DPD ne sera pas contrôlée par HHSC.

Le principe de HHSC est basé sur la combinaison de l'algorithme SA comme recherche d'optimisation globale et de l'approche de commande extrémale (en anglais : Extremum Seeking Control - ESC) comme contrôle adaptatif pour affiner les résultats optimisés. Le vecteur des paramètres libres à optimiser est défini par : $\Theta = [\mu; \alpha; \phi; V_{GS,m}; V_{GS,p}]$. Le HHSC converge vers la solution optimale, notée par Θ_{opt}, selon une fonction de coût J définie par :

$$J(\Theta) = \sum_i w_i \left| \frac{\text{FOM}_i}{\text{FOM}_{t,i}} \right| \quad i = \{1, 2, 3, 4\}$$

où w_i est le poids correspondant à FOM_i (figure de mérite), et $\text{FOM}_{t,i}$ est la valeur cible (target) de la figure de mérite que l'utilisateur essaie d'atteindre. Dans cette fonction de coût, les FOMs utilisés sont {EVM;ACPR;PAE;P_{out}}. La conception de J pour assurer un bon compromis entre linéarité et efficacité est faite pour que le HHSC maximise J à 1 selon les contraintes suivantes :

$$\sum_{i=1}^{4} w_i = 1 \quad \text{and} \quad \max \left| \frac{\text{FOM}_i}{\text{FOM}_{t,i}} \right| = 1 \tag{5.33}$$

où $[w_1; w_2; w_3; w_4]$ sont initialisés par $[0.1; 0.1; 0.4; 0.4]$.

Les cibles de FOMs sont définies selon les spécifications de l'utilisateur. Dans cette étude, Les cibles de FOMs sont : EVM_t = 3%, ACPR $_t$ = -50 dB, PAE $_t$ = 100 % et P $_{out,t}$ = 40 dBm.

Tableau 10 : Intervalle d'optimisation des paramètres libres

	Paramètre	Intervalle	Unité
CFR	μ	[0 2]	dB
Diviseur de puissance	α	[0 1]	–
	ϕ	[-180 20]	Degré
DUT	$V_{GS,m}$	[-4 -1]	Volt
	$V_{GS,p}$	[-4 -1]	Volt

Le tableau 10 présente l'intervalle d'optimisation de chaque paramètre libre par le HHSC.

Le DIDPA est excité par un signal LTE 64-QAM 20 MHz avec un PAPR de 8.2 dB. Le HHSC est appliqué pour optimiser les paramètres libres dans le tableau 10. La figure 10 présente la convergence de HHSC où le comportement de l'évolution de J en fonction des itérations de HHSC est illustrée. Le HHSC commence par une solution initiale $\Theta_0 = [0\text{dB } 0.5\ -90°\ -1.5\text{V}\ -2.5\text{V}]$ et converge vers $\Theta_{opt} = [1.1\text{dB } 0.7\ -135°\ -1.33\text{V}\ -2.87\text{V}]$.

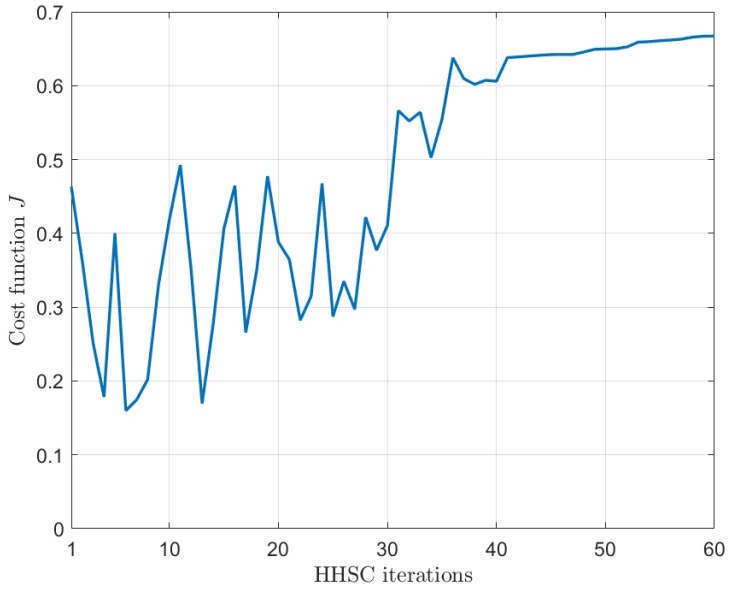


Figure 10 : Evolution de J par HHSC

Les FOMs correspondant à Θ_{opt} sont présentes dans le tableau 11. Le HHSC améliore les FOM d'efficacité en sacrifiant les FOM de linéarité. Ceci est dû au choix des coefficients de poids initiaux où w_3 et w_4 ont plus de poids que w_1 et w_2.

Tableau 11 : FOMs correspondant à la solution optimale Θ_{opt}.

EVM	ACPR	PAE	DE	P_{out}
10.63%	-27dB	59.29%	62%	36 dBm

La DPD est utilisée pour linéariser le DIDPA avec la configuration optimale Θ_{opt}. Le modèle DPD est basé sur le modèle DVR et dimensionné de manière optimale en utilisant l'algorithme HC. Le nombre de coefficients dans le modèle DVR est 30 coefficients. La figure 11 présente l'amélioration d'ACPR par le modèle DVR à la sortie de DIDPA.

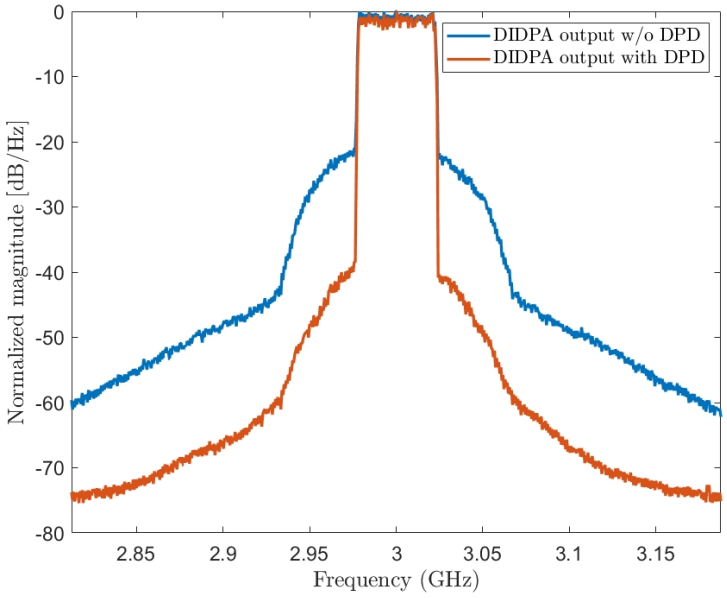
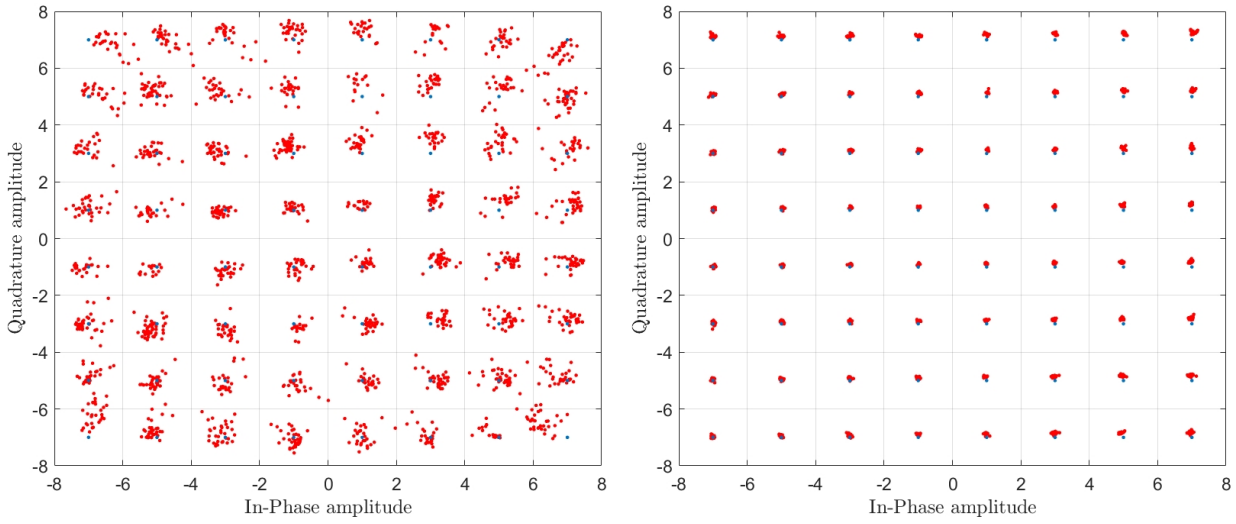


Figure 11 : Spectre de signal de sortie de DIDPA sans et avec DPD

La constellation des données I/Q avant et après l'application de la DPD est illustrée dans la figure 12 où l'EVM est amélioré de 10% à 2,5 %.



(a) (b)

Figure 12 : Constellation des donnees I/Q : (a) Sans DPD; (b): Avec DPD

Conclusion et perspectives

Dans cette thèse, nous nous sommes concentrés sur l'étude de la prédistorsion numérique pour linéariser les amplificateurs de puissance, et en particulier les modèles de segmentation. Cette thèse s'articule principalement sur trois aspects : l'étude et l'analyse comparative des modèles de la DPD, la conception d'un modèle optimal basé sur l'approche segmentale avec une stratégie d'implémentation matérielle, et l'optimisation et la linéarisation d'amplificateur de puissance Doherty à double-entrée avec un rendement énergétique élevé.

1. Une étude comparative est faite entre les modèles de la DPD, qui sont classés selon deux approches : l'approche globale et l'approche de segmentation. La comparaison

et l'analyse a été réalisée selon trois critères : les performances de la linéarisation (NMSE et ACPR), la complexité du modèle présentée par les coefficients du modèle et les propriétés numériques présentées par la dynamique des coefficients et le conditionnement de la matrice d'identification.

2. Pour les modèles basés sur l'approche segmentale, notamment le modèle DVR, deux études principales ont été développées : l'optimisation des seuils et la détermination de la structure optimale. Une nouvelle approche de l'optimisation des seuils est proposée en utilisant une approche itérative basée sur la décomposition du problème d'optimisation globale en un ensemble de sous-problèmes uni-modaux nécessitant une minimisation unidirectionnelle, comme la section dorée. Concernant la détermination de la structure optimale, l'étude de dimensionnement du modèle GMP par l'algorithme de hill-climbing a été étendu pour le modèle DVR avec deux aspects : le critère de recherche et la définition de voisinage.

3. Une approche système a été proposée pour concevoir un modèle DVR optimal qui est basée sur la combinaison de l'optimisation des seuils et du dimensionnement de la structure selon trois versions. La première version consiste à combiner le processus d'optimisation des seuils et le dimensionnement de la structure dans une recherche exhaustive globale. La seconde version consiste à intégrer le processus d'optimisation des seuils dans le dimensionnement du modèle mais avec une restriction : les seuils ne sont optimisés que si le nombre de segments d'une structure de modèle change. La troisième version est une mise à jour de la deuxième version, dans laquelle les termes du modèle sont considérés comme un paramètre conduisant à optimiser les seuils lorsque le nombre de segments et les termes du modèle sont modifiés.

4. Une approche d'auto-réglage a été proposée pour trouver la configuration optimale d'un amplificateur de puissance Doherty à double-entrée. Cette approche est obtenue en combinant deux méthodes d'optimisation : le recuit simulé comme étant un algorithme d'optimisation global et l'approche de commande extrémale comme étant un contrôle adaptatif. La convergence de l'approche proposée est guidé par une fonction de coût adaptative conçue pour conduire à un compromis entre linéarité et efficacité. La technique CFR et la linéarisation DPD sont intégrées de manière optimale dans l'architecture proposée afin de répondre aux spécifications de l'utilisateur en améliorant le rendement tout en maintenant un niveau satisfaisant de linéarité.

Bibliography

[1] P. M. Lavrador, T. R. Cunha, P. M. Cabral and J. C. Pedro, "The Linearity-Efficiency Compromise," in IEEE Microwave Magazine, vol. 11, no. 5, pp. 44-58, Aug. 2010.

[2] A. Katz, "Linearization: reducing distortion in power amplifiers," in IEEE Microwave Magazine, vol. 2, no. 4, pp. 37-49, Dec. 2001.

[3] Patrick Reynaert and Michiel Steyaert. Rf power amplifiers for mobile communications, Springer Netherlands, 2006.

[4] R. Raich and G. T. Zhou, "On the modeling of memory nonlinear effects of power amplifiers for communication applications," Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop., Pine Mountain, GA, USA, 2002, pp. 7-10.

[5] Slim Boumaiza and Fadhel M. Ghannouchi. Thermal memory effects modeling and compensation in RF power amplifiers and predistortion linearizers. IEEE Transactions on Microwave Theory and Techniques, Dec. 2003(51) :2427–2433, 12.

[6] Ghannouchi F.M. and Hammi O. Behavioral modeling and predistortion. IEEE Microwave Magazine, 10(7) :52–64, 2009.

[7] M. Isaksson, D.Wisell, and D. Ronnow, "A comparative analysis of behavioral models for rf power amplifiers," Microwave Theory and Techniques, IEEE Transactions on, vol. 54, no. 1, pp. 348–359, Jan. 2006.

[8] T. Reveyrand, D. Barataud, J. Lajoinie, M. Campovecchio, J.-M. Nebus, E. Ngoya,J. Sombrin, and D. Roques. A Novel Experimental Noise Power Ratio Characterization Method for Multicarrier Microwave Power Amplifiers. In 55th ARFTG Conference Digest, volume 37, pages 1–5, June 2000.

[9] J. B. Sombrin. On the formal identity of evm and npr measurement methods: Conditions for identity of error vector magnitude and noise power ratio. In 2011 41st European Microwave Conference, pages 337–340, Oct 2011.

[10] Ali Cheaito. Analyse théorique des distorsions dans la bande et en dehors de la bande de transmission pour les signaux à porteuses multiples : Impact conjoint des non-linéarités de l'amplificateur de puissance et effets de mémoire et prédistorsion. PhD thesis, INSA de Rennes, 2017.

[11] D. Guel and J. Palicot, "Analysis and comparison of clipping techniques for OFDM Peak-to-Average Power Ratio reduction," 2009 16th International Conference on Digital Signal Processing, Santorini-Hellas, 2009, pp. 1-6

[12] T. Jiang and Y.Wu. An overview : Peak-to-average power ratio reduction techniques for ofdm signals. IEEE Transactions on Broadcasting, 54(2) :257-268, June 2008.

[13] B. M. Popovic. Synthesis of power efficient multi-tone signals with flat amplitude spectrum. IEEE Transactions on Communications, 39(7) :1031-1033, Jul 1991.

[14] Tao Jiang, Guangxi Zhu, and Jianbin Zheng. Block coding scheme for reducing papr in ofdm systems with large number of subcarriers. Journal of Electronics, 21(6) :482-489, 2004.

[15] S. H. Muller and J. B. Huber. Ofdm with reduced peak-to-average power ratio by optimum combination of partial transmit sequences. Electronics Letters, 33(5) :368-369, Feb 1997.

[16] Robert Bauml, Robert Fischer, and Johannes Huber. Reducing the Peak to Average Power Ratio of Multicarrier Modulation by Selected Mapping. Electronics Letters 32(22) :2056-2057, 1996.

[17] T. May and H. Rohling. Reducing the peak-to-average power ratio in ofdm radio transmission systems. In Vehicular Technology Conference, 1998. VTC 98. 48th IEEE, volume 3, pages 2474-2478 vol.3, May 1998.

[18] J. Tellado-Mourelo, "Peak to average power reduction for multicarrier modulation," PhD thesis, Stanford University, 1999.

[19] D. L. Jones. Peak power reduction in ofdm and dmt via active channel modification. In Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers (Cat. No.CH37020), volume 2, pages 1076-1079 vol.2, Oct 1999.

[20] Kim J. and Konstantinou K. Digital predistortion of wideband signals based on power amplifier model with memory. Electronics Letters, 37(23) :1417–1418, 2001.

[21] Nagata Y. Linear amplification technique for digital mobile communications. In: IEEE 39th Vehicular Technology Conference; 1989. p. 159–164 vol.1.

[22] Cavers JK. Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements. IEEE Transactions on Vehicular Technology. 1990 Nov;39(4):374–382.

[23] Cavers JK. A linearizing predistorter with fast adaptation. In: 40th IEEE Conference on Vehicular Technology; 1990. p. 41–47.

[24] Wright AS, Durtler WG. Experimental performance of an adaptive digital linearized power amplifier [for cellular telephony]. IEEE Transactions on Vehicular Technology. 1992 Nov;41(4):395–400.

[25] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina. Memory polynomial predistorter based on the indirect learning architecture. Proc. of GLOBECOM, Taipei, Taiwan,, page 967–971, Mar 2002.

[26] R. Marsalek. Contributions to the power amplifier linearization using digital baseband adaptive predistortion. PhD thesis, Université Marne la Vallée, 2003.

[27] F. Raab, P. Asbeck, S. Cripps, P. Kenington, and Z. Popovic. Power amplifiers and transmitters for RF and microwave. IEEE Transactions on Microwave Theory and Techniques, pages 805–809, Mar 2002.

[28] J. Tapfuh Mouafo. Etude d'amplificaeurs faible niveau à haute linéarité en technologies intégrées HEMT ASGA pour applications spaciales. PhD thesis, Université de Limoges, 2008.

[29] Steve C. Cripps. RF Power Amplifiers for Wireless Communications ISBN 0-89006-989-1. Artech House Publishers, 1999.

[30] Lee SY, Lee YS, Hong SH, et al. An adaptive predistortion RF power amplifier with a spectrum monitor for multicarrier WCDMA applications. IEEE Transactions on Microwave Theory and Techniques. 2005 Feb;53(2):786 – 793.

[31] Katz A, Wood J, Chokola D. The Evolution of PA Linearization: From Classic Feedforward and Feedback Through Analog and Digital Predistortion. IEEE Microwave Magazine. 2016 Feb;17(2):32–40.

[32] Auer G, Giannini V, Desset C, et al. How much energy is needed to run a wireless network IEEE Wireless Communications. 2011 Oct;18(5):40–49.

[33] Baudoin, G., Venard, O., & Pham, D.-K. G. (2019). Digital predistortion. In Digitally Enhanced Mixed Signal Systems (pp. 65–123). Institution of Engineering and Technology.

[34] A. Zhu, J. C. Pedro and T. R. Cunha, "Pruning the Volterra Series for Behavioral Modeling of Power Amplifiers Using Physical Knowledge," in IEEE Transactions on Microwave Theory and Techniques, vol. 55, no. 5, pp. 813-821, May 2007.

[35] M. Schetzen ; The Volterra and Wiener Theories of Nonlinear Systems. Krieger Publishing Co., Inc., Melbourne, FL, USA, 2006.

[36] Zhou GT, Qian H, Ding L, et al. On the baseband representation of a bandpass nonlinearity. IEEE Transactions on Signal Processing. 2005 Aug;53(8):2953 – 2957.

[37] Lei Ding et al., "Memory polynomial predistorter based on the indirect learning architecture," Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, 2002, pp. 967-971 vol.1.

[38] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt and J. Pastalan, "A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers," in IEEE Transactions on Signal Processing, vol. 54, no. 10, pp. 3852-3860, Oct. 2006.

[39] Ngoya E, Gallou NL, Nebus JM, et al. Accurate RF and microwave system level modeling of wideband nonlinear circuits. In: 2000 IEEE MTT-S International Microwave Symposium Digest (Cat. No.00CH37017). vol. 1; 2000. p. 79–82 vol.1.

[40] A. Zhu, J. C. Pedro and T. J. Brazil, "Dynamic Deviation Reduction-Based Volterra Behavioral Modeling of RF Power Amplifiers," in IEEE Transactions on Microwave Theory and Techniques, vol. 54, no. 12, pp. 4323-4332, Dec. 2006.

[41] A. Zhu, J. Dooley and T. J. Brazil, "Simplified Volterra Series Based Behavioral Modeling of RF Power Amplifiers Using Deviation-Reduction," 2006 IEEE MTT-S International Microwave Symposium Digest, San Francisco, CA, 2006, pp. 1113-1116.

[42] L. Guan and A. Zhu, "Simplified dynamic deviation reduction-based Volterra model for Doherty power amplifiers," 2011 Workshop on Integrated Nonlinear Microwave and Millimetre-Wave Circuits, Vienna, 2011, pp. 1-4.

[43] Z. He, W. Ye and S. Feng, "Digital predistortion using modified dynamic deviation reduction-based volterra model," in Electronics Letters, vol. 49, no. 21, pp. 1342-1343, October 10 2013.

[44] B. Song, S. He, J. Peng and Y. Zhao, "Dynamic deviation memory polynomial model for digital predistortion," in Electronics Letters, vol. 53, no. 9, pp. 606-607, 4 27 2017.

[45] Chua LO, Kang SM. Section-wise piecewise-linear functions: Canonical representation, properties, and applications. Proceedings of the IEEE. 1977 June;65(6):915–929.

[46] A. Zhu, "Decomposed Vector Rotation-Based Behavioral Modeling for Digital Predistortion of RF Power Amplifiers," in IEEE Transactions on Microwave Theory and Techniques, vol. 63, no. 2, pp. 737-744, Feb. 2015.

[47] Bai EW. Non-Parametric Nonlinear System Identification: A Data-Driven Orthogonal Basis Function Approach. IEEE Transactions on Automatic Control. 2008 Dec;53(11):2615–2626.

[48] Magesacher T, Singerl P. Benchmarking of learning architectures for digital predistortion. In: 2016 50th Asilomar Conference on Signals, Systems and Computers; 2016. p. 648–651.

[49] Zhou D, DeBrunner VE. Novel Adaptive Nonlinear Predistorters Based on the Direct Learning Algorithm. IEEE Transactions on Signal Processing. 2007 Jan;55(1):120–133.

[50] Abi Hussein M, Bohara VA, Venard O. On the system level convergence of ILA and DLA for digital predistortion. In: 2012 International Symposium on Wireless Communication Systems (ISWCS); 2012. p. 870–874.

[51] H. Hamoud, K. El-Akhdar, S. Mons and E. Ngoya, "Evaluation of an optimal digital predistorter for multistandard systems," 2017 Integrated Nonlinear Microwave and Millimetre-wave Circuits Workshop (INMMiC), Graz, 2017, pp. 1-3.

[52] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson and C. Fager, "A Comparative Analysis of the Complexity/Accuracy Tradeoff in Power Amplifier Behavioral Models," in IEEE Transactions on Microwave Theory and Techniques, vol. 58, no. 6, pp. 1510-1520, June 2010.

[53] J. C. Pedro and S. A. Maas, "A comparative overview of microwave and wireless power-amplifier behavioral modeling approaches," in IEEE Transactions on Microwave Theory and Techniques, vol. 53, no. 4, pp. 1150-1163, April 2005.

[54] T. Gotthans, G. Baudoin and A. Mbaye, "Comparison of modeling techniques for power amplifiers," 2013 23rd International Conference Radioelektronika (RADIOELEKTRONIKA), Pardubice, 2013, pp. 232-235.

[55] Hadi, M. U., Kantana, C., Traverso, P. A., Tartarini, G., Venard, O., Baudoin, G., & Polleux, J. (2019). Assessment of digital predistortion methods for DFB-SSMF radio-over-fiber links linearization. Microwave and Optical Technology Letters, 62(2), 540–546.

[56] C. Kantana, O. Venard and G. Baudoin, "Linearization Performance vs Implementation Properties of Digital Predistorter Modelling for Three-Way Doherty Power Amplifier," 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), 2019, pp. 1-6.

[57] S. Wang, M. A. Hussein, O. Venard and G. Baudoin, "Performance analysis of multi-stage cascaded digital predistortion," 2017 40th International Conference on Telecommunications and Signal Processing (TSP), Barcelona, 2017, pp. 275-248.

[58] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran and J. M. Dilhac, "Multi-Lookup Table FPGA Implementation of an Adaptive Digital Predistorter for Linearizing RF Power Amplifiers With Memory Effects," in IEEE Transactions on Microwave Theory and Techniques, vol. 56, no. 2, pp. 372-384, Feb. 2008.

[59] S. Wang, M. A. Hussein, O. Venard and G. Baudoin, "Optimal sizing of generalized memory polynomial model structure based on Hill-Climbing heuristic," 2016 46th European Microwave Conference (EuMC), London, 2016, pp. 190-193.

[60] C. Kantana, O. Venard and G. Baudoin, "Comparison of GMP and DVR models" 2018 International Workshop on Integrated Nonlinear Microwave and Millimeter-wave Circuits (INMMIC), Brive La Gaillarde, 2018, pp. 1-3.

[61] C. Kantana, O. Venard, G. Baudoin (2021). Thresholds Optimization of Decomposed Vector Rotation Model for Digital Predistortion of RF Power Amplifier. Radioengineering, 30(1), 250-258.

[62] C. Kantana, O. Venard and G. Baudoin, "Decomposed Vector Rotation Model Sizing by Hill-Climbing Heuristic for Digital Predistortion of RF Power Amplifiers," 2020 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), San Antonio, TX, USA, 2020, pp. 22-25.

[63] J. Zhai, L. Zhang, Z. Yu, J. Zhou and W. Hong, "A Modified Canonical Piecewise-Linear Function-Based Behavioral Model for Wideband Power Amplifiers," in IEEE Microwave and Wireless Components Letters, vol. 26, no. 3, pp. 195-197, March 2016

[64] N. Yang, Y. Guo and C. Yu, "A decomposed vector rotation-based sideband distortion suppression model for wideband mmwave transmitter leakage compensation," 2017 Sixth Asia-Pacific Conference on Antennas and Propagation (APCAP), Xi'an, 2017, pp. 1-3,

[65] Q. Lu et al., "A modified decomposed vector rotation behavioral model for concurrent dual-band envelope tracking RF power amplifiers," 2017 IEEE Asia Pacific Microwave Conference (APMC), Kuala Lumpar, 2017, pp. 264-267,

[66] W. Cao and A. Zhu, "A Modified Decomposed Vector Rotation-Based Behavioral Model With Efficient Hardware Implementation for Digital Predistortion of RF Power Amplifiers," in IEEE Transactions on Microwave Theory and Techniques, vol. 65, no. 7, pp. 2443-2452, July 2017

[67] J. K. Cavers, "Optimum table spacing in predistorting amplifier linearizers," in IEEE Transactions on Vehicular Technology, vol. 48, no. 5, pp. 1699-1705, Sept. 1999.

[68] C.-H. Lin, H.-H. Chen, Y.-Y. Wang and J.-T. Chen, "Dynamically optimum lookup-table spacing for power amplifier predistortion linearization", IEEE Trans. Microw. Theory Techn., vol. 54, no. 5, pp. 2118-2127, May 2006.

[69] T. Magesacher, P. Singerl and M. Mataln, "Optimal Segmentation for Piecewise RF Power Amplifier Models," in IEEE Microwave and Wireless Components Letters, vol. 26, no. 11, pp. 909-911, Nov. 2016.

[70] K. Levenberg, "A method for the solution of certain non-linear problems in least squares", Quart. J. Appl. Math., vol. 2, no. 2, pp. 164-168, Jul. 1944.

[71] Carlos Mateo, Pedro L. Carro, Paloma García-Dúcar, Jesús De Mingo, and Íñigo Salinas, "Radio-over-fiber linearization with optimized genetic algorithm CPWL model," Opt. Express 25, 3694-3708 (2017).

[72] Kiefer, J. (1953), "Sequential minimax search for a maximum", Proceedings of the American Mathematical Society, 4 (3): 502–506.

[73] Evans, Merran, Nicholas Hastings, and Brian Peacock. Statistical Distributions. 2nd ed. New York: J. Wiley, 1993.

[74] Burr, Irving W. Cumulative Frequency Functions. Ann. Math. Statist. 13 (1942), no. 2, 215–232.

[75] Chhikara, Raj S.; Folks, J. Leroy (1989), The Inverse Gaussian Distribution: Theory, Methodology and Applications, New York, NY, USA: Marcel Dekker.

[76] Shoukri, M.M.; Mian, I.U.M.; Tracy, D.S. (1988), "Sampling Properties of Estimators of the Log-Logistic Distribution with Application to Canadian Precipitation Data", The Canadian Journal of Statistics, 16 (3): 223–236.

[77] H. Jiang, X. Yu, and P. Wilford, "Digital predistortion using stochastic conjugate gradient method," Broadcasting, IEEE Transactions on, vol. 58, no. 1, pp. 114 –124, march 2012.

[78] X. Yu and H. Jiang, "Digital predistortion using adaptive basis functions," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 12, pp. 3317–3327, Dec 2013

[79] J. Reina-Tosina, M. Allegue-Martinez, C. Crespo-Cadenas, C. Yu, and S. Cruces, "Behavioral modeling and predistortion of power amplifiers under sparsity hypothesis," Microwave Theory and Techniques, IEEE Transactions on, vol. 63, no. 2, pp. 745–753, Feb 2015.

[80] T. Gotthans, G. Baudoin, and A. Mbaye, "Optimal order estimation for modeling and predistortion of power amplifiers," in Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on, Oct 2013, pp. 1–4.

[81] S. Wang, M. A. Hussein, G. Baudoin, O. Venard and T. Gotthans, "Comparison of hill-climbing and genetic algorithms for digital predistortion models sizing," 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), Monte Carlo, 2016, pp. 289-292.

[82] K. Sullivan and S. Jacobson, "A convergence analysis of generalized hill climbing algorithms," Automatic Control, IEEE Transactions on, vol. 46, no. 8, pp. 1288–1293, Aug 2001.

[83] L. Rueda and V. Vidyadharan, "A hill-climbing approach for automatic gridding of cdna microarray images," Computational Biology and Bioinformatics,IEEE/ACM Transactions on, vol. 3, no. 1, pp. 72–83, Jan 2006.

[84] S. Wang, M. A. Hussein, O. Venard and G. Baudoin, "A Novel Algorithm for Determining the Structure of Digital Predistortion Models," in IEEE Transactions on Vehicular Technology, vol. 67, no. 8, pp. 7326-7340, Aug. 2018.

[85] Siqi Wang. Study on complexity reduction of digital predistortion for power amplifier linearization. PhD thesis, Université Paris-Est, 2018.

[86] P. Jardin and G. Baudoin, "Filter lookup table method for power amplifier linearization," IEEE Trans. Veh. Technol., vol. 56, no. 3, pp. 1076-1087, May 2007.

[87] J. K. Cavers, "Optimum indexing in predistorting amplifier linearizers," 1997 IEEE 47th Vehicular Technology Conference. Technology in Motion, Phoenix, AZ, USA, 1997, pp. 676-680 vol.2.

[88] H. Ben Nasr, S. Boumaiza, M. Helaoui, A. Ghazel and F. M. Ghannouchi, "On the critical issues of DSP/FPGA mixed digital predistorter implementation," 2005 Asia-Pacific Microwave Conference Proceedings, 2005, pp. 4 pp.

[89] J. Liszewski, B. Schubert, W. Keusgen and A. Kortke, "Low-complexity FPGA implementation of Volterra predistorters for power amplifiers," 2011 IEEE Topical Conference on Power Amplifiers for Wireless and Radio Applications, 2011, pp. 41-44.

[90] P. L. Gilabert and G. Montoro, "Look-Up Table Implementation of a Slow Envelope Dependent Digital Predistorter for Envelope Tracking Power Amplifiers," in IEEE Microwave and Wireless Components Letters, vol. 22, no. 2, pp. 97-99, Feb. 2012.

[91] B. Fehri and S. Boumaiza, "Automated symbolic optimization and high level synthesis of single- and multi- band digital pre-distortion hardware in an FPGA," 2015 IEEE MTT-S International Microwave Symposium, 2015, pp. 1-3.

[92] Zheren Long, Guohui Zheng, Yongliang Li, Dewei Yang and Hua Wang, "The design and implementation of baseband predistorter based on FPGA and ARM," 2015 10th International Conference on Communications and Networking in China (ChinaCom), 2015, pp. 626-631.

[93] W. Li, E. Guillena, G. Montoro and P. L. Gilabert, "FPGA Implementation of Memory-Based Digital Predistorters with High-Level Synthesis," 2021 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), 2021, pp. 37-40.

[94] Y. Li, W. Cao and A. Zhu, "Instantaneous Sample Indexed Magnitude-Selective Affine Function-Based Behavioral Model for Digital Predistortion of RF Power Amplifiers," in IEEE Transactions on Microwave Theory and Techniques, vol. 66, no. 11, pp. 5000-5010, Nov. 2018.

[95] Doherty W.H., "A New High Efficiency Power Amplifier for Modulated Waves," Proceedings of the Institute of Radio Engineers , vol.24, no.9, pp. 1163- 1182, Sept. 1936.

[96] Kahn L.R., "Single-Sideband Transmission by Envelope Elimination and Restoration," Proceedings of the IRE , vol.40, no.7, pp.803-806, July 1952.

[97] Wang, F.; Yang, A.H.; Kimball, D.F.; Larson, L.E.; Asbeck, P.M., "Design of wide-bandwidth envelope-tracking power amplifiers for OFDM applications," IEEE Trans. Microwave Theory & Tech, vol.53, no.4, pp. 1244- 1255, April 2005

[98] H. Chireix, "High power outphasing modulation", Proceedings of the IRE, vol. 23, no. 11, pp. 1370–1392, Nov. 1935.

[99] R. Darraji, F. M. Ghannouchi and O. Hammi, "A Dual-Input Digitally Driven Doherty Amplifier Architecture for Performance Enhancement of Doherty Transmitters," in IEEE Transactions on Microwave Theory and Techniques, vol. 59, no. 5, pp. 1284-1293, May 2011.

[100] R. Darraji and F. M. Ghannouchi, "Digital Doherty Amplifier With Enhanced Efficiency and Extended Range," in IEEE Transactions on Microwave Theory and Techniques, vol. 59, no. 11, pp. 2898-2909, Nov. 2011

[101] C. M. Andersson, D. Gustafsson, J. Chani Cahuana, R. Hellberg and C. Fager, "A 1–3-GHz Digitally Controlled Dual-RF Input Power-Amplifier Design Based on a Doherty-Outphasing Continuum Analysis," in IEEE Transactions on Microwave Theory and Techniques, vol. 61, no. 10, pp. 3743-3752, Oct. 2013

[102] A. Piacibello et al., "Comparison of S-band Analog and Dual-Input Digital Doherty Power Amplifiers," 2018 13th European Microwave Integrated Circuits Conference (EuMIC), Madrid, 2018, pp. 269-272.

[103] R. Darraji, P. Mousavi and F. M. Ghannouchi, "Doherty Goes Digital: Digitally Enhanced Doherty Power Amplifiers," in IEEE Microwave Magazine, vol. 17, no. 8, pp. 41-51, Aug. 2016.

[104] Horst, R., Pardalos, Panos, Nguyen Van Thoai, "Introduction to Global Optimization: Non-convex Optimization and Its Applications", Berlin, Heidelberg: Springer-Verlag, 2002.

[105] Y. Komatsuzaki, K. Nakatani, S. Shinjo, S. Miwa, R. Ma and K. Yamanaka, "3.0–3.6 GHz wideband, over 46% average efficiency GaN Doherty power amplifier with frequency dependency compensating circuits," 2017 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR), Phoenix, AZ, 2017, pp. 22-24.

[106] J. T. Olkkonen and H. Olkkonen, "Fractional Time-Shift B-Spline Filter," in IEEE Signal Processing Letters, vol. 14, no. 10, pp. 688-691, Oct. 2007.

[107] L. Anttila, M. Valkama and M. Renfors, "Blind Compensation of Frequency-Selective I/Q Imbalances in Quadrature Radio Receivers: Circularity -Based Approach," 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, Honolulu, HI, 2007, pp. III-245-III-248

[108] W. Kim, K. Cho, S. P. Stapleton and J. Kim, "Doherty Feed-Forward Amplifier Performance Using a Novel Crest Factor Reduction Technique," in IEEE Microwave and Wireless Components Letters, vol. 17, no. 1, pp. 82-84, Jan. 2007.

[109] D. Lopez, P. L. Gilabert, G. Montoro and N. Bartzoudis, "Peak cancellation and digital predistortion of high-order QAM wideband signals for next generation wireless backhaul equipment," 2014 International Workshop on Integrated Nonlinear Microwave and Millimetre-wave Circuits (INMMiC), Leuven, 2014, pp. 1-3.

[110] J. C. Cahuana, P. Landin, D. Gustafsson, C. Fager and T. Eriksson, "Linearization of dual-input Doherty power amplifiers," 2014 International Workshop on Integrated Nonlinear Microwave and Millimetre-wave Circuits (INMMiC), Leuven, 2014, pp. 1-3.

[111] P. Singerl, T. Magesacher and M. Mataln, "Predistortion- and development-platform for multi-input RF power amplifiers," 2016 11th European Microwave Integrated Circuits Conference (EuMIC), London, 2016, pp. 29-32.

[112] S. Niu, A. M. Koushik, R. Ma, K. H. Teo, S. Shinjo and Y. Komatsuzaki, "Stochastically approximated multiobjective optimization of dual input digital Doherty Power Amplifier," 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA), Hiroshima, 2017, pp. 147-152.

[113] R. Ma et al., "Machine-Learning Based Digital Doherty Power Amplifier," 2018 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT), Melbourne, VIC, 2018, pp. 1-3.

[114] Benosman, Mouhacine. Learning-based adaptive control : an extremum seeking approach - theory and applications. Oxford: Butterworth-Heinemann is an imprint of Elsevier, 2016.

[115] J. Peng, S. He, W. Shi, T. Yao, J. Wu and J. Wang, "Adaptive Signal Separation for Dual-Input Doherty Power Amplifier," in IEEE Transactions on Microwave Theory and Techniques, vol. 68, no. 1, pp. 121-131, Jan. 2020

[116] E. Guillena, W. Li, P. L. Gilabert and G. Montoro, "Prediction of the Optimal Phase Shift between Control Signals in Dual-Input Power Amplifiers," 2020 International Workshop on Integrated Nonlinear Microwave and Millimetre-Wave Circuits (INMMiC), Cardiff, United Kingdom, 2020, pp. 1-3.

[117] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220(4598), 671–680.

[118] Metropolis, Nicholas; Rosenbluth, Arianna W.; Rosenbluth, Marshall N.; Teller, Augusta H.; Teller, Edward (1953). "Equation of State Calculations by Fast Computing Machines". The Journal of Chemical Physics. 21 (6): 1087.

[119] Benosman, M., Farahmand, A.-M., & Xia, M. (2018). Learning-based iterative modular adaptive control for nonlinear systems. International Journal of Adaptive Control and Signal Processing, 33(2), 335–355.

[120] M. Leblanc, “Sur l’électrification des chemins de fer au moyen de courants alternatifs de fréquence élevée,” Revue générale de l’électricité, vol. 12, no. 8, pp. 275–277, 1922.

[121] V. Kazakevich, On extremum seeking. PhD thesis, PhD Thesis, Moscow High Technical University, 1944.

[122] M. Krstic, “Performance improvement and limitations in extremum seeking control,” Systems & Control Letters, vol. 39, no. 5, pp. 313–326, 2000.

[123] Ariyur, Kartik B., and Miroslav Krstic. Real time optimization by extremum seeking control. Hoboken, NJ: Wiley Interscience, 2003.

[124] N. Gunantara, ”A review of multi-objective optimization: Methods and its applications”, Cogent Engineering, vol. 5, no. 1, 2018.

[125] Hwang, C. L., and Abu S. Masud. Multiple objective decision making, methods and applications : a state-of-the-art survey. Berlin New York: Springer-Verlag, 1979.

[126] Murata T., Ishibuchi H., Gen M. (2001) Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms. In: Zitzler E., Thiele L., Deb K., Coello Coello C.A., Corne D. (eds) Evolutionary Multi-Criterion Optimization. EMO 2001. Lecture Notes in Computer Science, vol 1993. Springer, Berlin, Heidelberg.

[127] ”On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization,” in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-1, no. 3, pp. 296-297, July 1971

[128] A. Zhu, P. J. Draxler, J. J. Yan, T. J. Brazil, D. F. Kimball, and P. M. Asbeck, ”Open-loop digital predistorter for RF power amplifiers using dynamic deviation reduction-based Volterra series,” IEEE Trans. Microw. Theory Tech., vol. 56, no. 7, pp. 1524-1534, July 2008.

Abstract

This Ph.D. work contributes to the digital predistortion linearization technique of power amplifiers. Power Amplifier is one of the most critical elements of radiocommunication systems, which exhibits static nonlinearities and nonlinear memory effects. Achieving a good trade-off between the linearity of the power amplifier and its efficiency is becoming more crucial. Digital predistortion is a powerful linearization technique that aims to compensate for power amplifier distortions and provides linear amplification with good efficiency. The predistortion principle consists of implementing a nonlinear function, the so-called predistorter upstream of the power amplifier. The predistorter ideally has the inverse characteristics of the power amplifier. This operation allows us to consider the memory effects of the power amplifier, and in particular, the long-term memory. Several behavioral models have been used as predistorter. Most of them are classified into two families: global models derived from the Volterra series and models based on the segmentation approach. The comparative and analysis study of these models is one of the focuses of this dissertation, in which three aspects are used for comparison: linearization performance, complexity, and hardware implementation properties. By focusing on models based on the segmentation approach, this dissertation proposes an approach to design an optimal model according to a trade-off between linearization performance and model complexity. This model is used to linearize a dual-input Doherty power amplifier. A global optimization algorithm combined with a control process is proposed to enhance efficiency while maintaining a good linearity level according to a proposed adaptive cost function.

Résumé

Le travail de thèse présenté par ce manuscrit s'intéresse à la linéarisation des amplificateurs de puissance en utilisant la prédistorsion numérique. L'amplificateur de puissance est l'un des modules les plus critiques des équipements de communication radio qui présente des non-linéarités statiques ainsi que des effets de mémoire. Il devient de plus en plus crucial de réaliser un compromis entre la linéarité et le rendement énergétique. La prédistorsion numérique est une technique de linéarisation efficace qui compense les distorsions dues à la non-linéarité en appliquant une déformation sur le signal d'entrée, de manière à ce que le système global réalise une amplification linéaire. Le principe de la prédistortion numérique consiste à mettre en œuvre une fonction non linéaire dite prédistorteur en amont de l'amplificateur de puissance, et qui a idéalement les caractéristiques inverses de l'amplificateur de puissance. Cette opération permet également de prendre en compte les effets mémoire, et en particulier la mémoire à long terme. Plusieurs modèles comportementaux sont proposés pour la prédistorsion qui peuvent être classés en deux catégories : les modèles globaux et les modèles par segmentation. L'étude comparative de ces modèles est l'un des axes de travail traité par cette thèse, dont la comparaison repose sur trois aspects qui sont soulignés : les performances de linéarisation, la complexité et les propriétés de l'implémentation matérielle. En se concentrant sur les modèles basés sur l'approche par segmentation, un algorithme est proposé pour concevoir un modèle optimal en satisfaisant un bon compromis entre les performances de linéarisation et la complexité du modèle. Ce modèle est utilisé pour linéariser un amplificateur de puissance Doherty à double entrée pour lequel un algorithme d'optimisation globale associé à un processus de contrôle est proposé pour améliorer le rendement tout en conservant un bon niveau de linéarité selon une fonction de coût adaptative.