Signal-Aware Predistortion Configuration Using Unsupervised Learning for 5G–Advanced/6G Radios

Oskar Magnusson

Arvid Bergenfeldt

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6316
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2026

Abstract

Digital Predistortion (DPD) is an essential component of modern mobile communication systems ensuring that distortion, created by the nonlinearities of the Power Amplifier (PA) is reduced while allowing them to be operated at high efficiency levels. This thesis has investigated the use of an unsupervised machine learning method known as Gaussian Mixture Model (GMM) as a way to define and populate a suitable number of DPD configurations to satisfy the required use cases for specific radio products. The goal was to minimize the required resources by the DPD as well as the actual DPD configuration time spent in the hardware integration process. The proposed method was compared against a baseline implementation and evaulated on the mentioned metrics. The result of the thesis was a reduction in resources compared to the baseline while maintaining system requirements quantified by Adjacent Channel Leakage Ratio (ACLR). However, the proposed method had a significant increase in hardware integration time compared to the baseline. Overall, the most promising method was a combination between machine learning and traditional optimization which yielded the lowest DPD resources while still maintaining lower integration time than the strictly machine learning based approach. We conclude that a clear trade-off exists between DPD resource utilization and hardware integration time, where the optimal configuration is dependent on system-level priorities. These findings highlight the potential of hybrid approaches, combining machine learning with conventional optimization, as an effective strategy for balancing performance and implementation complexity in practical systems.

Keywords **–** Digital Predistortion, Unsupervised Learning, 6G, Digital Radio Tuning, 5G–Advanced, Machine Learning, State-of-the-art, Optimization, Clustering, GMM, Artificial Intelligence

Acknowledgements

Thanks to Lund University and the department of Automatic Control at LTH for letting us write our M.Sc. Thesis under your department. To our supervisor Pontus Giselsson, and our examinator Bo Bernhardsson, thank you for your valuable feedback and guidance during this project. It has been a valuable and great experience.

There are many people that have been a part of making this thesis possible. A major thank you to the company of Ericsson that enabled us the possibility to do our thesis under the supervision and expertise of your employees. Thus, we would like to give a special thank you to Sergio Gutiérrez, Ryan Feng, Yibo Wu, and Anahid Safavi at Ericsson and Vidit Saxena for their invaluable insights and feedback on our work. But the largest thank you is to our main supervisor at Ericsson, Simone Grimaldi. Without your essential insights, help, guidance, patience, knowledge, and supervision this thesis would never have been possible. Thank you for believing and guiding us, truly, thank you.

List of Abbreviations

3G	3rd Generation	17
3GPP	3rd Generation Partnership Project	17
4G	4th Generation	17
5G	5th Generation	13
6G	6th Generation	18
ACLR	Adjacent Channel Leakage Ratio	3
ACPR	Adjacent Channel Power Ratio	33
ADC	Analog-to-Digital Converter	56
AFE	Analog Front-End	55
AI	Artificial Intelligence	41
AM/AM	Amplitude-to-Amplitude Modulation	26
AM/PM	Amplitude-to-Phase Modulation	26
AMPS	Advanced Mobile Phone System	16
ASIC	Application Specific Integrated Circuit	14
AWGN	Additive White Gaussian Noise	79
BB	Baseband	56
BIC	Bayesian Information Criterion	50
BW	Bandwidth	20
CEV	Cumulative Explained Variance	53
CLARA	Clustering for Large Applications	63
CLARANS	Clustering for Large Applications based on Randomized Search	63
CFR	Crest Factor Reduction	55
DAC	Digital-to-Analog Converter	56

MSE	Mean Squared Error	34
mMTC	massive Machine-Type Communication	18
NMT	Nordic Mobile Telephony	16
NMSE	Normalized Mean Squared Error	34
NN	Neural Network	41
NR	New Radio	18
NTI	Non-Linear Time-Invariant System	29
OFDM	Orthogonal Frequency-Division Multiplexing	17
PA	Power Amplifier	3
PAPR	Peak-to-Average Power Ratio	79
PCA	Principal Component Analysis	51
PC	Principal Component	51
PSD	Power Spectral Density	22
QAM	Quadrature Amplitude Modulation	17
RC	Radio Communication	37
RF	Radio Frequency	16
RL	Reinforcement Learning	43
RMS	Root Mean Square	35
SEM	Spectral Emission Mask	28
SL	Supervised Learning	42
SNR	Signal-to-Noise Ratio	62
t.u	Time Unit	99
TACS	Total Access Communication System	16
TDD	Time-Division Duplex	17
UL	Unsupervised Learning	43
URLLC	Ultra-Reliable Low-Latency Communication	18
WCSS	Within-Cluster Sum of Squares	44
WCDMA	Wideband Code Division Multiple Access	17

Contents

1

Introduction

1.1 Background and Motivation

Wireless communication is a fundamental technology in modern society. It is present across nearly every industry and in a multitude of everyday devices such as phones, computers, and cars. As wireless systems continue to evolve, they place increasing demands on engineering, hardware, and signal-processing design, constantly pushing the boundaries of what is technically feasible [12]. Modern wireless communication has evolved through several technological generations, each introducing new capabilities and broader application areas. The current technical demands on 5th Generation (5G) mobile communication to achieve high-capacity, low-latency, and machine-oriented communication, create major engineering challenges. Modern wireless communication systems are required to support increasingly diverse services, wider bandwidths, and more complex signal-carrier configurations. These requirements place high demands on the radio transmitter chain to deliver sufficient output power while maintaining transmitter performance and signal quality [12, 47]. This paper will explore and focus on two central electronic components in wireless communication, the Digital Predistortion (DPD) and the Power Amplifier (PA). The PA and DPD work in symbiosis to create highly efficient transmission systems that enable long-distance, power-efficient, and reliable wireless communication. DPD is used to compensate for the nonlinear behavior of the PA, enabling more efficient amplification while reducing distortion in the transmitted signal [13]. However, generating a suitable set of DPD configurations capable of covering all use cases supported by a radio unit can be an elaborate and time-consuming process. This becomes especially important when the set of supported carrier configurations is large and the requirements on transmitted distortion are stringent. Moreover, the selected configurations can have a significant impact on the digital processing load of the radio unit, directly affecting runtime power consumption during operation. Consequently, efficient optimization methods are needed to flexibly balance multiple objectives, such as reducing hardware tuning time during integration and minimizing digital power consumption during runtime operation.

1.2 Purpose of the Thesis

The purpose of this thesis is to investigate whether machine-learning methods can support and improve the configuration process for the DPD. A key challenge is to determine how transmitted-signal characteristics, including carrier frequency, bandwidth, and spectral occupancy, affect the selection of a suitable DPD configuration and the resulting trade-off between resource consumption (digital-power consumption), configuration time, and stability. This creates an important practical challenge, since different signal configurations may place very different demands on the PA, and thus the DPD. In particular, this thesis focuses on identifying structure among different carrier-signal-configurations in feature-space, and studying whether similar carrier-signal-configurations can be grouped/clustered in a meaningful way based on these features. I.e. is there a correlation of distance in carrier feature space and distance in DPD-configuration (delay-term space) space. From this perspective, the purpose of this work is not to compete with the state-of-the-art DPD algorithms, but rather to complement and support existing methods in order to improve their efficiency and deepening the understanding how carriers affect the generated nonlinearities of a PA.

More specifically, the ambition of doing this is to explore two different trade-offs and minimization objectives (which will be presented in more detail in Section 5.2). The goal is to both minimize the total Hardware (HW) integration time, i.e. how long does it take to find, tune, and up-load a complete DPD-configuration in production, and the digital-power consumption of the Application Specific Integrated Circuit (ASIC) once the product is in the field. By minimizing the number of delay-terms (coefficients) for a DPD-configuration, fewer coefficients need to be updated/computed in the ASIC which results in less ASIC-area, memory, latency, bit-width, and MAC operations. This implies that less heat and hardware stress is induced, thus, less cooling material is needed, and the production cost, weight, and instalment complexity is reduced.

1.3 Research Questions

With the purpose of the thesis established, the following research questions were defined and explored throughout the work:

- **RQ1:** What efficient and effective approaches can be used to configure the parameters of a target DPD architecture across different use cases and requirements?
- **RQ2:** Can machine learning methods, such as reinforcement learning or unsupervised learning, complement or replace conventional optimization methods, and what synergies may exist between them?

1.4 Limitations

This work was carried out in an industrial research environment, where certain practical constraints affected the scope and level of detail presented in the thesis. Some implementation-specific aspects of the evaluated radio system, measurement setup, and configuration workflow are not described in full detail, since they are outside the intended academic scope of the report. Instead, the thesis focuses on the general methodology, the investigated optimization approach, and the performance trends observed from the evaluated datasets. Lastly, time also became a natural limitation of the work, meaning that some topics and ideas could not be explored further within the scope of the thesis. These aspects are discussed in more detail in the *Future Work* (Section 9).

1.5 Division of Work

This thesis has been largely done together, some material has been analyzed and studied separately but later explained and double-checked by the other. Hence, every aspect and part of the thesis can be explained and motivated by each author individually. The code that has been written has been written mostly together, with minor improvements and code-related optimizations added individually. The division of work on the paper has been quite mixed, however Arvid Bergenfeldt has put most of his effort in the Result, Methodology and Discussion, whereas Oskar Magnusson has written mostly about Introduction, Theory and about the DPD-framework. But all parts of the paper have been read through and edited by each member once the first draft is completed. Hence, no chapter or section in the paper is done by only one member.

2

Theory

This chapter will give a description of the technical background and theory relevant to this thesis. The goal is to give a broad and overall understanding of the theory, concepts and knowledge associated to the problem investigated in the thesis. It will cover the basic concepts of radio communication, power amplifiers, modeling of power amplifiers, and machine learning, to give the reader an understanding and familiarity of the relevant concepts. It also includes a more in-depth mathematical perspective to show how advanced mathematical concepts and tools can be used to describe, explain, and mitigate the complex behavior of power amplifiers in radio communication systems.

NOTE: Readers familiar with Radio Frequency (RF) transmitters and the underlying theory of DPD, may skim Section 2.1–2.4, and focus on Section 2.5–2.8, which introduces metrics, model structures, and clustering algorithms that are used for this thesis. Readers mainly interested in the related work and proposed method can continue directly to Chapter 4 and 5, after reading Sections 2.5, 2.6 and Chapter 3

2.1 The History and Evolution of Mobile Communication

During the last decades, the world has witnessed multiple generations of Mobile Communication Systems (MC). The first generation of MC emerged during the 1980s and was based on analog transmission, with the main technologies being Advanced Mobile Phone System (AMPS), developed in North America, Nordic Mobile Telephony (NMT), jointly developed by the government-controlled telephone operators in the Nordic countries, as well as Total Access Communication System (TACS) in the United Kingdom. This first generation of technology was limited mainly to voice services and made mobile telephony accessible to ordinary people, rather than only military and government use [12].

The second generation of MC started to emerge one decade later, during the 1990s, with the introduction of digital transmission on the radio link. As for the first generation, the main goal was still to enable voice communication services, but

digital transmission also allowed for limited data services. During this generation, several technologies were developed across the world. One of these technologies was Global System for Mobile Communication (GSM), developed in Europe, which gradually spread worldwide and eventually became the dominant second-generation MC technology. GSM is still used in some parts of the world today, despite the development of later mobile communication technologies [12].

The third generation of MC, most often referred to as 3rd Generation (3G), was introduced in the early 2000s. It marked the first major step toward mobile broadband by enabling faster wireless internet access and more advanced communication services. It was also during this generation that many central aspects of modern telecom systems began to take shape through the development of Wideband Code Division Multiple Access (WCDMA). Earlier MC technologies had primarily been designed for operation in paired spectrum, i.e., separate spectrum for network-to-device and device-to-network links, based on Frequency-Division Duplex (FDD). However, 3G also introduced operation in unpaired spectrum through the China-developed TD-SCDMA technology, based on Time-Division Duplex (TDD). This duplexing principle is still used in modern MC technologies such as 4G and 5G [12].

The fourth generation of MC, known as 4th Generation (4G), was introduced around 2010. The transition from 3G to 4G introduced Long-Term Evolution (LTE) as a new radio-access technology based on Orthogonal Frequency-Division Multiplexing (OFDM), enabling wider bandwidths, improved spectral efficiency, and higher data rates compared with earlier WCDMA-based systems. This was made possible by wider transmission bandwidths and more advanced multi-antenna technologies, as well as the improvements in Quadrature Amplitude Modulation (QAM), which enabled more data to be sent within the same Frequency Range (FR). Another major improvement compared to 3G was that both FDD and TDD were supported within the 4G framework, allowing a single global mobile communication technology to be adopted by a large majority of mobile-network operators worldwide [12]. This global standard was endorsed by the 3rd Generation Partnership Project (3GPP), which was established in 1998. It developed technical specifications and standards for MC, including 3G, 4G and 5G. 3GPP ensures worldwide compatibility and interoperability for all mobile and wireless communication systems. The main benefit of 3GPP standardization is that it provides a common technical framework for mobile communication systems, ensuring that equipment and networks from different vendors and operators can interoperate while satisfying the same regulatory and performance requirements [2].

Shortly after the introduction of 4G and LTE, discussions regarding the next generation of MC began to emerge. Around 2012, the development of 5G started to take shape. Compared to previous generations, the idea of 5G was to introduce a broader view and possibilities of wireless communication, supporting not only higher data rates but also a much wider range of services and application areas, as well as higher spectrum efficiency [12].

However, the development of 5G also included important steps to extend these use-cases to which LTE can be applied. For example enabling battery-life extension for user equipment thanks to energy efficient improvements. Hence, the evolution of 5G has had a focus and implementation approach that combines and improves the previous 4G LTE technology. Thus, LTE should be seen as an important part of the 5G evolution and overall radio-access. Although, despite LTE being a capable technology, it could not support all of the ambitious new use cases of 5G. To meet requirements of the new target use cases and to exploit the potential of modern MC, 3GPP initiated the development of a new radio-access technology known as New Radio (NR). NR is nowadays not only the backbone of 5G technology but also the underlying framework for the 6th Generation (6G) of Mobile Communication Systems (MC) [12].

2.2 Key 5G Use-Case Classes

To simplify the concept of 5G, it can be divided into three main classes of use-cases: enhanced Mobile Broadband (eMBB), massive Machine-Type Communication (mMTC), and Ultra-Reliable Low-Latency Communication (URLLC). These can be summarized as follows:

- *eMBB* corresponds to a continued evolution of traditional mobile broadband services, enabling larger data volumes, higher throughput, and an improved end-user experience.

- *mMTC* refers to services characterized by a massive number of connected devices, such as sensors, actuators, and monitoring equipment. The key requirements for such services are typically low device cost and low energy consumption, enabling battery lifetimes of several years.

- *URLLC* focuses on services that require very low latency and very high reliability, such as traffic safety systems, industrial automation, and automatic control applications.

Dividing 5G into these three categories is, of course, a simplification. In practice, many real-world applications span more than one class. Nevertheless, this categorization provides a broad and intuitive overview of the technical scope of 5G mobile communication and helps illustrate how different service requirements motivate different parts of the 5G framework [12].

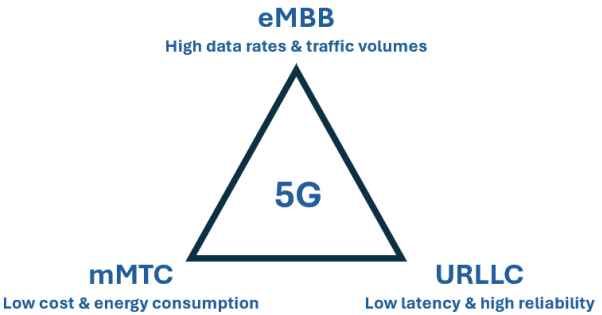


Figure 2.1 High-level 5G use-case classification. Adapted from [12]

The increasing diversity of 5G services and performance requirements also places higher demands on the underlying radio hardware, including the transmitter chain and the linearity of the power amplifier.

2.3 Signals and RF Representation

To understand how nonlinear RF effects arise in wireless transmitters, it is first necessary to introduce key concepts from signal theory that are directly relevant to radio communication. This subsection therefore introduces the mathematical representations that are most relevant for later discussions on PA behavior and digital predistortion. The presentation begins with the sinusoidal description of radio signals, followed by the complex baseband and In-phase and Quadrature (I/Q) framework, and finally the frequency-domain concepts of Fourier analysis, spectral representation, and bandwidth. Together, these concepts provide the foundation for understanding how communication signals are represented, analysed, and affected by nonlinear RF components.

Radio Signals and RF Fundamentals

Depending on the context, the term *signal* can have different meanings. However, at a fundamental level, analog signals can be represented as combinations of sinusoidal functions. A sinusoidal signal is fully characterized by its amplitude, frequency, and phase, and can be written as

$$y(t) = A\sin(2\pi ft + \phi), \tag{2.1}$$

where A denotes the amplitude, f the frequency, and ϕ the phase.

The same representation can be used to describe radio signals, which propagate as electromagnetic waves in the RF range of the radio spectrum. The existence of radio waves was experimentally demonstrated by Heinrich Hertz in the late 19th

century, which later enabled the development of radio, television, and eventually modern wireless communication [33].

However, practical wireless communication requires more than a single sinusoid. Information must be embedded into the transmitted waveform by varying its properties over time, and communication signals are therefore more naturally described using more advanced signal representations. This motivates the following discussions on complex baseband representation, Fourier analysis, and bandwidth, which together provide the mathematical foundation for later RF and PA-related modelling.

Complex Baseband and I/Q Representation

Although a sinusoidal carrier provides the fundamental structure of a radio signal, practical wireless communication requires the carrier to convey information. This is commonly done by allowing the amplitude and phase of the carrier to vary with time. A convenient way of representing such modulated RF signals is through their *complex baseband* equivalent, also referred to as the *complex envelope* [30, 15].

A bandpass signal centered around a carrier frequency f_c can be written as

$$x_{\mathrm{RF}}(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t), \tag{2.2}$$

where $I(t)$ and $Q(t)$ are referred to as the *in-phase* and *quadrature* components, respectively. These two components form an orthogonal pair and together provide a complete description of the modulated signal [30, 38].

By defining the complex baseband signal as

$$u(t) = I(t) + jQ(t), \tag{2.3}$$

the RF signal in (2.2) can equivalently be expressed as

$$x_{\mathrm{RF}}(t) = \Re\left\{u(t)e^{j2\pi f_c t}\right\}, \tag{2.4}$$

where $\Re\{\cdot\}$ denotes the real part [30, 38].

This representation is particularly useful because it separates the slowly varying information-bearing signal $u(t)$ from the rapidly oscillating carrier $e^{j2\pi f_c t}$. E.g., in the experiment of this thesis the carrier frequency (f_c) is 2.1 GHz, while the Bandwidth (BW) of the complex baseband signal is in the order of few hundreds of MHz. As a result, many operations in communication systems, such as modulation, filtering, amplification, and nonlinear modelling, can be analysed in complex baseband instead of directly at the RF carrier frequency. This greatly simplifies both mathematical analysis and practical signal processing [15, 23].

In addition, the complex baseband signal can be written in polar form as

$$u(t) = A(t)e^{j\phi(t)}, \tag{2.5}$$

where $A(t) = |u(t)|$ is the envelope and $\phi(t) = \arg\{u(t)\}$ is the instantaneous phase. This form is especially important in RF applications, since many transmitter impairments and PA nonlinearities are naturally described in terms of amplitude and phase distortions [23, 32]. The complex baseband and I/Q framework therefore forms a fundamental basis for later discussions on bandwidth, spectral behaviour, and nonlinear PA modelling.

Fourier Analysis

A central tool in signal processing and wireless communication is the Fourier transform, which makes it possible to represent a signal in the frequency domain rather than only in the time domain. While the time-domain representation describes how a signal evolves over time, the Fourier transform describes how the signal is composed of different frequency components [35, 30]. This is particularly important in RF systems, where signal bandwidth, spectral containment, filtering, and nonlinear distortion are most naturally interpreted in terms of frequency content [35, 23]. For this thesis, the main use of Fourier Analysis is to decide and meet the unwanted emission requirements targets expressed by 3GPP in terms of frequency and domain requirements. This is presented in more detail in Section 2.5.

For a continuous-time signal $x(t)$, the Fourier transform is defined as

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft}\, dt, \quad \forall f \in \mathbb{R} \tag{2.6}$$

with inverse transform

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft}\, df, \quad \forall t \in \mathbb{R} \tag{2.7}$$

The transform pair in (2.6)–(2.7) shows that a signal can be interpreted as a continuous superposition of complex exponentials, each associated with a specific frequency [35]. In this way, the Fourier transform provides the mathematical basis for analysing how signal energy is distributed over frequency.

Several important properties of the Fourier transform are especially relevant in communication systems. First, linear time shifts in the time domain correspond to phase shifts in the frequency domain:

$$x(t - t_0) \;\longleftrightarrow\; X(f) e^{-j2\pi ft_0}. \tag{2.8}$$

Second, convolution in time corresponds to multiplication in frequency:

$$x(t) * h(t) \;\longleftrightarrow\; X(f) H(f), \tag{2.9}$$

This is called *product-convolution duality*, which is fundamental for understanding filtering and channel responses [35, 30]. Third, multiplication by a complex exponential in time shifts the spectrum in frequency:

$$x(t) e^{j2\pi f_c t} \;\longleftrightarrow\; X(f - f_c), \tag{2.10}$$

which is the key mathematical operation behind frequency translation and carrier modulation [30].

Another important relation is *Parseval's Theorem*, which connects the signal energy in time and frequency:

$$\int_{-\infty}^{\infty} |x(t)|^2 \, dt = \int_{-\infty}^{\infty} |X(f)|^2 \, df, \tag{2.11}$$

where $|X(f)|^2$ is called Power Spectral Density (PSD) of the signal, which will be presented in more detail in Section 2.4. This identity is highly relevant in RF analysis, since it links waveform power and spectral power, and therefore underlies many practical performance metrics and spectrum-based evaluations [35]. In the present context, Fourier analysis provides the mathematical foundation for describing bandwidth, adjacent-channel leakage (Section 2.5), spectral regrowth (Section 2.4), and the frequency-domain effects of PA nonlinearities.

Bandwidth and Spectral Representation

As stated in Section Fourier Analysis useful way of analyzing communication signals is not only in the time domain, but also in the frequency domain. While the time-domain representation describes how a signal varies over time, the spectral representation describes how the signal energy is distributed over different frequencies [35, 30]. The BW of a signal refers to the range of frequencies occupied by the signal spectrum. There are various definitions of signal BW (e.g., 3 Decibel (dB) BW, null-to-null- BW, equivalent noise BW) but in general terms, if the signal spectrum is primarily concentrated over the interval $[f_L, f_H]$, the bandwidth can be defined as,

$$B = f_H - f_L, \tag{2.12}$$

where f_H and f_L are calculated depending on the specific definition of bandwidth that is used. In communication systems, BW is a central concept since it determines how much spectral space a signal requires for transmission. Wider transmission bandwidths are closely associated with the possibility of supporting higher data rates [30, 12]. For a complex baseband signal $u(t)$ with spectrum $U(f)$, a corresponding RF signal centered around the carrier frequency f_c occupies a frequency band around f_c with the same overall bandwidth. This makes the bandwidth concept directly relevant in both baseband and RF analysis.

In practical RF systems, spectral representation is also important because it reveals how the transmitted power is distributed around the desired carrier or channel. This makes it possible to study effects such as out-of-band radiation, adjacent-channel leakage, and spectral regrowth caused by nonlinear transmitter components. As a result, frequency-domain analysis is fundamental when evaluating PA behavior and the effectiveness of linearization techniques such as digital predistortion [35, 12].

For modern 4G and 5G signals, which often occupy relatively wide bandwidths and have to satisfy strict requirements on spectral containment, the spectral analysis becomes particularly important. It provides a direct link between signal representation, system capacity, and transmitter linearity, and therefore forms an important foundation for the later discussion of PA behavior in Section 2.4, as well as for frequency-domain performance metrics such as ACLR in Section 2.5.

Figure 2.2 presents an example of the PSD of a carrier signal around frequency f_c with a certain BW and how it is visualized in the frequency domain.

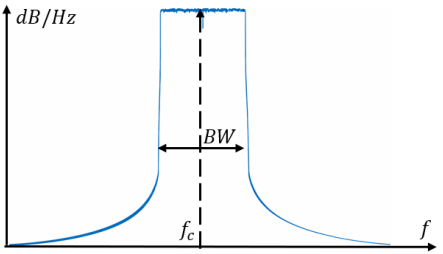


Figure 2.2 Example of PSD of a OFDM-like signal with a bandwidth BW and centered around a carrier frequency f_c.

2.4 Power Amplifier

A Power Amplifier (PA) is a fundamental electronic component in RF transmitters, where its purpose is to increase the amplitude and thereby the power of an input signal before transmission through the antenna [39]. In wireless communication systems, this amplification allows a relatively weak signal, generated within a compact device, to be boosted to a suitable transmission level such that it can be reliably detected at the receiver.

In an RF transmitter architecture, the PA is therefore one of the last components in the analog chain, before the signal is radiated through the antenna. Figure 2.3 illustrates this process in a simplified manner.

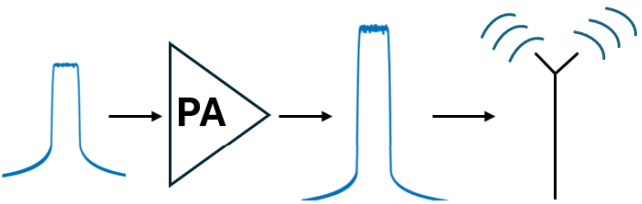


Figure 2.3 A simple example of an ideal and linear PA.

For an ideal linear PA, the output signal is simply a scaled version of the input signal, which can be written as

$$y(t) = G_{\text{PA}}\, u(t), \tag{2.13}$$

where $u(t)$ is the input signal, G_{PA} is the constant amplitude complex gain factor provided by the PA, and $y(t)$ is the amplified output signal.

However, this model describes only the ideal case and therefore constitutes a simplification of real PA behavior. In practical implementations, the amplification process is not perfectly linear. In particular, RF PAs are typically operated close to saturation to improve efficiency at the cost of increased nonlinear distortion, therefore physical HW components. within the PA, such as transistors, capacitors, and inductors, can introduce nonlinear effects. As a result, the actual output deviates from the ideal linear relationship, giving rise to distortion and reduced signal fidelity. This forms the basis for the nonlinear PA effects discussed next.

Nonlinear Effects in RF Power Amplifiers

As discussed above, an ideal PA behaves linearly, meaning that the output is an amplified but otherwise undistorted version of the input. In practice, however, real PAs exhibit nonlinear behavior, when operated close to their compression or saturation region. Under such conditions, the amplified signal is no longer a perfectly scaled copy of the input, and distortion is introduced (see example in Figure 2.4).

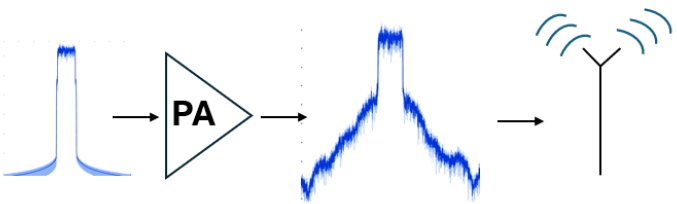


Figure 2.4 A simple example of the nonlinear effects of a PA.

This raises two natural questions, what consequences does this distortion introduce, and how can such nonlinearities be described mathematically?

At low input power levels, a PA typically behaves approximately linearly. However, as the input power increases, the PA gradually departs from linear operation and nonlinear distortion becomes more pronounced. Using the complex baseband representation introduced in Section 2.3, the input signal can be written as

$$u(t) = I(t) + jQ(t), \tag{2.14}$$

where $I(t)$ and $Q(t)$ denote the in-phase and quadrature components, respectively. The instantaneous signal power is then given by

$$P_u(t) = |u(t)|^2 = |I(t) + jQ(t)|^2 = I^2(t) + Q^2(t), \tag{2.15}$$

which describes how the signal power varies with time.

Combining this expression with the ideal linear PA model in (2.13) yields

$$|y(t)|^2 = |G_{\text{PA}}u(t)|^2 = |G_{\text{PA}}|^2 \left(I^2(t) + Q^2(t)\right) \tag{2.16}$$

which is valid as long as the PA remains in its approximately linear operating region. As the input power increases further, the PA eventually reaches its saturation (or compression) point. This is commonly illustrated by the input–output power characteristic shown in Figure 2.5.

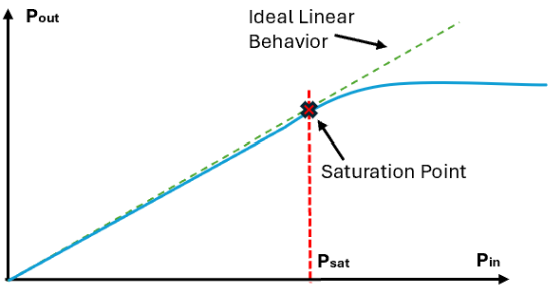


Figure 2.5 Nonlinear PA behavior illustrated through gain compression and saturation.

Before the saturation point, the PA can be approximated by the linear relation presented in Equation (2.13). Whereas beyond this region the output is more appropriately described by a nonlinear function $f_{\text{sat}}(\cdot)$:

$$y(t) = \begin{cases} G_{\text{PA}}u(t), & \text{for } P_{\text{in}} \leq P_{\text{sat}}, \\ f_{\text{sat}}(u(t)), & \text{for } P_{\text{in}} > P_{\text{sat}} \end{cases} \tag{2.17}$$

Several problematic effects arise in this nonlinear region, such as intermodulation distortion, in-band signal degradation, and out-of-band spectral regrowth. These effects may cause adjacent-channel interference and reduced communication quality. Despite this, operation close to saturation remains attractive in practice, since it improves PA efficiency. This creates a fundamental trade-off between efficiency and linearity in RF transmitter design.

To describe the nonlinear region after the compression point a simple mathematical way is to model the output as a nonlinear function of the input. In a memoryless approximation, this can be expressed using a polynomial expansion of the form

$$f_{\text{sat}}(u(t)) = a_1u(t) + a_2u^2(t) + a_3u^3(t) + \cdots, \tag{2.18}$$

where the complex coefficients $a_1, a_2, a_3, \ldots$ describe the nonlinear response of the amplifier. The linear term represents the desired amplification, while the higher-order terms give rise to nonlinear distortion. These higher-order contributions are responsible for effects such as gain compression, phase distortion, and intermodulation products, which will be presented in greater detail in Section 2.4.

Beyond the linear region, the PA gain is no longer constant and the output phase may also become amplitude dependent. A convenient and widely used way to describe these memoryless nonlinear effects is therefore through the Amplitude- and Phase-Modulation characteristics of the PA.

Amplitude and Phase Modulation

A standard way to describe memoryless nonlinear PA behavior is through Amplitude-to-Amplitude Modulation (AM/AM) and Amplitude-to-Phase Modulation (AM/PM) distortion. These functions characterize how the PA maps the *input envelope* to the *output envelope* and to an *amplitude-dependent phase shift*, respectively. The following equations explain how these functions are operating.

Let the complex baseband input be written in polar form as

$$\begin{aligned} u(t) &= A(t)\, e^{j\phi(t)}, \\ A(t) &= |u(t)| = |I(t) + jQ(t)|, \\ \phi(t) &= \arg\{u(t)\} = \arg\{I(t) + jQ(t)\}. \end{aligned} \tag{2.19}$$

A memoryless nonlinear PA can then be modeled by two real-valued static nonlinearities, an amplitude mapping $g(\cdot)$ (AM/AM) and a phase mapping $\psi(\cdot)$ (AM/PM), such that

$$y(t) = g\big(A(t)\big)\, e^{j(\phi(t) + \psi(A(t)))}. \tag{2.20}$$

AM/AM distortion describes the nonlinear relationship between input envelope $A(t)$ and output envelope $|y(t)|$

$$|y(t)| = g\big(A(t)\big). \tag{2.21}$$

For an ideal linear PA with constant gain G_{PA}, the mapping is proportional:

$$g(A(t)) = G_{\mathrm{PA}}\, A(t). \tag{2.22}$$

In practical PAs, $g(\cdot)$ deviates from proportionality as the input level increases, this is gain compression near the saturation point.

AM/PM distortion captures the amplitude-dependent phase rotation introduced by the PA. The output phase becomes

$$\arg\{y(t)\} = \phi(t) + \psi\big(A(t)\big), \tag{2.23}$$

where $\psi(A)$ is the AM/PM characteristic. For an ideal linear PA, $\psi(A) = 0$ for all A.

Together, $g(\cdot)$ and $\psi(\cdot)$ provide a compact polar description of PA nonlinearity, the envelope is distorted by AM/AM and the constellation is rotated in an amplitude-dependent manner by AM/PM. To get a visual understanding of how AM/AM and AM/PM help to describe these nonlinearities, the plots in Figure 2.6 based on measurements from an RF–PA .

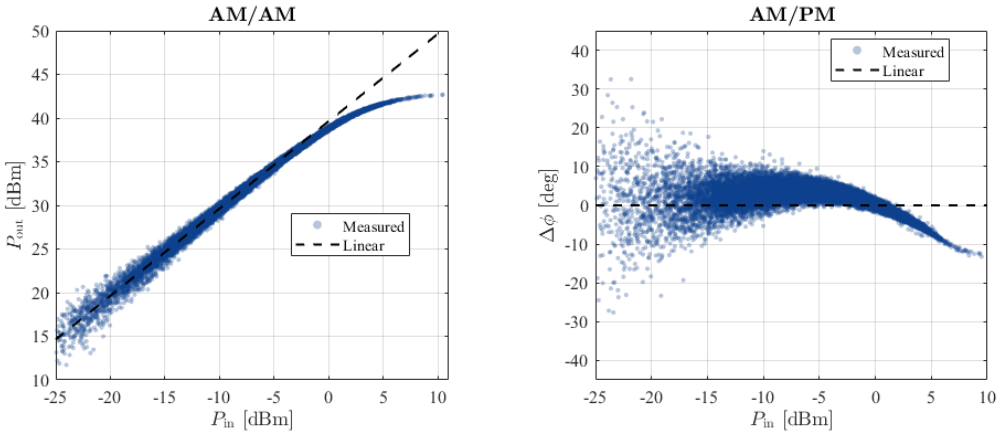


Figure 2.6 A visualization of AM/AM and AM/PM response from a RF PA.

These effects cause in-band distortion and out-of-band radiation (spectral regrowth), motivating the need of linearization methods of the PA to increase efficiency.

Spectral Regrowth

As previously stated, a nonlinear power amplifier distorts the instantaneous amplitude and phase of a signal. Therefore also changes how the signal energy is distributed in the frequency domain when operating close to its compression point. One important consequence of this is *spectral regrowth*, which refers to the widening of the output spectrum beyond the occupied signal bandwidth [11, 29]. When a modulated signal with a time-varying envelope passes through a nonlinear PA, nonlinear terms contribute to the output which generates unwanted frequency components. As previously stated, a simple odd-order memoryless polynomial description can for example be written as

$$y(t) = a_1 u(t) + a_3 u(t)|u(t)|^2 + a_5 u(t)|u(t)|^4 + \ldots, \tag{2.24}$$

where the nonlinear terms introduce additional spectral content outside the originally occupied signal band [11].

This spectral widening is due to the Convolution Theorem of the Fourier transform introduced in ((2.9)), where the time-domain products in (2.24) correspond to

convolution (and thus spectral expansion) in the frequency domain . These effects of spectral regrowth from measurements on a nonlinear PA are shown in Figure 2.7. This effect is important to mitigate for modern communication signals, since RF signals are digitally modulated waveforms with non-constant envelope and finite bandwidth. As a result, a single-carrier signal experiences spectral broadening when amplified by a PA working in its nonlinear region. In the frequency domain, this appears as energy spreading into neighboring frequency regions. This energy spreading is referred to as spectral regrowth [29], a more mathematically detailed explanation of energy distribution will be presented in Section 2.4. From a practical RF perspective, spectral regrowth is problematic because it increases out-of-band radiation and may cause interference in adjacent channels and bands. It is therefore closely related to performance metrics such as ACLR (Section 2.5) and Spectral Emission Mask (SEM) (Section 2.5). These performance metrics quantify how well the transmitted signal remains confined to its allocated frequency band [27, 29].

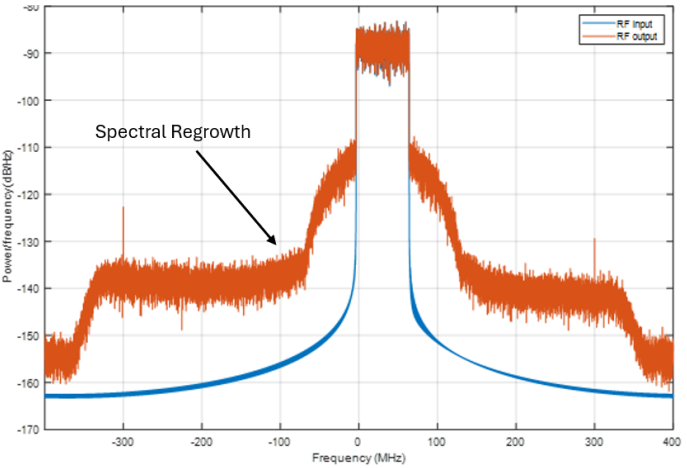


Figure 2.7 Spectral regrowth for a single carrier signal from a nonlinear PA.

However, for multi-tone and multi-carrier signals, these nonlinear spectral effects can be described more explicitly in terms of individual intermodulation products, which are introduced in the following section.

Intermodulation Distortion

Intermodulation Distortion (IMD) is associated with the amplitude modulation of a signal which contains two or more carrier frequencies, which are exposed to a nonlinear system.

The IMD the different carriers intermodulate with each other (for the same product-convolution duality principle described earlier) generating very distinct

clusters of spectral components at new frequencies. It is possible to mathematically represent where these intermodulated frequencies will emerge by the following assumptions and equations.

Let f_{c1} and f_{c2} denote two desired carrier frequencies. The two-tone input signal can be written as

$$u(t) = A_1 \sin(2\pi f_{c1} t + \phi_{c1}) + A_2 \sin(2\pi f_{c2} t + \phi_{c2}). \tag{2.25}$$

This signal is fed through a Non-Linear Time-Invariant System (NTI), RF system that can be represented using a power series expansion:

$$y(t) = f(u(t)) = a_1 u(t) + a_2 u^2(t) + a_3 u^3(t) + \dots \tag{2.26}$$

After inserting equation (2.25) into equation (2.26) one can, after some tedious calculations, expand the second-order term which yields the following,

$$\begin{aligned} u^2(t) = {} & A_1^2 \sin^2(2\pi f_{c1} t + \phi_{c1}) + A_2^2 \sin^2(2\pi f_{c2} t + \phi_{c2}) \\ & + 2A_1 A_2 \sin(2\pi f_{c1} t + \phi_{c1}) \sin(2\pi f_{c2} t + \phi_{c2}), \end{aligned} \tag{2.27}$$

using trigonometric identities, the new frequency components are generated at

$$f_{c1} \pm f_{c2}, \quad 2f_{c1}, \quad 2f_{c2}$$

Similarly, expanding the cubic term $u^3(t)$ produces third-order intermodulation components at

$$2f_{c1} \pm f_{c2}, \quad 2f_{c2} \pm f_{c1}, \quad 3f_{c1}, \quad 3f_{c2}.$$

These combinations represent the frequency location of the intermodulation products generated by the nonlinear system.

A natural question to ask is, since one can mathematically represent where these frequencies will arise, why are they so problematic for an RF system? Is it not possible to just filter them out?

This is a valid question to ask, and in real world applications most of the even-order frequencies are usually efficiently filtered out by analog filter blocks, since they are most often further away from the carrier frequencies. However, odd-order intermodulation products can appear close to the desired carriers (or in adjacent channels), meaning that aggressive filtering risks removing parts of the desired signal. The following figures will give an intuitive representation of what is happening when multiple high order intermodulation components are present.

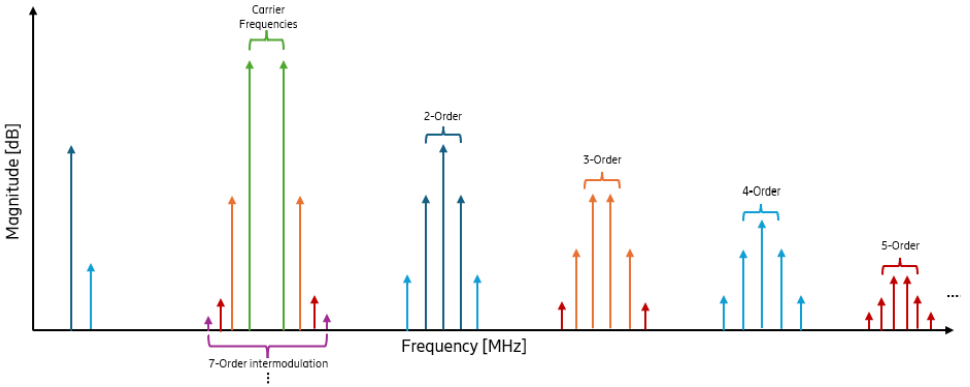


Figure 2.8 Intermodulation frequencies generated from a nonlinear PA.

From the illustration above the 3^{rd}, 5^{th}- and 7^{th}- order (odd-order) intermodulation products are close to the carrier frequencies f_{c1} and f_{c2}, which mean that they might be in-band or in the neighboring adjacent channels. As previously stated one can filter out most of the even-order intermodulation components by implementing a bandpass filter over the carrier frequencies or an analog filter in the analog domain. However, this will not filter out all of the odd-order intermodulation products close to the carrier frequency. The following figure will provide a visual representation.

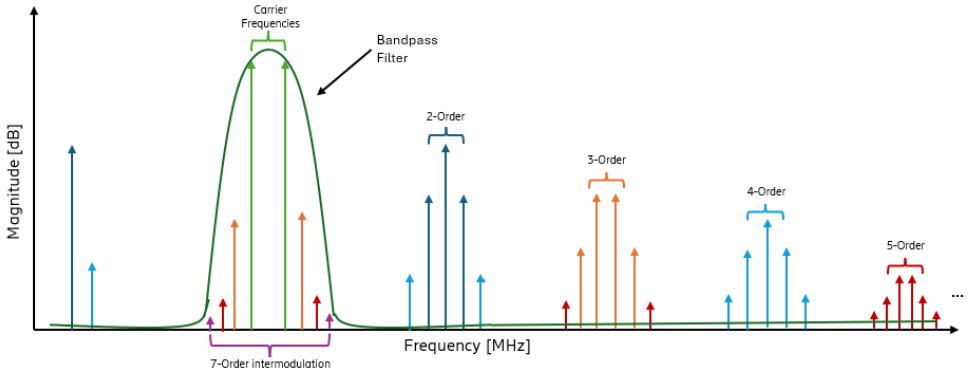


Figure 2.9 Intermodulation frequencies generated from a nonlinear PA.

The third- and fifth-order components are often among the most practically important, since they can be strong enough to affect in-band quality and adjacent-channel emissions in typical PA operating regions. A low-order polynomial model (e.g.,including up to 5th order) is therefore commonly used as a starting point to capture the dominant IMD behavior in a memoryless setting. The concepts of spectral

regrowth and IMD motivate the energy-distribution features used in the clustering method in Section 5.3.

Power Spectral Density and Energy Distribution

The previous section showed how nonlinear PA behavior gives rise to IMD products at predictable frequency locations. However, for RF analysis and DPD design, the key question is often not only *where* these distortion products appear, but also *how much signal energy or power* is distributed across those frequencies. This is important because non-linearities do not only introduce new spectral components, but also redistribute the signal energy away from the desired band and into adjacent frequencies, which is not desirable. A mathematical description of this energy or power distribution is therefore highly relevant when analyzing PA behavior, evaluating spectral regrowth, and constructing signal features for modelling and linearization [35, 34]. The following section will present the mathematics behind PSD but also Energy Spectral Density (ESD) to which gives a fundamental understanding of its use in PA behavior.

As presented in Section 2.3 for a finite-energy continuous-time signal $x(t)$ the Fourier transform is given by (2.6). From Parseval's Theorem the relationship presented in (2.11), motivates the following definition of the ESD,

$$S_x^{(E)}(f) = |X(f)|^2, \tag{2.28}$$

which describes how the total signal energy is distributed over frequency. The energy contained in any frequency interval $\mathscr{B}$ is then given by the integral of the interval set as the following,

$$E_{\mathscr{B}} = \int_{\mathscr{B}} S_x^{(E)}(f)\, df. \tag{2.29}$$

Hence, ESD provides a direct mathematical description of where the signal energy is concentrated in the spectrum [35, 34].

However in many communication and RF applications signals are better treated as *power signals* or as realizations of a stationary random processes. Hence, the more relevant quantity is the PSD, which describes how the average signal power is distributed over frequency. This is done by defining the autocorrelation function, which is mathematically presented as

$$R_{xx}(\tau) = \mathbb{E}\left[x(t)x^*(t-\tau)\right] \tag{2.30}$$

then the PSD is given by the Fourier transform of the autocorrelation function,

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau}\, d\tau, \tag{2.31}$$

with the inverse relation

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx}(f) e^{j2\pi f\tau}\, df. \tag{2.32}$$

This Fourier-pair relation is the Wiener–Khinchin theorem and forms one of the most important theoretical foundations for PSD analysis [34, 45].

An equivalent interpretation of the PSD can be obtained from finite-time observations. Let

$$x_T(t) = \begin{cases} x(t), & |t| < T, \\ 0, & \text{otherwise}, \end{cases} \tag{2.33}$$

and let its Fourier transform be

$$X_T(f) = \int_{-T}^{T} x(t) e^{-j2\pi ft}\, dt. \tag{2.34}$$

Then the PSD can be expressed as

$$S_{xx}(f) = \lim_{T\to\infty} \frac{1}{2T} \mathbb{E}\left[|X_T(f)|^2\right], \tag{2.35}$$

which shows that the PSD can be interpreted as the asymptotic power distribution of the signal in frequency. This viewpoint is particularly useful in practice, since measured spectra and periodograms are often based on finite observation windows [34, 45].

The average signal power can be recovered from the PSD by integrating over all frequencies:

$$P_x = R_{xx}(0) = \int_{-\infty}^{\infty} S_{xx}(f)\, df. \tag{2.36}$$

Similarly, the power contained within a specific frequency interval $\mathscr{B}$ is

$$P_{\mathscr{B}} = \int_{\mathscr{B}} S_{xx}(f)\, df. \tag{2.37}$$

These relations are directly relevant in RF systems, since they make it possible to quantify how much signal power remains in the desired channel and how much leaks into adjacent bands due to PA nonlinearities [34, 45]. To tie how PSD and ESD is associated with IMD the following figure represents how the energy is distributed around a dual carrier signal with 3$^{\text{rd}}$ and 5$^{\text{th}}$ order IMD products.

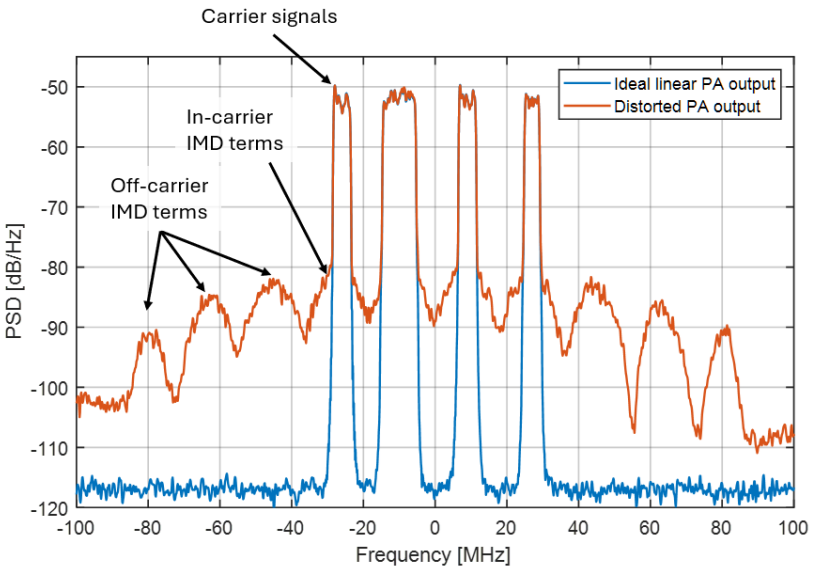


Figure 2.10 Example of PSD and ESD distribution for a dual carrier signal.

Ideally all of the energy of the carriers remains closely to the ideal linear PA output (the blue line in Figure 2.10). But as the IMD and nonlinear effects are present, the energy "leaks" outside of this region. This is *Spectral Regrowth* that was previously introduced in Section 2.4, but for a multi carrier baseband signals.

In the context of this work, PSD-based analysis provides a mathematically well-founded way of characterizing spectral energy distribution, identifying dominant occupied bands, and quantifying out-of-band distortion. These properties are closely linked to PA behavior, DPD performance and spectrum-based metrics such as Adjacent Channel Power Ratio (ACPR)/ACLR, which will be presented in Section 3.2 and 2.5 respectively. They are also highly relevant when constructing signal features that describe how energy is distributed across carriers and adjacent spectral regions [45, 35].

2.5 Performance Metrics

When evaluating, quantifying and comparing different models, a set of performance metrics is commonly used in RF PA modelling and DPD. This section introduces some useful metrics and their mathematical definition, together with a simplified example of technical specifications for base stations defined by 3GPP, which are referenced later in the report.

Normalized Mean Squared Error (NMSE)

Normalized Mean Squared Error (Normalized Mean Squared Error (NMSE)) quantifies how closely a model output y_{model} matches the measured (reference) output y_{measured}. A common definition is [42]

$$\text{NMSE(dB)} = 10\log_{10}\left(\frac{\sum_{i=1}^{M}|y_{\text{measured}}(i)-y_{\text{model}}(i)|^2}{\sum_{i=1}^{M}|y_{\text{measured}}(i)|^2}\right). \tag{2.38}$$

Compared to the standard Mean Squared Error (MSE), NMSE is normalized by the measured signal power, making it less sensitive to absolute signal scaling. This is particularly useful when comparing results across different operating points or different PAs, where the output power levels may vary.

Adjacent Channel Power Ratio (ACPR/ACLR)

Adjacent Channel Power Ratio (ACLR), often used interchangeably with ACPR, measures the amount of spectral leakage into an adjacent channel relative to the power in the main (allocated) channel. It can be expressed as [42]

$$\text{ACLR(dB)} = 10\log_{10}\left(\frac{\int_{f_{\text{adj}}}|Y(f)|^2\,df}{\int_{f_{\text{chan}}}|Y(f)|^2\,df}\right), \tag{2.39}$$

where $Y(f)$ denotes the Fourier transform of the signal, and f_{chan} and f_{adj} represent the frequency bands of the main channel and an adjacent channel, respectively (see Figure 2.11).

Unlike NMSE, which measures the overall waveform mismatch in the time domain, ACPR specifically evaluates *out-of-band* distortion. It is therefore a key metric for quantifying spectral regrowth and adjacent-channel interference in nonlinear PA operation. The ACLR definition introduced here is used as the pass/fail criterion for all sectioning methods in Section 6.4

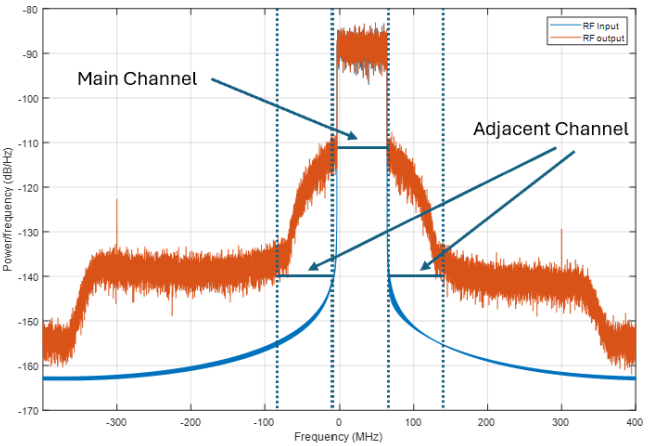


Figure 2.11 Illustration of a main channel and adjacent channels used in ACPR evaluation.

Error Vector Magnitude (EVM)

Error Vector Magnitude (EVM) is a widely used metric for quantifying the quality of digitally modulated signals. It captures the combined effect of amplitude and phase distortion in the complex I/Q plane, and therefore reflects both linear and nonlinear impairments introduced by the transmitter chain (e.g., filtering, frequency response variations, and PA nonlinearities) as well as measurement noise. [8, 9]

Conceptually, EVM is computed by comparing the received (measured) symbols to an ideal reference constellation and forming the *error vector* between them. The EVM is then defined as the Root Mean Square (RMS) magnitude of these error vectors, normalized by the reference signal power, and it is typically reported as a percentage or in dB. In a standard time-domain formulation, EVM can be expressed as [8]

$$\mathrm{EVM}_{\mathrm{RMS}} = \sqrt{\frac{\sum_{n=1}^{N} \left| x(t_n) - \left(y(t_n) \otimes e(t_n) \right) \right|^2}{\sum_{n=1}^{N} |x(t_n)|^2}}, \tag{2.40}$$

where $x(t_n)$ denotes the reference symbols, $y(t_n)$ the measured symbols, $e(t_n)$ an equalization filter (used to remove linear distortion), and N the number of evaluated symbols.

EVM is closely related to link-level performance, since high EVM typically implies degraded demodulation accuracy and increased error probability. In practice, communication standards, such as 3GPP, specify EVM limits that depend on modulation order (e.g., higher-order QAM requires lower EVM) and very specific procedures (e.g., equalization of received signal) for calculating the EVM [8].

Spectral Emission Mask (SEM)

SEM is a piecewise linear graph specifying the limit on the power over a band of spectrum that a transmitter may emit [44]. SEM is a common tool for signal processing and gives an industry standard for validation of DPD configuration. It can be used both as a binary validator (yes/no) whether a signal passes the mask, or used as a reference to compare the signal against. The figure showcases what a SEM might look like in practise, note that in this illustration the output signal is without linearisation and as a result does not meet the SEM which can be seen by the orange graph being above the SEM.

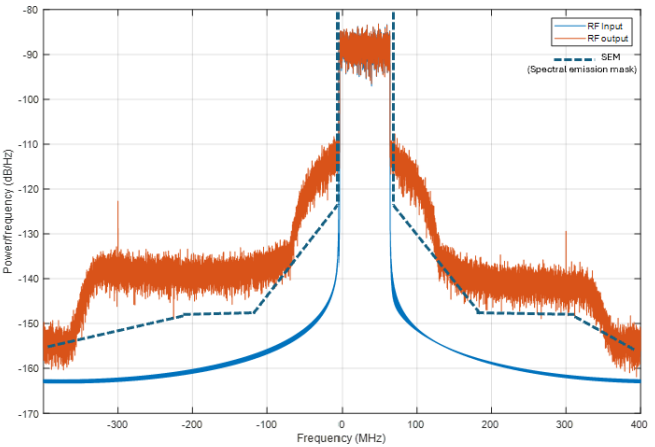


Figure 2.12 SEM for a single carrier signal.

Required Performance Protocol – 3GPP

As mentioned in Section 2.1 the development of wireless communications has developed drastically the past decades. This evolution has required common technical specifications to ensure that equipment, devices, and networks from different vendors and operators can interoperate reliably. The main organization which task is to specify the rules for all MC technologies is 3GPP. As of 2017 the technical specification protocols from 3GPP is the framework for more than 95% of the world's 7.8 billion mobile subscriptions [12].

As MC technologies develop, so does the 3GPP specification, it is an ongoing process to keep up with the advancements in the field. The 3GPP specifications referenced in this thesis specify unwanted emission limits (i.e., SEM, ACLR) and quality of the in the transmitted RF waveform (i.e., EVM).

Two example of unwanted emission requirements and signal quality for the 3GPP standard paper *Technical Specification Group Radio Access Network* [3] are,

- ACLR: ≤ -45 dB, for FR1
- EVM: $\leq 3.5\%$, for 256 QAM

Another metric used to ensure performance is NMSE, which is not a 3GPP requirement, but rather it is commonly used as an internal modelling metric for evaluating the accuracy of the PA/DPD model. In this work, an NMSE below approximately $-35\,\text{dB}$ is considered a good modelling result, while values around $-40\,\text{dB}$ or lower indicate strong model accuracy. However, the final linearization performance must still be evaluated using standardized RF metrics such as ACLR and EVM.

2.6 Nonlinear modeling

Most implementations of nonlinear models around the mid-2000s used a memoryless approach [32]. As discussed by Morgan *et al.* , such predistorters are typically implemented as instantaneous nonlinearities designed to compensate for the instantaneous nonlinear behavior of the PA [32]. But as Radio Communication (RC) evolved towards wider BW and multi-carrier signals, memoryless models became insufficient. The transition to downlink signals with wider bandwidth and consequent excitation of more dynamic effects for wider-band RF PAs eventually resulted in an industry shift from memoryless DPD models to *memory-based* ones. The key idea is to include past input samples to capture dynamic effects in the PA, providing a more accurate description of the required compensation. Memory effects become especially important in wide-band applications, where high efficiency and linearity is required and if not modeled for may degrade the system's performance drastically [41]. As a foundation model, the *Volterra series*, from which many practical behavioral memory models are derived, is introduced.

Volterra Series

The Volterra series is a general model for NTI systems with memory. Compared to memoryless polynomial expansions (e.g., a Taylor expansion), its defining feature is the ability to capture memory effects through delayed input terms [23]. A commonly used discrete-time Volterra representation can be written as

$$x_{\text{out}}(n) = \sum_{p=1}^{P} \sum_{i_1=0}^{M} \cdots \sum_{i_p=0}^{M} h_p(i_1,\ldots,i_p) \prod_{j=1}^{p} x_{\text{in}}(n-i_j), \tag{2.41}$$

where $h_p(i_1,\ldots,i_p)$ are the Volterra kernels (model coefficients), P is the nonlinear order, and M is the memory depth. The product term consists of delayed versions of the input signal. The number of parameters grows rapidly with increasing nonlinear order and memory depth, which motivates simplified model structures for practical DPD implementations.

Memory Polynomials

To address the complexity of the Volterra series, simpler model structures have been introduced that preserve the core idea but reduce the number of parameters signif-

icantly. One of the most widely used examples is the Memory Polynomial (MP) model [14].

$$x_{\text{out}}(n) = \sum_{j=0}^{M} \sum_{p=1}^{P} a_{jp} x_{\text{in}}(n-j) \left|x_{\text{in}}(n-j)\right|^{p-1}, \quad a_{jp} \in \mathbb{C}, \tag{2.42}$$

where P denotes the nonlinear order and M the memory depth.

The MP model can be viewed as a Volterra series where only the *diagonal* terms are retained, while cross-terms that mix different delays are removed. The basis functions therefore depend only on a single delayed input sample at a time.

Generalized Memory Polynomials

The Generalized Memory Polynomial (GMP) model extends the memory polynomial by introducing additional cross-terms between different delayed samples. This increased flexibility allows the model to capture more complex nonlinear behavior than a basic memory polynomial, which can be particularly important for wider bandwidths and multi-carrier signal [41]. Compared to the MP, the GMP model offers increased modelling capability by capturing additional interactions between delayed samples. This typically improves accuracy for signals where memory effects are significant, but it also increases the number of parameters and computational complexity. The choice between MP and GMP is therefore often guided by a performance–complexity trade-off. A commonly used GMP formulation is the following.

$$\begin{aligned} x_{\text{out}}(n) = &\sum_{j=0}^{M} \sum_{p=1}^{P} a_{jp} x_{\text{in}}(n-j) \left|x_{\text{in}}(n-j)\right|^{p-1} \\ &+ \sum_{k=1}^{K} \sum_{j=0}^{M} \sum_{p=1}^{P} b_{kjp} x_{\text{in}}(n-j) \left|x_{\text{in}}(n-j-k)\right|^{p-1} \\ &+ \sum_{k=1}^{K} \sum_{j=0}^{M} \sum_{p=1}^{P} c_{kjp} x_{\text{in}}(n-j) \left|x_{\text{in}}(n-j+k)\right|^{p-1}. \end{aligned} \tag{2.43}$$

The first term corresponds to the standard memory polynomial (2.42). The additional triple-sum terms introduce a lag parameter k that shifts the magnitude term relative to the complex sample. This seemingly small modification enables interactions between different time delays, resulting in a richer model structure that can better represent memory effects observed in practical PAs, especially as signal bandwidth increases [41, 32].

When using memory-based behavioral models such as GMP, two practical challenges commonly arise. The first is choosing suitable model dimensions, such as the memory depth and nonlinear order. In many cases, not all basis terms are required, and only a subset of coefficients contributes meaningfully to the observed behavior [32]. The second challenge is estimating the model coefficients, which is often the

more demanding step in practice since it typically relies on measured PA data and may need to track time-varying operating conditions. In this work, the focus is limited to the first problem, i.e., selecting and characterizing the model structure rather than estimating the coefficients in real time.

Iterative Learning Control (ILC)

A seemingly unrelated field which has practical applications as a reference when doing simulation work in DPD and PA modeling comes from the field of control theory. Iterative Learning Control (ILC) is a technique for improving transient response and tracking performance in systems that execute the same operation repeatedly over a fixed time interval. The fundamental idea is to use information from previous iterations in order to improve the performance in the next one. ILC is commonly used in applications such as robotics, mechanical systems, the chemical industry, and aerodynamic systems. As a result, the exact ILC formulation may differ slightly depending on the field of application [10].

From the Lund Institute of Technology (LTH) presentation by Bo Bernhardsson and Karl Johan Åström [7], a basic ILC problem can be represented as

$$\begin{aligned} y(t) &= Tu(t), \\ e(t) &= r(t) - y(t), \\ u_{\text{new}}(t) &= u(t) + L(q)e(t) \end{aligned} \tag{2.44}$$

where T denotes the system, $r(t)$ the reference signal, $e(t)$ the tracking error, and $L(q)$ the learning filter. If T is linear and disturbances are neglected, then the ideal choice $L = T^{-1}$ gives $e_{\text{new}}(t) \equiv 0$.

For repetitive operation, the process at iteration k can be described by

$$\begin{aligned} y_k(t) &= T_r(q)r(t) + T_u(q)u_k(t), \\ e_k(t) &= r(t) - y_k(t), \end{aligned} \tag{2.45}$$

and a commonly used ILC update law is

$$u_k(t) = Q(q)\left[u_{k-1}(t) + L(q)e_{k-1}(t)\right], \tag{2.46}$$

where $Q(q)$ and $L(q)$ are linear filters, which do not need to be causal [7].

Although ILC is generally associated with control theory, it is also relevant for RF applications such as PA linearization. The reason is that ILC can be used to iteratively approximate the inverse behavior of a dynamical system. In the context of DPD, this means that instead of directly identifying the predistorter parameters from the beginning, ILC first searches for the optimal PA input signal that drives the amplifier output as close as possible to a desired linear response. Only after this optimal input response has been found are the parameters of the predistorter estimated using a regression method, for example Least Squares (LS) [10].

Consider a discrete-time nonlinear system $\mathscr{F}$, representing the PA, such that

$$y(n) = \mathscr{F}[u(n)]. \tag{2.47}$$

Here, $u(n)$ is the PA input signal and $y(n)$ is the corresponding output signal. The objective is to find an optimal input signal $u^*(n)$ such that the PA output matches a desired response $y_d(n)$, i.e.,

$$\mathscr{F}[u^*(n)] \approx y_d(n). \tag{2.48}$$

In practice, exact equality is generally not achievable. Instead, the goal is to iteratively reduce the output error according to a suitable norm until the error is sufficiently small, at which point the method is said to have converged [10, 7].

For PA linearization using ILC, all signals are assumed to be defined over a finite discrete-time interval

$$n \in [0, N-1], \tag{2.49}$$

where N is the number of samples. At iteration k, the PA is excited by an input signal $u_k(n)$, which produces an output signal $y_k(n)$. The corresponding residual tracking error is then defined as

$$e_k(n) = y_d(n) - y_k(n). \tag{2.50}$$

This residual tracking error is then used to construct an improved input signal for the next iteration, which can be written as

$$u_{k+1}(n) = \mathscr{F}_L[u_k(n), e_k(n)], \tag{2.51}$$

where $\mathscr{F}_L(\cdot)$ denotes the learning update law [10]. Equivalently, the ILC process generates a sequence of input signals of the form

$$u_{k+1}(n) = \mathscr{F}_L[u_k(n), y_d(n) - y_k(n)], \quad n \in [0, N-1] \tag{2.52}$$

such that $u_k(n)$ converges to a fixed optimal point $u^*(n)$ that minimizes the tracking error.

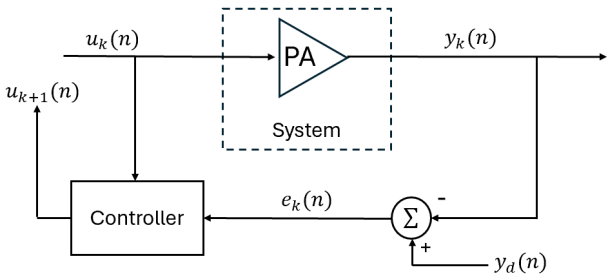


Figure 2.13 ILC scheme for a PA in an RF application.

Applying the ILC framework to PA linearization leads to the vector update expression

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \Gamma \mathbf{e}_k, \tag{2.53}$$

where

$$\mathbf{u}_k = [u_k(0), u_k(1), \ldots, u_k(N-1)]^T, \tag{2.54}$$

and

$$\mathbf{e}_k = [e_k(0), e_k(1), \ldots, e_k(N-1)]^T. \tag{2.55}$$

Here, Γ is the learning matrix that controls the convergence speed of the algorithm [10]. A deeper discussion of convergence conditions and learning algorithm design is outside the scope of this thesis, but is treated in detail by Chani-Cahuana et al. [10].

This highlights an important conceptual difference between ILC-based linearization and standard GMP-based DPD identification. In conventional DPD, the main objective is to directly estimate the predistorter coefficients. In ILC-based linearization, the first step is instead to iteratively identify the optimal PA input waveform $u^*(n)$. Only after this waveform has been found are the parameters of the predistorter estimated using a standard regression method, for example LS [10].

This provides target input–output data for fitting the predistorter, which is otherwise not directly available beforehand. This makes ILC not only a linearization tool, but also a useful framework for studying and identifying suitable predistorter structures for strongly nonlinear PAs [10].

Machine Learning Based PA Models

As with most things, machine learning (Machine Learning (ML)) has also been a part of the world of PA modeling. One of the earliest adaptions within this field used a Neural Network (NN) based approach and found it to be suitable for the then industry standard 3G telecommunication [16]. Since then more methods have been proposed and a comparison of the most common ML based methods was presented in 2019 where these were compared to the standard MP and GMP model [13]. Their result showed that out of the models presented Gradient Boosted Decision Trees (GBDT) performed the best, yielding similar or even better performance in the single carrier case that was studied. However, to our knowledge the experiment is yet to be remade using more complex carrier cases. The next chapter will dive more into the topic of Machine Learning and present the models necessary to understand the proposed method that does not target direct modeling of DPD or PA via ML but uses ML to improve DPD model selection.

2.7 Fundamentals of Machine Learning

ML is a subset of Artificial Intelligence (AI), as it focuses more specifically on enabling a computer to learn patterns from data without being explicitly programmed

with task-specific rules. ML is about learning and making predictions or decisions based on the data presented. Most ML algorithms rely on principles from statistics and optimization theory, which provide a robust foundation for learning. As a result, *in general*, the more high-quality data an ML algorithm is exposed to, the more accurately it can generalize and make predictions when presented with new, unseen data.

This makes ML a powerful tool with applications across a wide range of domains. In fact, ML is encountered in everyday life, often without the user being aware of it. Examples include item recommendations on websites, music suggestions, auto-correct spelling and word recommendation, medical diagnostics, weather patterns, traffic prediction and route optimization, and of course Large Language Models (LLM) chatbots such as ChatGPT. All of these applications rely, in one way or another, on ML, which highlights its usefulness and impact across a multitude of fields.

ML algorithms can be divided into three main categories:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Each category has its own strengths and areas of application, and the choice of which category to use depends on the specific problem that the ML algorithm is intended to solve. This section provides a brief presentation of the fundamentals of each category to give the reader an overview of how ML is used throughout this paper.

Supervised Learning

Supervised Learning (SL) is one of the most fundamental categories of ML, and it is often the first one encountered when learning about machine learning. In many practical applications, the relationship between an input x and an output y is difficult (or impossible) to describe explicitly using hand-crafted rules. Instead, SL learns this relationship from data that contains examples of observed input–output pairs (x_i, y_i) by applying some mathematical model/method. In other words, supervised learning amounts to *learning from examples*, where each input sample x_i is accompanied by a corresponding label y_i.

Given a set of labeled data, the goal is to fit a model that can predict the output for new, unseen inputs. Depending on the type of output, SL problems are typically divided into two main subtypes: *regression*, where y is a numerical value, and *classification*, where y belongs to a finite set of categories. Some simple examples when regression and classification are used:

- **Regression:** Predicting a continuous numerical value, e.g., estimating a house price based on input features such as living area, number of rooms, year of construction, and location. Here, the output y is a real-valued quantity.

- **Classification:** Predicting a discrete category, e.g., determining whether an email is *spam* or *not spam* based on features such as words in the subject line, the sender information, and the presence of links. Here, the output y belongs to a finite set of classes, in this case $\{spam,\ not\ spam\}$.

In practice, the available data is commonly divided into *training*, *validation*, and *test* sets. The training set is used to learn the mapping from x to y. The validation set is used during development to evaluate performance on unseen data and to tune model choices (e.g., model complexity and hyperparameters), producing output predictions $\hat{y}$. Finally, the test set is used as a final and unbiased evaluation to verify that the model generalizes well beyond the data it has been trained and tuned on [28].

Unsupervised Learning

Unsupervised Learning (UL) is a category of ML where the data is *unlabeled*, meaning that an observed input x does not have a corresponding output label y. In many applications, it is either difficult, expensive, or impossible to obtain labels, and the goal is therefore not to predict a known target value. Instead, UL aims to discover structure, patterns, or regularities that are inherent in the data itself. In this sense, unsupervised learning can be viewed as *learning from data without explicit answers*.

Given a dataset consisting of samples $\{x_i\}_{i=1}^{n}$, UL methods try to summarize or organize the data in a useful way. Common tasks include *clustering*, where the goal is to group similar data points together, and *dimensionality reduction*, where high-dimensional inputs are mapped to a lower-dimensional representation that preserves the most important information. These methods are often used for visualization, compression, feature extraction, and to gain intuition about complex datasets.

In practice, the performance of UL algorithms is typically assessed more indirectly than in supervised learning, since there is no ground-truth label to compare against. Instead, evaluation is often based on how well the discovered structure supports the intended purpose, e.g., whether clusters are meaningful, or whether a reduced representation retains relevant information. Unsupervised learning is therefore a useful tool when the goal is exploration, representation learning, or data-driven structure discovery rather than direct prediction [28].

Reinforcement Learning

Reinforcement Learning (RL) is a third major ML paradigm, alongside supervised and unsupervised learning, and it is centered around *learning from interaction*. Instead of learning from labeled input–output pairs (x, y), an RL agent interacts with an environment over time and learns how to choose actions in order to achieve a goal. The learning signal is provided in the form of a numerical *reward*, which reflects how desirable an outcome is.

The core idea of RL is to learn a mapping from situations to actions that maximizes the accumulated reward. The agent is not told which action is correct in a

given situation, instead, it must discover which actions lead to high reward through *trial and error*. In many practical settings, an action can influence not only the immediate reward but also the next situation and, consequently, future rewards. This introduces the important concept of *delayed reward*, which makes RL fundamentally different from standard supervised learning.

A key challenge in RL is the trade-off between *exploration* and *exploitation*. Exploitation means selecting actions that have previously yielded high reward, while exploration means trying actions that are less certain in order to gather more information and potentially find better strategies. Balancing these two behaviors is essential for effective learning, especially in uncertain or stochastic environments.

In summary, RL addresses goal-directed decision making under uncertainty, where learning takes place through sequential interaction with an environment. This makes RL particularly relevant for control and optimization problems where explicit labeled examples of correct behavior are difficult to obtain, but where performance can be evaluated through a reward signal [46].

2.8 Machine Learning Methods for Clustering and Dimensionality Reduction

In this section a deeper analysis of specific ML algorithms will be presented. These are relevant for this thesis and have been implemeted, explored and evaluated during this project.

K-Means – Unsupervised Clustering

The k-means method is a fundamental algorithm for UL. It is relatively simple and intuitive, and it is often a good first method to consider due to its low computational cost and straightforward implementation. The main idea is to partition a dataset into a fixed number of clusters. The number of clusters is denoted by K. K-means performs *hard clustering*, meaning that each data point is assigned to exactly one cluster. [28]

Given data points $\{\mathbf{x}_i\}_{i=1}^n$, the method partitions the dataset into κ distinct clusters $k_1, k_2, \ldots, k_K$ collected in a set $\kappa = \{k_1, \ldots, k_K\}$, where each $\mathbf{x}_i$ belongs to exactly one cluster. The clusters are chosen to minimize the sum of pairwise squared Euclidean distances within each cluster, which can be written as

$$\arg\min_{k_1,k_2,\ldots,k_K} \quad \sum_{i=1}^{K} \frac{1}{|k_i|} \sum_{\mathbf{x},\mathbf{x}' \in k_i} \|\mathbf{x} - \mathbf{x}'\|_2^2, \tag{2.56}$$

where $|k_i|$ is the number of data points in cluster k_i. This is also known as the Within-Cluster Sum of Squares (WCSS). The intention of this equation is to select the clusters such that all points within each cluster are as similar as possible. It can be

shown that this process is equivalent to selecting the clusters such that the summed distance from all data points to their respective cluster centres is minimized:

$$\arg \min_{k_1,k_2,\ldots,k_K} \sum_{i=1}^{K} \sum_{x \in k_i} \|x - \mu_i\|_2^2, \tag{2.57}$$

where $\hat{\mu}_i$ is the centre of cluster k_i, also known as a centroid, defined as the mean of all data points $\mathbf{x}_i \in k_i$.

Since K-Means is a "hard" clustering algorithm, meaning that every point will be designated to one, and only one, cluster. It is important to find the optimal number of clusters K. To find the optimal number of cluster there are a number of different algorithms to consider, but two key algorithms are the *Elbow Method* and the *Silhouette Method* [28].

Elbow Method. The elbow method is widely used in clustering to find the number of clusters relevant describing the data as well as possible. The elbow method is varying the number of clusters (K) in K-Means from one to K, where for each value of K the WCSS is calculated. Then one can plot WCSS against the number of cluster K, this will result in a curve that resembles an elbow. Due to as the number of clusters increase the WCSS value will start to decrease. Which is logical as the more clusters present the more points are closer to the centroids which implies that the value of total sum of squares among all centroids start to decrease.

This can be visiualized by a simple plot and example as the following.

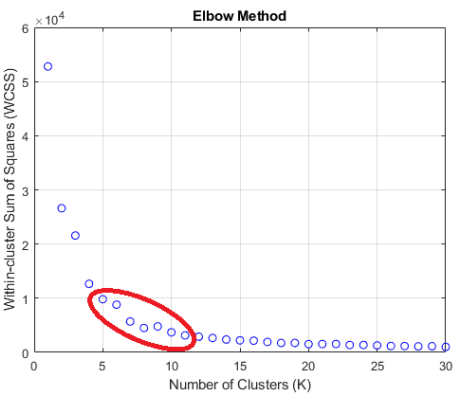


Figure 2.14 Illustration of the Elbow function, from our own data

From the figure the number of clusters to choose for K-means is located at the "elbow" of the plot, which is marked within the red region. By choosing the number of clusters from this region one gets a suitble number of clusters to for a specific dataset [28, 37].

However, this is not a guarantee to get an optimal performance but rather a "good enough", or an quantitative guess, for the amount of clusters, hence one usually summarize the result from multiple different cluster optimization methods to make a valid cluster estimation. Another cluster optimization method is *Sillhouette* which focuses on the internal cohesion and seperation of clusters, which quantifies how similar an object is to other surounding clusters, it works as the following.[37, 26]

Silhouette Method. The silhouette method is based on calculating the average distances between datapoints in the same cluster compared to the average distance between data points in other cluster. Define K clusters of data, $x(i)$ and $a_{x(i)}$ the average distance between $x(i)$ to every data point in the cluster K. The notation b_x is the minumum average distance between $x(i)$ and every data point in other clusters that are not a member in K. The calculation of the silhouette coefficients of $x(i)$, the silhouette average of each cluster, and the silhouette average of all clusters can be calculated by the following equation.

$$S_{x(i)} = \frac{b_{x(i)} - a_{x(i)}}{\max(a_{x(i)}, b_{x(i)})}, \tag{2.58}$$

where $x(i) =$ data points in the cluster, $i = 1, 2, 3, \ldots, n$
$a_{x(i)} =$ average distance between x_i and every data point in the same cluster and
$b_{x(i)} =$ minimum average distance between x_i and every data point in other clusters

$$S_k = \frac{1}{n} \sum_{i=1}^{n} S_{x(i)}, \tag{2.59}$$

where k is the number of clusters and n is the number of data points in the same cluster,

$$S_{avg} = \frac{1}{m} \sum_{k=1}^{m} S_k, \tag{2.60}$$

where m is the number of clusters. The silhouette score is then provided in a range between -1 and 1, where lower $a_{x(i)}$ and larger $b_{x(i)}$ values contribute to a higher silhouette score, which is interpreted as higher-quality clustering. I.e. points are strongly tied to their allocated cluster and well separated from other clusters. Hence, a larger silhouette score implies a more homogeneous clustering result [26]. The following figure is an example of how a silouette plot can look like.

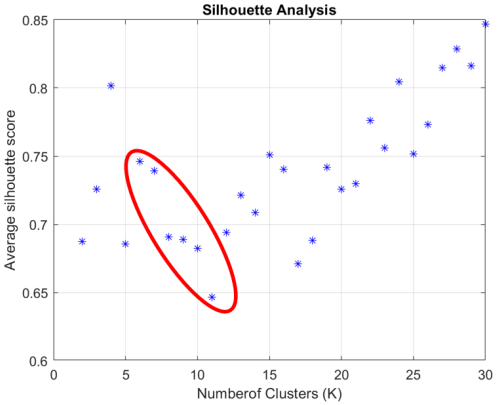


Figure 2.15 Illustration of the Silhouette function, from our own data

The red region contains the same datapoints as the red region in Figure 2.14. Hence, it showcase that the result of these cluster optimization algorithms should work in symbiosis when evaluating an arbitrary number of clusters, one does not simply exclude the other.

Gaussian Mixture Models – Unsupervised Clustering

The GMM is a probabilistic method for unsupervised clustering, based on the assumption that the observed data is generated from a mixture of K Gaussian distributions. Each Gaussian component represents one cluster in the data and the full dataset is modeled as a weighted combination of these components. Compared to methods such as K-means, GMM provides a more flexible representation of the data since each cluster is described not only by a center, but also by a covariance structure that captures its spread, orientation, and shape. Hence, GMM is a so called *soft clustering algorithm* whereas K-means is a *hard clustering algorithm*, due to its forced Euclidean based clustering [28].

Each Gaussian component is characterized by three parameters, a mean vector μ_k, a covariance matrix Σ_k and a mixing weight π_k, where

$$\sum_{k=1}^{K} \pi_k = 1, \qquad \pi_k \geq 0 \tag{2.61}$$

The mean and covariance describe the location and geometry of the component, while the mixing weight represents the probability that a randomly selected data point originates from that specific Gaussian.

Under the GMM assumption, the probability density of a data point x_n is given

by the mixture density

$$p(x_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k), \tag{2.62}$$

where the multivariate Gaussian density is

$$\mathcal{N}(x_n \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k)\right), \tag{2.63}$$

and d denotes the dimensionality of the data.

A convenient probabilistic interpretation is obtained by introducing a latent cluster variable indicating which Gaussian component generated each point. In that case, the joint density of a point and its latent cluster assignment is

$$p(x_n, k) = p(x_n \mid k)\, p(k) = \mathcal{N}(x_n \mid \mu_k, \Sigma_k)\, \pi_k. \tag{2.64}$$

Marginalizing over all possible latent cluster assignments then recovers the mixture density in (2.62).

The model parameters are estimated using the Expectation Maximization (EM) algorithm. In the E-step, the algorithm computes the posterior probability that component k generated point x_n. This quantity is referred to as the *responsibility* and is given by

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}. \tag{2.65}$$

The responsibility can be interpreted as the probability that data point x_n belongs to Gaussian component k, given the current parameter values.

In the M-step, these responsibilities are used to update the model parameters. First, the effective number of points assigned to component k is computed as

$$N_k = \sum_{n=1}^{N} \gamma_{nk}. \tag{2.66}$$

The updated mean, covariance, and mixing weight are then given by

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n, \tag{2.67}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T, \tag{2.68}$$

$$\pi_k = \frac{N_k}{N}. \tag{2.69}$$

These two steps are repeated iteratively until convergence. In the unsupervised setting, the parameter estimation problem is commonly formulated as a maximum-likelihood problem,

$$\hat{\theta} = \arg\max_{\theta} \log p(\{x_n\}_{n=1}^{N} \mid \theta), \tag{2.70}$$

where $\theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$ denotes the full set of model parameters. Using the mixture model in (2.62), this leads to the log-likelihood function

$$L(\mu_k, \Sigma_k, \pi_k) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right). \tag{2.71}$$

As the EM iterations proceed, this log-likelihood typically increases until little or no further improvement is obtained [28].

The key difference between GMM and k-means is that GMM performs *soft clustering*, meaning that each data point is assigned a probability of belonging to each cluster rather than being forced into exactly one cluster. In addition, the covariance matrices allow GMM to represent clusters with more general shapes than the roughly spherical clusters often associated with k-means. This makes GMM particularly attractive when the cluster structure is uncertain, overlapping, or anisotropic. This distinction is showcased below with a figure comparing the two cluster methods.

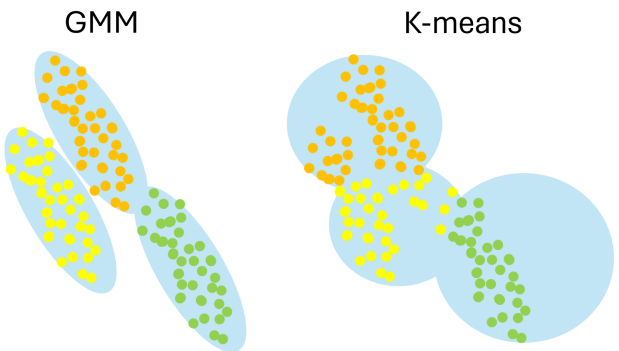


Figure 2.16 Simple comparison of K-means and GMM

Another important practical aspect is that the number of mixture components K must be chosen before training, and that the optimization problem is non-convex. As a result, different initializations may lead to different local optima. In practice, it is therefore common to run the algorithm multiple times with different initializations and retain the solution with the highest achieved log-likelihood [28].

Bayesian Information Criterion – BIC. As discussed in Section 2.8, the number of Gaussian components K must be selected before fitting a GMM. If K is chosen

too small, the model may fail to capture important structure in the data. If K is chosen too large, the model may instead overfit the data by describing noise or small variations as separate clusters. Therefore, a model selection criterion is needed to balance goodness of fit against model complexity.

The Bayesian Information Criterion (BIC) is a widely used criterion for this purpose and is commonly applied in model-based clustering with Gaussian mixture models [43, 21]. For a fitted model, the BIC can be written as

$$\mathrm{BIC} = -2\log L(\hat{\theta}) + q\log N, \tag{2.72}$$

where $L(\hat{\theta})$ is the maximized likelihood, q is the number of free parameters in the model, and N is the number of data points. The first term rewards models that fit the data well, while the second term penalizes unnecessary model complexity.

In practice, several GMMs are fitted using different values of K, and the corresponding BIC values are compared. Since adding more Gaussian components generally increases the likelihood, the penalty term is important for avoiding overly complex models. The selected number of components is therefore chosen as

$$K^* = \arg\min_K \mathrm{BIC}(K). \tag{2.73}$$

For a GMM with full covariance matrices in d dimensions, the number of free parameters is

$$q = Kd + K\frac{d(d+1)}{2} + (K-1), \tag{2.74}$$

where the three terms correspond to the mean vectors, covariance matrices, and independent mixing weights, respectively. Thus, BIC provides a practical way of selecting K by comparing how well each GMM explains the data while penalizing models with too many parameters. However, the selected value should still be interpreted as a statistical guideline rather than an absolute truth, and should be considered together with visual inspection and the practical interpretability of the resulting clusters [43, 21].

DBSCAN Clustering - Unsupervised Clustering

Density-Based Spatial Clustering of Applications (DBSCAN) is an unsupervised clustering algorithm designed to identify clusters based on local point density rather than predefined cluster centroids or a fixed geometric shape. Hence it is different from K-Means which is a Euclidean-based clustering algorithm and GMM which is a probabilistic clustering algorithm. A major strength of DBSCAN is that it can discover clusters of arbitrary shape while simultaneously identifying points that do not belong to any cluster, i.e., noise points [19].

The fundamental idea behind DBSCAN is that points belonging to the same cluster should lie in regions of sufficiently high density, while points in sparse regions should be classified as noise. To formalize this, the algorithm introduces the

ε-neighborhood of a point p,

$$N_\varepsilon(p) = \{q \in D \mid \text{dist}(p,q) \leq \varepsilon\}, \tag{2.75}$$

where ε defines the neighborhood radius and $\text{dist}(p,q)$ denotes the chosen distance measure between two points [19].

A point p is said to be *directly density-reachable* from a point q with respect to ε and MinPts if

$$p \in N_\varepsilon(q) \quad \text{and} \quad |N_\varepsilon(q)| \geq \text{MinPts}, \tag{2.76}$$

meaning that q lies in a sufficiently dense region. Thus, by such relations the notion of *density-reachability* is obtained, and clusters are then defined as maximal sets of density-connected points [19].

In practice, DBSCAN proceeds by selecting an arbitrary point and examining its ε-neighborhood. If the number of points in that neighborhood is at least MinPts, the point is treated as a *core point* and a new cluster is expanded by iteratively including all points that are density-reachable from it. If the neighborhood is too sparse, the point is labeled as noise (unless it is later found to belong to the neighborhood of another cluster) [19].

Compared to partition-based methods such as k-means, DBSCAN does not require the number of clusters to be specified in advance. This makes it particularly attractive in applications where the cluster structure is unknown beforehand or where the data may contain irregularly shaped groups and outliers [19]. The following figure gives a visual representation of how DBSCAN is able to find alternate shapes that follows the data point density pattern.

(a) DBSCAN clusters from example data. **(b)** K-Means clusters from example data.

Figure 2.17 Illustration of DBSCAN clustering. Adapted from [31].

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a popular multivariate statistical technique and is used in almost every major field of science, it is a mathematical tool to reduce the dimensions of a set of data points. The main idea of PCA is to apply an orthogonal linear transformation to project onto a new coordinate system. This results in the new coordinate system capturing the greatest variance by some scalar projections of the data. This scalar projection of the data is called a Principal Component (PC) [40].

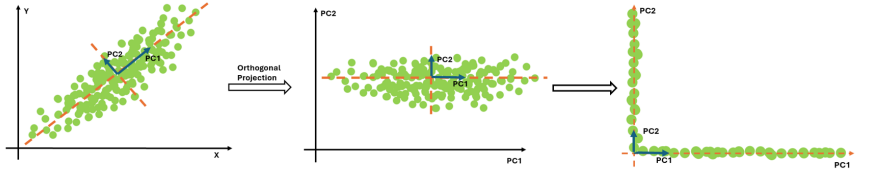


Figure 2.18 2D example of PCA and its orthogonal linear transformation.

The advantage of this is that noise and features that add little to no value to the data is removed, hence leaving only the most statistically significant features left. This reduces the dimensionality of the data set, making it more robust by removing non-relevant features for its implementation usecase. Making PCA useful for noise reduction as well as preprocessing before clustering or other machine learning methods [40].

The key mathematical framework for PCA is based on the eigenvalues and eigenvectors of the covariance matrix of the dataset. PCA then selects the most significant principal components with the largest eigenvalue and projects the data to the simplified PCA coordinate system. This is visualized in 2.18 in a simplified example of how PCA adapts its linear transform projection. The following equations gives a mathematical descripion of how PCA works [40],

Consider a dataset

$$X \in \mathbb{R}^{N \times p}, \tag{2.77}$$

where N is the number of observations and p is the number of features. The data is first normalized and centered

$$X_z = \frac{X - \bar{X}}{\sigma}, \tag{2.78}$$

where $\bar{X}$ denotes the mean of the dataset and σ the standard deviation, this is also known as Z-score normalization. In practice, to normalize or standardize the data before applying PCA is fundamental, especially when they have different physical scales, to ensure that every feature of the dataset contributes equally to the analysis [40].

Secondly, the covariance matrix of X is calculated on the centered data, which showcase how much one feature varies with repect to another, which is given by

$$\Sigma = \frac{1}{N-1} X_z^T X_z. \tag{2.79}$$

After the Covariance Matrix has been calculated the next step is to calculate the PC, as previously stated this is done by calculating the eigennvalues and eigenvectors of the Covariance matrix.

$$\Sigma v_i = \lambda_i v_i, \tag{2.80}$$

where λ_i and v_i are the eigenvalues and eigenvectors of the covariance matrix, respectively. The eigenvectors define the principal directions, while the eigenvalues quantify how much variance is captured along each direction [40].

To quantify how much of the total variation each principal component explains, the Explained Variance Ratio (EVR) is used. For the ith principal component, this is defined as

$$\mathrm{EVR}_i = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}, \tag{2.81}$$

which gives the proportion of the total dataset variance captured by the ith principal component. The Cumulative Explained Variance (CEV) of the first m principal components can similarly be written as

$$\mathrm{CEV}(m) = \sum_{i=1}^{m} \mathrm{EVR}_i = \sum_{i=1}^{m} \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}. \tag{2.82}$$

This metric is commonly used to determine how many principal components should be used in the reduced representation. Usually the first few PCs are often selected such that they explain a sufficiently large proportion of the total variance, while the remaining lower-variance components are discarded [40]. The following figure illustrates how CEV can look

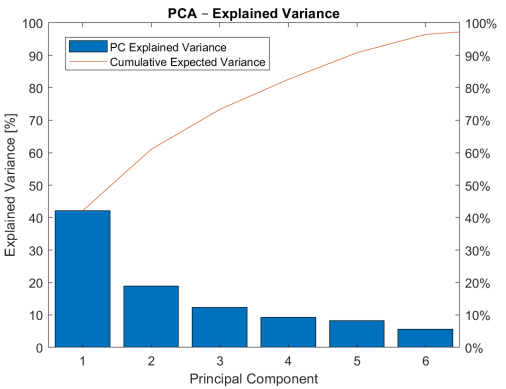


Figure 2.19 Example of Cumulative Explained Variance (CEV) with 6 PC's explaining $> 95\%$ of Explained Variance.

After sorting the eigenvalues in descending order and selecting the first m number of PCs based on their EVR, the corresponding eigenvectors are collected into a matrix

$$W_m = [v_1\ v_2\ \ldots\ v_m]. \tag{2.83}$$

The dataset can then be projected onto the reduced PCA space as

$$Z = X_z W_m, \tag{2.84}$$

where Z is the lower-dimensional representation of the original data [40].

Thus, PCA does not perform clustering by itself, but instead provides a transformed and often more compact feature space in which the dominant variation of the data is preserved. This makes PCA a useful preprocessing algorithm before applying unsupervised clustering methods such as GMM, DBSCAN, or K-means. To be able to reduce the dimension of the dataset before applying unsupervised clustering algorithms is key to ensure robust clustering in ML. Reducing the dimensions of the dataset also has other practical applications such as improving visualization however important to note that the interpretation of the axes changes when plotting the PCA components as suppose to the original features.

3

Technical background to Digital and RF Transmitter Architecture

Modern radio transmitter architectures are based on a close interaction between digital baseband processing and analog/RF hardware. The transmitted waveform is commonly represented by complex I/Q samples in the digital domain, where it can be filtered, resampled, shaped, and compensated before digital-to-analog conversion and RF up-conversion. This makes the digital part of the transmitter well suited for functions that require programmability and adaptation to different signal conditions. The Digital Front-End (DFE) therefore plays a central role in preparing the signal for transmission, typically including operations such as interpolation, Crest Factor Reduction (CFR) and DPD. Among these, DPD is particularly important because it pre-compensates the signal for nonlinear distortion introduced by the PA, enabling improved transmitter linearity without sacrificing PA efficiency [47].

To get a general idea of how the transmission architecture is structured the simplified block diagram in Figure 3.1 illustrates a general and typical example of the transmission architecture of a modern radio.

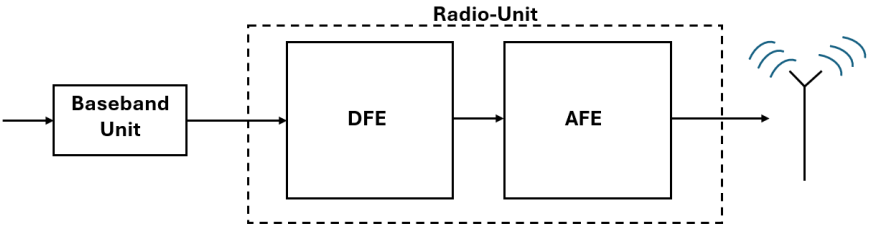


Figure 3.1 Simplified block diagram of transmission branch of a radio unit including interface with the baseband processing unit, the digital front-end (DFE), the analog front-end (Analog Front-End (AFE)).

The transmitter chain starts from input data, such as encoded user or control information, which is mapped in the Baseband (BB) unit into a digital complex I/Q waveform, as described in Section 2.3. It then enters the DFE, where the signal is processed by a digital filter. After filtering, CFR is applied in order to reduce the signal peaks and thereby improve the operating conditions for the subsequent transmitter stages. The signal is then processed by the DPD, to pre-compensate for the nonlinear effects introduced by the PA, before the signal is converted from the digital to the analog domain by the Digital-to-Analog Converter (DAC). This is visualized in Figure 3.2 and explained more rigorously in Section 3.1.

After the signal has been converted through the DAC it enters the AFE, where the signal gets up-converted to a desired carrier frequency, to then go through a PA before the analog filter. After the analog bandpass filter the signal goes through the PA to be amplified before being transmitted through the antenna. Worth mentioning is that this paper will not explain or go into depth of the AFE, as it is outside the scope of this thesis. For the interested reader one can look at Mohamed Ferazi's paper [20].

However, after the PA the signal will be fed back to the DFE-block through an attenuator and Analog-to-Digital Converter (ADC). This information is then used in the DPD-block where it updates the coefficients of the DPD model (2.6) and adapt the linearization to reduce the nonlinear distortion effects caused by the PA. It terms of control theory it is a closed-loop system with feedback to continuously adapt the linearization.

The linearization properties of the DPD is crucial as the nonlinear effects of the PA distorts the signal and broadens the frequency spectrum. A clearer understanding of the transmitter architecture is therefore important before studying the DFE, DPD, and Digital Radio Tuning (DRT) in greater detail in the following sections.

3.1 Digital Front-End (DFE)

The Digital Front-End (DFE) is a central block of the transmission architecture, since it contains all of the digital signal processing components and functions applied to a signal before transmission. In the transmission architecture chain the DFE is placed before the analog part of the transmission chain. A possible implementation of the DFE includes digital filters, CFR, interpolation stages, DPD and more to treat the signal in the digital domain before converting it in the DAC. These operations help prepare the signal for the AFE and improve the conditions for subsequent transmitter stages [36], Figure 3.2 illustrates this.

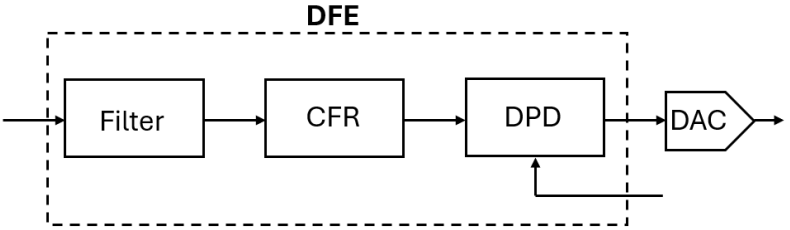


Figure 3.2 An exemplary block diagram of Digital Front-End (DFE) in the transmission architecture.

The benefit of placing the digital compensation circuits inside the radio transceiver is to mitigate the RF analog impairments, impairments, which are generally not limited to nonlinearity but can also include Direct Current (DC)-offset, I/Q imbalance, and variations due to component aging. The pre-processing in the DFE might also include increasing the sample rate of the digital signal compared to the rate of the baseband signal. The increased sampling rate enables numerous benefits in RF. To name a few, 1) compensation of impairments in frequency regions further away from the transmission signals (e.g., out-of-band frequencies for unwanted emission mitigation). 2) more accurate digital nonlinear processing due to increased Nyquist frequency allowing larger spectral expansion (see convolution theorem in Section 2.3, Equation (2.9)). This is especially important for the DPD block, which is a key compensation for the nonlinear effects in the analog domain [36].

Thus, the DFE is a comprehensive functional block rather than a single algorithmic block. It consists of several digital signal processing stages whose purpose is to improve signal quality, ease the interaction with the AFE, and provide the necessary conditions for efficient and robust radio transmission.

3.2 Digital Pre-Distortion (DPD)

As discussed earlier in Section 2.4, practical RF PAs exhibit nonlinear behavior, especially when operated close to saturation in order to improve power efficiency. This creates a fundamental trade-off between linearity and efficiency, highly efficient PA operation tends to introduce unacceptable nonlinear distortion, while highly linear operation typically requires significant output back-off and therefore reduced efficiency [24, 23].

DPD is one of the most widely used linearization techniques for addressing this trade-off. The basic idea is to apply a complementary nonlinear transformation to the signal *before* it enters the PA, such that the cascade of the predistorter and the PA behaves approximately as a linear amplification system [23, 24]. In this sense,

the predistorter intentionally distorts the baseband signal in a controlled way, so that the nonlinear effects introduced by the PA are pre-compensated.

From a behavioral modeling perspective, DPD can be interpreted as an inverse modeling problem. If the PA is described by a nonlinear function $f_{\mathrm{PA}}(\cdot)$ relating its input and output, then the goal of the predistorter is to realize an approximate inverse (or pre-inverse) function $f_{\mathrm{DPD}}(\cdot)$ such that

$$f_{\mathrm{PA}}\big(f_{\mathrm{DPD}}(u(t))\big) \approx Gu(t), \tag{3.1}$$

where $u(t)$ denotes the the given input signal (e.g., baseband or post-CFR signal) and G is the gain of the overall linearized transmitter [23]. The following figure represents this process.

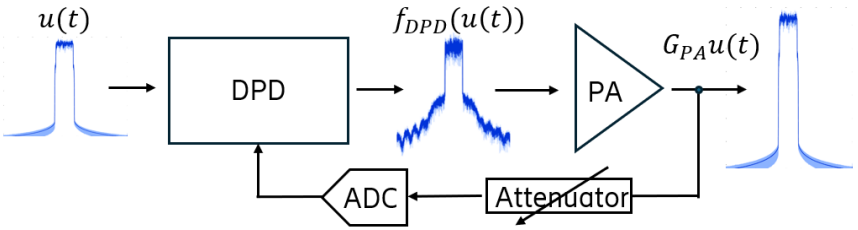


Figure 3.3 General principle of linearization.

In practical scenarios the DPD might needed to be adapted at runtime (e.g., the coefficients of the polynomial model can be re-estimated), which can be implemented using an observation path, where a digitized version of the PA output is compared to the reference input signal in order to identify and update the predistorter model [24, 23]. This can be relevant in many implementations as AFE nonlinearity can vary at runtime depending on output power, temperature, and components aging. This is important to understand, since the memory effects of DPD are present on the ∼nano-second timescale, and estimation of DPD coefficients on the ∼milli-second scale [24].

Another important practical aspect is that efficient DPD depends on accurate observation of the PA behavior. The measured input and output waveforms must be properly acquired, synchronized, and time aligned. Otherwise, mismatches in delay can be misinterpreted as memory effects and degrade the quality of the extracted model [23]. Hence, DPD is not only a modeling problem, but also a signal-processing and system-identification problem.

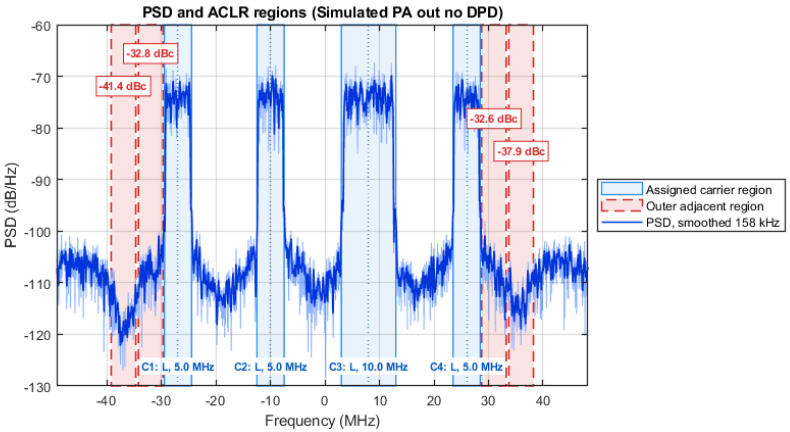


Figure 3.4 PA output with non-predistorted input.

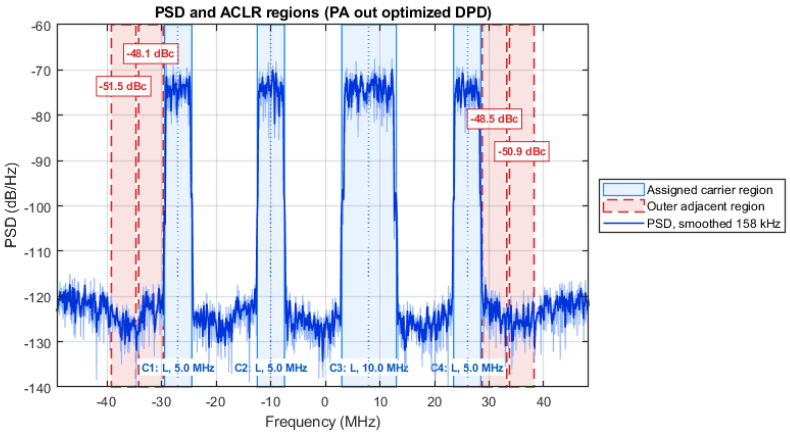


Figure 3.5 PA output with GMP-based predistorted input.

Presented in Figure 3.4, the pure distorted output signal from the PA has larger distortions from the spectral regrowth. For example the ACLR level is as high as -32 dB, which for comparison is much higher than the allowed limit imposed by 3GPP specification discussed in Section 2.5 (i.e., -45 dB).Once a GMP-DPD model is pruned (see upcoming section on DRT) adapted and applied to the input signal, the ACLR at PA output falls in the -48 dB region which is safely below the 3GPP ACLR limit. From Figure 3.5 the ACLR has decreased to below -48 dB in both adjacent regions, thus complying with the industry standards of ACLR from 3GPP.

3.3 Digital Radio Tuning (DRT)

While DPD is the algorithm that actually linearizes the PA, the DPD itself must still be configured. In practice, this configuration is not universal. Different radio signals stress the PA differently depending on properties such as carrier configuration, bandwidth, occupied spectrum and energy distribution. As a result the same DPD structure is not equally suitable for all radio scenarios.

In the context of this work, *Digital Radio Tuning* (DRT) refers to the offline process of devising and testing a suitable set of DPD configurations to satisfy a well-defined set of target use cases (e.g., target signal types, supported carrier configurations) for a specific radio unit. The purpose of DRT is to provide the DPD algorithm with better initial conditions, such that a valid and efficient linearization states can be reached faster at runtime and with lower computational and hardware cost. The following figure gives a visual representation of the core idea and building blocks of DRT.

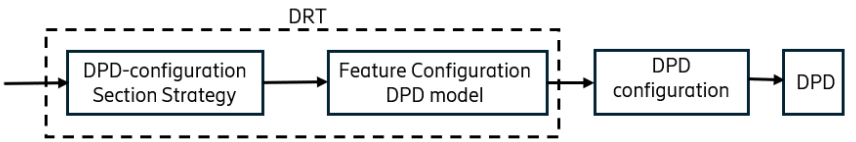


Figure 3.6 Configuration of DPD model fetures (per section)

If the target linearization case is relatively simple (e.g. due to narrower carrier configuration or lower target output power), an unnecessarily large and complex predistorter structure may lead to wasted computational effort, higher power consumption and unnecessary thermal stress. Conversely, if the linearization case is more demanding, an overly simple initialization may fail to meet the required spectral constraints. An example of relevant contraints and performance metrics were presented in greater detail in Section 2.5.

From this perspective, DRT can be viewed as a signal-aware configuration framework for DPD. In this work, this is approached through data-driven characterization of the input signals, where similar carrier configurations and signal conditions are grouped together and linked to suitable initial GMP structures. The goal is to reduce the reliance on manual rule-based selection and greedy search procedures, and instead provide a more systematic, automated and efficient way of configuring the DPD.

This is particularly important because DRT is performed before the radio unit is deployed in the field. A more streamlined and automated tuning procedure can therefore reduce configuration time, improve repeatability and enable more efficient use of hardware resources. In a broader system perspective, this may translate into lower energy consumption, reduced cooling and material requirements and thus,

improved scalability when multiple radio signals must be supported.

In summary, DPD is the linearization "engine" itself, whereas DRT is the process that helps that engine start from a better operating point and bring out more and better performance. The central idea of this thesis is therefore not only to compensate PA nonlinearities, but also to make the DPD configuration process faster through signal-driven initialization of the predistorter structure, which implies better linearization for the DPD.

There are a multitude of different linearization algorithms for DPD, and for this thesis the Doubly Orthogonal Matching Pursuit (DOMP) algorithm is used. In the field of RF, DOMP is a commonly used model-based realization of the DPD (GMP, Look-Up Table (LUT), etc) the "configuration of DPD model" block in Figure 3.6 can in practice be seen as a feature-selection block. Such problem has root in system identification theory and several specific paper exist, which also cover nonlinear modeling for RF applications [4]. One of the most popular and well-performing feature-selection algorithms in this context is the Doubly Orthogonal matching pursuit (DOMP) [4]. For the interested reader the Institute of Electrical and Electronics Engineers (IEEE) paper by Juan A. Becerra and María J. Madero-Ayora [6] gives the mathematics behind the method and how it works in great detail.

But for the purpose of this thesis, the mathematics will be left out, and we will instead just rely on an implementation of the DOMP algorithm to implement the model-feature selection block in our experiments which are presented later in Section 5.

4

Related Work

This chapter is structured as follows. The first section focuses on the current state of research in Digital Predistortion (DPD), outlining established methods and recent developments within the field. The second section shifts attention towards approaches most closely related to the proposed method, with particular emphasis on signal clustering and techniques relevant to the implementation presented in this thesis.

4.1 Nonlinear modeling for DPD

In the field of DPD research, most efforts are focused on understanding and effectively modeling the PA under fixed or predefined conditions. In recent years Neural network (NN)-based DPD approaches have shown strong performance improvements over classical methods such as Volterra and LUT models. In [25], RNN-based architectures achieved significant Signal-to-Noise Ratio (SNR) gains while highlighting the inherent trade-off between performance and computational complexity. These findings motivate the need for complementary techniques that can improve efficiency without significantly increasing system complexity. Other methods aim at improving the tuning procedures, known as feature selection [5]. Given a fixed DPD model (e.g GMP) the objective is about finding the minimum number of terms necessary to linearize and to this in the fastest manner possible. One such well established method is the DOMP, a greedy-pursuit-based approach which enables faster pruning than its peers while generally ensuring good convergence properties and excellent modeling performance for a given number of GMP model terms compared to other feature selection techniques (e.g, greedy-pursuit such as Random Forests [6], regularizaion algorithms such as Ridge or LASSO [6], or heuristic methods, such as Hill Climbing [6]). For an highlight over those methods we refer to Table 3 in [6].

The reviewed literature indicates that DPD optimization and feature-selection methods are already well-established research areas, with several approaches aiming to reduce model complexity and improve tuning efficiency. Therefore, the ob-

jective of this thesis was not to develop a competing DPD optimizaion algorithm, but rather to investigate a complementary approach. In particular, the thesis explores whether similarities between carrier configurations can be used to support more efficient DPD configuration. This direction was identified as a research gap in the related literature, since signal clustering for signal-aware nonlinear models selection have not been extensively studied in the context of DPD configuration. Consequently, the optimization of use-case-dependent DPD configuration became the natural focus of this work.

4.2 Clustering signals

Carrier sectioning using clustering is to our knowledge an unexplored topic, however clustering of signals has been studied before.

One such example is the thesis paper by [18]. In their paper, Eriksson and Abdelnaeim used two k-medoid based methods known as Clustering for Large Applications (CLARA) and Clustering for Large Applications based on Randomized Search (CLARANS) to try to cluster signals. These types of algorithms specialise in spherical clustering and also perform well with outliers in the data. In their work the main measurement of success was how well defined the clusters were and the amount of overlap between the different clusters. Their result showed good capabilities in clustering similar events but also deemed that the chosen model was far to complex. The authors proposed that further work includes investigating simpler models, as this opens up possibilities to work with larger dataset to get more robust results. Despite their result the use cases of the clustering was never tested in any application, an interesting research gap that our work hopes to improve upon. Thus our work will have the same fundamental idea with the goal of recreating similar successful results, however within the context of DPD. Also validating using the previously mentioned performance metrics as suppose to human analysis and experience, is an unexplored idea which this thesis aim to cover.

Another example comes from the field of *radar deinterleaving*, which involves separating mixed signals to retrieve individual signals from radar emitters [17]. Building on this problem, [22] proposes a new Neural Network (NN) based clustering method where the incoming radar signals are first encoded to images before clustering. Their result showed that their proposed method outperformed baseline clustering methods in the two metrics they used to evaluate. They also concluded that the problem is a trade–off between correct grouping and minimizing the number of clusters. Although this work is related through its use of clustering on radio-frequency signal data, it differs from the present thesis both in application and objective. The method is developed for radar deinterleaving and relies on supervised learning together with an image-based signal representation, which is not directly applicable to carrier sectioning for DPD configuration. Furthermore, the reviewed clustering approaches focus mainly on grouping or classifying signals, without us-

ing the resulting clusters for subsequent model selection. This distinction is central to the present thesis, where the clusters are intended to support the assignment of suitable DPD models to different carrier configurations.

5

Signal-Aware DPD Configuration Methods

As described in Section 1.2, the purpose of this thesis is to investigate whether machine-learning methods can support the configuration process of DPD for different carrier configurations. Regardless of the specific DPD model implemented in the DFE, a key step is to configure, or tune, the predistortion block such that the nonlinear effects of the PA are sufficiently reduced while the required transmitter performance metrics are fulfilled (such as 3GPP performance protocol presented in Section 2.5).

In this chapter, the term *carrier* is used as shorthand for a carrier configuration or use case that must be supported by the radio product. In principle, each carrier could be assigned its own individually tuned DPD configuration. However, this would be inefficient in practice, since the number of supported carriers can be very large and each configuration require additional tuning, validation, and HW integration effort. Thus, a individually tuned DPD configuration is not feasible in practice, as it takes too much time. Instead, several carriers can be grouped and assigned to the same DPD configuration. This grouping is referred to as *DPD sectioning* or *carrier sectioning*.

A *section* is defined as a group of carriers that share the same DPD configuration for a given PA. More specifically, all carriers within a section use the same selected GMP-DPD configuration, including the active delay-term structure and corresponding coefficients, and must satisfy the required performance constraints when evaluated with that configuration. In this work, ACLR is used as the main validation criterion, the same sectioning principle could also be formulated using other transmitter performance metrics, such as EVM or other 3GPP-related requirements.

The purpose of DPD sectioning is therefore to avoid tuning a unique DPD configuration for every carrier. Instead, one representative DPD configuration is tuned for a selected carrier and then evaluated on other carriers. The carriers that pass the required performance constraints are assigned to the same section. This creates an important trade-off. Using fewer sections can reduce the number of configurations

that must be tuned and validated, thereby reducing HW integration time. However, each section may then need a more complex DPD configuration in order to cover a wider range of carriers. Conversely, using more sections may reduce the average DPD complexity per section, e.g., the number of active GMP terms, but can increase the total integration and validation effort.

The central problem studied in this chapter is therefore *how* carriers should be grouped into sections. The proposed idea for this thesis is that carriers with similar signal characteristics may have similar linearization requirements, and may therefore be suitable to share the same DPD configuration. This motivates the signal-aware approach investigated in this thesis, where carrier features are used to guide the sectioning process.

This chapter first presents two baseline methods for DPD sectioning in Section 5.1. The sectioning problem is then formulated as a conceptual optimization problem in Section 5.2, before the proposed clustering-based method is introduced in Section 5.3. All candidate methods are evaluated using the same reference DPD structure, based on a GMP model, and the same feature-selection engine, DOMP, as described in Chapter 4.

5.1 Baseline Methods

A baseline DPD sectioning method was selected in this thesis as a reference benchmark for the proposed clustering-based method. The baseline method, illustrated in Figure 5.1, is based on a simple and robust iterative procedure where new sections are created until all carriers have been assigned to a section that satisfies the required performance constraints.

The method starts with all carriers unassigned. In each iteration, one carrier is selected as the representative carrier for a new section. A DPD configuration is then tuned for this representative carrier using the selected DPD structure and term-selection method, in this thesis a GMP model together with DOMP. Once the DPD configuration has been obtained, it is evaluated on the remaining unassigned carriers using the chosen validation metric, here ACLR as described in Section 2.5. All carriers that satisfy the defined ACLR threshold with this configuration are assigned to the current section. The carriers that fail the validation remain unassigned and are passed on to the next iteration, where a new representative carrier is selected and a new section is created. This procedure is repeated until all carriers have been assigned to a valid section.

The main design choice in this baseline procedure is how to select the representative carrier for each new section. In this thesis, two baseline variants are evaluated, denoted Baseline V1 and Baseline V2. Baseline V1 is stochastic and selects a random carrier from the remaining unassigned carriers. This gives a simple reference method, but the resulting sections may vary between runs due to the random selection.

Baseline V2 is deterministic and instead selects an arbitrarily defined *hardest carrier* from the remaining unassigned carriers. In general, the hardest carrier to linearize can depend on several factors, including the PA, the carrier placement, the occupied bandwidth, the DFE setup, and the relevant performance metric. For example, carriers placed close to the edge of the supported band may be particularly difficult in practice, since nonlinear spectral regrowth can more easily leak into adjacent channels. However, in the evaluated dataset the supported band is relatively limited, and the main optimization objective is related to the number of active DPD terms. Therefore, the hardest carrier in Baseline V2 is approximated by the carrier with the largest total occupied BW. This should be interpreted as a practical heuristic rather than a general definition of DPD difficulty.

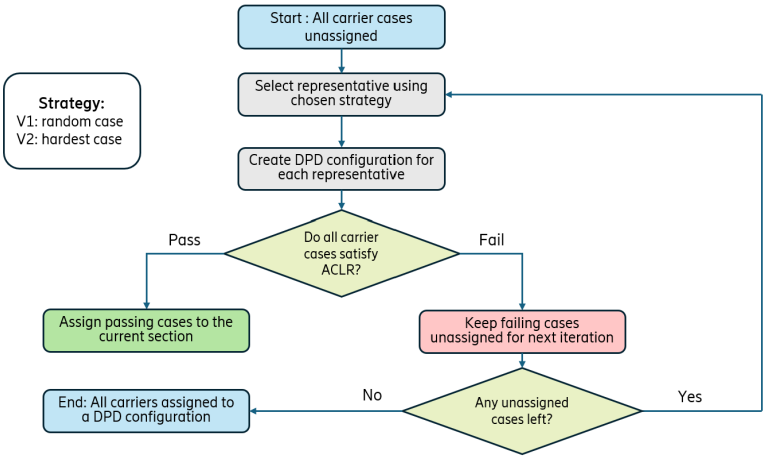


Figure 5.1 Flowchart of the baseline sectioning method.

The baseline methods assume that, for the considered PA and DPD model, there exists at least one DPD configuration that can linearize each carrier sufficiently to satisfy the required performance level, here represented by the ACLR limit. If the sectioning strategy is inefficient, simple carriers may be assigned to overly complex DPD configurations due to poor grouping. This can increase the average number of active GMP terms per section, which is used in this thesis as a proxy for digital DPD complexity and runtime resource usage. Another potential issue is that the number of validation tests depends on which carrier is selected as representative for each section. If a representative carrier generalizes poorly to the remaining carriers, many validation tests may fail before a valid section is found, resulting in a longer and less efficient integration workflow.

To motivate the heuristic used in Baseline V2, each carrier was first linearized

individually using a dedicated GMP-DPD configuration. This makes it possible to estimate the number of active GMP terms required for each carrier to meet the target −48 dB ACLR requirement. Figure 5.2 shows the required number of active GMP-DPD terms for all carriers in the dataset when each carrier is assigned its own dedicated DPD configuration.

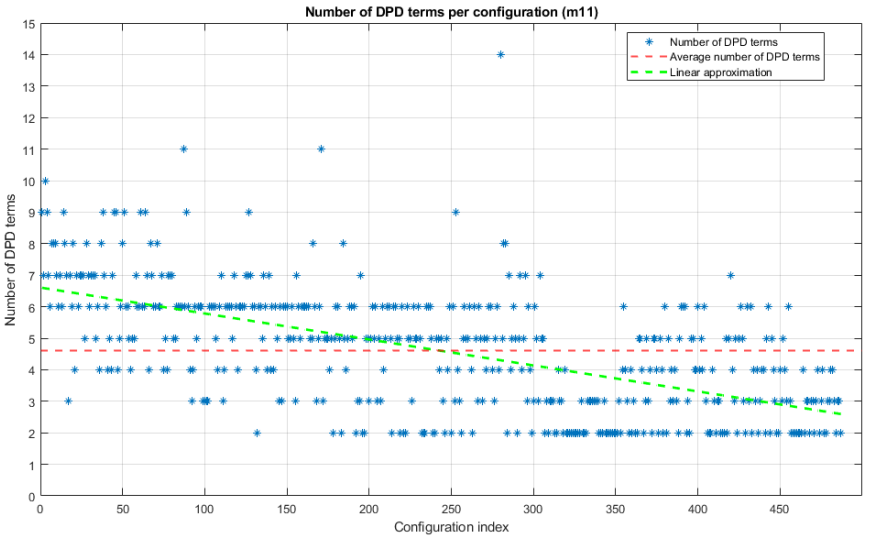


Figure 5.2 Number of active GMP-DPD terms required when each carrier is linearized using a dedicated DPD configuration. The carriers are sorted according to decreasing total occupied BW.

The blue points (∗) show the individual carriers, the red dashed line shows the average number of active GMP terms, and the green dashed line shows a linear approximation of the trend. The carriers are sorted with respect to the total occupied BW, meaning that the first carriers (positioned at index $1, 2, 3, \ldots$) in the plot have the largest sum of carrier BW.

The trend shown in Figure 5.2 supports the use of total occupied BW as a practical heuristic for estimating DPD complexity in this set of carriers. However, the presence of outliers also shows that this simple ordering criterion does not capture all difficult carriers. Carrier difficulty is not only determined by bandwidth, but may also depend on carrier placement, spacing, nonlinear distortion products, and the specific PA behavior. Therefore, Baseline V2 should be interpreted as a deterministic and physically motivated baseline heuristic, not as a generally optimal sectioning strategy.

5.2 Optimization Problem

To address some of the limitations of the baseline methods, a more general formulation of the DPD sectioning problem is introduced. As discussed in Section 2.5, DPD performance is commonly evaluated using linearization-related metrics such as ACLR and EVM. In the sectioning problem, these metrics can be interpreted as constraints that each carrier must satisfy when evaluated with its assigned DPD configuration. The objective of the sectioning process can then be chosen depending on the system-level priority, for example reducing DPD complexity, reducing the number of sections, or reducing the total HW integration time.

The optimization formulations presented in this section should be interpreted as conceptual descriptions of the trade-offs involved in DPD sectioning. The methods evaluated in this thesis do not solve these optimization problems globally. Instead, the baseline and clustering-based methods are heuristic sectioning strategies designed to approximate different objectives under the same validation constraints.

A general formulation of the DPD sectioning problem can be written as

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & \mathrm{ACLR}_{k(m),m}(x) \leq \mathrm{ACLR}_{\mathrm{req}}, \quad \forall m = 1,\ldots,M, \\ & k(m) \in \{1,\ldots,K\}, \quad \forall m, \end{aligned} \tag{5.1}$$

where x denotes a complete sectioning strategy, including the number of sections, the assignment of carriers to sections, and the DPD configuration associated with each section. The function f denotes the system-level cost to be minimized. The term $\mathrm{ACLR}_{k(m),m}(x)$ denotes the ACLR obtained when carrier m is evaluated using the DPD configuration assigned to section $k(m)$, while $\mathrm{ACLR}_{\mathrm{req}}$ denotes the required ACLR limit. Furthermore, M denotes the total number of carriers and K denotes the number of sections generated by the sectioning strategy.

Several objective functions could be considered depending on the desired system behavior. In this thesis two objectives are considered, minimizing DPD complexity and minimizing estimated HW integration time. The main focus is on DPD complexity, which is represented by the number of active GMP terms in the selected DPD configurations. This should be interpreted as a practical proxy for digital DPD resource usage, since a larger number of active terms generally implies more coefficients to store, update, and evaluate in the ASIC, which was presented in Section 1.2. However, it is not a direct measurement of physical implementation cost, since actual area, memory usage, latency, bit width, and power consumption also depend on the specific hardware architecture.

Using this objective, the sectioning problem can be written as

$$\begin{aligned} \min_{x} \quad & \mathrm{DPD}_{\mathrm{resources}}(x) \\ \text{subject to} \quad & \mathrm{ACLR}_{k(m),m}(x) \leq \mathrm{ACLR}_{\mathrm{req}}, \quad \forall m = 1,\ldots,M, \\ & k(m) \in \{1,\ldots,K\}, \quad \forall m. \end{aligned} \tag{5.2}$$

In the evaluation, $\mathrm{DPD}_{\mathrm{resources}}(x)$ is quantified using the mean number of active GMP terms per section, as described in Section 6.4. This metric captures the relative complexity of the selected DPD configurations and is used to compare the different sectioning methods.

The second objective considered in this thesis is the estimated HW integration time. This time is divided into two parts. The first part, denoted $T_{\mathrm{conf}}(x)$, represents the time required to configure the DPD models for the generated sections. The second part, denoted $T_{\mathrm{tests}}(x)$, represents the time required to validate the carriers, for example by measuring ACLR after loading a specific DPD configuration. The corresponding conceptual optimization problem can be written as

$$\begin{aligned} \min_{x} \quad & T_{\mathrm{conf}}(x) + T_{\mathrm{tests}}(x) \\ \text{subject to} \quad & \mathrm{ACLR}_{k(m),m}(x) \leq \mathrm{ACLR}_{\mathrm{req}}, \quad \forall m = 1,\ldots,M, \\ & k(m) \in \{1,\ldots,K\}, \quad \forall m. \end{aligned} \tag{5.3}$$

The integration-time objective should also be interpreted as a model-based estimate rather than a directly measured hardware integration time. It is used to compare how the different sectioning strategies affect the expected configuration and validation effort.

5.3 Proposed Method – Clustering

In line with the optimization formulation in Section 5.2, the proposed method aims to guide the DPD sectioning process using carrier features. The main objective is to reduce the DPD complexity, represented by the number of active GMP terms in each configuration, while still maintaining a reasonable HW integration time. The proposed method is based on an unsupervised clustering approach, where GMM, described in Section 2.8, is used to group carriers in feature space before any DPD validation tests are performed.

In contrast to the baseline methods, where the representative carrier is selected either randomly or from a simple hardest case criterion, the clustering-based method uses information from the carriers themselves. The hypothesis is that carriers with similar characteristics in feature space may also have similar linearization requirements, in DPD configuration space, and may therefore be suitable to share the same DPD configuration. The clustering should therefore be interpreted as an initial data-driven proposal for how the carriers may be grouped. The final sections are still determined through validation, a cluster only becomes a valid section after the selected DPD configuration has been tested and the carriers satisfy the required performance constraints.

Similar to the baseline method, some carriers may fail the ACLR requirement after the first clustering attempt. The proposed method handles this by iteratively

applying clustering to the remaining failed carriers. In each iteration, a new clustering is performed, a representative carrier is selected from each cluster, and the resulting DPD configuration is validated on the carriers assigned to that cluster. The procedure is repeated until all carriers have been assigned to a valid section. Figure 5.3 shows the proposed method on a high level.

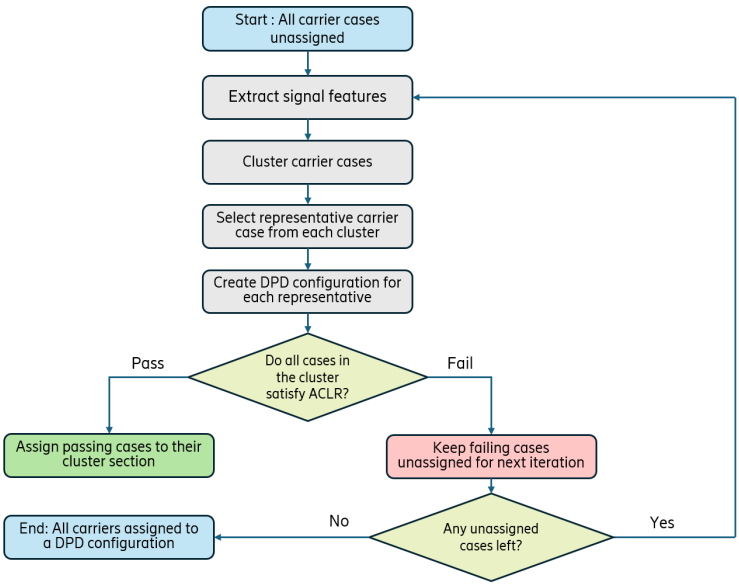


Figure 5.3 Flowchart of the proposed clustering-based sectioning method.

To better explore the potential of the proposed approach, three variants were implemented. These are denoted Clustering V1, Clustering V2, and Clustering V3. All three variants are based on the same clustering idea, but differ in how the representative carrier is selected from each cluster. Clustering V1 selects the hardest carrier in each cluster, here approximated by the largest total occupied BW, similar to Baseline V2. Clustering V2 instead selects the carrier closest to the cluster centroid, measured using Euclidean distance in the reduced feature space. Clustering V3 is a hybrid between Clustering V2 and Baseline V2. It starts with the centroid-based clustering strategy, but once the number of remaining failed carriers falls below a specified fraction of the original dataset, the method switches to Baseline V2. In this implementation, this switching threshold was set to 35%. This value was selected empirically and should be interpreted as a practical design choice rather than an optimized hyper parameter.

Feature Extraction

In order to perform clustering, relevant features must be extracted from each carrier. These features should describe properties that are expected to influence the nonlinear behavior of the PA and the resulting DPD requirements. The following features were selected as the initial feature set. Figure 5.4 gives a visual representation of the features.

- Number of carriers
- Sum of BW (sumBW)
- Minimum carrier BW (minBW)
- Maximum carrier BW (maxBW)
- Minimum distance between carriers (minDist)
- Maximum distance between carriers (maxDist)
- Maximum carrier spacing (CSmax)
- Power distribution of IMD (imd1, … , imd6)

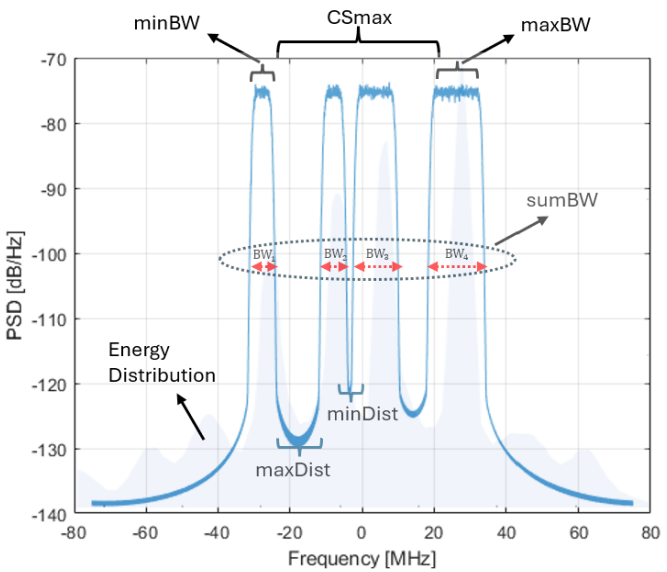


Figure 5.4 Visualization of the extracted carrier features.

The first seven features are direct properties of the carrier configuration and can be extracted from the reference carrier dataset. These include occupied bandwidth, carrier spacing, and the number of carriers. The IMD power distribution feature is more elaborate and is intended to give a simple spectral description of where nonlinear distortion products are expected to appear.

The IMD power distribution is calculated by passing the carrier through a memoryless polynomial approximation with third- and fifth-order nonlinear terms. This

model should not be interpreted as the GMP model used for the actual DPD configuration, nor as a complete model of the measured PA. Instead, it is a simplified Taylor-series-like approximation of a memoryless nonlinear system, as discussed in Sections 2.4 and 2.4. The purpose is to obtain a fast and physically motivated estimate of how the nonlinear distortion energy is distributed over frequency for each carrier. After the nonlinear transformation, the PSD of the resulting signal is evaluated and divided into N discrete frequency regions within the considered linearization bandwidth. In this implementation, $N = 6$ was selected after manual tuning.

As a starting point, the full feature set was considered for the clustering algorithm. However, by studying the correlation between the features using a correlation matrix, several features were found to be strongly correlated. Highly correlated features can add unnecessary complexity to the clustering problem and may cause certain carrier properties to be overrepresented. The correlation matrix of the full feature set is shown in Figure 5.5.

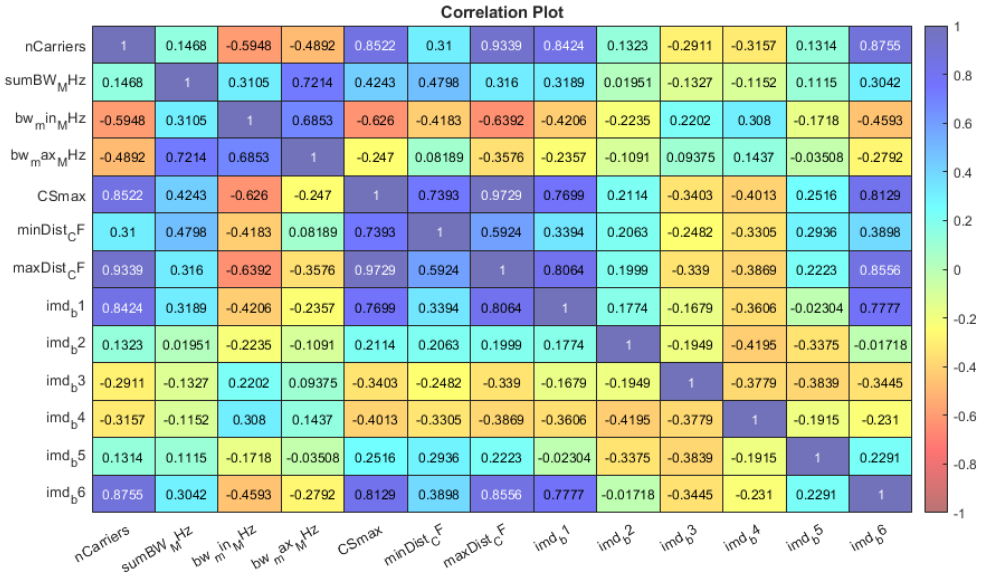


Figure 5.5 Correlation plot of the full feature set.

In Figure 5.5, blue squares indicate positive correlation, green indicates weak or no correlation, and red indicates negative correlation. Based on this analysis, the total number of features was reduced from 13, where the IMD power distribution is counted as six separate features, to the following seven features:

- Sum of BW
- Maximum carrier spacing
- Minimum carrier BW
- Energy distribution 2
- Energy distribution 3
- Energy distribution 4
- Energy distribution 5

The correlation matrix of the reduced feature set is shown in Figure 5.6.

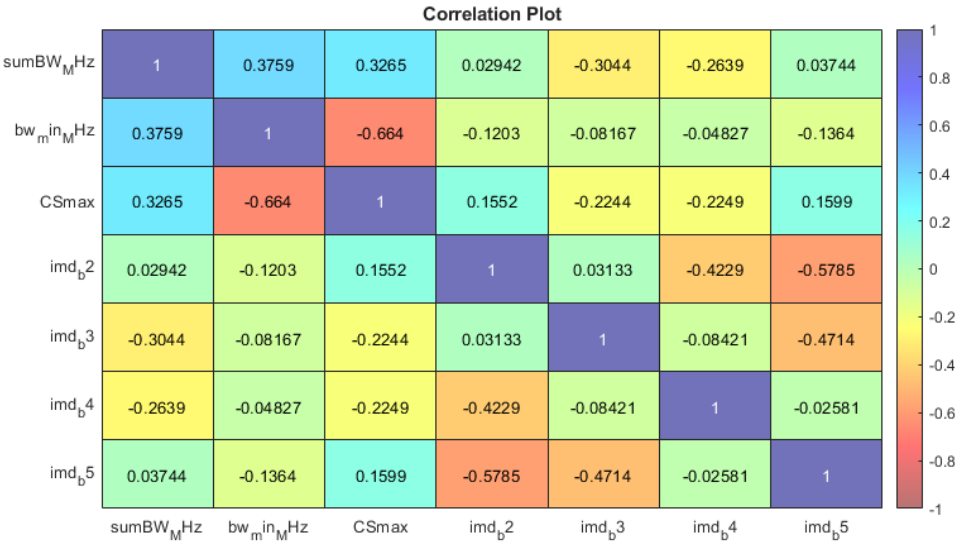


Figure 5.6 Correlation plot of the reduced feature set.

The retained features were selected to capture different aspects of the carrier configuration. The sum of BW describes how much of the operating band is occupied by carriers. A larger occupied bandwidth generally increases the amount of spectral content that must be linearized and was shown in Section 5.1 to be related to the number of active GMP terms required for a dedicated DPD configuration.

The maximum carrier spacing, denoted CSmax, describes the distance between the leftmost and rightmost carrier. A large CSmax indicates that carriers are spread over a wider part of the operating band, which may affect the location and strength of nonlinear distortion products. The minimum carrier BW was retained since it helps distinguish between configurations with several narrow carriers and configurations with fewer wider carriers.

The IMD power distribution features act as a simplified spectral footprint of the nonlinear distortion. They provide information about which frequency regions are

expected to contain more nonlinear distortion energy for a given carrier. This was considered useful since carriers with similar occupied bandwidth may still generate different distortion patterns depending on their placement and spacing.

In the full correlation plot, shown in Figure 5.5, the outer IMD regions imd1 and imd6 showed strong correlation with other features and were therefore removed. Similarly, maxBW, minDist, and maxDist were removed since they were strongly correlated with retained features and also introduced ambiguities for single-carrier cases. In such cases, maximum and minimum bandwidth become identical, while distance-based features become zero since no spacing exists between carriers. The number of carriers was also removed, since much of this information is already indirectly captured by the IMD energy distribution and spacing-related features.

Since the proposed clustering method is iterative, the feature set may be slightly adjusted between iterations. The motivation is that the remaining failed carriers after each iteration may represent a more specific and difficult subset of the original dataset. Therefore, features that are useful in the first clustering iteration may not always be equally informative in later iterations. This should be interpreted as an empirical design choice, and a broader discussion of this aspect is provided in the discussion chapter.

Dimension Reduction and Clustering

Before applying GMM, the feature data was standardized and reduced to a lower-dimensional representation. This was done both to reduce the number of input dimensions used for clustering and to improve interpretability of the feature space. A lower-dimensional representation also makes it easier to visualize the data using one- or two-dimensional plots, instead of studying each feature separately.

For this purpose, Principal Component Analysis (PCA), described in Section 2.8, was used. The number of retained principal components was selected based on the cumulative explained variance. The smallest number of components explaining more than 95% of the variance was retained. This gives a compact representation of the feature space while preserving most of the variation in the original data. A visual example is shown in Figure 2.18.

After dimensionality reduction, GMM was fitted to the reduced feature dataset. The number of mixture components, corresponding to the number of clusters, was selected by comparing models with 1 to 10 clusters using the BIC, described in Section 2.8. An example BIC curve is shown in Figure 5.7.

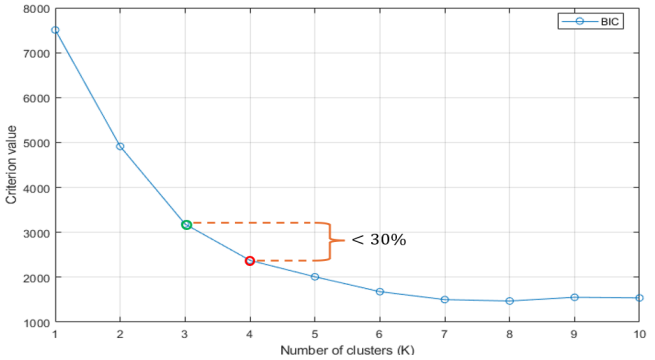


Figure 5.7 BIC curve for models with 1–10 clusters, including the 30% improvement criterion.

The number of clusters was selected by analyzing the relative decrease in the BIC score as the number of mixture components increased. The selected model was chosen as the first model for which the relative BIC improvement was below 30%. This conservative criterion was introduced to limit the number of clusters and reduce the risk of generating unnecessarily many candidate sections. However, the 30% criterion is an empirical design choice and not a generally optimized threshold.

The choice of GMM was motivated by its ability to model clusters with different shapes and covariance structures, while still providing a probabilistic and repeatable clustering framework. This is useful in the present setting since carrier configurations may not form spherical or equally sized groups in feature space. Nevertheless, the use of GMM should be interpreted as one possible clustering approach rather than the only valid choice. A broader comparison with other clustering algorithms is left as future work.

Figure 5.8 provides an overview of the complete workflow, from carrier generation and feature extraction to dimensionality reduction, model selection, and final clustering.

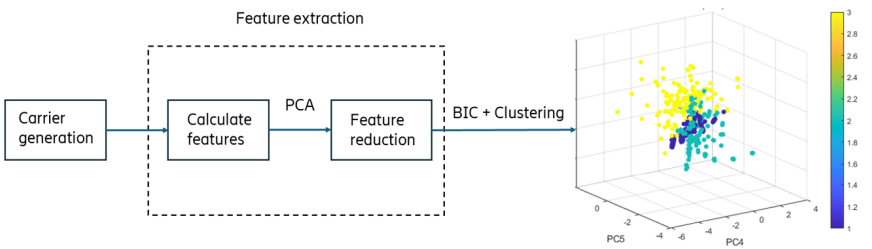


Figure 5.8 Illustration of all steps from carrier generation to clustering.

6

Methodology

The methodology is simulation based and uses measured output data from a real PA. The process can be divided into two main stages, the first is the measurement stage where a baseband signal dataset is first generated and then RF-PA response measured using a HW testbench the second stage is the simulations, where the proposed method is tested. The following figures give an overview of what steps are performed in each stage, while we describe each step in details in the next sections.

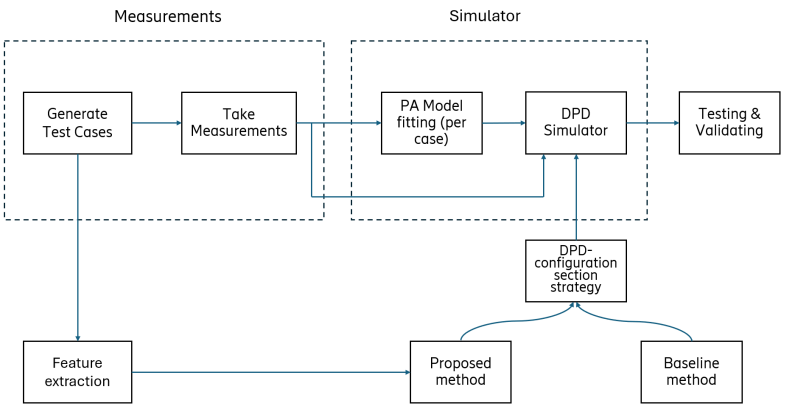
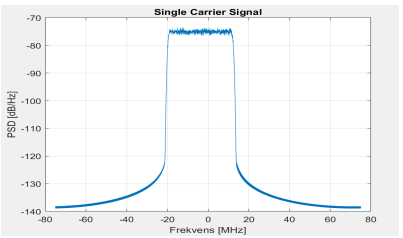


Figure 6.1 Block diagram the methodology used in this thesis to test the proposed DPD sectioning method

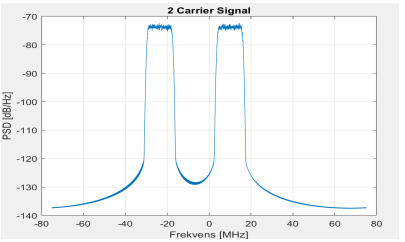
6.1 Data generation

The necessary data was extracted in two seperate steps. The first consisting of creating all the permutations of carrier signals that were needed and the second was generating the signals in a lab and measuring the true signal. The generation process started by selecting the possible carrier types that could be sent on the chosen

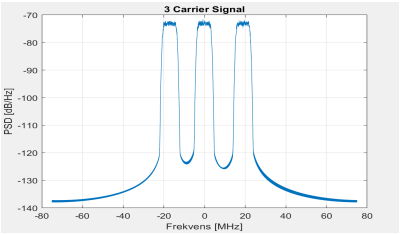
frequency band. The target band was Band 1 (or n1 in NR terminology), which is a FR1 FDD band with downlink spectrum allocated between 2110 MHz - 2170 MHz. The most common carrier-types to use in this band are WCDMA, LTE and 5G NR. WCDMA is an older 3G network that is being phased out in order to make space for the faster and more powerful 4G and 5G connectivity [1]. As a result of this it was decided to limit the carrier-types specified to LTE in 4G-advanced, and NR in 5G-advanced. For LTE signals the following channel bandwidths were allowed $[5, 10, 15, 20]$ MHz while for FR1 we allowed all the channel bandwidths fitting into the BW downlink spectrum, that is: $[5, 10, 15, 20, 25, 30, 40, 45, 50]$ MHz. We limited the band to combinations of these two carrier-types, as well as the total number of carriers in each signal was between one to four carriers. This choice was motivated by the fact that any combinations with more carriers where not consistently able to be placed within the specified operating band. These carriers were then placed randomly within the given interval, with stepsize of 200 kHz (0.2 MHz). In accordence with 3GPP standard the spacing is dependent on the channel raster, which in the case of FR1 is 200 kHz [3]. 500 singe/multi-carrier configurations with variable channel bandwidth, carrier frequency and carrier type (also allowing mixed LTE/NR configurations) were generated. Below are examples of what different carrier combinations could look like. It should be noted that the baseband signals in the figures are centered around 0 Hz baseband frequency, which will in this case correspond to a central frequency of 2140 MHZ (B1 Down-Link (DL) center).



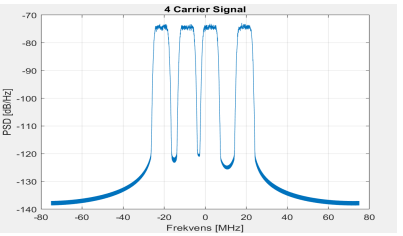
(a) Single Carrier Signal



(b) Dual Carrier Signal



(c) Triple Carrier Signal



(d) Quad Carrier Signal

Figure 6.2 Example of four (out of 500) single/multi-carrier signals used in HW measurements and simulations.

6.2 Measurements

The second step consisted of generating the actual Baseband (BB) waveforms corresponding to each test case (see Figure 6.2 for an example) and measuring the response of the target hardware RF-PA. The procedure was carried out in several stages. First, multicarrier BB signals were generated in Matlab according to the carrier BWs and carrier frequency allocations specified in the dataset description file. The signals were generated as band-limited Additive White Gaussian Noise (AWGN), with the purpose of reproducing both the spectral occupancy and the Peak-to-Average Power Ratio (PAPR) characteristics of practical OFDM signals. This approach allowed the different test cases to be easily generated while maintaining waveform properties representative of realistic downlink signals.

A Crest Factor Reduction (CFR) algorithm was then applied to the generated signals to obtain a PAPR of 7.1 dB. This value was selected as a realistic pre-DPD operating condition for signals in DFE processing chains. After CFR, the signals were passed through a predistortion stage. In this context, the purpose of the predistortion stage was not to define the final PA model used in the simulator, but rather to generate waveforms with an expanded dynamic range and the associated spectral widening typically introduced by predistorted signals. This ensured that the hardware PA was excited with signals having dynamic-range and spectral-content characteristics similar to those of the predistorted waveforms later used for PA-model evaluation in simulation. Each resulting input waveform was then applied to the PA using the RF measurement setup illustrated in Figure 6.3.

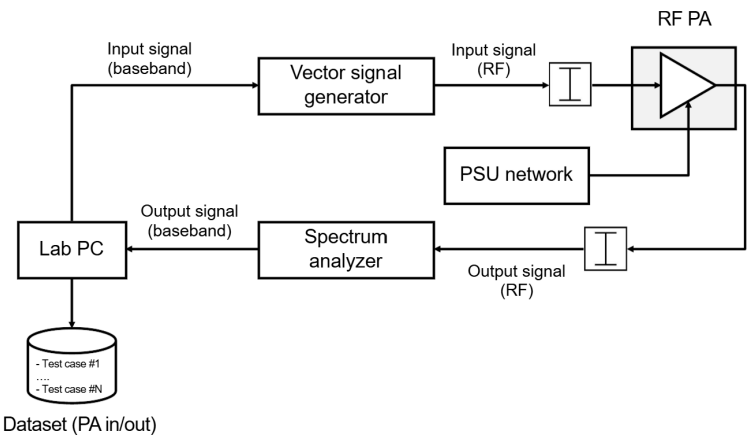


Figure 6.3 Block diagram of the measurement setup

The baseband samples were transferred from the computer to a vector signal generator, which performed Digital-to-Analog Converter (DAC)-conversion and RF upconversion. The RF signal was routed through coaxial cables to the target RF-PA. After amplification, the signal passed through the output RF chain, including attenuation and directional-coupler stages, before being RF-downconverted and digitized by the spectrum analyzer. The captured signals were subsequently post-processed and stored in the dataset. The resulting dataset contains the CFR-processed baseband input waveforms, which were used as source signals for the measurements, together with their corresponding predistorted versions and measured PA output signals. These signal pairs were then used for PA modeling and evaluation in the simulator. For the dataset of 500 cases three different measurements were taken with output levels of the RF-PA varying (low/medium/high). This resulted in a total of 1500 carrier cases incurring in increasing distortion level in the output and thus presenting an increasingly difficult linearization task. In the specific the maximum power gain compression was 1.5 dB, 2.1 dB and 3.2 dB respectively.

6.3 Simulator

The first step was taking PA input and output signals for each of the measured cases (500*3) and fit a different GMP model (Section 2.6) to each of the measurements, to be able to accurately simulate PA response to the tested carrier cases offline. Each model was generated ensuring a modeling NMSE of -40 dB or lower.

With reference to Figure 6.1, once the PA models are generated, a simulated DPD equipped with a tunable GMP model is used as a central block to generate pre-distorted signals.

Running the configuration of DPD model features (see Figure 3.5) consists in this implementation in selecting the number and configuration of GMP terms to be used. This corresponds to the selection of the delay and exponential parameters (j, k, p in (2.43)) for each term. Such feature selection process is done via the DOMP approach described earlier in this work. The DPD coefficients of the model are estimated by a Least Squares (LS) estimator, and the goodness of linearization is evaluated via running the pre-distorted signal through the PA model corresponding to the specific test signal and computing EVM and ACLR metrics.

Such simulator and DRT process is then used as basis to test both the Baseline and Proposed DPD sectioning approaches, as illustrated in Figure 6.1.

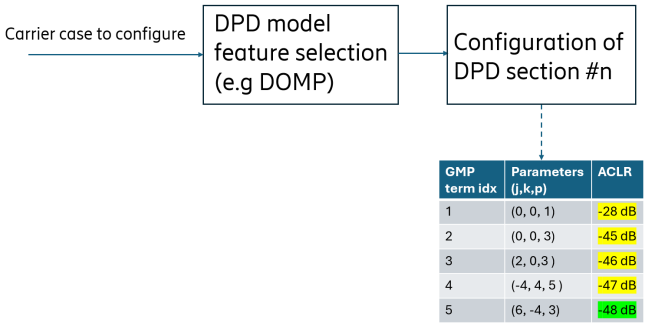


Figure 6.4 Each section in the DPD consists of a mapping of carriers and a GMP model. This model is generated using a DPD model feature selection algorithm that takes in a carrier and configures the parameters of the GMP. The model is built in steps and is always picked to be the first model with an ACLR of less than -48 dB

6.4 Testing and Validating

After the simulator has generated a proposed sectioning of the carrier configurations, each section must be tested and validated. The purpose of this step is to verify whether the selected DPD configuration for a given section is also suitable for the remaining cases assigned to that section.

For each section, the simulator first selects a representative case and generates a corresponding DPD configuration using the Doubly Orthogonal Matching Pursuit (DOMP) algorithm. The remaining cases in the same section are then tested using this specific configuration. This is done by passing each test case through the DPD model generated by the simulator, followed by the individual PA model associated with that specific case. The output signal is then evaluated using ACLR, which is used as the main validation criterion for spectral leakage.

In Radio Communication (RC), typical ACLR requirements for signals in FR1 are below -45 dB (Section 2.5). For this thesis, a stricter threshold of -48 dB is used in order to include an additional performance margin. A case is therefore considered passed if its resulting ACLR is below -48 dB. Cases with ACLR values above this threshold are considered to have too much spectral leakage and are therefore not accepted within the current section.

Once a valid set of sections has been generated, the next step is to evaluate the required DPD resources. To make the different sectioning strategies comparable, four metrics are used. The first metric is the *mean number of terms* per section, which describes how many DPD terms are used on average. This value is weighted by the number of data points (i.e., carrier cases) in each section, giving a measure of the expected DPD resource usage across the complete dataset.

The second metric is the *maximum number of terms*. This corresponds to the largest DPD model size required by the most complex section, and can therefore be interpreted as a worst-case. It describes the most computationally expensive DPD configuration that may be required to be loaded at runtime by the radio unit.

The final two metrics are *the number of DRT tests* required to obtain a valid configuration for every carrier case, and the *total number of sections* to do so. These two metrics are important because they directly affect the overall DPD configuration time for a specific product. As discussed in Section 5.2, the DPD configuration time can be expressed as the sum of configuration time and testing time ($T_{conf} + T_{tests}$). Assuming that the time needed to run DRT for a single section is the sum of baseline processing time T_0 (e.g, dictionary buildup in case of DOMP) and a model pruning/testing time which grows approximately linear with DPD model size S_c, then the total time to populate all DPD section can be a approximated as :

$$T_{\text{INT}} = T_0 N_S + \sum_{n=1}^{N_S} T_{\text{LT}} S_C(n) + T_{\text{Test}} N_{\text{tests}} \tag{6.1}$$

Where N_S denotes the total number of sections, $S_C(n)$ is the number of DPD terms (model size) for a section n, T_0 is the baseline processing time for feature selection and comes from the initialization step in the DOMP algorithm [6], T_{LT} is the time it takes to load and test a DPD configuration, T_{test} is the time required to perform one validation test, and N_{tests} is the total number of performed tests.

This validation procedure therefore ensures that each proposed section satisfies the required spectral performance, while also providing quantitative measures of DPD resource usage, worst-case complexity, and expected hardware integration time.

7

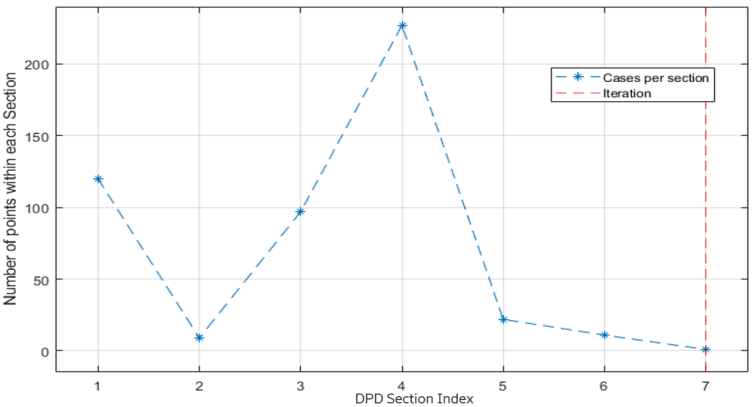
Results

In this section the performance and results from the five different methods will be presented. Both figures and tables will be used to present the result. As described in Section 6.2 three different datasets are used for testing, corresponding to testing three different output levels of the PA called low/medium/high, presenting increasing distortion level and therefore increasing stress on the DPD model. This enables the clustering method to be tested on various levels of PA non-linearites to showcase generalization of performance PA operating points requiring different amounts of linearization resources.

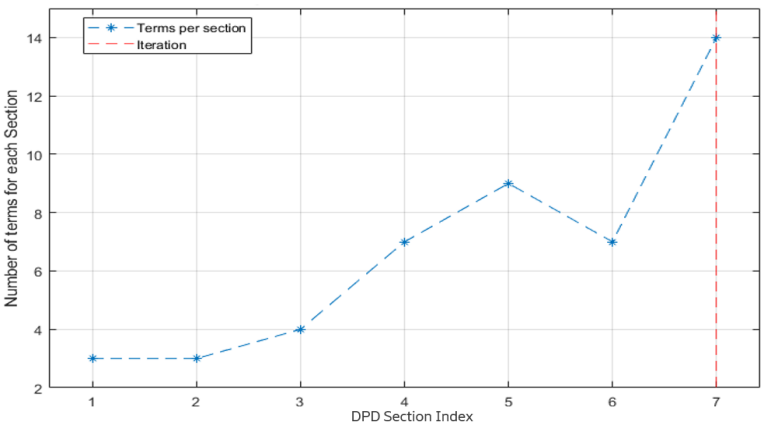
7.1 Baseline V1

Table 7.1 Result from 50 iterations for Baseline V1 using random sectioning allocation.

Dataset	No. validation tests	Mean No. DPD-terms	Max No. DPD-terms	No. of sections
`low`	1001	5.93	14	7.42
`medium`	1018	5.94	18	8.62
`high`	1318	6.92	23	15.56

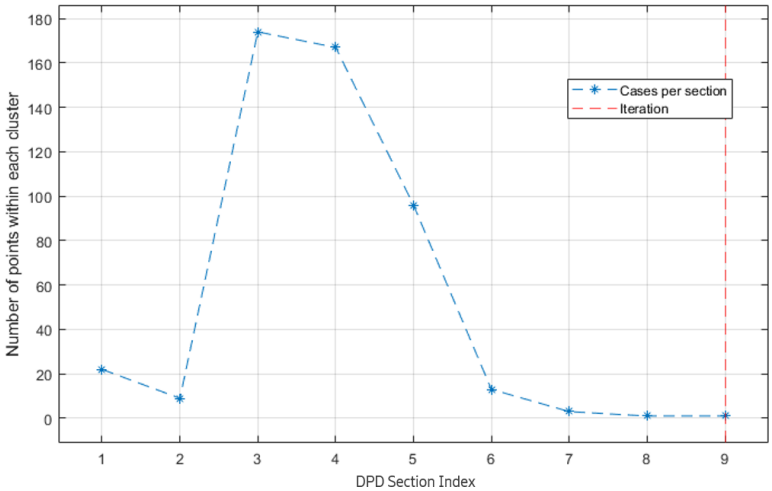


(a) Number of carrier cases assigned to each DPD section.

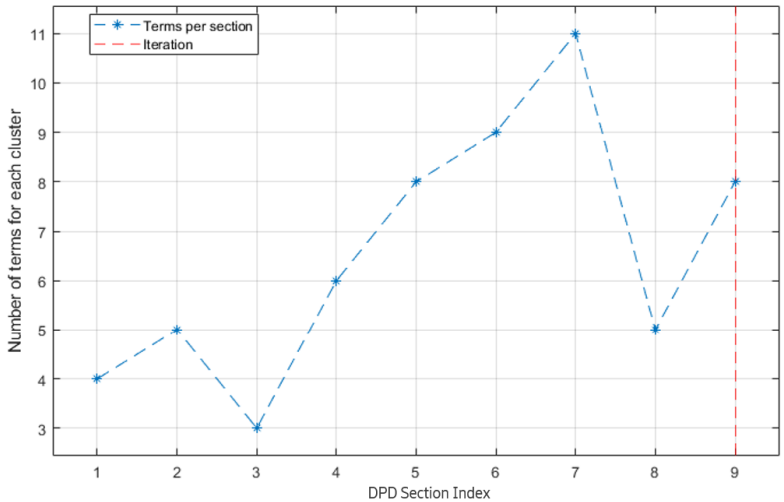


(b) Number of DPD terms for each DPD section.

Figure 7.1 Performance statistics for a single random run of Baseline V1 on `low`.

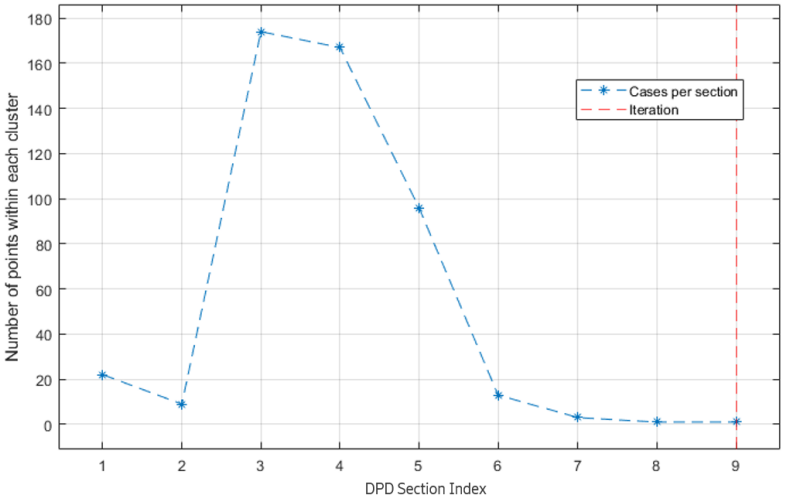


(a) Number of carrier cases assigned to each DPD section.

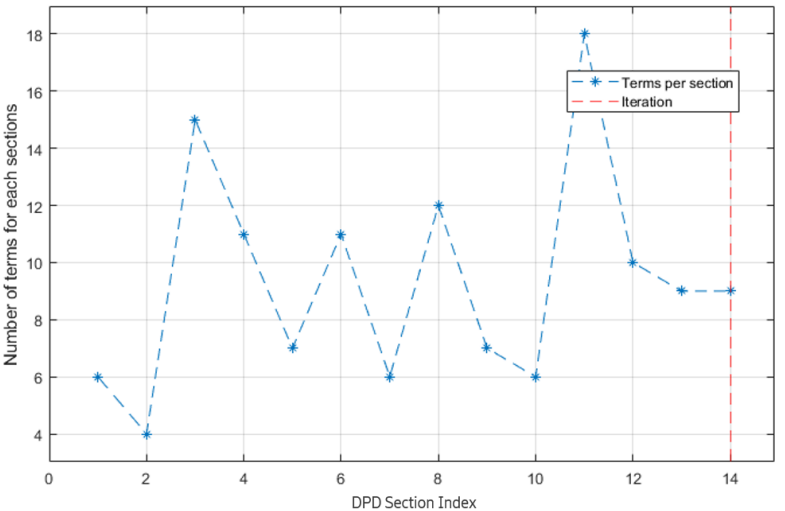


(b) Number of DPD terms for each DPD section.

Figure 7.2 Performance statistics for a single random run of Baseline V1 on `medium.mat`.



(a) Number of carrier cases assigned to each DPD section.



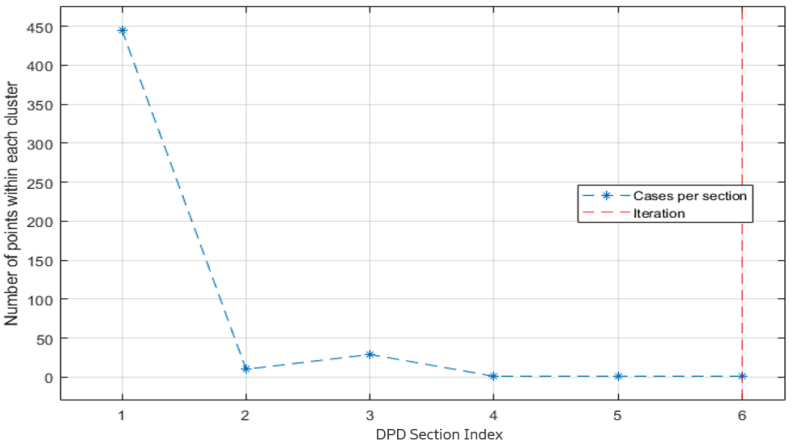
(b) Number of DPD terms for each DPD section.

Figure 7.3 Performance statistics for a single random run of Baseline V1 on `high.mat`.

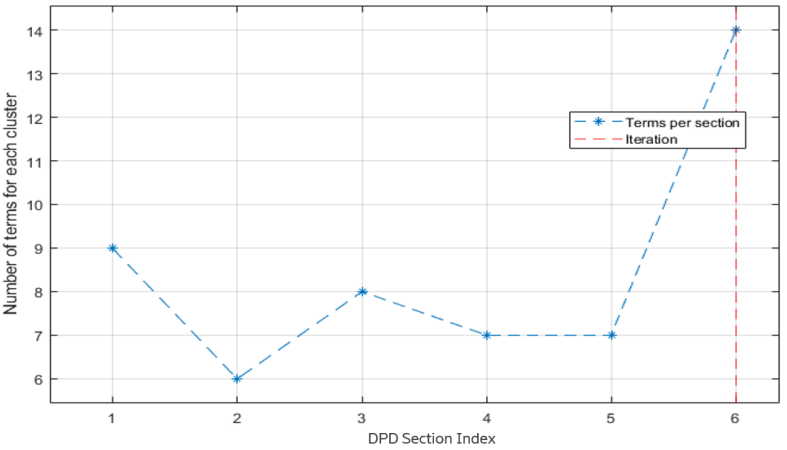
7.2 Baseline V2 – Hardest Case Carrier Allocation

Table 7.2 Results for Baseline V2 using hardest case sectioning allocation.

Dataset	No. validation tests	Mean No. DPD-terms	Max No. DPD-terms	No. of sections
`low`	567	8.88	14	6
`medium`	592	9.11	18	6
`high`	1013	10.63	23	10

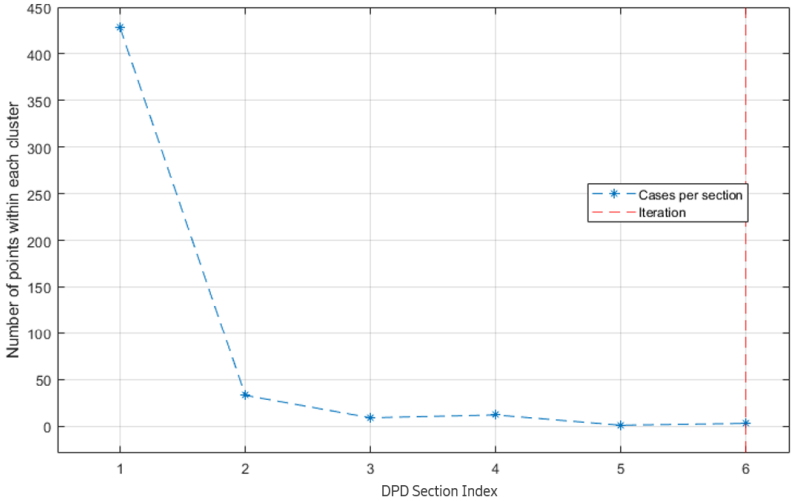


(a) Number of carrier cases assigned to each DPD section.

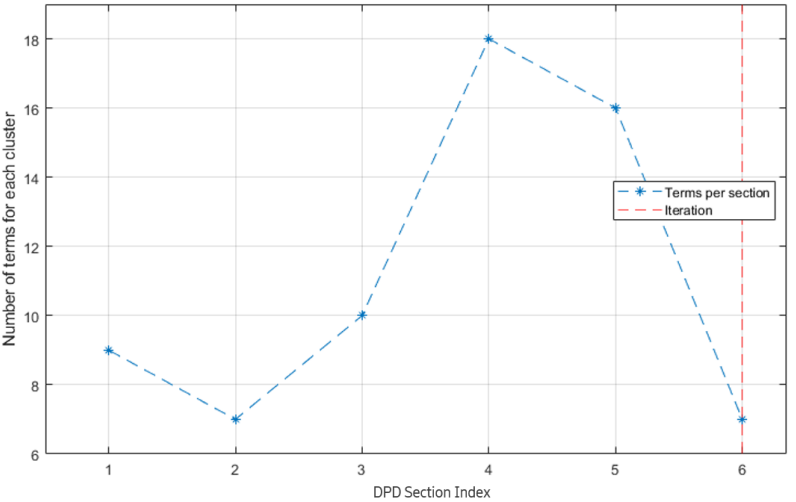


(b) Number of DPD terms for each DPD section.

Figure 7.4 Performance statistics for a single hardest case run of Baseline V2 on `low`.

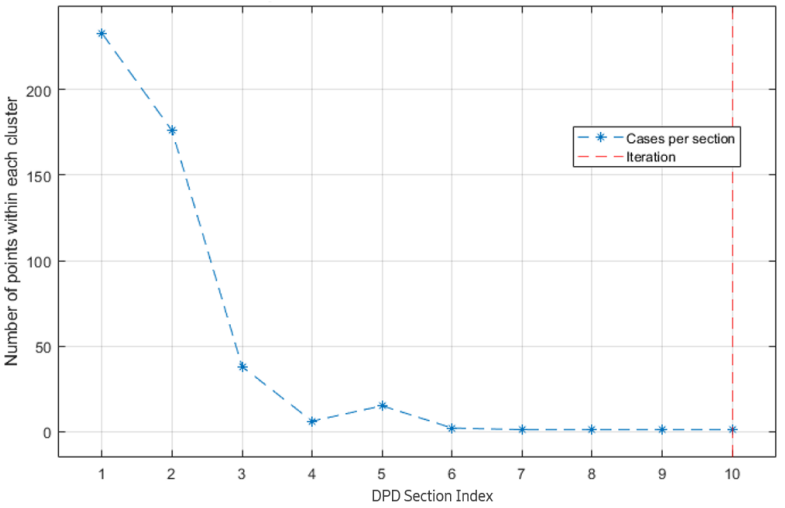


(a) Number of carrier cases assigned to each DPD section.

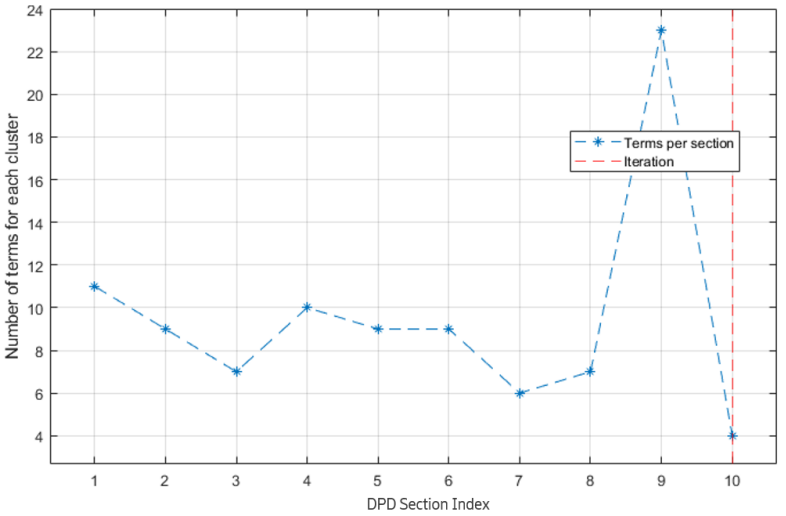


(b) Number of DPD terms for each DPD section.

Figure 7.5 Performance statistics for a single hardest case run of Baseline V2 on `medium`.



(a) Number of carrier cases assigned to each DPD section.



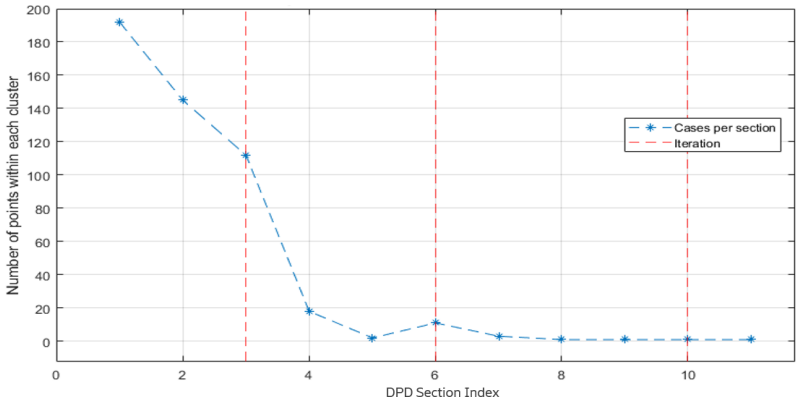
(b) Number of DPD terms for each DPD section.

Figure 7.6 Performance statistics for a single hardest case run of Baseline V2 on `high`.

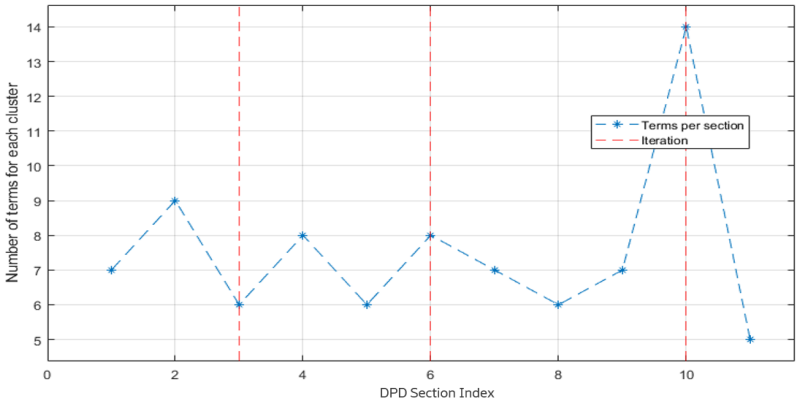
7.3 Clustering V1 – Hardest Case Carrier Selection

Table 7.3 Results for Clustering V1 using hardest case with iterative features.

Dataset	No. validation tests	Mean No. DPD-terms	Max No. DPD-terms	No. of sections
`low`	532	7.43	14	11
`medium`	551	7.60	18	13
`high`	802	8.80	23	27

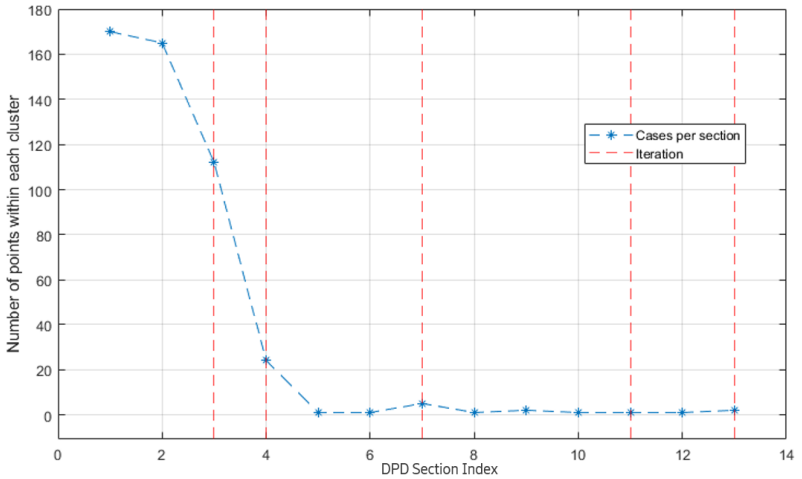


(a) Number of carrier cases assigned to each DPD section.

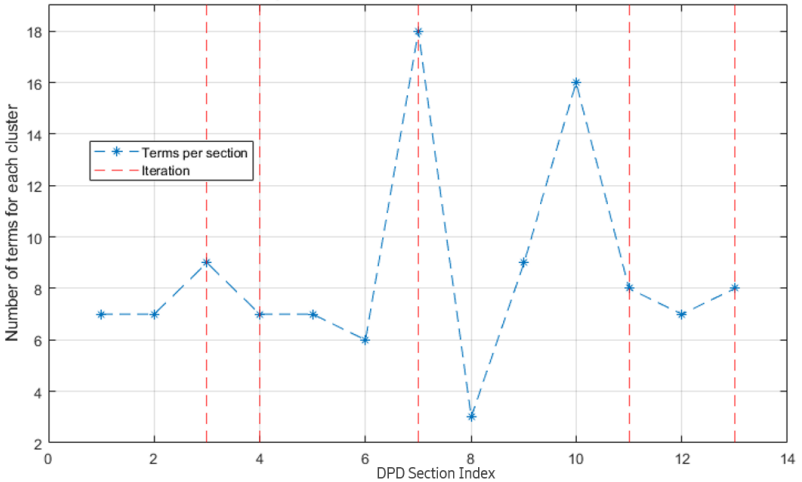


(b) Number of DPD terms for each DPD section.

Figure 7.7 Performance statistics for a single hardest case run of Clustering V1 on `low`.

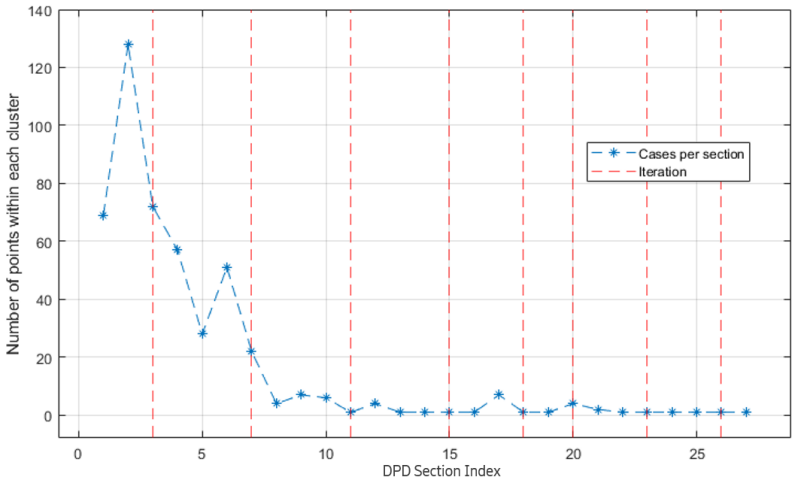


(a) Number of carrier cases assigned to each DPD section.

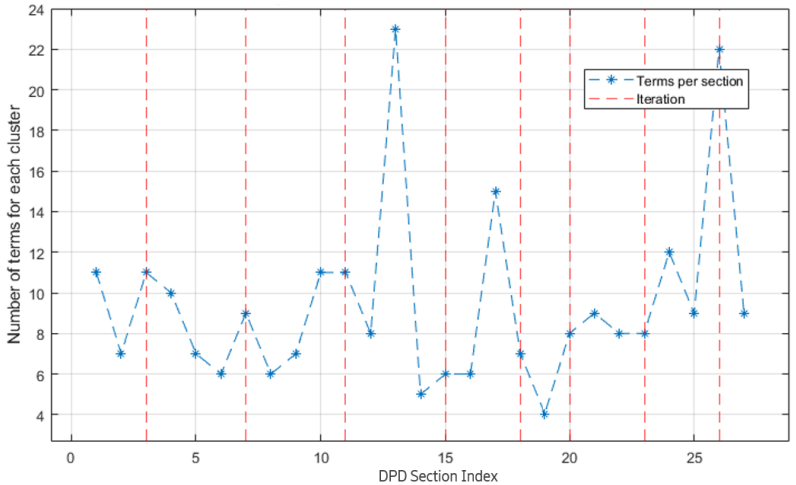


(b) Number of DPD terms for each DPD section.

Figure 7.8 Performance statistics for a single hardest case run of Clustering V1 on `medium`.



(a) Number of carrier cases assigned to each DPD section.



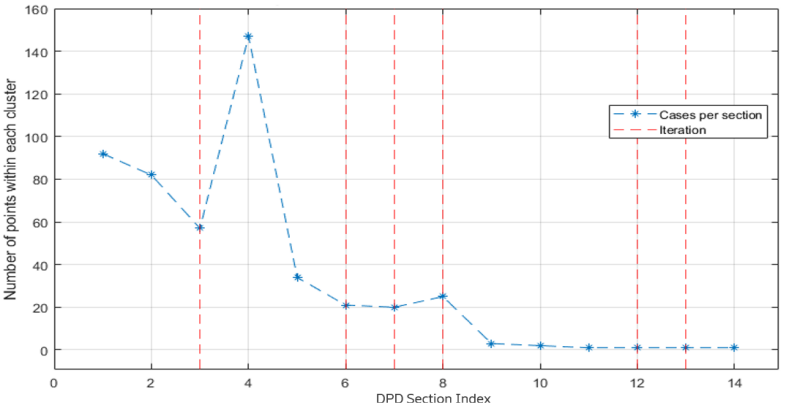
(b) Number of DPD terms for each DPD section.

Figure 7.9 Performance statistics for a single hardest case run of Clustering V1 on `high`.

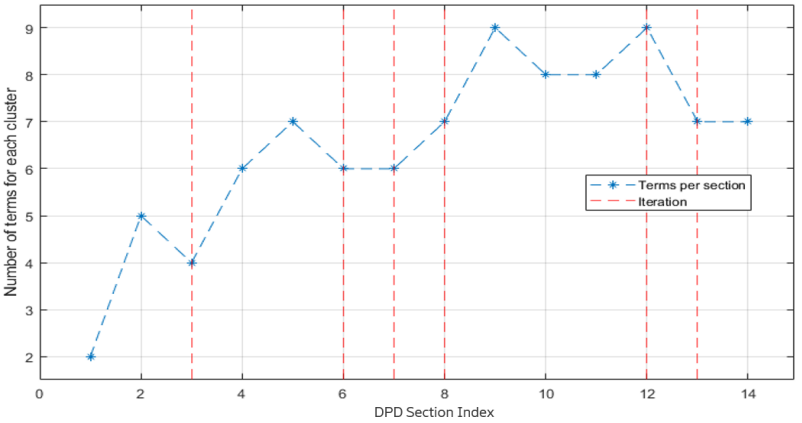
7.4 Clustering V2 – Centroid Case Carrier Selection

Table 7.4 Results for Clustering V2 using centroid case with iterative features.

Dataset	No. validation tests	Mean No. DPD-terms	Max No. DPD-terms	No. of sections
`low`	842	5.00	9	14
`medium`	879	5.13	13	18
`high`	945	6.20	23	23

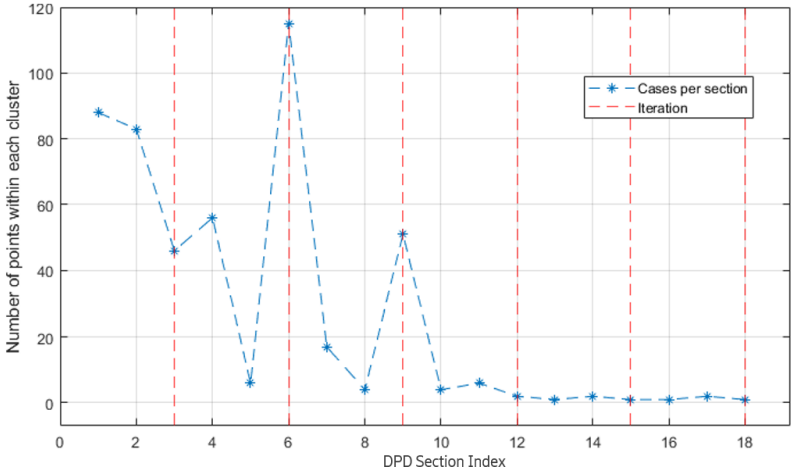


(a) Number of carrier cases assigned to each DPD section.

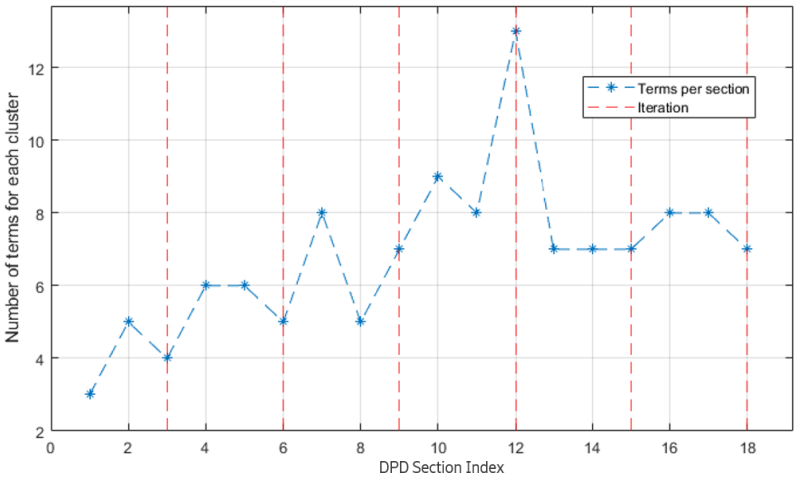


(b) Number of DPD terms for each DPD section.

Figure 7.10 Performance statistics for a single centroid case run of Clustering V2 on `low`.

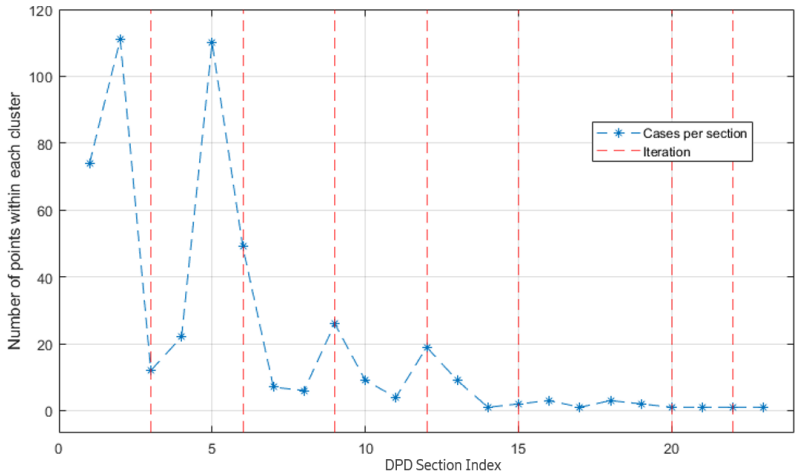


(a) Number of carrier cases assigned to each DPD section.

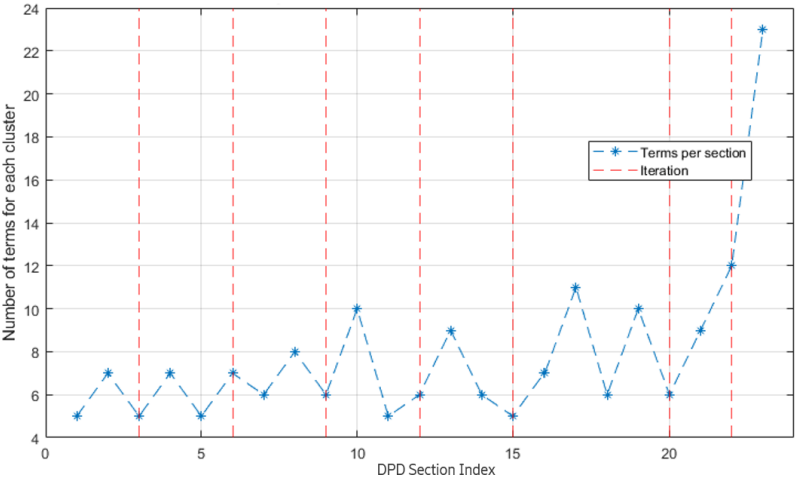


(b) Number of DPD terms for each DPD section.

Figure 7.11 Performance statistics for a single centroid case run of Clustering V2 on `medium`.



(a) Number of carrier cases assigned to each DPD section.



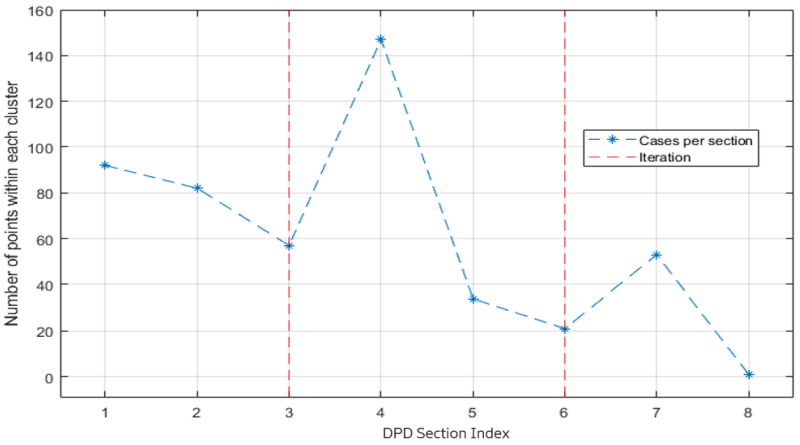
(b) Number of DPD terms for each DPD section.

Figure 7.12 Performance statistics for a single centroid case run of Clustering V2 on `high`.

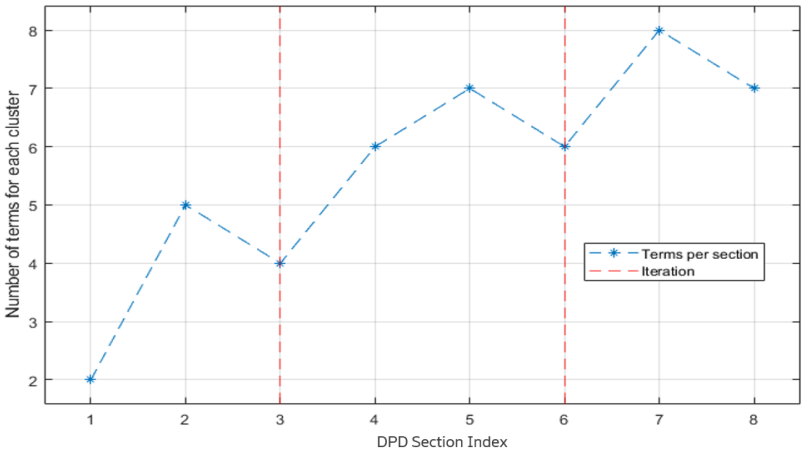
7.5 Clustering V3 – Hybrid Method

Table 7.5 Results for Clustering V3 using hybrid case with iterative features.

Dataset	No. validation tests	Mean No. DPD-terms	Max No. DPD-terms	No. of sections
`low`	798	5.13	8	8
`medium`	875	5.41	9	11
`high`	931	6.71	11	13

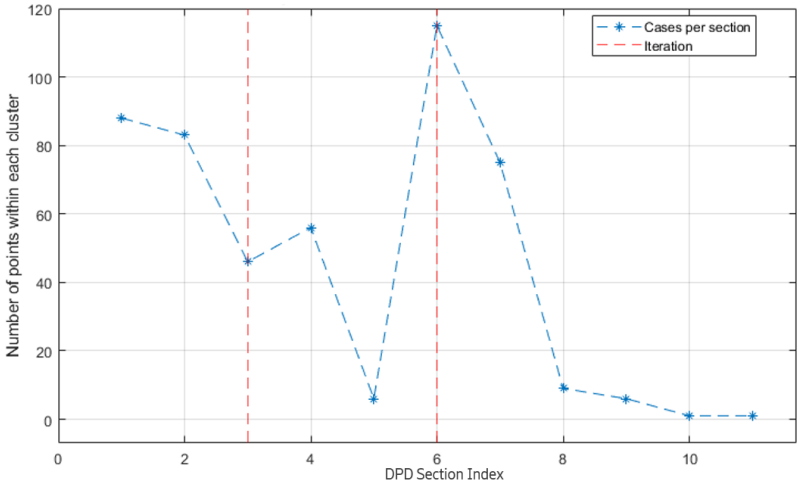


(a) Number of carrier cases assigned to each DPD section.

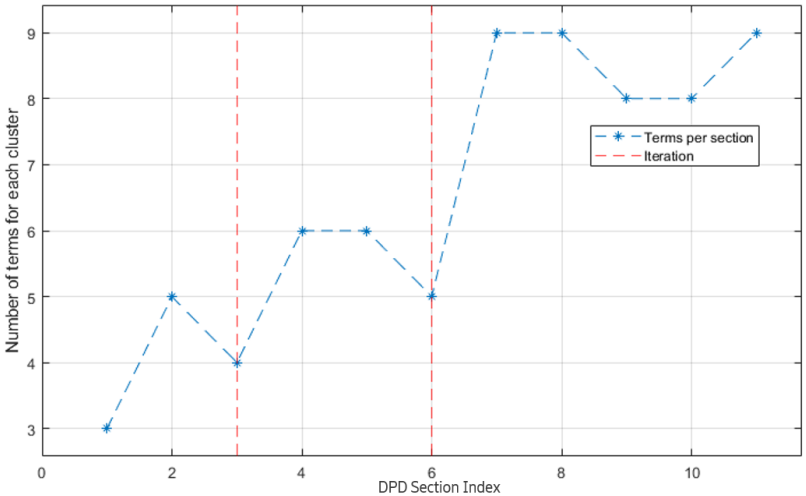


(b) Number of DPD terms for each DPD section.

Figure 7.13 Performance statistics for a single hybrid case run of Clustering V3 on `low`.

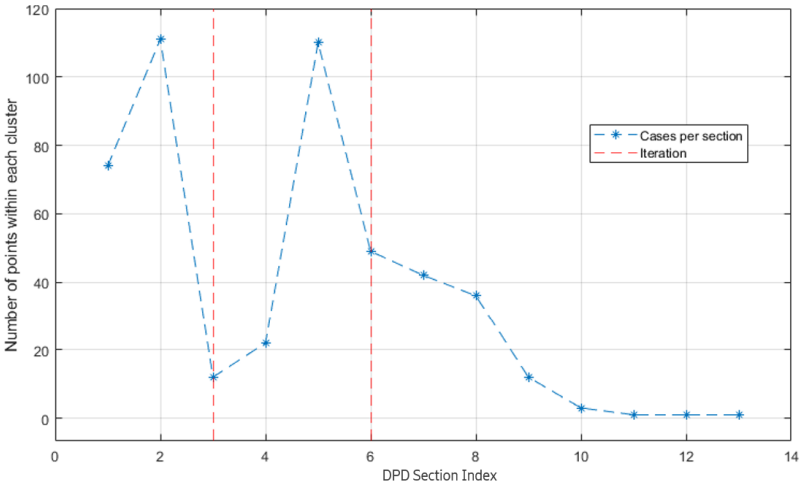


(a) Number of carrier cases assigned to each DPD section.

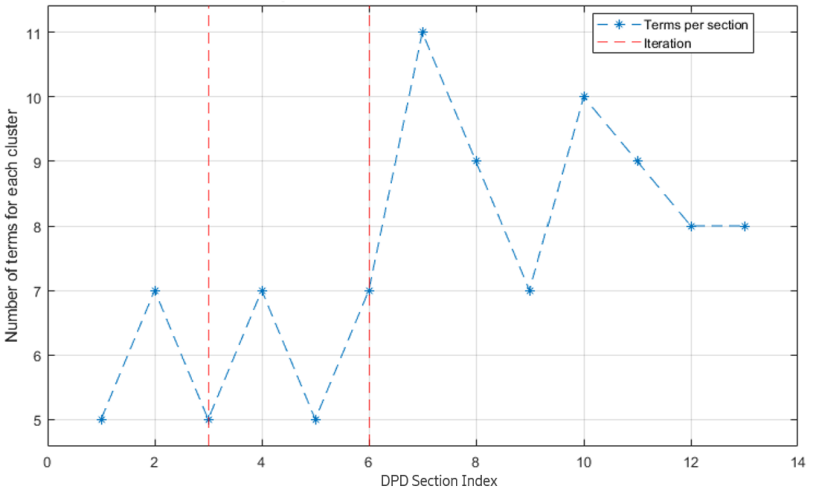


(b) Number of DPD terms for each DPD section.

Figure 7.14 Performance statistics for a single hybrid case run of Clustering V3 on `medium`.



(a) Number of carrier cases assigned to each DPD section.



(b) Number of DPD terms for each DPD section.

Figure 7.15 Performance statistics for a single hybrid case run of Clustering V3 on `high`.

7.6 Integration time

We have estimated approximate durations in (time units, t.u.) for the time constants in Equation (6.1) from typical DPD testing times and from a textbook implementation of DOMP for DPD model selection. The estimated values corresponds to the following DPD integration time according to (6.1) Time Unit (t.u), $T_0 = 199.8$ t.u , $T_{LT} = 3.7$ t.u and $T_{TEST} = 1.85$ t.u. By using these constants the following table of approximative integration time values could be calculated,

Table 7.6 Estimated DPD integration time for different methods, in Time Unit (t.u).

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
`low`	3229	2436	3489	4691	3241
`medium`	4193	2541	4031	5677	4082
`high`	5632	3910	7821	7013	4682

7.7 Summary

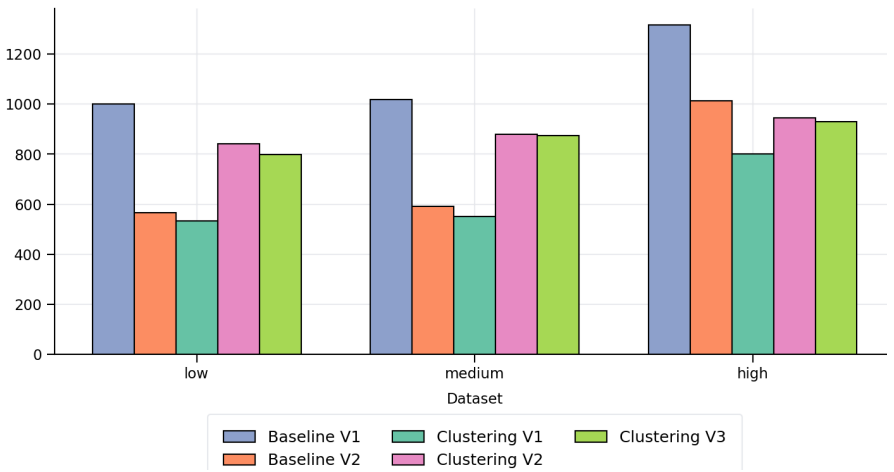


Figure 7.16 Comparison of required number of performance validation tests for all methods.

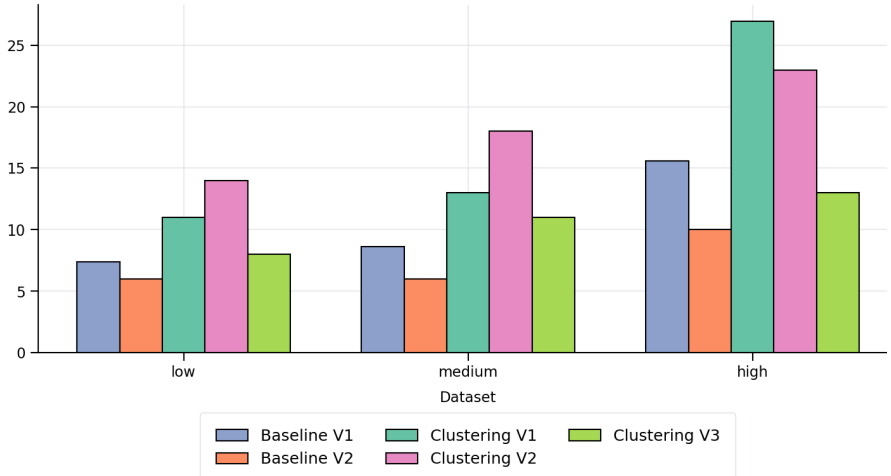


Figure 7.17 Comparison of required number of DPD sections for all methods.

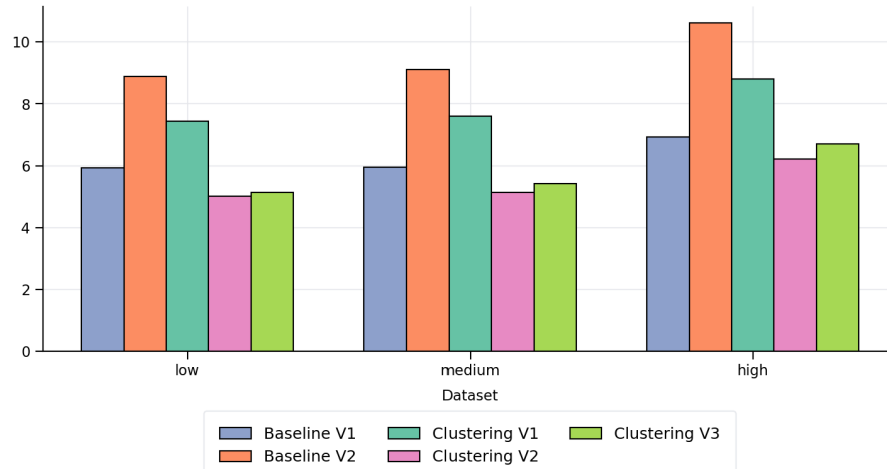


Figure 7.18 Comparison of average number of DPD terms for all methods

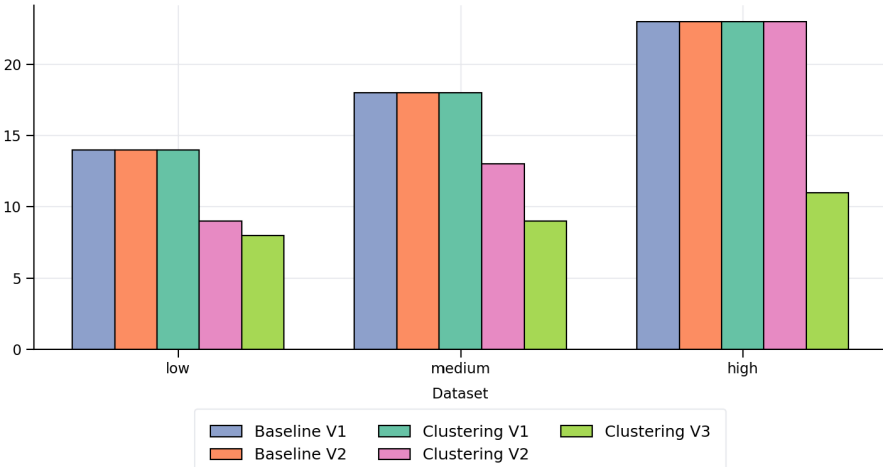


Figure 7.19 Comparison of max number of DPD terms for all methods

Table 7.7 Average number of sections for different methods.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	7.42	6	11	14	8
m8	8.62	6	13	18	11
m5	15.56	10	27	23	13

Table 7.8 Mean number of DPD-terms for different methods.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	5.93	8.88	7.43	5.01	5.13
m8	5.94	9.11	7.60	5.13	5.41
m5	6.92	10.63	8.80	6.20	6.71

Table 7.9 Maximum number of DPD-terms for different methods.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	14	14	14	9	8
m8	18	18	18	13	9
m5	23	23	23	23	11

Table 7.10 Estimated DPD integration time for different methods, in Time Unit (t.u).

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
low	3229	2436	3489	4691	3241
medium	4193	2541	4031	5677	4082
high	5632	3910	7821	7013	4682

Table 7.11 Percentage difference, average number of sections compared to Baseline V2.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	+23.67%	0.00%	+83.33%	+133.33%	+33.33%
m8	+43.67%	0.00%	+116.67%	+200.00%	+83.33%
m5	+55.60%	0.00%	+170.00%	+130.00%	+30.00%

Table 7.12 Percentage difference, mean number of DPD-terms compared to Baseline V2.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	-33.23%	0.00%	-16.35%	-43.65%	-42.22%
m8	-34.76%	0.00%	-16.58%	-43.70%	-40.56%
m5	-34.88%	0.00%	-17.14%	-41.61%	-36.88%

Table 7.13 Percentage difference, maximum number of DPD-terms compared to Baseline V2.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
m11	0.00%	0.00%	0.00%	-35.71%	-42.86%
m8	0.00%	0.00%	0.00%	-27.78%	-50.00%
m5	0.00%	0.00%	0.00%	0.00%	-52.17%

Table 7.14 Percentage difference in estimated DPD integration time compared to Baseline V2.

Dataset	Baseline V1	Baseline V2	Clustering V1	Clustering V2	Clustering V3
low	+32.55%	0.00%	+43.23%	+92.57%	+33.05%
medium	+65.01%	0.00%	+58.64%	+123.42%	+60.65%
high	+44.04%	0.00%	+100.03%	+79.36%	+19.74%

8

Discussion

Comparing the Baseline methods with the Clustering methods, a clear difference can be observed in how the different approaches balance DPD complexity, number of sections, and total integration time. On a general level, the Baseline methods tend to produce configurations with a higher number of terms per section, which can be viewed as a form of "overfitting". This is especially visible when comparing Baseline V2 with Clustering V2 in Figure 7.18. The increased complexity of Baseline V2 can be understood as a consequence of its conservative selection strategy, where carrier cases with larger BW are used to configure the sections. Since larger bandwidths generally require a higher number of terms, as shown in Figure 5.2, this results in configurations that satisfy the performance requirements for a large portion of the carrier signals, but often use more DPD resources than necessary.

One advantage of the clustering-based approach is that it provides a more systematic way of selecting representative candidates for each section. In the Baseline methods, the choice of representative cases is limited to either random selection or predefined rules (e.g. Hardest Case). Since no direct information about the underlying structure of the carrier configurations is used, the selection becomes less analytical. By contrast, the clustering methods use the structure of the feature space to identify groups of similar carrier configurations. This makes it possible to select representative cases based on the cluster structure itself.

This effect can be seen when comparing Clustering V1 and Clustering V2. In Clustering V1, the hardest case within each cluster is used to define the section, whereas Clustering V2 instead uses the carrier case closest to the cluster centroid. The results show that Clustering V2 reduces the mean number of terms by 2.4 compared to Clustering V1, as shown in Figure 7.18 and Tables 7.3–7.4. This indicates that the centroid point represents the general cluster characteristics well enough to produce a less complex configuration, while still allowing a large portion of the cases in the cluster to fulfil the performance requirements. In other words, selecting the hardest case gives a more conservative configuration, whereas selecting the centroid gives a more resource-efficient configuration.

However, the comparison between Clustering V2 and Clustering V3 shows that the clustering approach is not equally effective throughout all iterations. The clus-

tering algorithm performs well during the first iterations, but its performance decreases as the remaining dataset becomes smaller. This can be explained by the fact that clustering methods rely on having enough data to identify meaningful structure. When fewer data points remain, the statistical reliability of the clusters decreases, and the resulting clusters become more sensitive to individual cases. This is especially relevant for GMM, where the estimated Gaussian components become less reliable when only a few samples are available.

This behaviour is visible in Figures 7.7a and 7.7b, where the last sections contain only a small number of points each, while the complexity of these sections varies from low to high. This suggests that the remaining dataset in the later iterations is not necessarily well suited for further clustering. As a result, the clustering methods are most useful in the begining when there is still enough data available to identify stable and meaningful clusters.

Looking at the total integration time in Table 7.10, Baseline V2 performs significantly better than the clustering methods. This is mainly because the total integration time is heavily influenced by the number of sections. Each additional section introduces a fixed cost through the term T_0, since a new section requires initialization and configuration steps. In comparison, the contributions from model complexity and validation tests are less dominant in the integration-time model. Therefore, methods that produce fewer sections, such as Baseline V2, obtain a clear advantage when total integration time is the primary optimization objective.

At the same time, the results reveal an interesting, and important, trade-off between the complexity of the DPD configuration and the number of generated sections. Reducing the number of sections tends to increase the number of terms required per section, while reducing the number of terms tends to increase the number of sections. It is outside the scope of this thesis to investigate this result further, as it was an observation from the proposed clustering method, but it could be worth investigting further in the future if there is a correlation between the number of sections and the average number of terms. However no grounded conclusion of this can be made from the result presented in this thesis.

The preferred solution therefore depends on the practical priorities, such as e.g., configuration time and DPD complexity/runtime power consumption. This trade-off is directly related to the two optimization formulations introduced in Section 5.2. If the goal is to reduce offline tuning, testing, and configuration time, then minimizing the number of sections becomes the most important objective. As discussed in Section 7.6, the fixed cost T_0 is the dominant contributor to the total integration time, since it corresponds to the time required to initialize and configure a new section. From this perspective, Baseline V2 is the best-performing method across the tested datasets, as shown in Table 7.10.

If the goal instead is to reduce digital power consumption, in the radio unit, then minimizing the number of DPD terms becomes more important. The number of terms is related to the number of coefficients that the DPD-ASIC must update and evaluate during operation. A larger number of terms therefore requires more

calculations, which increases energy consumption, heat generation, and stress on the hardware. From this perspective, Clustering V2 is the strongest method, since it achieves the lowest mean number of terms across the tested datasets, as shown in Figure 7.18.

The disadvantage of Clustering V2 is that it produces a large number of sections, as shown in Figure 7.17. This leads to a high total integration time, which makes the method less attractive from a production perspective. This is also confirmed in Table 7.10, where Clustering V2 has one of the highest integration times among the evaluated methods. The result therefore highlights a central finding of the work, optimizing only for model complexity increases the number of sections, while optimizing only for integration time increases the model complexity.

This analysis motivates the introduction of Clustering V3, which was designed as a hybrid between Baseline V2 and Clustering V2. The purpose of this method is to combine the strengths of both approaches by reducing the number of sections while still keeping the number of terms lower than in the pure Baseline approach. The results in Table 7.10 and Figures 7.16, 7.17, and 7.18 indicate that Clustering V3 behaves as a compromise between the two extremes. It produces slightly more terms than Clustering V2, but fewer sections than the pure clustering approach. At the same time, it requires more sections than Baseline V2, but generally avoids the highly conservative term usage of the baseline configuration.

This indicates that Clustering V3 gives a favourable balance between model complexity and section count. However, the total integration time is still noticeably higher than for Baseline V2, as shown in Table 7.10. This was expected, since Clustering V3 still introduces more sections than Baseline V2, but the intention was that the hybrid method would become a stronger competitor also with respect to integration time. The results therefore show that Clustering V3 improves the balance between the objectives, but does not fully overcome the strong influence of the number of sections in the current integration-time.

Overall, the results show that the preferred method depends strongly on the final optimization objective. If production-time efficiency and short offline integration are the most important criteria, Baseline V2 remains the most attractive method. If reduced DPD complexity, lower digital power consumption, and reduced hardware load during operation are more important, then the clustering-based methods become more interesting, especially Clustering V2 and Clustering V3. This gives an alternative system-level perspective: should more time be spent offline during production in order to reduce the complexity and energy consumption of the DPD once the radio unit is deployed in the field, or should production time be minimized even if the resulting configuration is more complex? This question is outside the scope of this thesis, but the results indicate that it is an important and interesting direction to take for future investigation.

9

Conclusions and Future Work

This thesis has investigated whether machine-learning methods can support and improve the configuration process for DPD in modern radio systems. Two baseline methods were compared against three proposed unsupervised learning clustering-based methods across three datasets, ranging from lower to higher nonlinear behavior and from simpler single-carrier cases to more complex multi-carrier configurations. The methods were evaluated with respect to section count, DPD complexity, number of validation tests, and total integration time.

The results show that no single method is optimal for all objectives. Baseline V2 achieved the lowest total integration time, mainly due to its lower number of generated sections, but at the cost of higher DPD complexity. In contrast, the clustering-based methods, especially Clustering V2, reduced the number of DPD terms and therefore performed better when model complexity was the main objective. Clustering V3 provided the most balanced result among the proposed methods, acting as a hybrid between the low section count of Baseline V2 and the lower DPD complexity of Clustering V2.

Thus, Clustering V3 cannot be concluded to be an objective improvement over the Baseline methods in all aspects. Instead, the preferred method depends on the optimization objective. If total integration time is prioritized, Baseline V2 remains the strongest alternative. If reduced DPD complexity and resource usage are prioritized, the clustering-based methods become more attractive. Despite not outperforming in all aspects the proposed method provides a more structured foundation and can be more easily parametrized and tuned using the optimization approaches in Section 5.2.The main conclusion is therefore that carrier configurations contain useful structure that can be exploited to support more efficient DPD configuration. This indicates that ML-based methods can play an important complementary role in future automated, resource-efficient, and scalable DRT methods.

RQ1: What efficient and effective approaches can be used to configure the parameters of a target DPD architecture across different use cases and requirements?.

The results indicate that the most effective approach depends on the selected optimization objective. If the goal is to minimize total integration time, a baseline-style method with fewer sections is more efficient. If the goal is instead to reduce DPD complexity and resource usage, clustering-based methods are more effective. Therefore, an efficient DPD configuration strategy should not be evaluated using only one metric, but should account for the trade-off between section count, model complexity, and validation. In this work, the hybrid method Clustering V3 provided the most promising balance between these objectives.

RQ2: Can machine learning methods, such as reinforcement learning or unsupervised learning, complement or replace conventional optimization methods, and what synergies may exist?. The results show that unsupervised learning can complement conventional optimization methods in the DPD configuration process. Clustering methods were able to identify structure among carrier configurations and use this structure to reduce the required DPD complexity compared to the baseline methods. However, the results also show that clustering alone is not always sufficient, especially when the remaining dataset becomes small and the cluster structure becomes less reliable. The strongest result was obtained when Clustering V2 was combined with Baseline V2 to create a Hybrid-Method, Clustering V3. This suggests that machine learning is most useful as a complementary tool rather than a replacement for conventional optimization in this application.

Future Work

Future work could explore alternative clustering methods, such as BIRCH or OPTICS, as well as further development of the features used for clustering. Since this thesis only evaluated a limited set of generated datasets, future studies should also include additional datasets with different channels, carrier configurations, and operating conditions to better assess the generalization of the proposed method.

Another interesting direction is to make better use of the information obtained during each clustering iteration. In the current implementation, each iteration provides new information about valid configurations, section behavior, and remaining unassigned cases. This information could potentially be used to guide later iterations more intelligently.

A further important aspect is to quantify the potential energy savings obtained by reducing the number of DPD terms. In this thesis, the number of terms was used as an indication of DPD complexity, but the direct relation between reduced complexity, energy consumption, and hardware load was not measured. Investigating this relation would provide a clearer understanding of the practical benefits of the proposed methods.

Finally, an initial research question considered during the early phase of the thesis was whether machine-learning methods could predict and avoid unstable DPD configurations, thereby improving stability and reducing optimization time. This question was not investigated further due to time limitations and because the the-

sis direction became more focused on configuration efficiency and clustering-based sectioning. However, this remains an interesting topic for future work, since reliable prediction of unstable configurations could have important implications for more robust and automated DPD configuration. For this type of work one would prefer a radio product-based testbench over a simulation based approach taken in our thesis.

Bibliography

[1] 3G Mobile. *The big switch: why moving from 3g to 4g/5g matters in 2026*. Accessed: 2026-04-22. 2026. URL: https://www.3gmobile.com/2026/03/06/the-big-switch-why-moving-from-3g-to-4g-5g-matters-in-2026/.

[2] 3GPP. *About 3gpp*. Accessed April 23, 2026. 2026. URL: https://www.3gpp.org/about-us.

[3] 3rd Generation Partnership Project (3GPP). *NR; Base Station (BS) Radio Transmission and Reception*. Technical Specification TS 38.104 V18.5.0. Release 18. 3GPP, 2024. URL: https://www.3gpp.org/.

[4] A. Barry, W. Li, J. A. Becerra, and P. L. Gilabert. "Comparison of feature selection techniques for power amplifier behavioral modeling and digital predistortion linearization". *Sensors* **21**:17 (2021), p. 5772. DOI: 10.3390/s21175772. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC8433820/.

[5] A. Barry, W. Li, J. A. Becerra, and P. L. Gilabert. "Comparison of feature selection techniques for power amplifier behavioral modeling and digital predistortion linearization". *Sensors* **21**:17 (2021), p. 5772. DOI: 10.3390/s21175772. URL: https://www.mdpi.com/1424-8220/21/17/5772.

[6] J. A. Becerra, M. J. Madero-Ayora, J. Reina-Tosina, C. Crespo-Cadenas, J. García-Frías, and G. Arce. "A doubly orthogonal matching pursuit algorithm for sparse predistortion of power amplifiers". *IEEE Microwave and Wireless Components Letters* **28**:8 (2018), pp. 726–728. DOI: 10.1109/LMWC.2018.2845947.

[7] B. Bernhardsson and K. J. Åström. *Iterative learning control (ilc)*. Course notes. 2016.

[8] D. Brown and Y. Rahmat-Samii. "Error vector magnitude as a performance standard for antennas in the millimeter-wave era: part 1: metric comparisons and measurement concepts". *IEEE Antennas and Propagation Magazine* **65**:5 (2023), pp. 25–34. DOI: 10.1109/MAP.2023.3272838.

[9] D. Brown and Y. Rahmat-Samii. “Error vector magnitude as a performance standard for antennas in the millimeter-wave era: part 2: modeling, simulation, and measurement methods”. *IEEE Antennas and Propagation Magazine* **65** (2023), pp. 10–24. DOI: 10.1109/MAP.2023.3272842.

[10] J. Chani-Cahuana, P. N. Landin, C. Fager, and T. Eriksson. “Iterative learning control for rf power amplifier linearization”. *IEEE Transactions on Microwave Theory and Techniques* **64**:9 (2016), pp. 2778–2790. DOI: 10.1109/TMTT.2016.2588483.

[11] G. Collins and D. W. Runton. “Nonlinear analysis of power amplifiers”. *Microwave Journal* (2007). Online article.

[12] E. Dahlman, S. Parkvall, and J. Sköld. *5G NR: The Next Generation Wireless Access Technology*. Academic Press, 2018. ISBN: 978-0-12-814323-0.

[13] S. Dikmese, L. Anttila, P. P. Campo, M. Valkama, and M. Renfors. “Behavioral modeling of power amplifiers with modern machine learning techniques”. *IEEE Transactions on Microwave Theory and Techniques* **67**:7 (2019), pp. 2813–2824. DOI: 10.1109/TMTT.2019.2912663.

[14] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina. “A robust digital baseband predistorter constructed using memory polynomials”. *IEEE Transactions on Communications* **52**:1 (2004), pp. 159–165. DOI: 10.1109/TCOMM.2003.822188.

[15] A. Djordjevi. *RF and Microwave Circuit Design*. Artech House, 2022. ISBN: 978-1-63081-826-5.

[16] M. Doufana and C.-W. Park. “Neural network based power amplifier dynamic modeling for wireless communications”. In: *Proceedings of the IEEE International Conference on Microwave and Millimeter Wave Technology (ICMMT)*. 2008, pp. 1–4. DOI: 10.1109/ICMMT.2008.4540592.

[17] A. Erdogan, J. Lin, K. George, and J. Juliano. “Pulse on pulse deinterleaving radar algorithm”. In: IEEE. 2020. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9031148.

[18] T. Eriksson and M. M. Abdelnaeim. *Waveform Clustering: Grouping Similar Power System Events*. Master’s Thesis. Mälardalen University, Västerås, Sweden, 2019. URL: https://www.diva-portal.org/smash/get/diva2:1325944/FULLTEXT01.pdf.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. “A density-based algorithm for discovering clusters in large spatial databases with noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996, pp. 226–231.

[20] M. Fezari. *Analog Front-End Design and Simulation*. Tech. rep. Technical report. Badji Mokhtar Annaba University, Faculty of Technology, 2025. URL: https://www.researchgate.net/publication/389846351_Analog_Front-End_Design_and_Simulation.

[21] C. Fraley and A. E. Raftery. “Model-based clustering, discriminant analysis, and density estimation”. *Journal of the American Statistical Association* **97**:458 (2002), pp. 611–631. DOI: 10.1198/016214502760047131.

[22] S. Gasperini, M. Paschali, C. Hopke, D. Wittmann, and N. Navab. “Signal clustering with class-independent segmentation”. In: IEEE. 2020. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9053409.

[23] F. M. Ghannouchi and O. Hammi. “Behavioral modeling and predistortion”. *IEEE Microwave Magazine* **10**:7 (2009), pp. 52–64. DOI: 10.1109/MMM.2009.934513.

[24] M. A. Hussein, O. Venard, B. Feuvrie, and Y. Wang. “Digital predistortion for rf power amplifiers: state of the art and advanced approaches”. In: *Proceedings of the European Microwave Conference*. 2013.

[25] H. Imtiaz, A. Rusch, Z. Zheng, R. H. Nejad, Leslie, and M. Zeng. “Performance vs. complexity in nn pre-distortion for a nonlinear channel”. *Optics Express* **31**:23 (2023), pp. 38513–38527. DOI: 10.1364/OE.500467.

[26] N. Kaoungku, K. Suksut, R. Chanklan, K. Kerdprasop, and N. Kerdprasop. “The silhouette width criterion for clustering and association mining to select image features”. *International Journal of Machine Learning and Computing* **8**:1 (2018), pp. 69–72. DOI: 10.18178/ijmlc.2018.8.1.665. URL: https://www.researchgate.net/publication/323588077_The_silhouette_width_criterion_for_clustering_and_association_mining_to_select_image_features.

[27] M. Y. Li, I. Galton, L. E. Larson, and P. M. Asbeck. “Nonlinearity estimation and spectral regrowth prediction of power amplifiers using correlation techniques”. In: *2005 IEEE Annual Wireless and Microwave Technology Conference*. 2005.

[28] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön. *Machine Learning: A First Course for Engineers and Scientists*. Online draft version, March 4, 2026. Cambridge University Press, 2026. URL: https://smlbook.org/.

[29] S. A. Maas. “Volterra analysis of spectral regrowth”. *IEEE Microwave and Guided Wave Letters* **7**:7 (1997), pp. 192–193. DOI: 10.1109/75.605596.

[30] U. Madhow. *Fundamentals of Digital Communication*. Cambridge University Press, 2008. ISBN: 978-0-521-87532-4.

[31] Mattt. *Dbscan*. GitHub repository, accessed April 8, 2026. URL: https://github.com/mattt/DBSCAN.

[32] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan. “A generalized memory polynomial model for digital predistortion of rf power amplifiers”. *IEEE Transactions on Signal Processing* **54**:10 (2006), pp. 3852–3860. DOI: 10.1109/TSP.2006.879264. URL: https://ieeexplore.ieee.org/document/1703853.

[33] NASA Science. *Radio waves*. Page last updated: Aug 03, 2023. 2023. URL: https://science.nasa.gov/ems/05_radiowaves/.

[34] A. V. Oppenheim and G. C. Verghese. *Chapter 10: power spectral density*. MIT OpenCourseWare, 6.011 Introduction to Communication, Control, and Signal Processing. Available online. 2010.

[35] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab. *Signals and Systems, edition 2*. Pearson Education Limited, 2014. ISBN: 978-1-292-02590-2.

[36] C. S. Park, Y. Gim, J. Park, and S. Park. "Design and implementation of a digital front-end with digital compensation for low-complexity 4g radio transceivers". *IEEE Access* **9** (2021), pp. 111432–111454. DOI: 10.1109/ACCESS.2021.3102949.

[37] P. Patel, B. Sivaiah, and R. Patel. "Approaches for finding optimal number of clusters using k-means and agglomerative hierarchical clustering techniques". In: *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*. IEEE, 2022. DOI: 10.1109/ICICCSP53532.2022.9862439. URL: https://ieeexplore.ieee.org/document/9862439.

[38] J. G. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill, 2008. ISBN: 978-0-07-295716-7.

[39] F. H. Raab, P. Asbeck, S. Cripps, P. B. Kenington, Z. B. Popović, N. Pothecary, J. F. Sevic, and N. O. Sokal. "Power amplifiers and transmitters for rf and microwave". *IEEE Transactions on Microwave Theory and Techniques* **50**:3 (2002), pp. 814–826. DOI: 10.1109/22.989965. URL: https://ieeexplore.ieee.org/document/989965.

[40] S. Sahin. *Principal component analysis (pca) made easy: a complete hands-on guide*. Accessed 2026-04-13. 2024. URL: https://medium.com/@sahin.samia/principal-component-analysis-pca-made-easy-a-complete-hands-on-guide-e26a3680c0bc.

[41] A. A. M. Saleh and J. Salz. "Adaptive linearization of power amplifiers in digital radio systems". *The Bell System Technical Journal* **62**:4 (1983), pp. 1885–1890.

[42] D. Schreurs, M. O'Droma, A. A. Goacher, and M. Gadringer. *RF Power Amplifier Behavioral Modeling*. Information cited from page 99. Cambridge University Press, Cambridge, UK, 2009. ISBN: 978-0-521-88173-9. URL: https://picture.iczhiku.com/resource/eetop/sYIGQJoOirFjAnXB.pdf.

[43] G. Schwarz. "Estimating the dimension of a model". *The Annals of Statistics* **6**:2 (1978), pp. 461–464. DOI: 10.1214/aos/1176344136.

[44] J. A. Stine. “Spectrum-consumption models”. In: B. A. Fette (Ed.). *Cognitive Radio Technology*. Copyright © 2009, Elsevier Inc. Elsevier, Burlington, MA, 2009. Chap. 20, pp. 645–687. URL: https://www.sciencedirect.com/science/chapter/monograph/abs/pii/B9780123745354000205?fr=RR-2&ref=pdf_download&rr=9e97c386099809af.

[45] P. Stoica and R. Moses. *Spectral Analysis of Signals*. Pearson Prentice Hall, 2005. ISBN: 0-13-113956-8.

[46] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, Massachusetts, 2018. ISBN: 9780262039246. URL: http://incompleteideas.net/book/the-book-2nd.html.

[47] J. Wood. “A glimpse of the future”. *IEEE Microwave Magazine* **13**:7 (2012), pp. 60–69. DOI: 10.1109/MMM.2012.2216094.

Popular Science Summary

Signal-Aware Predistortion Configuration Using Unsupervised Learning for 5G–Advanced/6G Radios

Can mobile networks become more energy efficient without sacrificing quality? This project shows how combining machine learning with traditional methods can optimize radio systems—while revealing an important trade-off.

Modern mobile networks are under constant pressure to deliver more data, faster and more efficiently. At the heart of this challenge is the radio transmitter's power amplifier, which boosts signals before transmission. However, these amplifiers are imperfect—they distort the signal and can cause interference with nearby channels.

To address this, engineers use a technique called digital predistortion (DPD). It works by "pre-correcting" the signal before it enters the amplifier, so that the final transmitted signal remains clean and accurate. The challenge is that configuring DPD is both time-consuming and resource-intensive.

This thesis explores whether machine learning can make this process more efficient. Using a method that automatically groups similar signal conditions, the system can determine how many DPD configurations are needed. The results show that this approach reduces resource usage—requiring less computation and memory—while still maintaining signal quality.

However, there is a catch: the method increases the time required to integrate and configure the system in hardware. In industrial settings, where time-to-deployment is critical, this becomes a significant drawback. The most promising solution turned out to be a hybrid approach, combining machine learning with traditional optimization techniques. This achieved both low resource usage and reasonable integration time.

The key takeaway is that there is a clear trade-off: improving efficiency in one area can make another more challenging. The optimal solution therefore depends on system priorities. At the same time, the results highlight how combining modern machine learning with established engineering methods can pave the way for smarter and more efficient future communication systems

Signalbaserad Konfigurering av Digital Predistorsion med hjälp av Oövervakad Inlärning för 5G–Avancerad/6G-Radio

Hur kan mobilnät bli både mer energieffektiva och behålla hög kvalitet? Genom att kombinera maskininlärning och klassiska metoder visar detta arbete hur radiosystem kan optimeras – men också att det finns en viktig avvägning.

Mobilnät världen över pressas att leverera mer data snabbare, samtidigt som energiförbrukningen måste hållas nere. En central komponent i detta är radiosändarens effektförstärkare, som gör signalen stark nog att nå fram. Problemet är att förstärkaren inte är perfekt – den förvränger signalen och kan orsaka störningar.

För att motverka detta används en teknik som kallas digital predistorsion (DPD). Den fungerar som en ”förhandskorrigering” av signalen, så att slutresultatet blir rätt trots förstärkarens brister. Men att konfigurera denna korrigering är både tidskrävande och resursintensivt.

I detta examensarbete undersöks om maskininlärning kan göra processen smartare. Genom en metod som grupperar liknande signalförhållanden automatiskt kan systemet avgöra hur många olika DPD-inställningar som behövs. Resultatet visar att detta kan minska resursanvändningen – mindre beräkningar och minne krävs – samtidigt som signalens kvalitet bibehålls.

Det finns dock en hake: metoden gör att det tar längre tid att ställa in systemet i praktiken. För industrin, där tid till färdig produkt är viktig, är detta en avgörande faktor.

Den mest lovande lösningen visade sig vara en kombination av maskininlärning och traditionella optimeringsmetoder. Denna hybridmetod gav både låg resursanvändning och rimlig integrationstid.

Slutsatsen är att det finns en tydlig kompromiss: att spara resurser kan kosta tid. Vilken lösning som är bäst beror därför på vad som prioriteras i systemet. Resultaten visar samtidigt att framtidens radiosystem kan bli både smartare och effektivare genom att kombinera nya och etablerade tekniker.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
June 2026

Document Number
TFRT-6316

Author(s)
Oskar Magnusson
Arvid Bergenfeldt

Supervisor
Simone Grimaldi, Ericsson, Sweden
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Signal-Aware Predistortion Configuration Using Unsupervised Learning for 5G–Advanced/6G Radios

Abstract

Digital Predistortion (DPD) is an essential component of modern mobile communication systems ensuring that distortion, created by the nonlinearities of the Power Amplifier (PA) is reduced while allowing them to be operated at high efficiency levels. This thesis has investigated the use of an unsupervised machine learning method known as Gaussian Mixture Model (GMM) as a way to define and populate a suitable number of DPD configurations to satisfy the required use cases for specific radio products. The goal was to minimize the required resources by the DPD as well as the actual DPD configuration time spent in the hardware integration process. The proposed method was compared against a baseline implementation and evaulated on the mentioned metrics. The result of the thesis was a reduction in resources compared to the baseline while maintaining system requirements quantified by Adjacent Channel Leakage Ratio (ACLR). However, the proposed method had a significant increase in hardware integration time compared to the baseline. Overall, the most promising method was a combination between machine learning and traditional optimization which yielded the lowest DPD resources while still maintaining lower integration time than the strictly machine learning based approach. We conclude that a clear trade-off exists between DPD resource utilization and hardware integration time, where the optimal configuration is dependent on system-level priorities. These findings highlight the potential of hybrid approaches, combining machine learning with conventional optimization, as an effective strategy for balancing performance and implementation complexity in practical systems.

Keywords
Digital Predistortion, Unsupervised Learning, 6G, Digital Radio Tuning, 5G–Advanced, Machine Learning, State-of-the-art, Optimization, Clustering, GMM, Artificial Intelligence

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-116

Recipient's notes

Security classification

http://www.control.lth.se/publications/