# Efficient non-linear ISI compensation techniques for commercial receivers

Esteban Gabriel Cabanillas

**N° d'ordre** : 2014telb0313

**Sous le sceau de l'Université européenne de Bretagne**

# Télécom Bretagne

**En accréditation conjointe avec l'Ecole Doctorale Sicma**

---

# Efficient non-linear ISI compensation techniques for commercial receivers

---

## Thèse de Doctorat

Mention : Sciences et Technologies de l'Information et de la Communication


Présentée par **Esteban Cabanillas**

Département : Electronique

Laboratoire : Lab-STICC (UMR6285)


Directeur de thèse : Michel Jézéquel

Soutenue le 14 février 2014

**Jury :**

M. Olivier Romain, Professeur, Université de Cergy Pontoise (Rapporteur)
M. Jean-François Hélard, Professeur, INSA Rennes (Rapporteur)
M. Michel Jézéquel, Professeur, TELECOM Bretagne (Directeur de thèse)
M. Cyril Lahuec, Maître de conférences (HDR), TELECOM Bretagne (Examinateur)
M. Didier Lohy, Development Director, NXP Semiconductors France (Examinateur)
M. Pascal Scalart, Professeur, Université de Rennes 1 (Examinateur)

A ma femme Anne-Flore

A mes parents, à mes frères

# Acknowledgments

Je tiens tout d'abord à remercier le directeur de cette thèse, M.Michel Jézéquel, pour m'avoir guidé, encouragé et conseillé.

Mes remerciements vont également à M.Cyril Lahuec, pour m'avoir fait confiance, pour sa gentillesse et sa patience, pour tous ses conseils et toutes les heures passées à corriger chaque publication et ce mémoire de thèse. Il est difficile de trouver les mots pour lui exprimer ma reconnaissance et mon immense gratitude.

Je remercie M. Didier Lohy, mon tuteur chez NXP Semiconductors, pour m'avoir accueilli au sein de cette entreprise, pour ses conseils stimulants et pour chaque discussion qu'on a pu avoir pendant ces trois années.

Je remercie énormément M. Carol Hervé, pour toutes les connaissances transmises et pour son aide pour implanter le travail présenté dans cette thèse. Je remercie aussi M. Didier Coulibally et M. Philippe Durieux pour tous leurs conseils concernant le design numérique. Je ne veux pas oublier non plus M. Didier Greignic pour avoir fait cette démarche de collaboration et m'avoir reçu au sein de son équipe.

Je passe ensuite une dédicace spéciale à tous les jeunes gens qui m'ont accompagné pendant ces trois années au sein de la société NXP semiconductors: Sylvain, Amandine, Jean-Marie et Nico. Je remercie aussi tous mes collègues un peu moins jeunes mais aussi importants que les autres: Olivier, Xavier, Markus et Denis.

# Abstract

The aim of this Thesis is to study how the impairments induced by the Power Amplifier (PA) non-linearity over digital communication systems can be mitigated. These impairments are the spectral spreading and the non-linear inter-symbol interference (ISI). Both adverse effects can be easily compensated at the transmitter side by implementing pre-distortion techniques. The implementation of pre-distorters is not a mandatory characteristic in transmitters. Hence, it is convenient to compensate for the non-linear inter-symbol interference at receiver. This technique is known as post-distortion. The most effective post-distortion technique introduced in the literature is the Volterra Canceller, which estimates and eliminates the non-linear ISI from the received symbols. An optimal mitigation of the non-linear ISI is reached when the Volterra Canceller works with error-free symbols.

This work proposes an original post-distortion technique based on the linearization of the transmitter PA at receiver side. The most relevant challenge associated with this technique is the blind estimation of the inverse of the PA transfer function from noisy information. The linearizer is carried out from orthogonal polynomial regression. In terms of ability to mitigate the non-linear ISI and complexity, the most efficient linearizer is obtained from Hermite polynomials. Then, the optimal behaviour of the Volterra Canceller working with error-free symbols can be reached by combining the Hermite polynomial linerizer with an actual Volterra Canceller. This behaviour is verified by computer simulations. These simulations are done from a satellite TV system model based on the Digital Video Broadcast Satellite 2 (DVB-S2) standard, which is the application case of this Thesis. The performance of the receiver compensating for the non-linear ISI is stated in terms of gain of $E_S/N_0$ with respect to the uncompensated case. Simulation results show that the ability to mitigate the non-linear ISI shown by the combined technique is near that of an ideal Volterra Canceller.

A digital implementation for the Hermite polynomial linearizer and the Volterra Canceller is then presented. This implementation is optimised for commercial DVB-S2 receivers. The combined compensation technique and the DVB-S2 receiver are embedded into an FPGA Altera STRATIX IV. This implementation is tested in a Test-Bench emulating a DVB-S2 chain considering a DVB-S2 transmitter, a Gaussian channel noise (AWGN) and the receiver. The results obtained from the Test-Bench are correlated with simulation results. It is shown that the Hermite polynomial linearizer improves in 60% the gain of a stand-alone Volterra Canceller. This gain is achieved with only 40% of additional complexity. Hence, the combined technique is effective in terms of performance and complexity.

# Résumé

L'objectif de cette Thèse est d'étudier différentes méthodes de compensation des dégradations induites par un amplificateur de puissance (AP) non-linéaire sur des systèmes de communications numériques. Il existe deux effets adverses induits par un AP non-linéaire: l'étalement spectral et l'interférence inter-symbole (IIS) non-linéaire. Ces dégradations peuvent être corrigées au niveau du transmetteur avec une méthode de pré-distorsion. Malgré son efficacité, la pré-distorsion n'est pas implantée dans tous les APs. Il est par conséquent important de compenser l'IIS non-linéaire au niveau du récepteur. Cette technique est nommée post-distorsion. A ce jour, la méthode de post-distorsion la plus efficace est le Volterra Canceller. Une telle technique estime et élimine l'IIS non-linéaire associée aux symboles reçus. Une compensation optimale peut être faite à partir d'un Volterra Canceller travaillant avec des symboles idéaux, ce qui n'est en pratique jamais possible.

Cette Thèse propose une méthode innovante de post-distorsion plus efficace que le Volterra Canceller. Elle est basée sur la linéarisation du canal non-linéaire. La principale difficulté liée à l'implantation de cette technique est l'estimation aveugle de la fonction de transfert inverse du canal. De plus, cette estimation est faite à partir de données bruitées, ce qui complique la tâche. La linéarisation est basée sur une régression polynômiale d'Hermite. La compensation optimale caractérisant le Volterra Canceller idéal est atteinte par la combinaison de cette linéarisation polynomiale et d'un Volterra Canceller. Ce comportement a été vérifié par simulations faites à partir du modèle mathématique d'un système de TV numérique satellite Digital Video Broadcasting Satellite 2 (DVB-S2), cas d'application de cette Thèse. Les performances du récepteur qui compensent l'IIS non-linéaire sont établies en termes du gain de $E_S/N_0$ par rapport à un récepteur sans compensation.

L'architecture est optimisée pour un récepteur commercial DVB-S2. La méthode proposée a été implantée sur FPGA dans un récepteur DVB-S2 et testée. Le récepteur a ensuite été inclus dans une chaîne de transmission émulant un système DVB-S2. Les résultats obtenus sont corrélés avec les résultats des simulations. Par rapport à un Volterra Canceller seul, la méthode proposée améliore le gain de 60% pour une complexité ajoutée de 40%. Ceci confirme l'efficacité de cette compensation en termes de performance et de complexité.

# Contents

# List of Figures

## LIST OF FIGURES

## LIST OF FIGURES

# List of Tables

# 1 Introduction

This Thesis deals with the adverse effects induced by the non-linearity of power amplifiers (PA) working in digital communication systems. In order to achieve high power efficiency, the PA must work near the saturation power level, inducing a strong non-linearity in the system. An example of a digital communication system working with a non-linear PA is a satellite TV system, figure 1.1. On figure 1.1, the terrestrial station transmits via the up-link antenna a digital TV signal. As the power supply in the terrestrial station is not a constraint, the power efficiency of such a transmission is not a critical factor. Such a condition is different in the satellite side, which power supply is limited by the capacity of the solar panels and the accumulators. Thus, the high power amplifier (HPA) of the satellite needs to work near saturation power levels in order to transmit the digital TV signal to terrestrial satellite receivers with maximal power.

There are two impairments associated with the PA non-linearity. The first one is the spectral spreading, which is a transmitter side impairment and can be only mitigated at the transmitter. The second impairment is the non-linear inter-symbol interference (ISI), which degrades the receiver's performance in terms or error-rate for a given $E_S/N_0$. Such an impairment can be compensated at the transmitter or at the receiver sides.

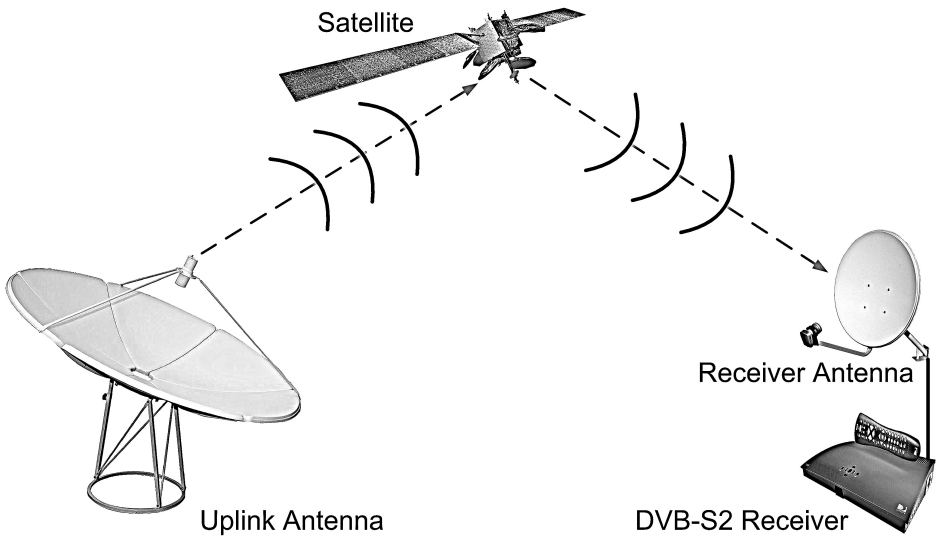



Figure 1.1: Satellite TV transmission system

The target of this Thesis is the correction of the non-linear ISI by means of receiver side compensation techniques also called post-distortion techniques. Compensating the non-linear ISI with post-distortion techniques is not new. A non-linear equalizer used to compensate for the non-linear ISI in digital satellite systems has been firstly introduced by [BB83]. Two other post-distortion techniques outperforming the non-linear equalizer are the Volterra Canceller [Bur06] and the non-linear turbo canceller [BC05]. The non-linear turbo canceller shows better performance than the Volterra Canceller. However, the Volterra Canceller later presents the best trade-off between performance and system Complexity.



This Thesis presents a novel solution outperforming the state-of-the-art in post-distortion techniques. Such a solution is based on the linearization of the PA. The linearization is implemented at the receiver side by means of an orthogonal polynomial regression, and works together with a simplified Volterra Canceller. The polynomial linearizer and the Volterra Canceller have been designed to work with commercial receivers. Thus, the architecture and the digital design of the novel solution are focused on the minimization of the system complexity.

The digital design of the polynomial linearizer and the Volterra Canceller has been used to compensate the non-linear ISI associated with a satellite TV receiver, based on the Digital Video Broadcasting - Satellite 2 (DVB-S2) standard. The digital base-band part of the DVB-S2 receiver and the post-distortion technique have been embedded into an FPGA. The gains observed in the measurements of the FPGA implementation working with the polynomial linearizer and the Volterra Canceller correspond to the gains obtained from computer simulations.

## 1.1 THESIS STRUCTURE

The Thesis is composed of six chapters and one main appendix.

Chapter 2 introduces the different sources of non-linearity existing in digital communication systems. It is shown that due to the required power efficiency, the PA is the most critical source of non-linearity. Thus, chapter 2 presents a few relevant PA definitions and characterizes the two impairments associated with the PA non-linearity. The first impairment is the spectral spreading, which expands the transmission bandwidth. The second impairment is the non-linear inter-symbol interference (ISI), which degrades the performance of the receiver in terms bit-error rate (BER) or symbol-error rate (SER) vs. $E_S/N_0$. Chapter 2 also shows the simulation results of a DVB-S2 system working with a non-linear HPA, confirming the characterization proposed for the PA impairments.

Chapter 3 introduces two families of compensation techniques for the adverse effects of the PA non-linearity. Firstly, a brief introduction of the compensation techniques implemented at the transmitter side, the pre-distortion techniques, is given. Secondly, chapter 3 presents four different post-distortion techniques: the non-linear Zero Forcing

Equalizer, the Volterra Canceller, the turbo canceller and finally the Neural Network based post-distortion techniques. Chapter 3 also introduces three adaptation algorithms needed to train the post-distortion techniques. These algorithms are the least-squares, the least-mean squares and the recursive least-square algorithms. Then, a comparison of the performance of the post-distortion techniques from the results presented in the literature is done. Among all the post-distortion techniques introduced, the Volterra Canceller shows better trade-off between system complexity and performance. Finally, a brief analysis of the possibilities of implementation of each post-distortion technique into commercial DVB-S2 receivers is given.

Chapter 4 presents the novel technique proposed in this work based on the receiver side linearization of the non-linear channel to avoid the undesirable non-linear ISI. This linearization is implemented from Laguerre or Hermite polynomial regressions. The linearizer showing lower system complexity is the one carried out from Hermite polynomials. This chapter also proposes a non-linear ISI compensation technique which completes a polynomial linearizer with a Volterra Canceller to improve the effectiveness of the compensation of the non-linear ISI. The performance achieved for the combined technique is near the performance of an ideal Volterra Canceller. Then, this chapter proposes two different methods to generate the training data necessary to carry out the adaptation of the polynomial linearizer and the Volterra Canceller. Finally, chapter 4 shows the simulation results of a DVB-S2 system working with a non-linear HPA and including a few of the post-distortion techniques introduced in this Thesis. The results are assessed from a novel method based on the average number of LDPC iterations needed to decode with no error the transmitted information. This method helps saving simulation time. Two conclusions are drawn from the computer simulation results. The first one is that the proposed solution based on the combination of a polynomial linearizer and a Volterra Canceller outperforms the state-of-the-art. The second one is that the polynomial linearizer implemented from Hermite polynomials is more efficient in terms of performance and system complexity than the one carried out from Laguerre polynomials.

An innovative digital design for the Volterra Canceller and the Hermite polynomial linearizer is presented in chapter 5. This digital design is focused on the minimization of the complexity of this compensation technique. The Volterra Canceller is designed to be optimal for the 8-PSK digital modulation. A smart combination of look-up tables avoids the complexity of digital multipliers without performance degradation. In addition, the Volterra Canceller is trained from a least-mean squares algorithm which has been simplified to avoid unnecessary operations. The design of the Hermite polynomial linearizer also avoids complex multipliers by means of simplified solutions based on bit-shifting and approximations. The training system implemented is the same as that of the Volterra Canceller.

Chapter 6 presents the measurement results obtained from the FPGA implementation of the Volterra Canceller and the Hermite polynomial linearizer proposed in chapter 5. The compensation system works with a DVB-S2 receiver. The capacity to compensate for the non-linear ISI of the Volterra Canceller and the Hermite polynomial linearizer is assessed from the innovative method based on the average number of LDPC iterations proposed in

chapter 4. The measurement results confirm the simulation results presented in chapter 4.

Chapter 7 summarizes this Thesis and introduces some perspectives for this work.

## 1.2 Overview of the Thesis contributions

The following list summarizes the major contributions of this Thesis to the state-of-the-art:

- Post-distortion technique based on Hermite and Laguerre polynomial linearization
- Combination of a polynomial linearizer with a Volterra Canceller to outperform the state-of-the-art on post-distortion techniques
- Self training data generation system
- Performance analysis based on the average number of LDPC iterations with respect to the $E_S/N_0$
- Innovative digital design for a Volterra Canceller combined with a Hermite polynomial linearizer focused on the minimization of the system complexity

The following list summarizes the methodology considered to carry out this Thesis:

- Mathematical model for the spectral spreading and the non-linear ISI induced by a non-linear PA into digital communication systems
- State-of-the-art on post-distortion techniques
- Matlab and System-C models of a whole a DVB-S2 system including the non-linear HPA
- FPGA implementation of the Volterra Canceller and the Hermite polynomial linearizer integrated into a DVB-S2 receiver
- FPGA implementation of a block approximating a non-linear HPA

## 1.3 List of related publications

- E. Cabanillas, D. Lohy, C. Lahuec, M. Jézéquel,
"Compensating the High Power Amplifier Nonlinearity for a DVB-S2 System,"
IEEE NEWCAS 2012, Montréal, Canada, June 2012.
- E. Cabanillas, D. Lohy, C. Lahuec, M. Jézéquel,
"Efficient NISI Compensation Technique for Low-Cost Satellite Video Receivers,"
IEEE ICC 2013, Budapest, Hungary, June 2013.

# 2

# Non-linearity in a digital communication system

This chapter deals with the sources of non-linearity characterizing digital communication systems and the associated adverse effects. After introducing a typical digital communication system model, this chapter presents the most relevant sources of non-linearity. Those sources are the analogue blocks placed at transmitter or at receiver sides. Thus the non-linearity associated with the analogue-to-digital and digital-to-analogue converters, the mixers, the analogue filters, the low-noise amplifier and the power amplifier (PA) is analyzed. As the power efficiency is a critical factor for PAs, this block needs to work near saturation zone, i.e. the PA is the most significant source of non-linearity in digital communication systems. Then, the rest of this chapter describes and characterizes the most relevant PA generalities and definitions, and introduces the two impairments induced by the PA non-linearity: the spectral spreading and the non-linear inter-symbol interference.

The organisation of this chapter is the following. Section 2.1 describes a typical digital communication system. Section 2.2 presents the different sources of non-linearities existing in digital communication systems. Section 2.3 defines the most relevant aspects characterizing non-linear PAs. Section 2.4 deals with the two adverse effects induced by the PA non-linearity: the spectral spreading and the non-linear inter-symbol interference. In Section 2.5, the impact of the adverse effects is analysed in the context of an application case (a satellite video receiver). Section 2.6 summarizes this chapter.

## 2.1 GENERALITIES OF DIGITAL COMMUNICATION SYSTEMS

The simplified base-band model for typical digital communication systems is illustrated on figure 2.1. The channel encoder processes the information bits $a_i$. The encoded bits $c_j$ are digitally modulated onto constellation points $s_n$. Then, up-sampling and shaping by a root-raised cosine (RRC) digital matched filter take place. The filter output signal $x_t$ is defined as:

$$x_t = \sum_{j=-N}^{N} r_j s_{n+j}. \tag{2.1}$$

Where $N$ represents half the length of the RRC filter coefficients vector and $r_j$ is the $j^{th}$ RRC filter coefficient. The value of each coefficient is characterized by the roll-off factor associated with the filter, which determines the bandwidth of the filter output signal (section A.6). After amplification and transmission over an Additive White Gaussian Noise (AWGN) channel, the received signal $x''_t$ is defined as

$$x''_t = TXM(x_t) + n_t = x'_t + n_t. \tag{2.2}$$

Where function $TXM(\cdot)$ represents the transfer function of the mismatches induced by the transmitter's analogue components. Signal $n_t$ is the Gaussian channel noise (AWGN) at instant $t$. Then, signal $y_t$ results from:

$$y_t = RXM(x''_t). \tag{2.3}$$

Where function $RXM(\cdot)$ represents the transfer function of the mismatches induced by the receiver's analogue components. The received symbol $s'_n$ is the result of applying

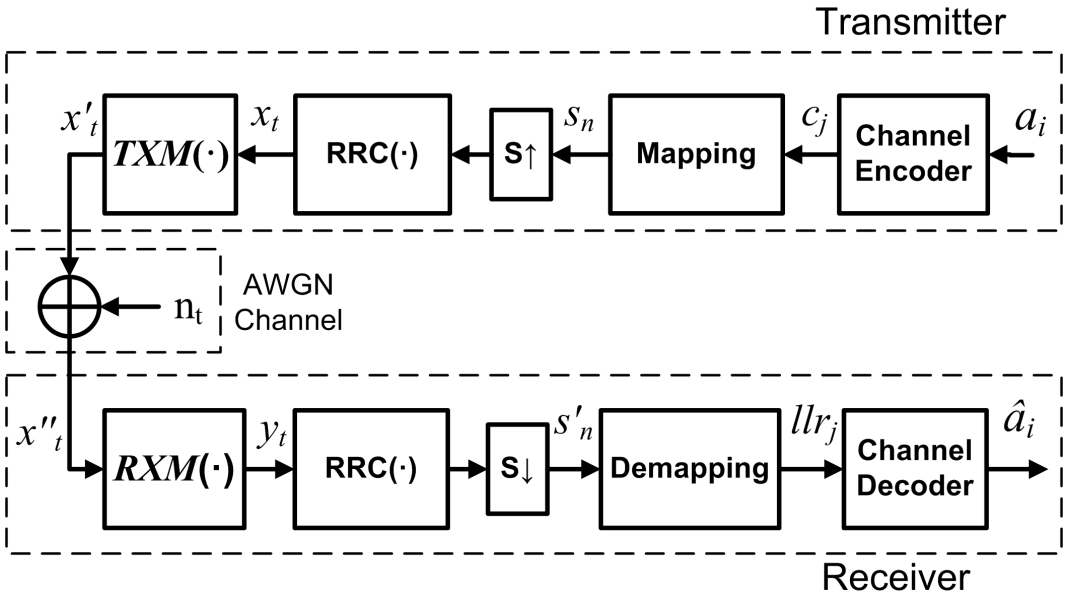



Figure 2.1: Typical base-band block diagram for digital communication systems

digital matched filtering to $y_t$ and further decimation. Assuming perfect synchronization and non-multi-path channel, the received symbol is

$$s'_n = \sum_{j=-N}^{N} r_j y_{t+j}. \tag{2.4}$$

The received symbols is soft-decision demapped, producing the log-likelihood ratios ($llr_j$) used to decode the received $\hat{a}_i$ bits.

As demonstrated in [FB08], a band-limited free inter-symbol interference (ISI) transmission is possible when digital matched filters are implemented at transmitter and receiver sides. The combination of such filters is characterized by having a constant transfer function for any frequency in the transmission band, which is the necessary condition to avoid ISI. Any perturbation induced by the analogue components of the transmitter or the receiver leads to a frequency selective channel. In other words, the combination of the two digital matched filter transfer functions and the perturbation transfer function is not a constant value in the frequency domain. Therefore, the perturbation induces ISI in the received symbols. The ISI is characterized as the second term of:

$$s'_n = \alpha_0 s_n + \sum_{j=-M_{ISI}, j\neq 0}^{M_{ISI}} \alpha_j s_{n+j}. \tag{2.5}$$

Where $\alpha_j$ is the weight associated with the interferer symbol with delay $j^{th}$, and $M_{ISI}$ is half the value of the memory of the ISI. Such a memory represents the maximal delay of the interferer symbols.

## 2.2 Sources of non-linearity in digital communication systems

The modern standards related to digital communication are characterized on the one hand by the power requirements, and on the other hand by the increasing demand for high data rates.

The power requirements respond to the limited autonomy of the batteries in portable devices, and the policies of reduction of power consumption associated with the development of green sustainable technologies.

As the RF spectrum is limited, the high data rate must be accomplished by using efficiently the given bandwidth associated with the target digital communication system. Indeed, the transmission must be done with a high spectral efficiency. Such an efficiency is achieved with the implementation of high digital modulation orders (8-PSK, 16-QAM/APSK, 32-QAM/APSK, etc). The tolerated ratio between the signal power and the noise power increases with the digital modulation order, [Smi98]. Then, the trade-off between the high data rate demanded and the low power consumption leads to the necessity of high power efficiency.

The power efficiency is a metric indicating how much of the power consumed by an electronic component is used in the output signal power. As presented in [BM10], the power efficiency $\eta$ is computed as:

$$\eta = \frac{P_o}{P_i + P_{DC}}. \tag{2.6}$$

Where $P_i$ is the input signal power, $P_{DC}$ is the DC power supply, and $P_o$ is the power of the output signal. A linear transmission is achieved when $P_{DC}$ is higher than $P_o$, which is associated with a low efficiency. Therefore, in order to reach a high value of $\eta$, most of the power supply must be used to amplify the output signal, indeed, the component must be driven into saturation, inducing a non-linear behaviour.



The analysis of power efficiency and non-linearity can be extended to each analogue component characterizing a standard digital communication system. Those components are introduced on figure 2.2, which illustrates a band-pass model of a digital communication system with the most relevant analogue components.

On figure 2.2, the transmitter is composed of a Digital Modulator, which encodes and maps the information bits $a_i$ onto constellation points and then applies digital matched filtering. The resulting digital signal is transformed into an analogue signal by means of a digital-to-analogue converter (DAC). After filtering, the analogue signal is translated in frequency by mixing it with an RF carrier signal. The carrier frequency is generated by the local oscillator (LO). The power amplifier (PA), which include the band filter, gives the signal the necessary power to reach the receiver after passing through the channel. At the receiver side, the received signal is amplified by a low-noise amplifier (LNA) and then translated in frequency by means of a mixer. The signal is then filtered and converted to the digital domain by means of an analogue-to-digital converter (ADC). The received information bits $\hat{a}_i$ are finally obtained from the Digital Demodulator.

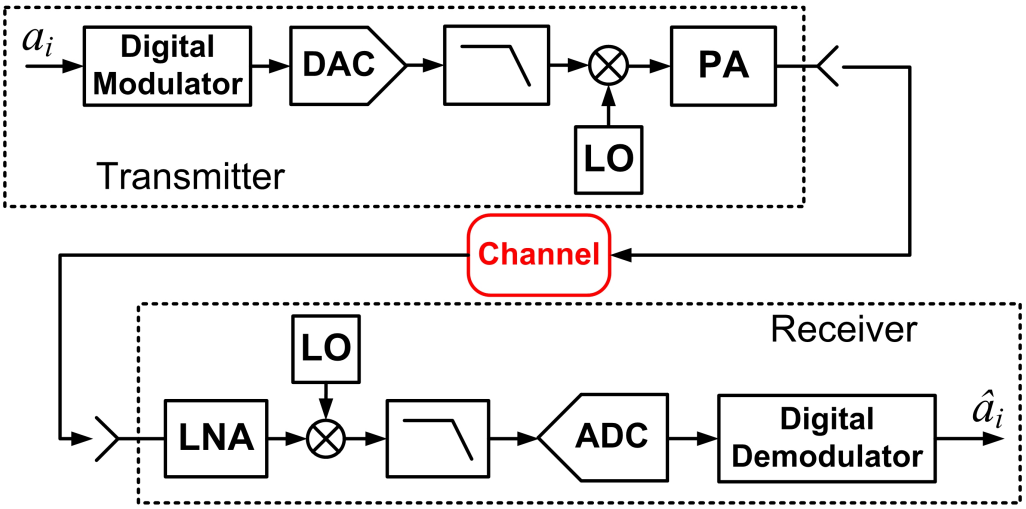



Figure 2.2: Most relevant analogue blocks of a typical digital communication system

Excepting the Digital Modulator and the Digital Demodulator, all the blocks constituting the transmitter and the receiver are potential sources of non-linearity. The non-linearity associated with each analogue block, i.e. the LNA, the mixers, the low-pass filters and finally the PA, is determined by the efficiency desired for each block.

Compared to conventional mixers, LNAs or analogue filters, which output powers are typically few tens of milliwatts, the output power driven from the PA attains few watts or kilowatts, depending on the application case. Therefore, the power efficiency of this block practically determines the whole power efficiency of the digital communication system. Then, the saturation level of the PA needed to reach such an efficiency is the key source of non-linearity. In other words, in order to reach an optimal power efficiency, the limited amount of non-linearity tolerated for the entire system is beforehand induced by the PA.

The ADC and the DAC are the other sources of non-linearity in digital communication systems. As presented in [RVR+10], [FNCS04] and [SR12], the ADC and the DAC are characterized by two kinds of non-linearities, the integral non-linearity (INL) and the differential non-linearity (DNL). The INL describes the deviation of the converter's output signal with respect to the ideal transfer curve, i.e. a straight-line of slope +1. The DNL of an ADC is defined as the difference between any step width and the LSB value. Finally, the DNL of a DAC is defined as the difference between any step height and the LSB value. As the degradation of the INL and the DNL is determined by the LSB value characterizing the converter, it can be controlled by setting an optimal number of bits in the converter.

Consequently, this work assumes that the non-linearity characterizing the communication system is solely induced by the PA at the transmitter side. The remaining of this chapter characterizes the non-linear PA and analyses the adverses effects induced by this non-linear component.

## 2.3 Characterizing the PA non-linearity

As discussed in section 2.2, the PA is the most important source of non-linearity in digital communication systems. Therefore, before introducing the adverse effects of the non-linearity, this section presents the most significant concepts, definitions and models associated with the PA non-linearity.

### 2.3.1 Amplifiers definitions

The power of either the input or output PA's signals is characterized by four metrics. With respect to the input signal power $P_{in}$, the input saturation power is defined as:

$$P_{in,sat} = max(P_{in}). \tag{2.7}$$

The input average power is obtained from:

$$\bar{P}_{in} = E|P_{in}|^2. \tag{2.8}$$

Where $E|\cdot|$ represents the function computing the average value. The same metrics can be obtained from the output signal power $P_{out}$. Thus, the output saturation power is computed as:

$$P_{out,sat} = max(P_{out}). \tag{2.9}$$

And the output average power is defined as:

$$\bar{P}_{out} = E|P_{out}|^2. \tag{2.10}$$

### 2.3.2 Approximating the amplitude and phase transfer functions of a PA



The PA transfer function is approximated by means of the amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) curves. The AM/AM curve describes how the instantaneous output signal power changes with respect to the variations of the instantaneous input signal power. Figure 2.3 shows the typical AM/AM transfer function modelling for satellite high power amplifiers (HPA).

On figure 2.3, both the input power $\hat{P}_{in}$ and the output power $\hat{P}_{out}$ are the instantaneous input and output powers normalised with respect to the power saturation values, i.e. $P_{in,sat}$ and $P_{out,sat}$ respectively.

The AM/PM curve describes the variations of the instantaneous phase of the HPA output signal with respect to the variations of the instantaneous input signal power. In order to illustrate such a transfer function, figure 2.4 shows the AM/PM curve characterizing satellite HPAs.

The AM/AM and AM/PM characteristics can be also represented using complex notation. In this case, those functions are not expressed in terms of input and output

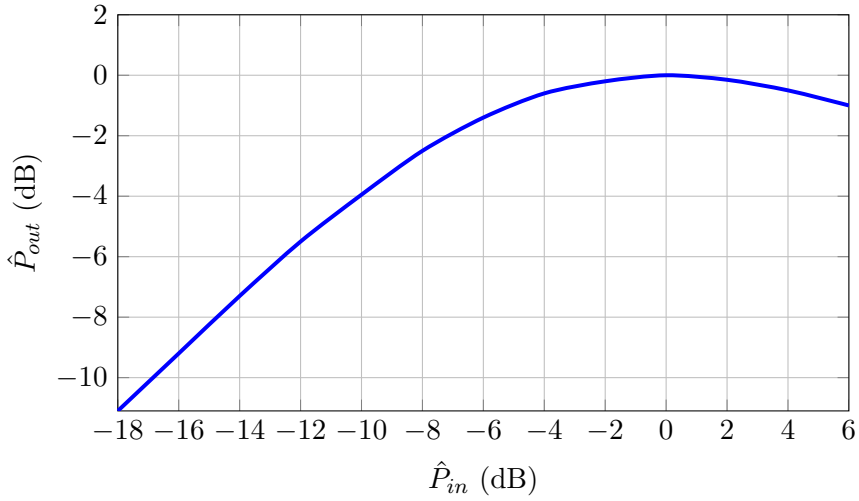



Figure 2.3: AM/AM curve for Satellite HPAs

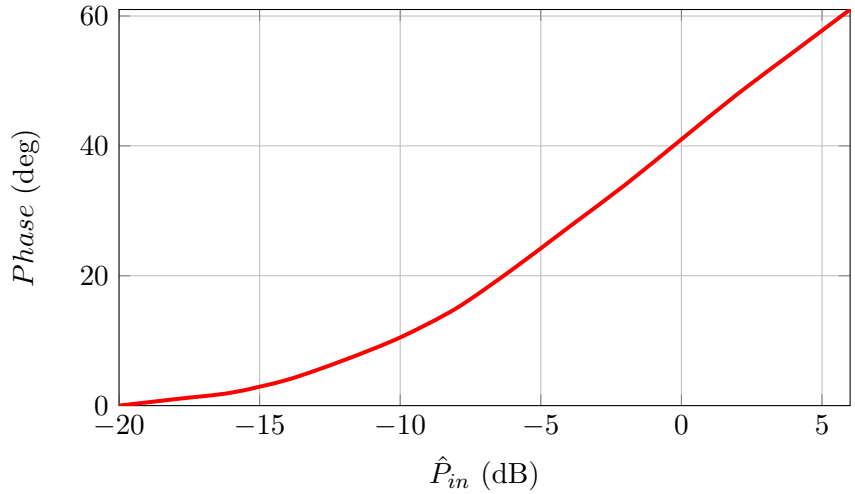



Figure 2.4: AM/PM curve for Satellite HPAs



powers but in terms of magnitudes and phases. Therefore, the input signal $x_t$ is expressed as:

$$x_t = |a_t|e^{j\phi_t}. \tag{2.11}$$

Where $|a_t|$ is the absolute value of the instantaneous magnitude of $x_t$ and $\phi_t$ is its instantaneous phase. Then, the PA output is a function of the input magnitude $a_t$ and the input phase $\phi_t$ computed as:

$$x'_t = Am(|a_t|)e^{j(\phi_t + Ph(|a_t|))}. \tag{2.12}$$

Where $Am(\cdot)$ and $Ph(\cdot)$ are functions modelling the AM/AM and AM/PM characteristics. The mathematical model expressed as (2.12) is shown as a block diagram on figure 2.5.

PAs can be classified in two main families: the solid-state power amplifiers (SSPA) and the traveling wave tube amplifiers (TWTA), [Str94]. The general model for any family of

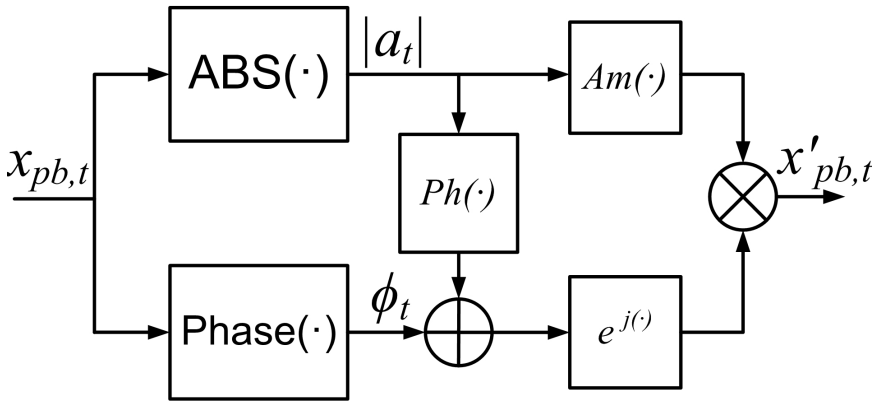



Figure 2.5: Mathematical model for non-linear PAs

non-linear PA is the polynomial model. As presented in [SL08], both the AM/AM and the AM/PM characteristics can be modelled as a $N-1$ order polynomial as follows:

$$x'_t = x_t \sum_{n=0}^{N-1} \zeta_{n+1}|a_t|^n = e^{j\phi_t} \sum_{n=0}^{N-1} \zeta_{n+1}|a_t|^{n+1}. \tag{2.13}$$

Where $\zeta_i$ is a complex coefficient used to fit the polynomial regression to the actual AM/AM and AM/PM characteristics.

The literature also presents different PA models used to approximate specifics families of PA. The AM/AM and AM/PM characteristics for the TWTA have been modelled by [Sal81]. For such a model, the AM/AM characteristic is approximated as:

$$Am_S(|a_t|) = \frac{g_0|a_t|}{(1+g_1|a_t|^2)}. \tag{2.14}$$

The AM/PM is approximated as:

$$Ph_S(|a_t|) = \frac{p_0|a_t|^2}{(1+p_1|a_t|^2)}. \tag{2.15}$$

Equations (2.14) and (2.15) are known as the Saleh's model. The parameters $g_0, g_1, p_0$ and $p_1$ are obtained from TWTA measured points and curve fitting methods.

In the case of a SSPA, the generalised method to model its AM/AM characteristic has been introduced by [Rap91]. Such a model is named Rapp's model and is defined as:

$$Am_R(|a_t|) = \frac{\nu|a_t|}{(1+(\frac{\nu|a_t|}{|a_{sat}|})^{2\rho})^{\frac{1}{2\rho}}}. \tag{2.16}$$

Where $|a_{sat}|$ is the saturation level, $\nu$ is the amplifier's gain and $\rho$ is a curve-fitting parameter. Due to the very small phase distortion presented by the SSPAs, the AM/PM conversion is neglected by the Rapp's model, [FJBQ10].

### 2.3.3 The PA back-off

Another important parameter characterizing PAs is the back-off. The back-off represents the difference in dB between the power saturation point and the average power point. Therefore, the back-off can be referenced to either the input power, leading to the input back-off (IBO), or to the output power, defining the output back-off (OBO). As presented in [SG08], the IBO and the OBO are computed as:

$$\text{IBO} = 10\log_{10}\frac{P_{in,sat}}{\bar{P}_{in}}. \tag{2.17}$$

$$\text{OBO} = 10\log_{10}\frac{P_{out,sat}}{\bar{P}_{out}}. \tag{2.18}$$

The values of IBO and OBO can be also defined with respect to the normalized values $\hat{P}_{in}$ and $\hat{P}_{out}$, as illustrated on figure 2.6. As seen on this figure, the back-off is an indicator of

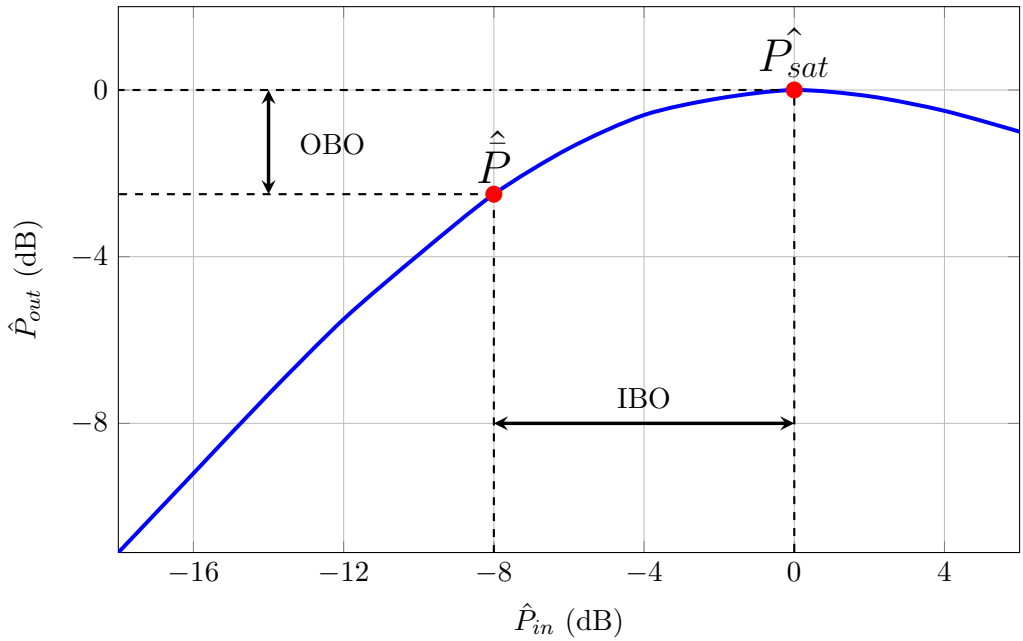



Figure 2.6: Representation of the back-off from the AM/AM curve for a Satellite HPA

the non-linearity induced by a HPA. A low value of back-off (IBO or OBO near 0 dB) leads to a HPA working in the saturation zone, inducing a strong non-linearity. In contrast, a big value of IBO or OBO guaranties a linear HPA behaviour.

The back-off is also an indicator of the efficiency of a HPA. As stated in (2.6), the maximal value of $\eta$ is obtained when the output power is the saturation power. Thus, high values of $\eta$ are associated with low values of back-off and with strong non-linearities. In contrast, working in linear regions, i.e. high values of back-off, degrades the efficiency of the HPA.

## 2.4 Adverse effects of PA non-linearity in digital communication systems

This section deals with the adverse effects induced by the PA non-linearity into digital communication systems. When a PA is driven to low values of back-off, two effects degrading the performance of the system appear: the spectral spreading [BWM+10] and the non-linear inter-symbol interference (ISI) [LA08]. The next subsections introduce such non-linear issues.

### 2.4.1 The spectral spreading

The spectral spreading is a transmitter side impairment associated with the saturation of the PA. To describe this phenomena, it is convenient to define the input-output PA pass-band signals. Such signals are illustrated on figure 2.7. On figure 2.7, the signal $x_t$

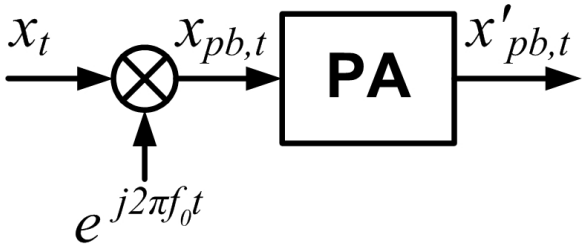



Figure 2.7: Pass-band signals for the PA

is the complex transmitted base-band signal defined as:

$$x_t = x_{I,t} + jx_{Q,t}. \tag{2.19}$$

Where $x_{I,t}$ and $x_{Q,t}$ are the real and the imaginary parts of $x_t$. The Fourier transform of $x_t$ is defined as $X_f$ and it bandwidth is $\Delta B$. Before being amplified by the PA, the base-band signal is translated in frequency by a mixer. Such a translation is done by taking the real part of the product between $x_t$ and the carrier signal of frequency $f_0$. Thus, the pass-band signal $x_{pb,t}$ is obtained from:

$$x_{pb,t} = \Re(e^{j2\pi f_0 t} x_t) = \frac{1}{2}(x_t e^{j2\pi f_0 t} + x_t^* e^{-j2\pi f_0 t}). \tag{2.20}$$

The spectral analysis of the input-output PA signal is only done for positive frequency values. All the concepts introduced can be applied to negative frequency values. After the frequency translation, the spectrum of $x_{pb,t}$ is the same spectrum as $x_t$ but shifted from 0 to frequency $f_0$. Such a spectrum is shown on figure 2.8.

Signal $x_{pb,t}$ is the input of the PA. Considering the example of an order five PA polynomial

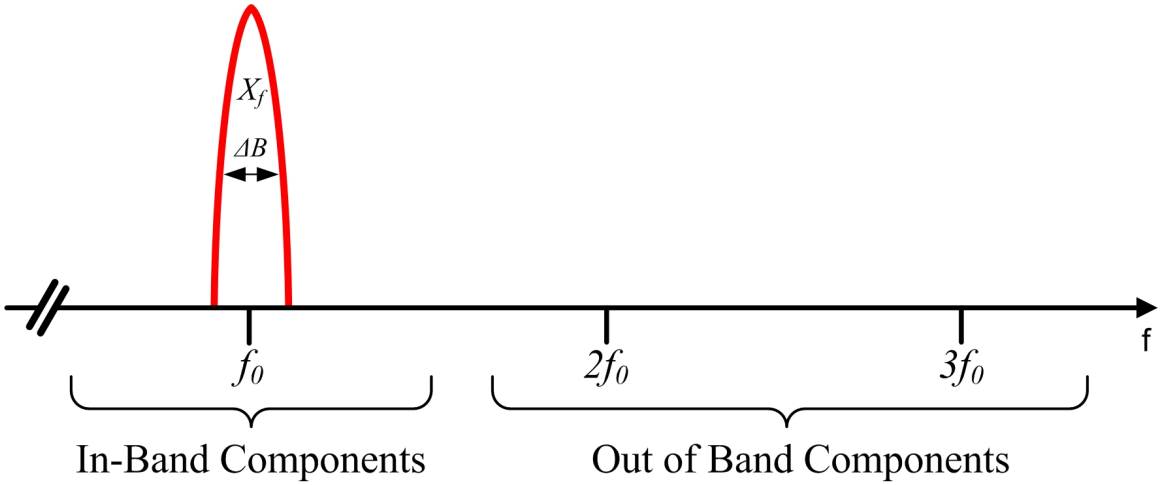



Figure 2.8: PA input spectrum for positive frequencies

Table 2.1: Multidimentional combinations of $x_t$ and $x_t^*$ associated with frequencies from $f_0$ to $5f_0$

| Frequency | Combinations |
|---|---|
| $f_0$ | $x_t$ |
| | $x_t x_t x_t^*$ |
| | $x_t x_t x_t x_t^* x_t^*$ |
| $2f_0$ | $x_t x_t$ |
| | $x_t x_t x_t x_t^*$ |
| $3f_0$ | $x_t x_t x_t$ |
| | $x_t x_t x_t x_t x_t^*$ |
| $4f_0$ | $x_t x_t x_t x_t$ |
| $5f_0$ | $x_t x_t x_t x_t x_t$ |



model, the amplifier output $x'_{pb,t}$ is computed as:

$$x'_{pb,t} = \zeta_0 + \zeta_1 x_{pb,t} + \zeta_2 x_{pb,t}^2 + \zeta_3 x_{pb,t}^3 + \zeta_4 x_{pb,t}^4 + \zeta_5 x_{pb,t}^5. \tag{2.21}$$

As $x_{pb,t}$ is defined as half the value of $(x_t e^{j2\pi f_0 t} + x_t^* e^{-j2\pi f_0 t})$, the different powers of $x_{pb,t}$ result in multidimensional combinations of $x_t$ and $x_t^*$ at frequencies multiples of $f_0$ taking values from $-5f_0$ to $5f_0$. All the multidimensional combinations for the positives multiples of $f_0$ are shown in table 2.1. As observed in this table, the multidimensional combinations of signal $x_t$ and $x_t^*$ leads to spectral spreading since a multiplication in the time domain results in a convolution in the frequency domain. Then, the bandwidth associated with each combination is determined as a linear function of the combination order. As an example, figure 2.9 illustrates the result of the order two convolution of the Fourier transform of $x_t$. As shown on figure 2.9, the resulting spectrum is twice as large as the original spectrum.

Consequently, the positive spectrum of the PA output signal $x'_{pb,t}$ is obtained by applying the convolution theorem to every combination shown in table 2.1. The resulting spectrum is depicted on figure 2.10 for frequencies $f_0, 2f_0$ and $3f_0$.

As the spectral components centered on $2f_0$ and $3f_0$ are out of the transmission band, these components must be filtered out by the PA. Therefore, the PA output spectrum surrounds frequency $f_0$, the center frequency of the transmission band. The bandwidth characterizing this component is a linear function of the polynomial modelling the PA.

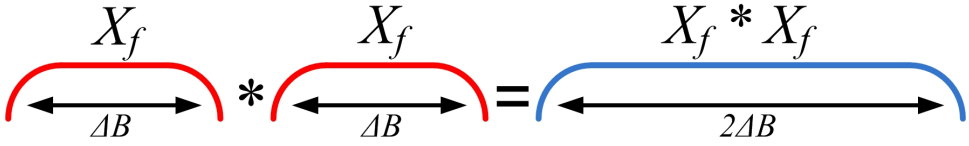



Figure 2.9: Frequency domain convolution between two signals having equal bandwidth

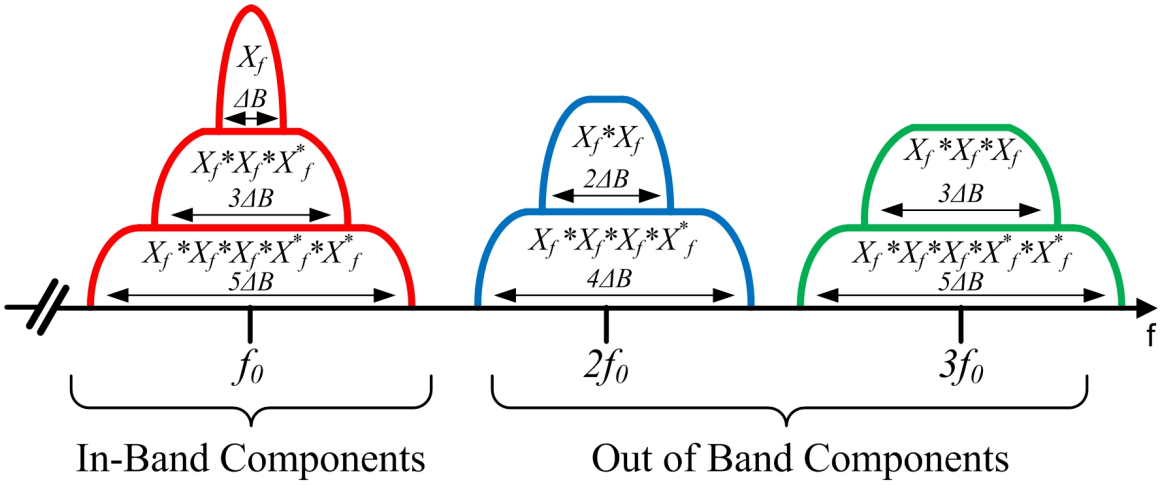



Figure 2.10: PA output spectrum with order five non-linearity

Thus, for a given polynomial model of order $N$ (for $N$ equal to an odd value), the spectrum bandwidth of $x'_{pb,t}$ is $N$ times wider than the spectrum of $x_{pb,t}$. The level of each one of the multidimensional combinations varies as a function of the back-off. As stated in table 2.1, the spectral components associated with frequency $f_0$ are those of order one, three and five. The power level of each component is determined by the values of the polynomial model coefficients $\zeta_i$s of (2.21), which grows inversely with respect to the back-off. Indeed, the lower the back-off is the higher the level of the multidimensional combinations is.

### 2.4.2 The non-linear inter-symbol interference

The characterization of the non-linear ISI induced by the PA non-linearity is done from the simplified base-band model of a satellite digital communication system, figure 2.11. Such a system considers a non-linear HPA, which typically works with values of back-off near 0 dB.

As introduced in section 2.1, any perturbation induced by the analogue components placed between the transmitter and the receiver digital matched filters induces the ISI. In the case of a HPA working with a low value of back-off (saturation zone), the ISI becomes non-linear. In order to characterize the non-linear ISI, the HPA non-linear transfer function is approximated with the polynomial model given by (2.13). To avoid high complexity in the computation of the non-linear ISI, the order of the polynomial model is three. Then, the output of the HPA for the base-band model is:

$$x'_t = \zeta_1 x_t + \zeta_3 x_t x_t x_t^*. \tag{2.22}$$

As the order two combinations and the combinations of order three different to $x_t x_t x_t^*$ are out of the transmission band (see section 2.4.1), these combinations are not taken into account in (2.22). Then, the HPA output signal is rewritten as the combination of the

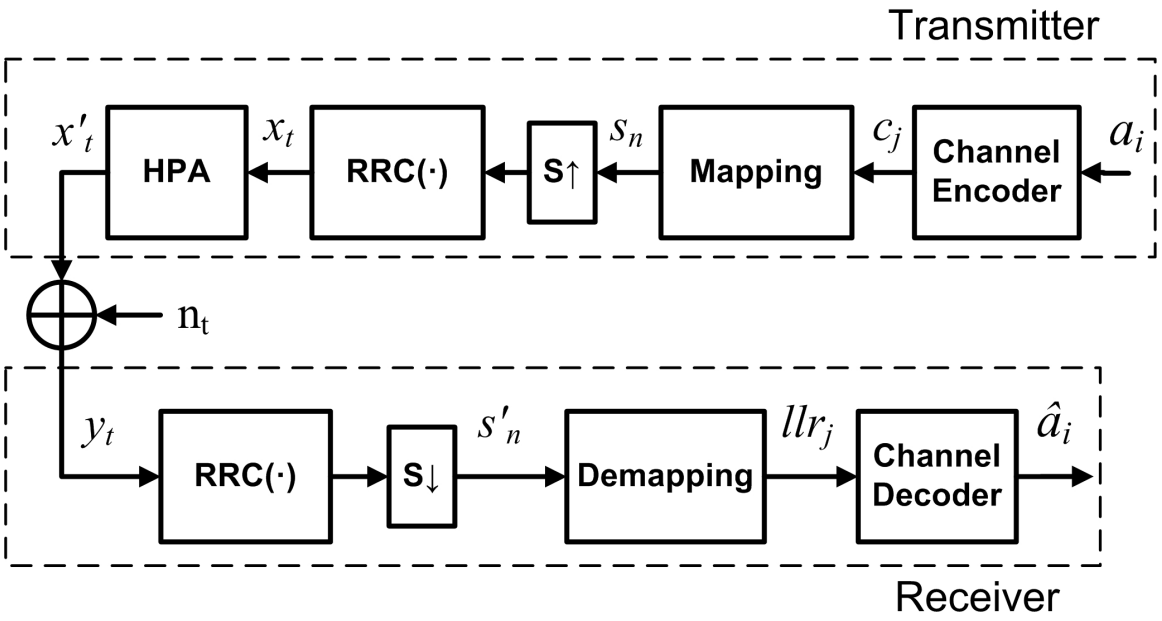



Figure 2.11: Satellite digital communication systems with non-linear HPA

transmitter side RRC filter and the HPA model (combination of (2.22) and (2.1)):

$$x'_t = \zeta_1 \sum_{j=-N}^{N} r_j s_{n+j} + \zeta_3 \sum_{i=-N}^{N} r_i s_{n+i} \sum_{j=-N}^{N} r_j s_{n+j} \sum_{k=-N}^{N} r_k^* s_{n+k}^*. \tag{2.23}$$

Then, the second term of (2.23) is arranged as:

$$x'_t = \sum_{j=-N}^{N} \zeta_1 r_j s_{n+j} + \sum_{i=-N}^{N} \sum_{j=-N}^{N} \sum_{k=-N}^{N} \zeta_3 r_i r_j r_k^* s_{n+i} s_{n+j} s_{n+k}^*. \tag{2.24}$$

Finally, (2.24) is simplified as follows:

$$x'_t = \sum_{j=-N}^{N} h_j s_{n+j} + \sum_{i=-N}^{N} \sum_{j=-N}^{N} \sum_{k=-N}^{N} h_{ijk} s_{n+i} s_{n+j} s_{n+k}^*. \tag{2.25}$$

Where

$$h_j = \zeta_1 r_j, \tag{2.26}$$

$$h_{ijk} = \zeta_3 r_i r_j r_k^*. \tag{2.27}$$

After the transmission of $x'_t$ through the Gaussian channel noise (AWGN), the received signal $y_t$ is expressed as:

$$y_t = x'_t + n_t = \sum_{j=-N}^{N} h_j s_{n+j} + \sum_{i=-N}^{N} \sum_{j=-N}^{N} \sum_{k=-N}^{N} h_{ijk} s_{n+i} s_{n+j} s_{n+k}^* + n_t. \tag{2.28}$$

The received symbol $s'_n$ is obtained by applying RRC filtering to the received signal. Therefore, $s'_n$ is computed as:

$$s'_n = \sum_{\varrho=-N}^{N} r_\varrho y_{t+\varrho}. \tag{2.29}$$

Replacing $y_t$ by (2.28) leads to:

$$s'_n = \sum_{\varrho=-N}^{N} r_\varrho (x'_{t+\varrho} + n_{t+\varrho}). \tag{2.30}$$

$$s'_n = \sum_{\varrho=-N}^{N} r_\varrho x'_{t+\varrho} + \sum_{\varrho=-N}^{N} r_\varrho n_{t+\varrho}. \tag{2.31}$$



The value of $n_{t+\varrho}$ is independent of $n_{t+\varrho+i}$ for any value of $i$ different than 0. Then, as $n_t$ is a random value, the linear combination of the second term of the right side of (2.31) results in a new random value, defined as the received noise at instant $n$, $nn_n$. Recombining (2.31) with (2.25) leads to:

$$s'_n = \sum_{\varrho=-N}^{N} r_\varrho \Big( \sum_{j=-N}^{N} h_j s_{n+j+\varrho} + \sum_{i=-N}^{N} \sum_{j=-N}^{N} \sum_{k=-N}^{N} h_{ijk} s_{n+i+\varrho} s_{n+j+\varrho} s^*_{n+k+\varrho} \Big) + nn_n. \tag{2.32}$$

The distribution of the first term on the right side results in:

$$s'_n = \sum_{\varrho=-N}^{N} r_\varrho \sum_{j=-N}^{N} h_j s_{n+j+\varrho} + \sum_{\varrho=-N}^{N} r_\varrho \sum_{i=-N}^{N} \sum_{j=-N}^{N} \sum_{k=-N}^{N} h_{ijk} s_{n+i+\varrho} s_{n+j+\varrho} s^*_{n+k+\varrho} + nn_n. \tag{2.33}$$

The sum with index $\varrho$ can be recombined with the others sums. Then, by defining

$$i' = i + \varrho \tag{2.34}$$

$$j' = j + \varrho \tag{2.35}$$

$$k' = k + \varrho \tag{2.36}$$

equation (2.33) can be rewritten as:

$$s'_n = \sum_{j'=-2N}^{2N} \alpha_{j'} s_{n+j'} + \sum_{i'=-2N}^{2N} \sum_{j'=-2N}^{2N} \sum_{k'=-2N}^{2N} \alpha_{i'j'k'} s_{n+i'} s_{n+j'} s^*_{n+k'} + nn_n. \tag{2.37}$$

The ISI and non-linear ISI can be extracted from (2.37) by isolating the term containing $s_n$:

$$s'_n = \alpha_0 s_n + \sum_{j'=-2N, j'\neq 0}^{2N} \alpha_{j'} s_{n+j'} + \sum_{i'=-2N}^{2N} \sum_{j'=-2N}^{2N} \sum_{k'=-2N}^{2N} \alpha_{i'j'k'} s_{n+i'} s_{n+j'} s^*_{n+k'} + nn_n. \tag{2.38}$$

Where $\alpha_0$ is the attenuation of the distortion-free received symbol $s_n$. The second term of (2.38) represents the ISI, which is a linear combination of $s_{n+j}$ for $j \neq 0$. Finally, the third term describes the non-linear ISI, resulting from the linear combination of the order

three products of the transmitted symbols, computed as $s_{n+i'}s_{n+j'}s^*_{n+k'}$.

The ISI and non-linear ISI associated with $s'_n$ are characterized by coefficients $\alpha'_j$ and $\alpha_{i'j'k'}$ respectively. Coefficients $\alpha'_j$ are defined as:

$$\alpha_{j'} = \sum_{j,\varrho/(j+\varrho=j')} h_j r_\varrho = \sum_{j,\varrho/(j+\varrho=j')} \zeta_1 r_j r_\varrho. \tag{2.39}$$

Coefficients $\alpha_{i'j'k'}$ are also defined as a combination of the RRC filter coefficients $r_i$ and the HPA polynomial model parameter $\zeta_3$

$$\alpha_{i'j'k'} = \sum_{i,j,k,\varrho/(i+\varrho=i',j+\varrho=j',k+\varrho=k')} h_{ijk} r_\varrho = \sum_{i,j,k,\varrho/(i+\varrho=i',j+\varrho=j',k+\varrho=k')} \zeta_3 r_i r_j r_k r_\varrho. \tag{2.40}$$

Therefore, both the ISI and the non-linear ISI associated with the received symbol, are determined on the one hand by the RRC filter placed at the transmitter and at the receiver side, and on the other hand by the HPA non-linearity.

Concerning the HPA non-linearity, the model used to characterize the non-linearity was an order three polynomial. Such a polynomial was parametrized by the order one polynomial coefficient $\zeta_1$ and the order three polynomial coefficient $\zeta_3$. As seen in (2.39), $\zeta_1$ is a factor included in the computation of each one of the $\alpha_i$. Therefore, $\zeta_1$ is a scale factor of both the linear ISI and the received free-ISI symbol $s_n$.

As shown in [HB08], the value of $\zeta_3$ increases when the value of back-off associated with the HPA decreases. Then, as the value of each $\alpha_{i',j',k'}$ is a linear function of $\zeta_3$, the non-linear ISI associated with the received symbols is also a linear function of $\zeta_3$.

With respect to the RRC filters, these blocks are characterized by two parameters. The first parameter is the filter memory, determining the filter delay and the memory of the ISI and non-linear ISI. For the model given in (2.1) and in (2.29), the memory considered is $N$. Therefore the filter is implemented by combining $s_n$ with the $N$ symbols transmitted before $s_n$ and the $N$ symbols transmitted after $s_n$. The second parameter is the roll-off factor. As stated before, the roll-off factor, which takes values from zero to one, determines the transmission bandwidth and the magnitude of the ISI and non-linear ISI. A low roll-off leads to a narrow bandwidth, but induces a strong ripple in the time domain filter transfer function. Such a ripple determines the values of the filter coefficients $r_i$. In contrast, when the roll-off is high (value near the unity), the occupied band is large, attenuating the ripple and the values of $r_i$. As examples, figures 2.12 and 2.13 illustrate the absolute value of the coefficients for two RRC filters composed of 13 coefficients with roll-off equal to 0.2 and 0.8 respectively. The absolute values of the coefficients for each filter are detailed in table 2.2.

As seen in table 2.2, for a roll-off equal to 0.8, only the coefficients with memory in the interval [-2,2] are significant, which is not the case for the roll-off equal to 0.2. Therefore, by comparing the examples given on figures 2.12 and 2.13 it can be stated that a low value of roll-off induces higher level with higher memory order of ISI and non-linear ISI.

Table 2.2: RRC coefficients values for roll-off equal to 0.2 and 0.8

| roll-off | $r_{-6}$ | $r_{-5}$ | $r_{-4}$ | $r_{-3}$ | $r_{-2}$ | $r_{-1}$ | $r_0$ | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.2 | 0 | 0.15 | 0.19 | 0 | 0.4 | 0.82 | 1 | 0.82 | 0.4 | 0 | 0.19 | 0.15 | 0 |
| 0.8 | 0 | 0.01 | 0.06 | 0 | 0.31 | 0.77 | 1 | 0.77 | 0.31 | 0 | 0.06 | 0.01 | 0 |

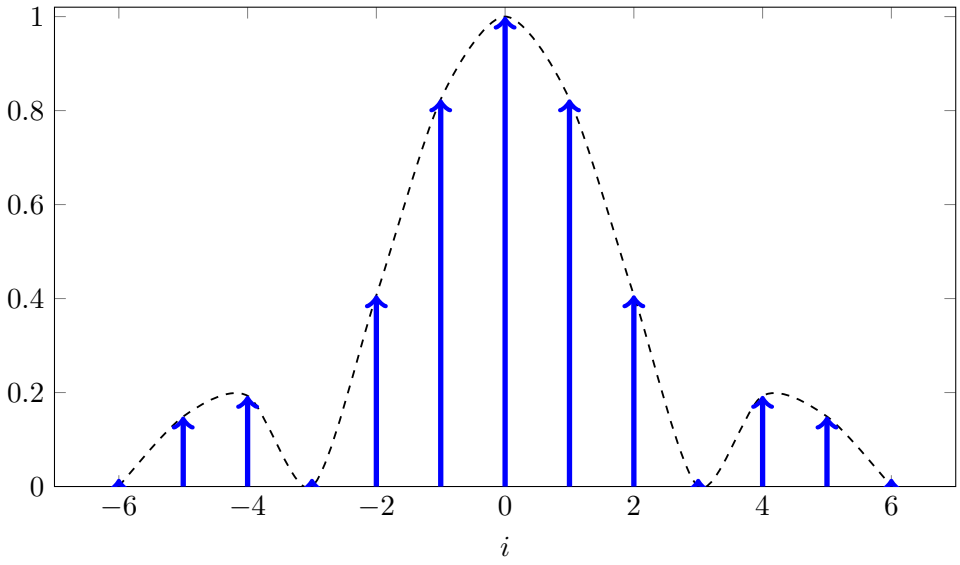



Figure 2.12: absolute value of the RRC filter with roll-off=0.2

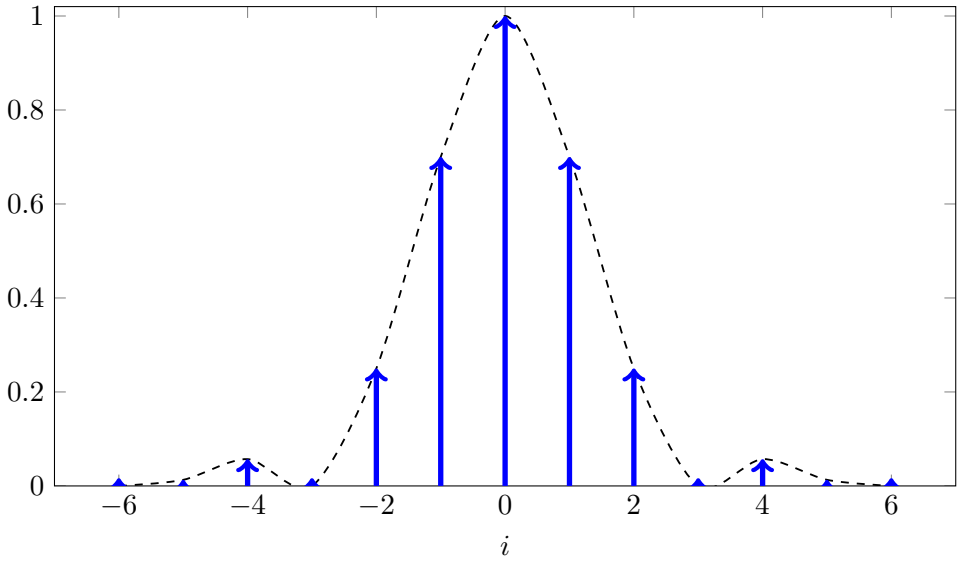



Figure 2.13: absolute value of the RRC filter with roll-off=0.8

However, the reduced level of ISI characterizing the high values of roll-off is obtained at the expense of an increase in the transmission bandwidth.

## 2.5 NON-LINEAR INTER-SYMBOL INTERFERENCE AND SPECTRAL SPREADING IN DVB-S2 SYSTEMS

As the application case of this work is the DVB-S2 system, it is interesting to analyze the issues induced by the HPA non-linearity in such a system. Thus, this section characterizes and analyses the non-linear ISI and the spectral spreading associated with DVB-S2 systems. Such an analysis is done based on results obtained from the computer simulations of the base-band model described on figure 2.1. The DVB-S2 characteristics are modelled according to the standard described in appendix A. The impact of the two issues over the DVB-S2 system is qualitatively evaluated.

The non-linear HPA considered in the simulations is obtained from the AM/AM and the AM/PM functions proposed in the DVB-S2 standard [DS06]. Such functions have been approximated with a Saleh's model by means of curve fitting.



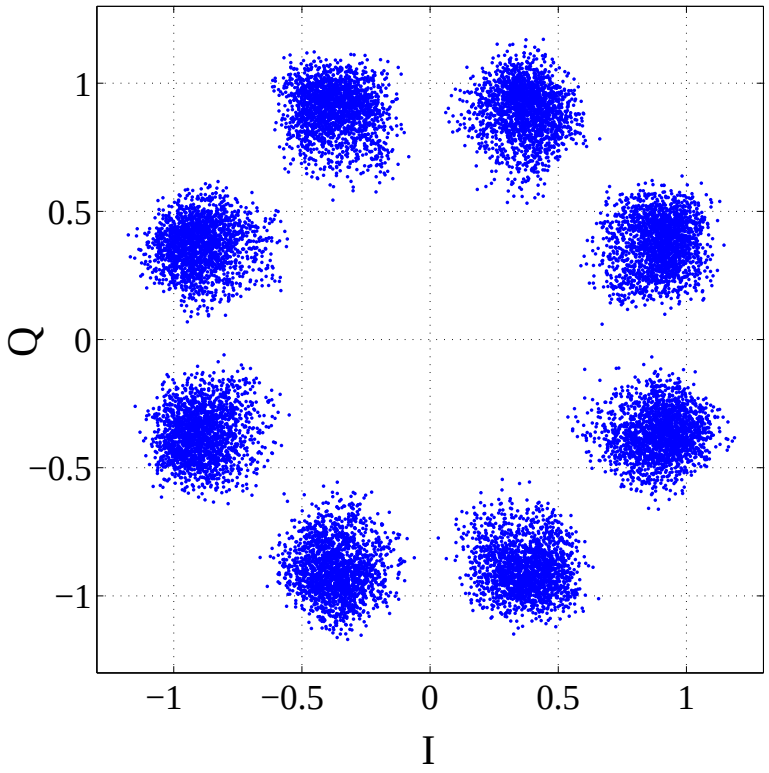
### 2.5.1 Analysis of the non-linear inter-symbol interference

As presented in section 2.4, the level of non-linear ISI associated with the received symbols is on the one hand a function of the roll-off of the transmitter and the receiver digital matched filters (RRC), and on the other hand a function of the back-off characterizing the HPA. Thus, this section studies the impact of the non-linear ISI with respect to the variations of both the roll-off and the back-off.

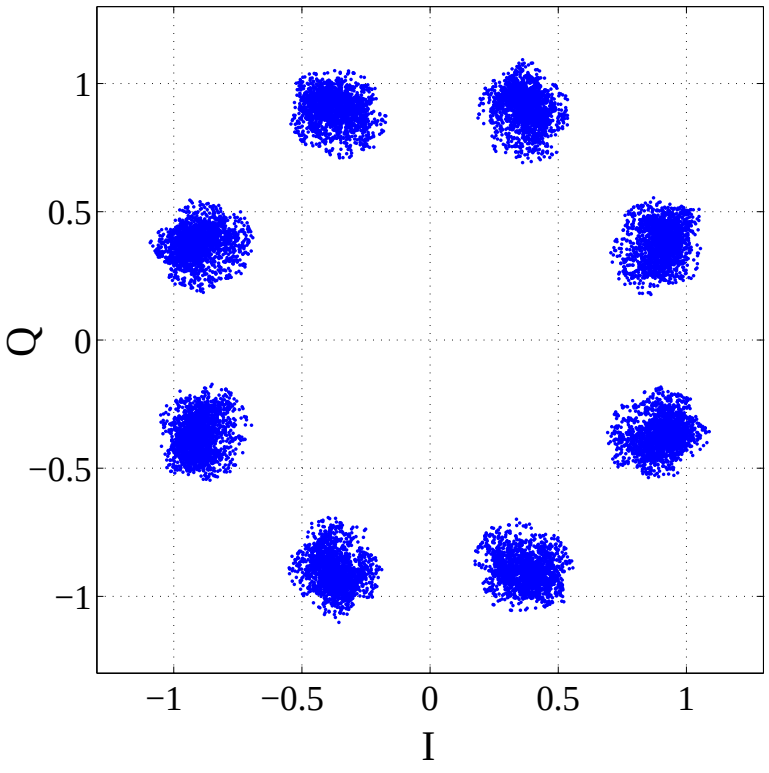
The variations in the dispersion of the received symbols due to the non-linear ISI with respect to the value of roll-off are shown on figures 2.14a and 2.14b for values of roll-off equal to 0.20 and 0.35 respectively. Such values of roll-off are specified in the DVB-S2 standard. The back-off considered in simulations is referenced in terms of the input back-off, with a value of IBO = 0 dB, which is the worst case.

In order to show only the degradation induced by the non-linear ISI, the $E_S/N_0$ considered in simulations is high (100 dB). As illustrated on figures 2.14a and 2.14b, the dispersion of the received symbols varies with the value of roll-off, the lower the roll-off the larger the dispersion.

An equivalent qualitative analysis is done for the variations of IBO. The value of roll-off for simulations is 0.35, which is the mandatory case for the DVB-S2 standard. The values of IBO considered in the simulations are 0 dB, 3 dB and 10 dB. The simulation results are illustrated on figures 2.14b (previously used to compare the dispersion as a function of the roll-off), 2.15a, and 2.15b respectively. As done in the precedents simulations, the $E_S/N_0$ considered is 100 dB.

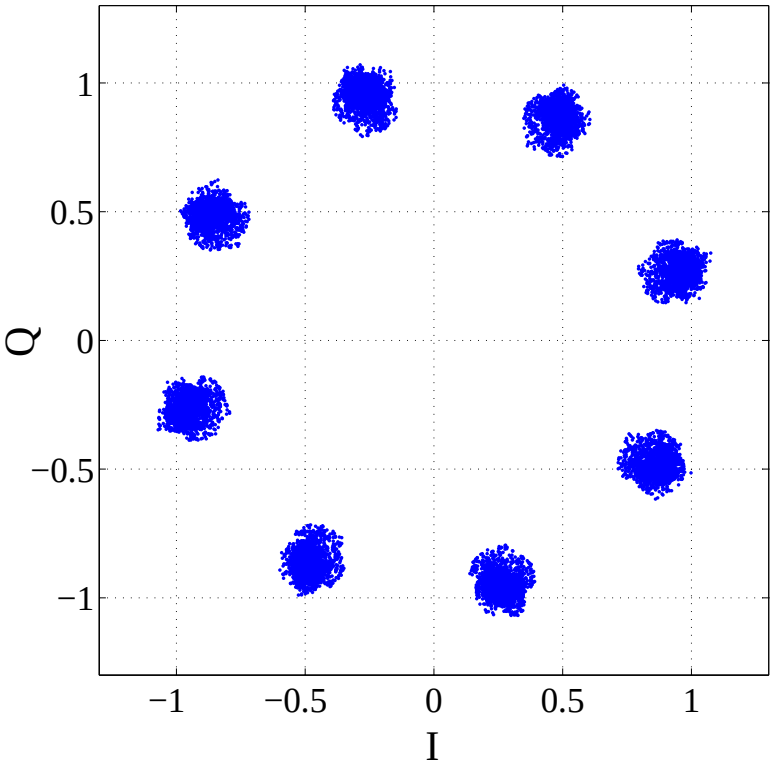



(a) IBO = 0 dB and roll-off=0.20

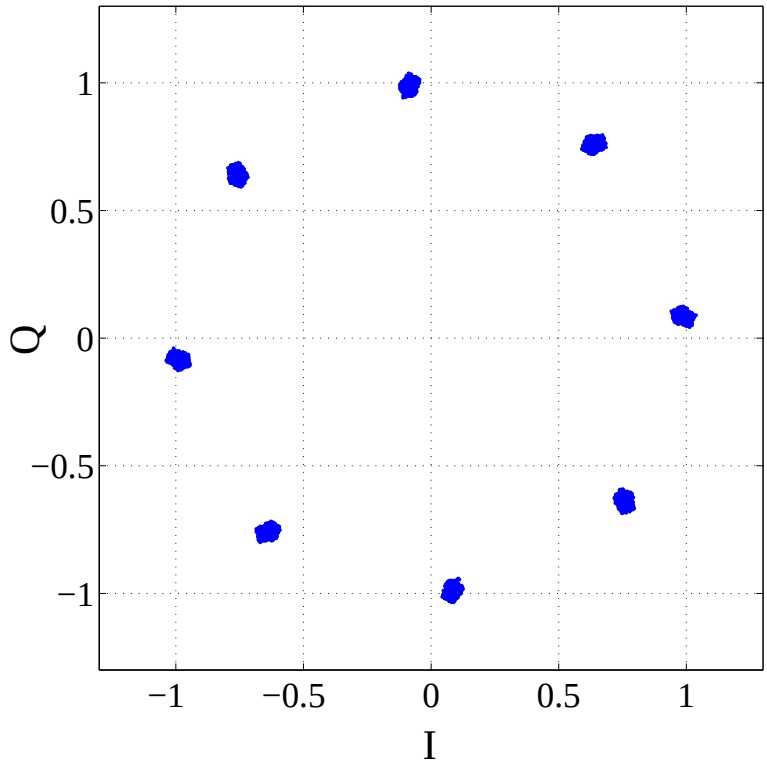



(b) IBO = 0 dB and roll-off=0.35

Figure 2.14: Received symbols for different values of roll-off




(a) IBO =3 dB and roll-off = 0.35




(b) IBO = 10 dB and roll-off = 0.35

Figure 2.15: Received symbols for different values of IBO

As observed on figures2.14b, 2.15a and 2.15b the non-linear ISI induces the higher dispersion in the received symbols points when the HPA works in saturation (IBO = 0 dB). This is clearly exemplified for the case of IBO = 10 dB. In this case, the HPA works in a linear region, avoiding any kind of non-linear issue. Figures 2.14b, 2.15a and 2.15b also depict the phase error associated with the received symbols. This error is due to the AM/PM transfer function characterizing the HPA.

### 2.5.2 Analysis of the spectral spreading



The analysis of the spectral spreading induced by the HPA non-linearity is also done from computer simulations of a DVB-S2 system. The regrowth of the transmission bandwidth is stated by observation of the spectrum of signal $y_t$, just at the output of the Gaussian channel noise (AWGN), figure 2.11. For the simulations, the original transmission bandwidth $\Delta B$, i.e. the transmission band before applying the HPA, is equal to:

$$\Delta B = f_s R. \tag{2.41}$$

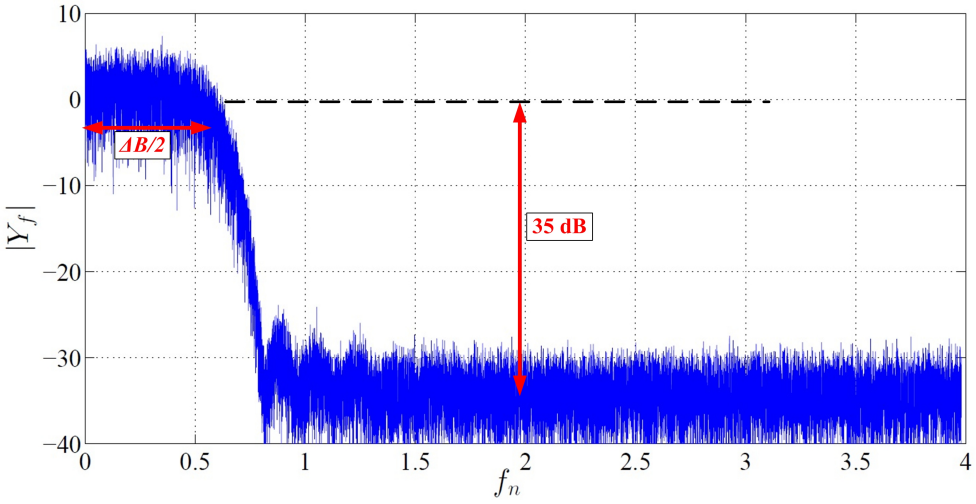
Where $f_s$ is the symbol rate and R is the roll-off value (R=0.35 for the simulations of this subsection). The simulations are carried out with different values of IBO in order to analyze the variations of the spectral regrowth. Figures 2.16a, 2.16b, and 2.16c present the Fourier transform of signal $y_t$ for values of IBO equal to 20 dB, 6 dB and 0 dB respectively. The $E_S/N_0$ considered in the simulations is 35 dB. On figures 2.16a, 2.16b, and 2.16c, the variable $f_n$ is the normalized value of the frequency defined as:

$$f_n = \frac{f}{f_s}. \tag{2.42}$$

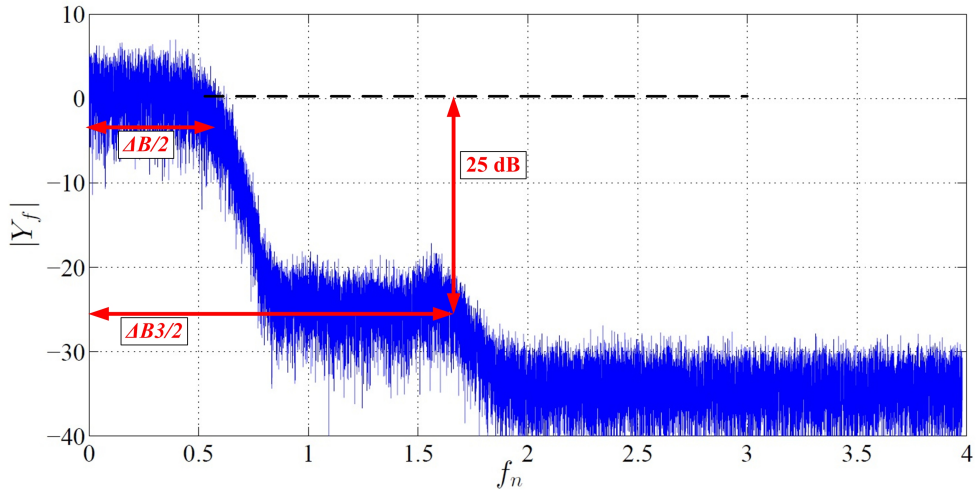
For high values of IBO (20 dB), figure 2.16a, the components associated with the spectral spreading are below the noise floor. Thus, the levels of the order three, order five or higher orders combinations are lower than the level of noise.

The order three combination component can be clearly discriminated in the case of IBO = 6 dB, figure 2.16b. In this case, the difference between the order one and order three component is 25 dB.

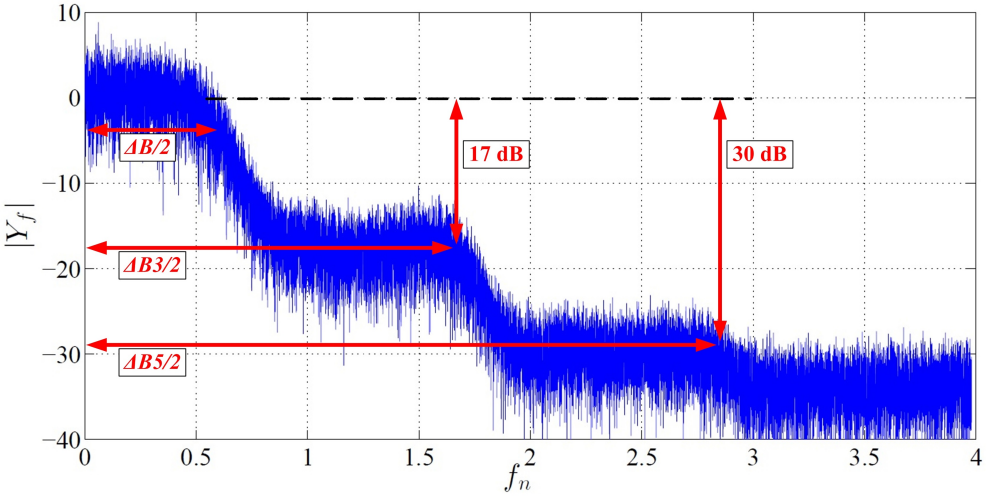
For the worst IBO case (IBO = 0 dB), both the order three and the order five combinations present a power spectrum higher than the noise power spectrum. The difference between the level of the order one and the order three combination is about 17 dB. In addition, the difference between the level of the order one and the order five combination is almost 30 dB. The bandwidth for this case is five times $\Delta B$.




(a) IBO = 20 dB




(b) IBO = 6 dB




(c) IBO = 0 dB

Figure 2.16: Base-band HPA output spectrum for different values of IBO

## 2.6 Summary

This chapter has characterized a typical digital communication system, showing that any perturbation induced by the analogue components results in an undesirable ISI. The necessity of high power efficiency in the analogue components induces a non-linear behaviour. Such a non-linearity is critical in the PA, which is the most significant source of non-linearity. After defining the most relevant PA definitions, different models for different types of PA have been introduced. Then, the two impairments associated with the PA non-linearity, i.e. the spectral spreading and the non-linear ISI, have been characterized.

The spectral spreading has been described from a polynomial PA model. It has been demonstrated that the spectral spreading in the transmission band is composed of odd order combination. The level of each odd order combination is a function of the back-off. The bandwidth of the transmitted signal depends on the order of the polynomial model.



The non-linear ISI has been characterized from a simplified base-band satellite system model. Such a characterization has demonstrated that the non-linear ISI is a function of the roll-off associated with the digital matched filters at the transmitter and at the receiver sides, and the PA back-off.

Finally, this chapter has presented the computer simulations of a satellite digital TV system (DVB-S2 standard) working with a non-linear HPA. The simulation results have validated the concepts introduced for the spectral spreading and the non-linear ISI.

# 3 Non-linear ISI Compensation Techniques

Chapter 2 has described the adverse effects induced by the PA non-linearity into digital communication systems. These effects are the spectral spreading and the non-linear ISI. As the spectral spreading is a transmitter side issue and this work focuses on receiver side compensation techniques, the remaining of this Thesis deals with the compensation of the non-linear ISI.

Therefore, this chapter presents an overview on the different non-linear ISI compensation techniques found in the literature. The compensation techniques are categorized into two families, pre-distortion techniques and post-distortion techniques. The first one is implemented at transmitter side while the post-distortion technique is applied at received side.

As the pre-distortion techniques are used at transmitter side and as this Thesis focuses only solutions implemented at the receiver side, the pre-distortion techniques are only introduced with a short description.

This chapter is organised as follows. Section 3.1 introduces the Volterra series, which are implemented in several pre and post-distortion techniques. Section 3.2 reviews the pre-distortion techniques. In section 3.3, four different types of post-distortion techniques are presented. These techniques are the non-linear Zero Forcing Equalizer, the Volterra Canceller, the turbo canceller, and finally the Neural Network based post distortion techniques. The post-distortion techniques presented in this chapter need to be adapted to the non-linear channel. Then, section 3.4 details the adaptation method used to train the post-distortion techniques. The performance of the different post-distortion techniques implementing different adaptations methods are compared in section 3.5. Section 3.6 analyses the implementation of the post-distortion techniques presented in section 3.3 into commercial DVB-S2 receivers,determining which one is the most adapted for such a receiver. Finally, section 3.7 summarizes this chapter.

## 3.1 Modelling the non-linear ISI with Volterra series

The non-linear ISI associated with the received symbols at instant $n$ ($s'_n$) can be expressed as the linear regression of multidimensional combinations of the transmitted symbols $s$ at instant $n$ and different than $n$, as shown in section 2.4.2. This linear regression corresponds to a Volterra series regression [BB83]. An order $N$ Volterra series regression is defined as:

$$s'_n = \sum_{i=-M}^{M} q_i s_{n+i} + \sum_{i=-M}^{M}\sum_{j=-M}^{M} q_{i,j} s_{n+i} s_{n+j} + \sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M} q_{i,j,k} s_{n+i} s_{n+j} s_{n+k} + O^3. \tag{3.1}$$

Where $M$ is the memory of the non-linear ISI channel, $q_i$ is the $i^{th}$ Volterra coefficient and $O^3$ makes reference to terms with order bigger than three.



Initially proposed by [BB83] to carry out a non-linear equalizer, the Volterra series have been widely used in different architectures of pre-distortion and post-distortion techniques. Some of these architectures are presented in the next subsections.

## 3.2 The pre-distortion techniques

Compensating the non-linear impairments by means of pre-distortion techniques has been the main topic of several works presented in the literature. The attractiveness of such a technique is the absence of channel noise in the data, which is the ideal scenario to carry out any compensation technique. Another strength of this technique is the fact that the transmitted information used to train the pre-distortion system is known at the transmitter side. This is specially useful for communication systems which do not considers pilot information. Moreover, implementing pre-distortion compensates for the two impairments induced by the PA non-linearity, the non-linear ISI and the spectral spreading. A pre-distortion block working into a digital communication system is shown on figure 3.1.

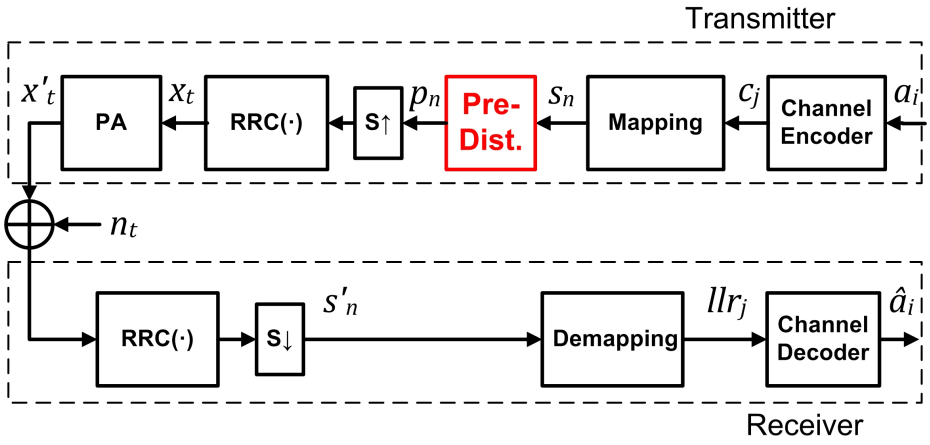



Figure 3.1: Pre-distortion technique into digital communication systems

The implementation of the pre-distorter into a typical transmitter is detailed on figure 3.2. The principle of a pre-distortion technique is that the combination of the pre-distorter, the matched digital filter (block RRC(·)) and the PA presents the same behaviour than the matched digital filter alone. Indeed, the pre-distorter transfer function fits with the inverse of the non-linear PA transfer function. The pre-distortion technique works as follows: The Pre-distortion block distorts the intrinsic symbols $s_n$, resulting in symbols $p_n$. These symbols are up-sampled and filtered by means of a digital matched filter, which output signal $x_t$ is the input of the PA. Then, the amplifier output $x'_t$ is fed-backed, filtered and down-sampled resulting in symbols $s'_n$, an estimation of the transmitted symbols. The error between the fed-backed symbols $s'_n$ and the symbols $s_n$ is used by the Training Block to adapt the Pre-distorter block to the non-linear PA. Indeed, the Training Block minimizes the error:

$$e_n = |s'_n - s_n|. \tag{3.2}$$

Meaning that when $e_n$ is minimal, the distortion added by the pre-distorter matches optimally with the PA non-linearity.



As shown in the state-of-the-art, the Pre-distortion block on figure 3.2 is implemented applying the inverse of the PA transfer function to the transmitted symbols [KS89]. Such an inverse transfer function has been implemented in [BE03], [AGA06] and [GLL04] by means of a polynomial regression, while [MMK+06], [GMC06] and [JP10] have used Volterra series regression.

The pre-distortion techniques presented in the state-of-the-art show the best performance among all the techniques used to compensate for the issues induced by the PA non-linearity. As stated before, this fact is due to the absence of noise and to the knowledge of the transmitted data at the transmitter side. However, the implementation of pre-distortion techniques is not a generalised/mandatory characteristic in transmitters. Then, in order to guarantee the compensation for non-linear ISI, receivers need perform post-distortion. Thus, the next sections present several post-distortion techniques which mitigate the non-linear ISI taking account of the receiver side disadvantages: the Gaussian channel noise and the ignorance of the transmitted data.

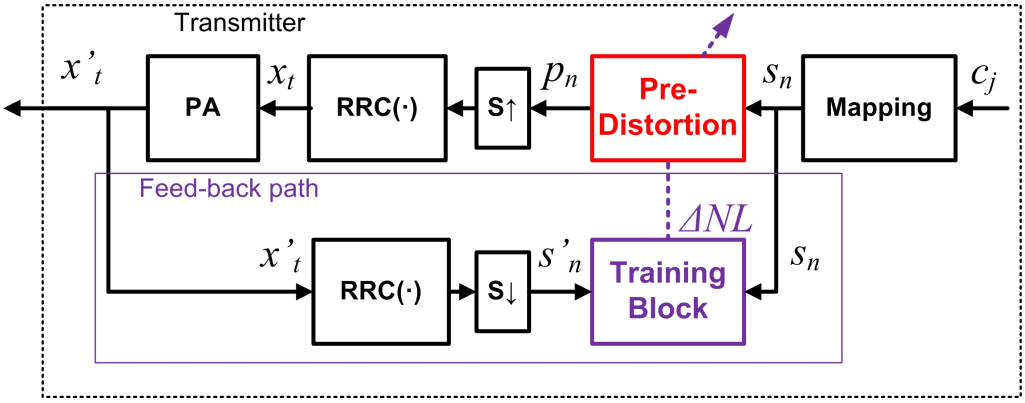



Figure 3.2: The pre-distortion technique

## 3.3 THE POST-DISTORTION TECHNIQUES

This section presents the state-of-the-art in post-distortion techniques. As stated before, these techniques are implemented at receiver after the digital matched filter and the down-sampling block, figure 3.3, block Post-Dist. Hence, the post-distortion techniques compensate for the non-linear ISI by processing the received symbols $s'_n$. The rest of this section introduces four techniques to implement the Post-Dist. block.

### 3.3.1 The non-linear zero-forcing equalizer

As mentioned in section 2.4.2, any kind of ISI (linear or non-linear) rises when the Nyquist frequency criterion is not respected. This criterion states that the ISI is induced when a transmission channel is frequency selective. Figure 3.4 shows both a frequency selective channel and a non-selective one. As depicted on this figure, the frequency domain transfer function of a non-selective channel is constant for any frequency in the band considered. The transfer function of the frequency selective channel varies with frequency.

A frequency selective channel can be compensated by means of a zero-forcing equalizer (ZFE). The principle of an equalizer is to apply the inverse of the channel transfer function $C(f)$ to the incoming symbols $s'_n$, resulting in a constant transfer function (non-selective channel). The non-selective channel resulting from the combination of a frequency selective channel and its inverse function is depicted on figure 3.5.

In order to simplify the notation, this chapter defines a non-linear channel $C(f)$ as the combination of the transmitter and the receiver matched digital filters (RRC), the Gaussian channel noise (AWGN) and the PA, which is the source of non-linearity, figure 3.6. Block V-ZFE represents the non-linear ZFE, which applies the inverse of the non-linear channel transfer function $1/C(f)$. Therefore, if the transfer function of the

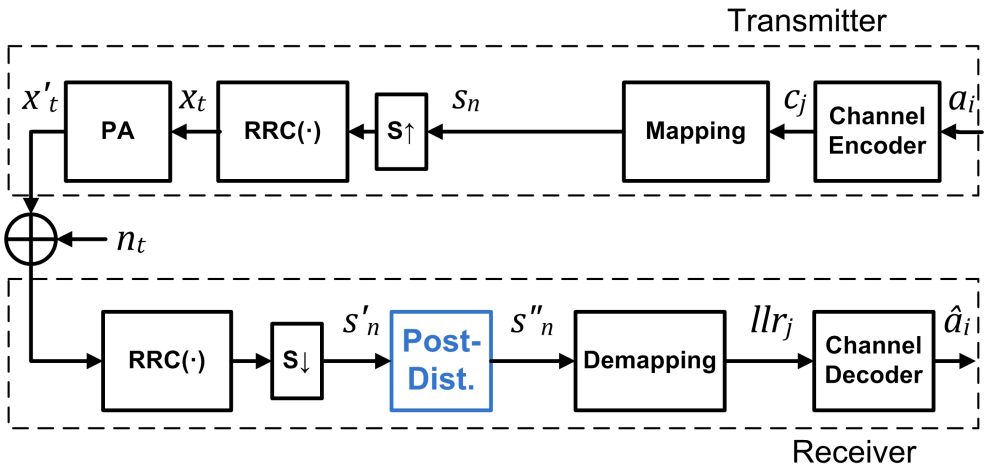



Figure 3.3: Post-distortion technique into digital communication systems

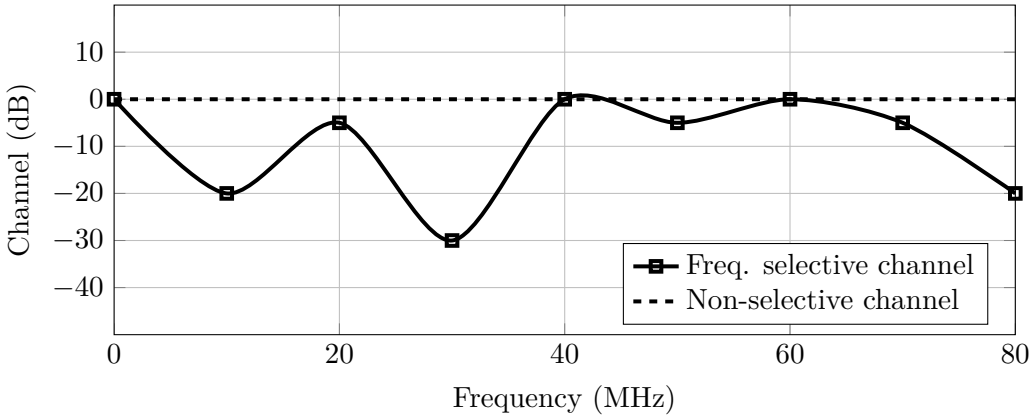



Figure 3.4: Frequency selective and non-selective channels



non-linear channel can be determined, the non-linear ISI in the received symbols can be eliminated.

The-state-of-the-art presents different architectures of non-linear ZFE based on Volterra series. The first approach of Volterra ZFE was introduced by [BB83]. This technique

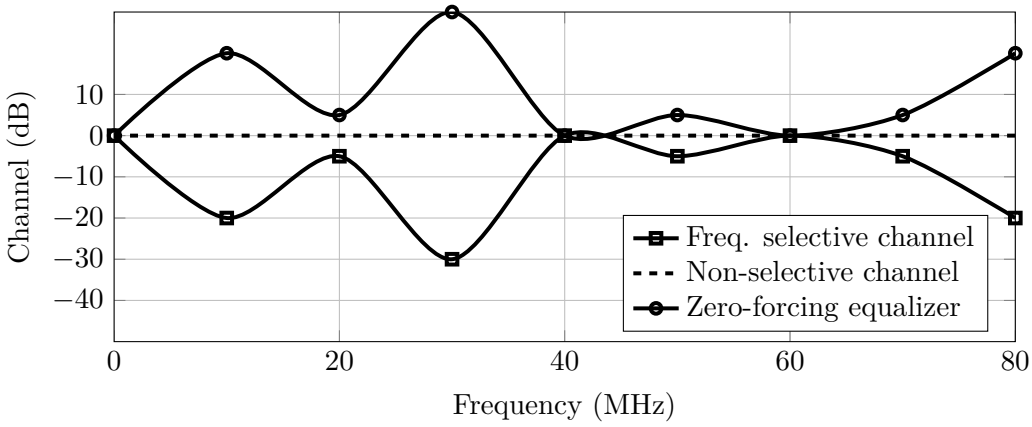



Figure 3.5: The ZFE compensating frequency selective channels

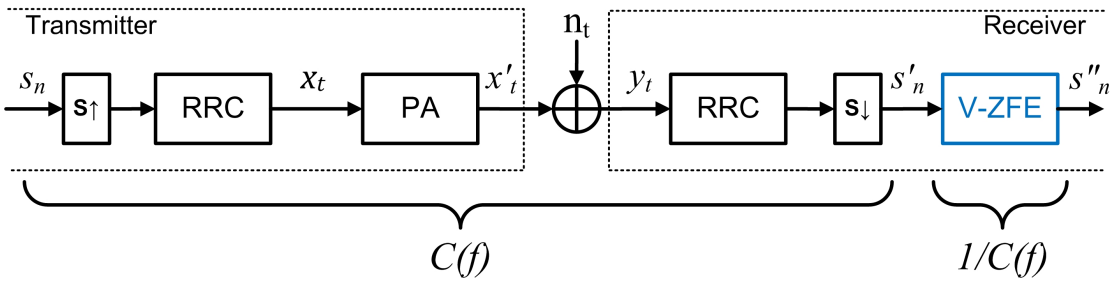



Figure 3.6: The non-linear channel with zero forcing equalizer

is also implemented in [KS89], [BS10] and [GMH+09]. A Volterra ZFE is based on the estimation and application of the inverse of the non-linear channel transfer function by means of a Volterra series regression. The Volterra ZFE is integrated into a receiver as shown on figure 3.7. On this figure, the input signal of the receiver, $y_t$, is the PA output $x'_t$ after applying the Gaussian channel noise $n_t$. The received symbols $s'_n$ are the result of applying digital matched filtering and down-sampling to $y_t$. The symbols $s'_n$ are the input of the V-ZFE, which represents the Volterra ZFE. The output symbol of the Volterra ZFE $s''_n$ is expressed as a Volterra regression as follows:

$$s''_n = \sum_{i=-M}^{M} b_i s'_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} b_{i,j,k} s'_{n+i} s'_{n+j} s'^{*}_{n+k}. \tag{3.3}$$

Where $M$ is the memory of the Voterra ZFE and the $b_i$'s are the coefficients of the Volterra regression. The order $N$ considered in (3.3) is three and the even order combinations are not considered in post-distortion methods because they are out of the transmission band.

As seen on figure 3.7, the received symbol $s'_n$ is also the input of the Training block. Its function is to adapt the coefficients $b_i$ of the V-ZFE block to the non-linear channel. The adaptation is done by minimizing:

$$e_n = |\hat{p}_n - s''_n|. \tag{3.4}$$

Where $s''_n$ is the output of the Volterra ZFE and $\hat{p}_n$ is a training symbol (replica of the transmitted symbol $s_n$). To carry out the adaptation, the Training block implements an algorithm based on the minimization of the error $e_n$. The different types of algorithm implemented in the Training block are detailed in section 3.4.

The architecture of the Volterra ZFE based on (3.3) is depicted on figure 3.8. As shown on figure 3.8, to carry out the Volterra regression, the signal $s'_n$ is delayed by a duration equals to $2 \times M$, $M$ being the memory length. The Volterra Combiner calculates the required multidimensional combinations.

The Volterra ZFE is characterized by having bad performance in $E_S/N_0$ scenarios lower than the maximal attenuation level of the non-linear channel [CLLJ12]. Indeed, when

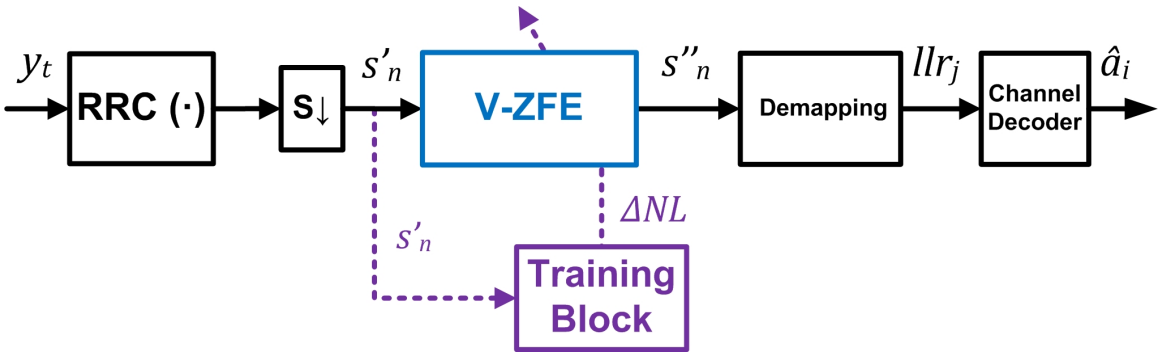



Figure 3.7: The non-linear zero forcing equalizer into a typical receiver

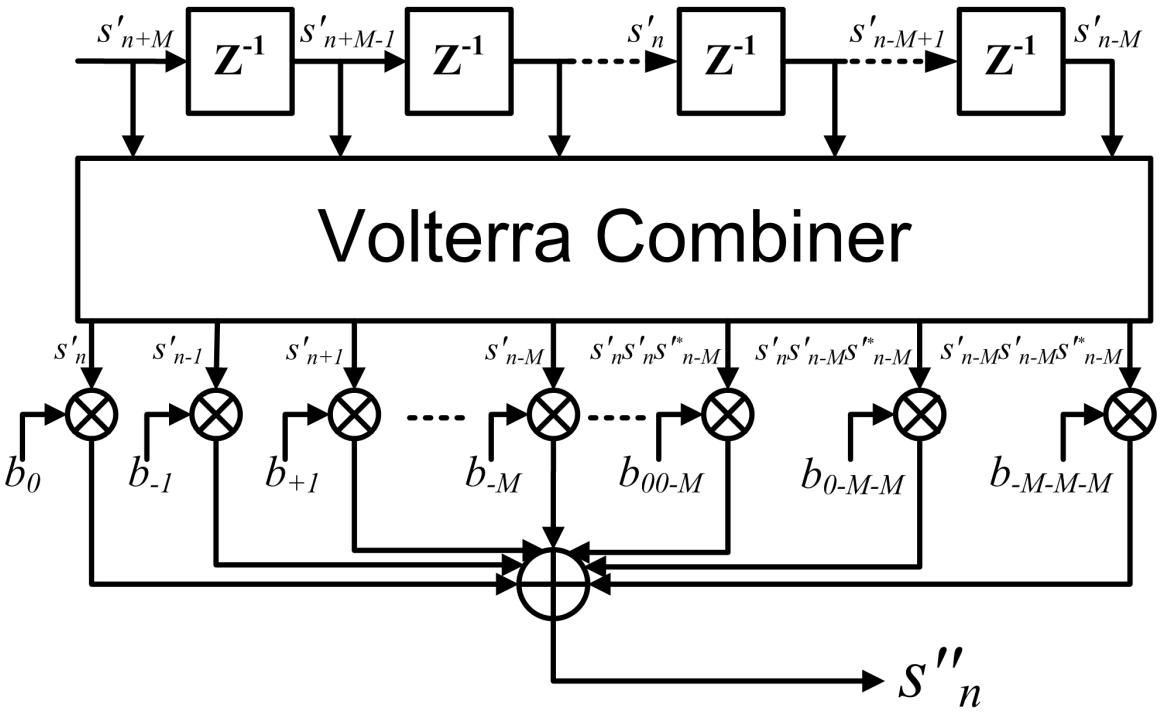



Figure 3.8: Architecture of the Volterra zero-forcing equalizer

the attenuation level of the non-linear channel is below the noise floor, the estimation of the inverse of the non-linear channel transfer function is inaccurate. Then, the resulting compensated channel is not constant in the frequency domain. Thus, the literature presents several approaches for post-distortion techniques outperforming the Volterra ZFE. These techniques are detailed in the next subsections.

### 3.3.2 The Volterra Canceller

Introduced by [Bur06], the Volterra Canceller is another post-distortion technique based on Volterra series regression. Such a technique has been also implemented in [BS10], [ARB+08] and [GWCW07]. Unlike the Volterra ZFE which applies the inverse of the non-linear channel transfer function to the incoming symbols $s'_n$, the Volterra Canceller estimates and eliminates the term of non-linear ISI associated with $s'_n$. The block diagram describing the implementation of a Volterra Canceller into a typical receiver is shown on figure 3.9.

The Volterra Canceller works as follows. The incoming symbols $s'_n$ affected by the non-linear ISI and Gaussian channel noise (AWGN) are a-priori mapped onto minimum distance constellation points $\hat{s}_n$. In other words, the a-priori mapping block assigns in a hard-decision mode a constellation point to each received symbol. Such a mapping is illustrated on figure 3.10, where the noisy non-linear ISI affected symbols are a-priori mapped onto 8-PSK constellation points.

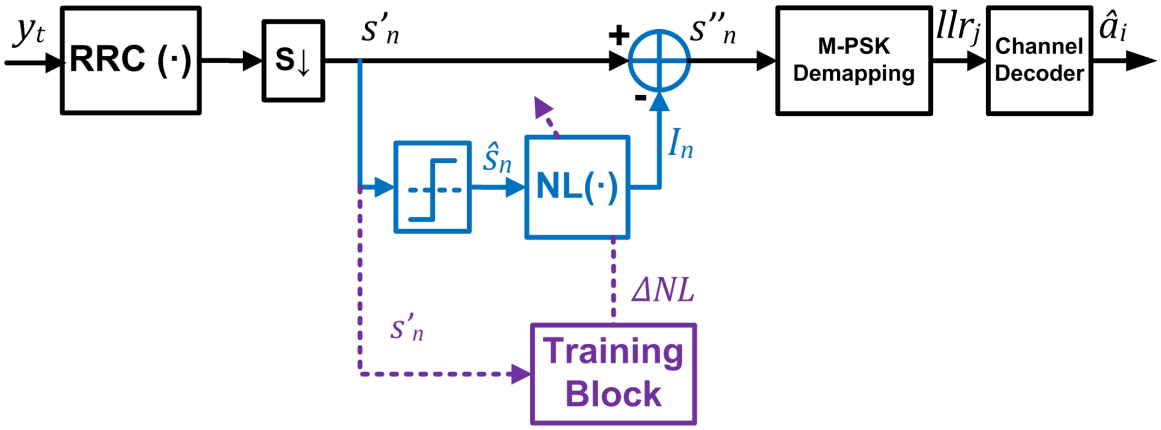



Figure 3.9: The Volterra Canceller into a typical receiver



The non-linear ISI term is estimated by block NL(·), which inputs are the a-priori mapped constellation points. The Volterra series based model characterizing block NL(·) estimates $I_n$, which is the non-linear ISI associated with the received symbol $s'_n$. It is done by means of a Volterra series regression as follows:

$$I_n = \sum_{i=-M}^{M} b_i \hat{s}_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} b_{i,j,k} \hat{s}_{n+i} \hat{s}_{n+j} \hat{s}^*_{n+k} - C_n. \tag{3.5}$$

Where the order of the Volterra Canceller is limited to three and the memory considered is $M$. $C_n$ represents the centroid value of symbol $\hat{s}_n$. The computation of a single centroid value is defined in [BC02] as:

$$C_i = \frac{1}{Nc} \sum_{n=0}^{Nc} s'_n \qquad \text{for any} \quad \hat{s}_n = s_i. \tag{3.6}$$

Where $Nc$ is the number of received symbols considered to compute a single Centroid value. The value of $i$ depends on the number of points in the constellation implemented,

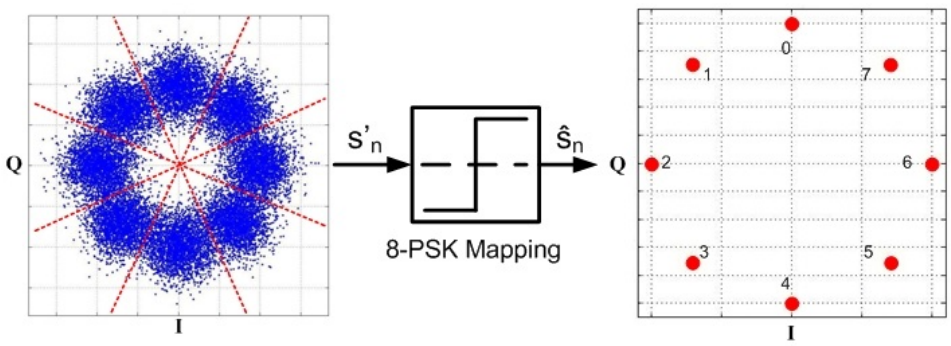



Figure 3.10: 8-PSK a-priori mapping

i.e. $\Omega$ centroids values to compute for an $\Omega$-point constellation. Finally, the compensated symbol $s''_n$ results from subtracting term $I_n$ to the intrinsic received symbol $s'_n$. The architecture of block NL$(\cdot)$ is similar to that of the Volterra ZFE, shown on figure 3.8 .

The Volterra Canceller coefficients $b_i$ also need to be adapted to the non-linear channel. Such an adaptation is done by means of the Training Block shown on figure 3.9. This block adapts the coefficients $b_i$ by minimizing the error function defined as follows:

$$e_n = |s'_n - \sum_{i=-M}^{M} b_i \hat{s}_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} b_{i,j,k} \hat{s}_{n+i} \hat{s}_{n+j} \hat{s}^*_{n+k}|. \tag{3.7}$$

In (3.7), the error function is defined as the difference between the received symbol affected by the non-linear ISI at the instant $n$ and the Volterra series based estimation of this symbols at instant $n$. The algorithms implemented in the Training Block are introduced in section 3.4.

3

### 3.3.3 The non-linear turbo canceller

The turbo coding technique, introduced by [BGT93], has become a generalised characteristic in modern digital communication systems. Performing turbo coding allows a digital communication system to work with lower BER for even lower $E_S/N_0$ scenarios. The combination of turbo decoders with equalizers gives origin to a novel technique of ISI compensation, the turbo equalizer. The turbo equalization of channels impaired by ISI has been covered in [DJB95] and [GLL97].

The turbo equalization has been also implemented in [BC05] and [ARB+08] to eliminate non-linear ISI from a turbo code based communication system by means of a turbo canceller. In [Bei11] and [LF12], a Volterra Canceller is combined with a LDPC decoder to eliminate the non-linear ISI from satellite communication systems. The architecture of the turbo canceller implemented in both [BC05] and [Bei11] is described on figure 3.11.

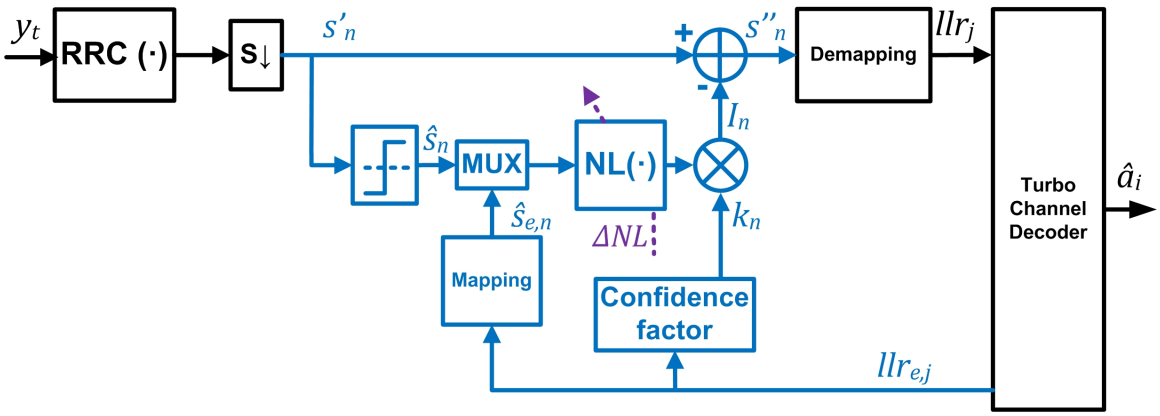



Figure 3.11: The turbo Canceller into a typical receiver

The turbo canceller depicted on figure 3.11 is placed into a turbo code based receiver. The principle of the turbo canceller is to combine the Volterra Canceller with the turbo decoder output $llr_{e,j}$ generated at the $k^{th}$ decoder iteration. Before the first decoder iteration, the Volterra Canceller eliminates part of the non-linear ISI as explained in section 3.3.2. The compensated symbols $s''_n$ are demapped in a soft-decision mode, generating the intrinsic $llr_j$s, which are the input of the turbo decoder. After each iteration, a set of extrinsic decoded $llr_{e,j}$s are mapped onto constellation points $\hat{s}_{e,n}$ and then sent to the Volterra Canceller. Term $I_n$, which represents the non-linear ISI, is again computed from the Volterra model in block NL($\cdot$). A confidence factor $k_n$ is obtained from the $llr_{e,j}$. Such a factor help to ponderate the estimations of $I_n$ associated with the values of $llr_{e,j}$ larger than one. This ponderation helps to improve the performance of the turbo canceller as shown in [BC05].

Once the decoder converges or stops because of the maximum number of iterations is reached, the decoded bits $\hat{a}_i$ are generated.

### 3.3.4 Neural networks based post-distortion techniques

The last technique introduced in this review of post-distortion techniques is the Neural Network based non-linear equalization. Neural Networks have been used by [YWL04], [JSS02], [PPBP99] and [PCMC08] to eliminate the non-linear ISI in different types of digital communication systems. The architecture implemented in those works is based on the Multilayer Perceptron (MLP) described on figure 3.12.

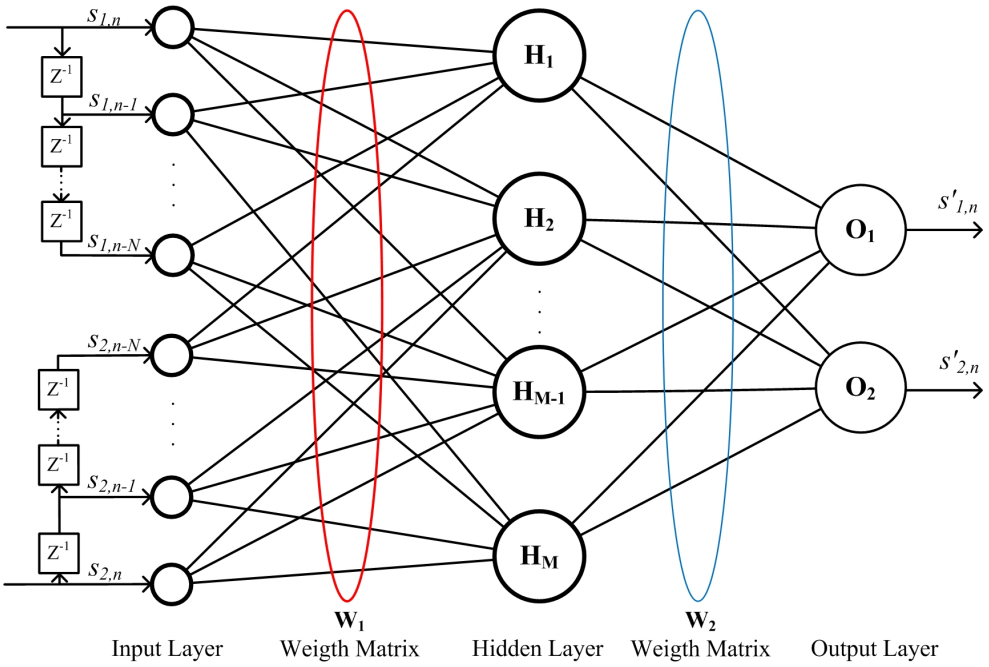



Figure 3.12: NISI compensation with Neural Networks: the MLP architecture

The neurons are organised in layers (three layers on figure 3.12). A standard architecture is composed of an input layer, one or more hidden layers and an output layer.

On figure 3.12, the MLP has only one hidden layer. The MLP based non-linear equalizer delays the inputs $s_{1,n}$ and $s_{2,n}$ by means of $N$ delay blocks, where $N$ represents the memory of the non-linear ISI channel. The input layer distributes the delayed input signals onto multiple paths going to the neurons of the hidden layer. The architecture of a single neuron of the hidden layer ($H_i$) is described on figure 3.13. This type of neuron computes a linear regression with the delayed inputs, and then applies a non-linear function. The output of the $i^{th}$ neuron of the hidden layer is computed as:

$$h_{i,n} = HLF(\sum_{j=0}^{N-1} w1_{i,j} s_{i,n-j}). \tag{3.8}$$

Where $s_{i,n-j}$ represents the $n-j$ delayed inputs$s_i$ and $i$ is the number of the neuron, and $w1_{i,j}$ are the regression coefficients. Generally, the non-linear function $HLF(\cdot)$ is a sigmoid function or a hyperbolic tangent function [RKJ+12].

The architecture of an output layer neuron is almost similar to that of a neuron in the hidden layer, figure 3.14. The output $s'_i$ of an output layer neuron ($O_i$) for an instant $n$ is determined by:

$$s'_{i,n} = OLF(\sum_{k=0}^{M-1} w2_{j,k} h_{k,n}). \tag{3.9}$$

Where $w2_{j,k}$ is the coefficient matrix multiplying the $M$ different hidden layer neuron outputs $H_{k,n}$, and $j$ is the number of the output layer neuron (1 or 2 for figure 3.12).

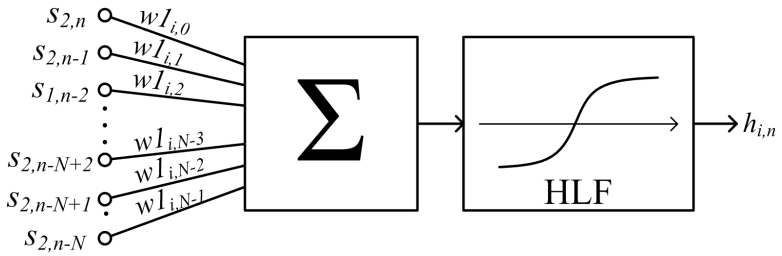



Figure 3.13: The hidden layer neuron

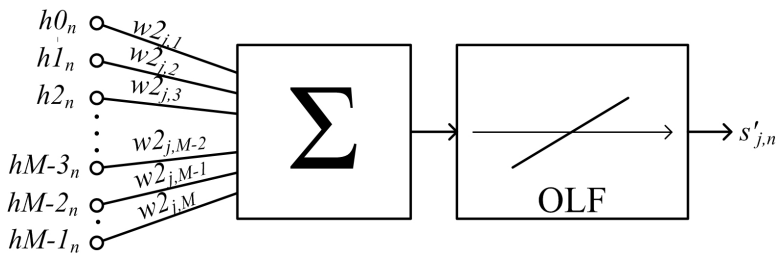



Figure 3.14: The output layer neuron

Finally, the output $s'_{i,n}$ is the result of applying the function $OLF(\cdot)$ to the regression. Function $OLF(\cdot)$ is always a linear function.

The set of coefficients $w1_{i,j}$ and $w2_{j,k}$ are adapted to the non-linear channel by means of the gradient back-propagation algorithm described in [MZ09], which requires training data. This adaptation is done without the need of prior knowledge of the non-linear channel.

## 3.4 Adaptation methods for post-distortion techniques

This section gives an overview of the different adaptation methods for post-distortion techniques. Such methods are used to adapt the linear transversal filter depicted on figure 3.15, which is the architecture used in the techniques presented in sections 3.3.1, 3.3.2 and 3.3.3.



On figure 3.15, $x_n$ is the filter input, $w_i$ is the $i^{th}$ filter coefficient and $\hat{y}_n$ is the filter output. The principle of the adaptation methods is based on the minimization of the square-error value defined as:

$$e_n^2 = |y_n - \hat{y}_n|^2. \tag{3.10}$$

In (3.10), the square-error $e_n^2$ is defined as the squared value of the difference between a training signal $y_n$ and the estimate of the training signal determined by the filter output $\hat{y}_n$ defined as:

$$\hat{y}_n = \sum_{i=0}^{M-1} w_i x_{n-i}. \tag{3.11}$$

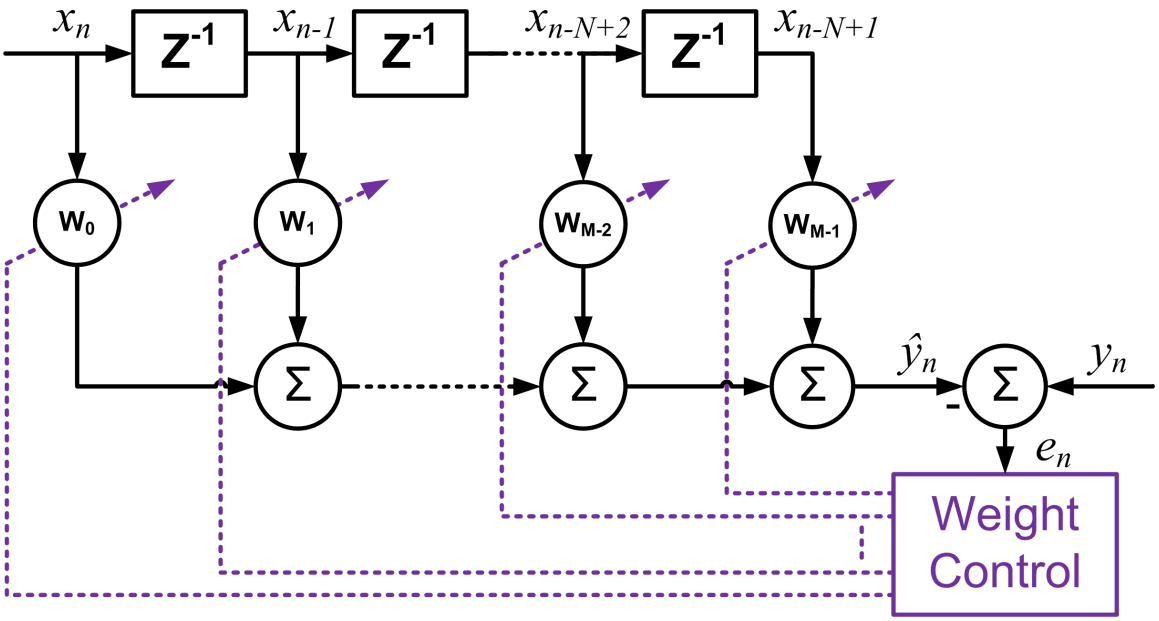



Figure 3.15: Linear transversal filter

The following subsections introduce three adaptation methods based on the minimization of the square-error: the least-squares (LS) algorithm, the least-means squares (LMS) algorithm and finally the recursive least-squares (RLS) algorithm.

### 3.4.1 The least-squares algorithm

The principle of the least-squares algorithm is to optimally estimate the set of $M$ filters coefficients $w_i$ from a set of $M+N$ samples of the input signal $x_n$, sampled at time $n = 0, 1, 2, ..., N+M-1$; and the set of desired training data $y_n$ corresponding to $x_n$. To obtain the optimal estimation of $w_i$s it is necessary to define the sum of the square-error $e^2$ [Hay96]:

$$e^2 = \sum_{n=M}^{N+M-1} |y_n - \hat{y}_n|^2. \tag{3.12}$$

The minimization of $e^2$ is done by partial derivation of (3.12) with respect to each coefficient $w_i$. The minimization for the coefficient $w_r$ is computed as follows:

$$\frac{\partial e^2}{\partial w_r} = -2 \sum_{n=M}^{N+M-1} x_{n-r}(y_n - \sum_{i=0}^{M-1} w_i x_{n-i}) = 0. \tag{3.13}$$

Which leads to:

$$\sum_{n=M}^{N+M-1} x_{n-r} y_n = \sum_{n=M}^{N+M-1} x_{n-r} \sum_{i=0}^{M-1} w_i x_{n-i}. \tag{3.14}$$

Rearranging (3.14) yields:

$$\sum_{n=M}^{N+M-1} x_{n-r} y_n = \sum_{i=0}^{M-1} w_i \sum_{n=M}^{N+M-1} x_{n-r} x_{n-i}. \tag{3.15}$$

Applying partial derivation of $e^2$ with respect to each coefficient $w_i$ leads to a system of $M$ equations. As shown in [Hay96], such a system of equation can be easily represented by its matrix form derived from (3.15). The left side term in (3.15) represents the cross-correlation vector $\mathbf{z}$ and its matrix form is defined as:

$$\mathbf{z} = \mathbf{A^T y}. \tag{3.16}$$

In (3.16) $\mathbf{y}$ is a $N$ length vector composed of training data $y_n$ and $\mathbf{A}$ is the $(N \times M)$ data matrix defined as:

$$\mathbf{A} = [\mathbf{x_M}, \quad \mathbf{x_{M+1}}, \quad \cdots \quad, \mathbf{x_{M+N}}]. \tag{3.17}$$

Where each tap input vector $\mathbf{x_i}$ of length $M$ for an instant $n$ is defined as:

$$\mathbf{x_n} = [x_n, \quad x_{n-1}, \quad \cdots \quad, x_{n-M+1}]^T. \tag{3.18}$$

The right side of (3.15) can be seen as the product between the auto-correlation matrix $\mathbf{\Phi}$ of the signal $x_n$ multiplied by the vector of coefficients $w_i$. The $(k, l)$ element of $\mathbf{\Phi}$ is computed as:

$$\phi_{k,l} = \sum_{n=M}^{N+M-1} x_{n-k} x_{n-l}. \tag{3.19}$$

Matrix $\mathbf{\Phi}$ can be rewritten as the product between $\mathbf{A}$ and its transposed matrix:

$$\mathbf{\Phi} = \mathbf{A^T}\ \mathbf{A}. \tag{3.20}$$

Therefore the matrix system representing (3.15) can be written as:

$$\mathbf{z} = \mathbf{\Phi w}. \tag{3.21}$$

$$\mathbf{A^T y} = \mathbf{A^T A w}. \tag{3.22}$$

In concequence, assuming than the inverse of the square matrix $(\mathbf{A^T A})$ exists, the filter coefficients vector $\mathbf{w}$ can be computed as:

$$\mathbf{w} = (\mathbf{A^T A})^{-1}\mathbf{A^T y}. \tag{3.23}$$



The method of least-squares is limited by the fact that it must be carried out by $N + M$ sets of data $(x_i, y_i)$, which prevent the use of real time applications. The other limitation of this technique is the number of multiplications and additions needed in each matrix multiplication and matrix inversion. Such a factor increases exponentially if the coefficients to estimate and the sets of data $(x_i, y_i)$ are complex.

### 3.4.2 The least-mean squares algorithm

Widely implemented to adapt equalizers in a real-time fashion, the LMS algorithm has also been used to adapt different techniques compensating the non-linear ISI as shown in [MMK+06], [GWCW07] and [CLLJ13]. Such a technique is based on the minimization of the mean square-error (MSE), which is defined in [PS08] as the cost function:

$$J = E|y_n - \hat{y}_n|^2 = E|y_n - \sum_{i=0}^{M-1} w_i x_{n-i}|^2. \tag{3.24}$$

Where $E|\cdot|$ is the mean value operator. To adapt the set of coefficients $w_i$, the LMS algorithm refreshes each coefficient with the negative value of the gradient vector of the cost function $J$, which is composed of $M$ elements. The gradient vector is defined as:

$$\vec{\nabla} J = [\frac{\partial J}{\partial w_0}, \frac{\partial J}{\partial w_1}, \quad \cdots \quad , \frac{\partial J}{\partial w_{M-1}}]^T. \tag{3.25}$$

Where the $l^{th}$ partial derivation of $J$, $\frac{\partial J}{\partial w_l}$ is written as:

$$\frac{\partial J}{\partial w_l} = E[-2x_{n-l}(y_n - \sum_{i=0}^{M-1} w_i x_{n-i})]. \tag{3.26}$$

Then, the partial derivation can be rearranged as:

$$\frac{\partial J}{\partial w_l} = E[-2x_{n-l}y_n] - E[-2x_{n-l}\sum_{i=0}^{M-1} w_i x_{n-i}]. \tag{3.27}$$

$$\frac{\partial J}{\partial w_l} = E[-2x_{n-l}y_n] + 2\sum_{i=0}^{M-1} w_i E[x_{n-l}x_{n-i}]. \tag{3.28}$$

Thus, the gradient $\vec{\nabla}J$ can be rewritten as a matrix:

$$\vec{\nabla}J = -2\mathbf{z} + 2\mathbf{w}\mathbf{\Phi}. \tag{3.29}$$

Where $\mathbf{z}$ is the cross-correlation vector of length $M$ and $\mathbf{\Phi}$ is the auto-correlation matrix of size $M \times M$ [Hay96].

The adaptation of the set of coefficient $\mathbf{w}$ is obtained iteratively. The principle of the algorithm is to periodically add a refresh term to $\mathbf{w}$. The refresh term is determined by the negative value of $\vec{\nabla}J$. As $\vec{\nabla}J$ describes the growing direction of the error function, the negative value of this gradient allows the minimization of the error. Thus, the estimates of $\mathbf{w}$ at an instant $n+1$ is computed as:

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \frac{1}{2}\mu(-\vec{\nabla}J). \tag{3.30}$$

Where $\mu$ is the LMS step which determines the behaviour of the algorithm [Hay96]. Combining (3.29) and (3.30) leads to a recursive equation, which is a function of the two statistics $\mathbf{z}$ and $\mathbf{\Phi}$:

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \frac{1}{2}\mu(-2\mathbf{z} + 2\mathbf{w}\mathbf{\Phi}). \tag{3.31}$$

The LMS algorithm replaces $\mathbf{z_n}$ and $\mathbf{\Phi_n}$ by the instantaneous values $\hat{\mathbf{z}}_\mathbf{n}$ and $\hat{\mathbf{\Phi}}_\mathbf{n}$. The instantaneous value of the cross-correlation vector is defined as:

$$\hat{\mathbf{z}}_\mathbf{n} = \mathbf{x_n}y_n. \tag{3.32}$$

Where $\mathbf{x_n}$ is the tap input vector defined in (3.18). The instant value of the auto-correlation matrix is computed as:

$$\hat{\mathbf{\Phi}}_\mathbf{n} = \mathbf{x_n}\mathbf{x_n^T}. \tag{3.33}$$

Therefore, combining (3.32) and (3.33) with (3.31), the recursive expression for the coefficient vector adaptation can be rearranged as:

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \mu(\mathbf{x_n}y_n - \mathbf{x_n}\mathbf{x_n^T}\mathbf{w_n}). \tag{3.34}$$

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \mu\mathbf{x_n}(y_n - \mathbf{x_n^T}\mathbf{w_n}). \tag{3.35}$$

$$\mathbf{w_{n+1}} = \mathbf{w_n} + \mu\mathbf{x_n}e_n. \tag{3.36}$$

Where $e_n$ is the error between the ideal data $y_n$ and the estimate data resulting from multiplying $\mathbf{x_n^T}$ and $\mathbf{w_n}$.

### 3.4.3 The recursive least-squares algorithm

The last adaptation method presented in this chapter is the recursive least-squares (RLS) algorithm. Such an algorithm has been used in [BS10], [CLLJ12] and [BE03] to adapt post-distortion techniques to non-linear channels. This method is in fact a recursive approach of the least-squares algorithm presented in section 3.4.1. Therefore, the principle of the algorithm is to minimize the cost function defined by the square-error presented in (3.12), which leads to solve the matrix equation:

$$\mathbf{z} = \mathbf{\Phi w}. \tag{3.37}$$

Where $\mathbf{z}$ is the cross-correlation vector defined in (3.16) and $\mathbf{\Phi}$ is the auto-correlation matrix presented in (3.20). In contrast to the LMS algorithm which uses the instantaneous values of $\mathbf{z}$ and $\mathbf{\Phi}$, the RLS algorithm implements a forgetting factor $\gamma$ [Hay96] to compute both statistics. The value of $\gamma$ must be close to one but never larger. Thus, the auto-correlation matrix for the RLS algorithm is computed as:

$$\mathbf{\Phi_n} = \sum_{i=1}^{n} \gamma^{n-i} \mathbf{x_i x_i^T}. \tag{3.38}$$

Where $\mathbf{x_i}$ is the tap input vector defined in (3.18). Then, if the term corresponding to $i = n$ is isolated, (3.38) can be rearranged as:

$$\mathbf{\Phi_n} = \gamma[\sum_{i=1}^{n-1} \gamma^{n-1-i} \mathbf{x_i x_i^T}] + \mathbf{x_n x_n^T}. \tag{3.39}$$

Which leads to:

$$\mathbf{\Phi_n} = \gamma \mathbf{\Phi_{n-1}} + \mathbf{x_n x_n^T}. \tag{3.40}$$

Where $\mathbf{\Phi_{n-1}}$ is the auto-correlation matrix computed at instant $n-1$ and $\mathbf{x_n x_n^T}$ is the refresh correction term updating the value of $\mathbf{\Phi_n}$.

Such a recursive expression can be used to redefine the cross-correlation vector $\mathbf{z_n}$, leading to:

$$\mathbf{z_n} = \gamma \mathbf{z_{n-1}} + \mathbf{x_n} y_n. \tag{3.41}$$

The value of the vector of coefficients $\mathbf{w}$ is determined from (3.37) as:

$$\mathbf{w} = \mathbf{\Phi^{-1} z}. \tag{3.42}$$

The complexity of the inverse matrix operation $\mathbf{\Phi^{-1}}$ can be avoided if such an operation is realized by using the "Matrix Inversion Lemma" [Hay96]. If $\mathbf{A}$ and $\mathbf{B}$ are two positive-definite $M \times M$ matrix related by:

$$\mathbf{A} = \mathbf{B^{-1}} + \mathbf{C D^{-1} C^T}. \tag{3.43}$$

Where $\mathbf{D}$ is a $N \times M$ positive matrix and $\mathbf{C}$ is a $M \times N$ positive matrix, the inverse matrix $\mathbf{A^{-1}}$ can be expressed as:

$$\mathbf{A^{-1}} = \mathbf{B} - \mathbf{BC}(\mathbf{D} + \mathbf{C^T BC})^{-1} \mathbf{C^T B}. \tag{3.44}$$

In order to apply (3.44) into (3.42), the followings identities are defined:

$$\begin{aligned}\mathbf{A} &= \mathbf{\Phi_n}\\ \mathbf{B^{-1}} &= \lambda\mathbf{\Phi_{n-1}}\\ \mathbf{C} &= \mathbf{x_n}\\ \mathbf{D} &= 1\end{aligned}$$

Then, $\mathbf{\Phi_n^{-1}}$ can be written as:

$$\mathbf{\Phi_n^{-1}} = \lambda^{-1}\mathbf{\Phi_{n-1}^{-1}} - \frac{\lambda^{-2}\mathbf{\Phi_{n-1}^{-1}}\mathbf{x_n}\mathbf{x_n^T}\mathbf{\Phi_{n-1}^{-1}}}{1+\lambda^{-1}\mathbf{x_n^T}\mathbf{\Phi_{n-1}^{-1}}\mathbf{x_n}}. \tag{3.45}$$

For the sake of simplicity let us define:

$$\mathbf{k_n} = \frac{\lambda^{-1}\mathbf{P_{n-1}}\mathbf{x_n}}{1+\lambda^{-1}\mathbf{x_n^T}\mathbf{P_{n-1}}\mathbf{x_n}}. \tag{3.46}$$

Where

$$\mathbf{P_n} = \mathbf{\Phi_n^{-1}}. \tag{3.47}$$

Then, (3.45) is written as:

$$\mathbf{P_n} = \lambda^{-1}\mathbf{P_{n-1}} - \lambda^{-1}\mathbf{k_n}\mathbf{x_n^T}\mathbf{P_{n-1}}. \tag{3.48}$$

Where $\mathbf{P_n}$ is defined as the inverse correlation matrix and $\mathbf{k_n}$ is the gain vector. The vector $\mathbf{k_n}$ can be rearranged from (3.46), leading to:

$$\mathbf{k_n} = [\lambda^{-1}\mathbf{P_{n-1}} - \lambda^{-1}\mathbf{k_n}\mathbf{x_n^T}\mathbf{P_{n-1}}]\mathbf{x_n}. \tag{3.49}$$

$$\mathbf{k_n} = \mathbf{P_n}\mathbf{x_n}. \tag{3.50}$$

The next step of the algorithm is to determine a recursive expression for the computation of the coefficient vector. This is done by combining (3.42) with (3.41) and (3.47), which results in:

$$\mathbf{w_n} = \lambda\mathbf{P_n}\mathbf{z_{n-1}} + \mathbf{P_n}\mathbf{x_n}y_n. \tag{3.51}$$

Substituing $\mathbf{P_n}$ by (3.48) only in the first term of the right side of (3.51) leads to:

$$\mathbf{w_n} = \mathbf{P_{n-1}}\mathbf{z_{n-1}} - \mathbf{k_n}\mathbf{x_n^T}\mathbf{P_{n-1}}\mathbf{z_{n-1}} + \mathbf{P_n}\mathbf{x_n}y_n. \tag{3.52}$$

Replacing $\mathbf{P_{n-1}}$ by $\mathbf{\Phi_{n-1}^{-1}}$ results in:

$$\mathbf{w_n} = \mathbf{\Phi_{n-1}^{-1}}\mathbf{z_{n-1}} - \mathbf{k_n}\mathbf{x_n^T}\mathbf{\Phi_{n-1}^{-1}}\mathbf{z_{n-1}} + \mathbf{P_n}\mathbf{x_n}y_n. \tag{3.53}$$

$$\mathbf{w_n} = \mathbf{w_{n-1}} - \mathbf{k_n}\mathbf{x_n^T}\mathbf{w_{n-1}} + \mathbf{P_n}\mathbf{x_n}y_n. \tag{3.54}$$

As $\mathbf{P_n}\mathbf{x_n}$ is equal to vector $\mathbf{k_n}$, (3.54) leads to the recursive expression for the estimation of $\mathbf{w_n}$:

$$\mathbf{w_n} = \mathbf{w_{n-1}} - \mathbf{k_n}[y_n - \mathbf{x_n^T}\mathbf{w_{n-1}}]. \tag{3.55}$$

$$\mathbf{w_n} = \mathbf{w_{n-1}} - \mathbf{k_n}e_n. \tag{3.56}$$

Where $e_n$ is the estimation error defined as:

$$e_n = y_n - \mathbf{x_n^T}\mathbf{w_{n-1}}. \tag{3.57}$$

To be carried out, the RLS algorithm has an initialization phase. Matrix $\mathbf{P_0}$ is initialized as $\mathbf{P_0} = \delta\mathbf{I}$, where $\delta$ is a positive value and $\mathbf{I}$ is the identity matrix. The vector $\mathbf{w_0}$ is generally initialized to 0.

### 3.4.4 Convergence analysis

The convergence and the accuracy of the LMS algorithm are controlled by step $\mu$. The LMS algorithm converges if and only $\mu$ ranges as:

$$0 < \mu < \frac{1}{\lambda_{max}} = \mu_{max}. \tag{3.58}$$

Where $\lambda_{max}$ is the maximum eigenvalue of the auto-correlation matrix $\mathbf{\Phi}$. Any value of $\mu$ larger than $\mu_{max}$ induces a divergent behaviour of the algorithm, [PS08].

Another important characteristic determined by $\mu$ is the speed of convergence. The larger the value of $\mu$ is the higher the speed of convergence is. In contrast, the estimation error is inversely proportional to $\mu$. Such a behaviour can be seen on figure 3.16. On figure 3.16, the estimation of a coefficient of the Volterra Canceller presented in section 3.3.2 is carried out by means of a LMS algorithm. The behaviour of the estimation for values of $\mu$ equal to 0.001, 0.003, 0.007 and 0.01 are determined as a function of the number of LMS iterations. These results have been obtained by means of computer simulations.

As seen on figure 3.16, the fastest convergence corresponds to $\mu = 0.01$. The curve representing the convergence with $\mu = 0.01$ also presents the biggest oscillations, degrading

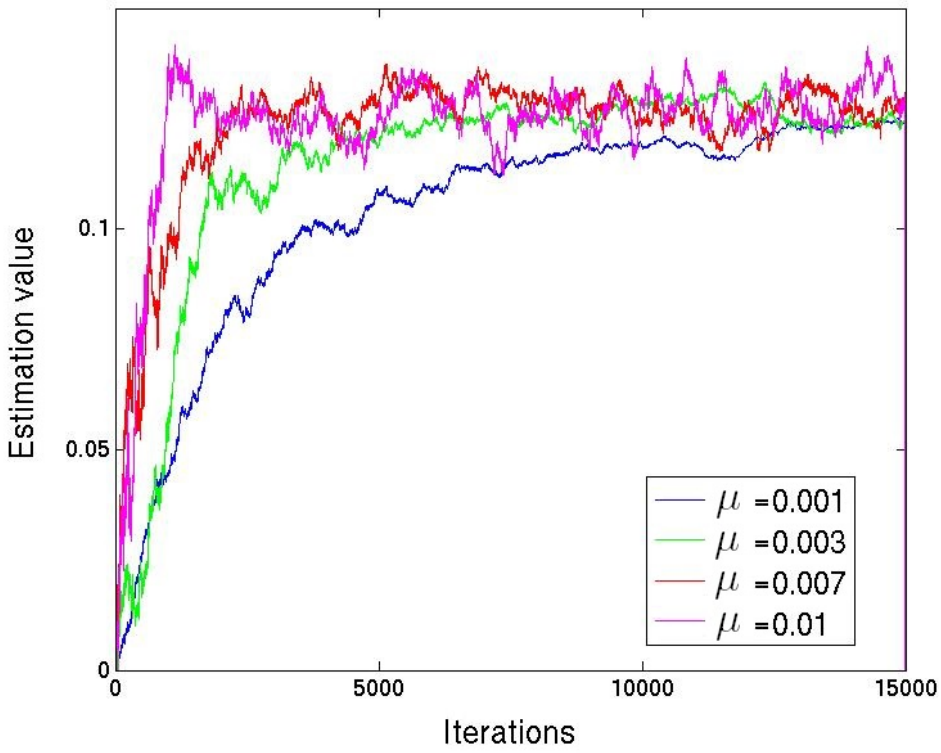



Figure 3.16: LMS convergence and speed as a function of $\mu$

the accuracy of the estimation. The opposite case is shown for $\mu = 0.001$, where the convergence is slower but the oscillations are strongly attenuated.

In contrast to LMS algorithm whose convergence is a function of the step value $\mu$, the RLS algorithm's convergence is independent of the forgetting factor $\gamma$. The convergence of the RLS algorithm is attained in $2M$ iterations, where $M$ is the number of coefficients to estimate. The speed of convergence is typically an order of magnitude higher than the speed of the LMS algorithm. To illustrate this behaviour, figure 3.17 shows the convergence of a Volterra Canceller coefficient trained by either a RLS algorithm or a LMS algorithm implementing values of $\mu$ equals to 0.001 and 0.007.

On figure 3.17, which considers $E_S/N_0$=20 dB, the convergence of the RLS algorithm is faster than the LMS algorithm carried out with $\mu = 0.001$ and $\mu = 0.007$, and the oscillations shown by the RLS curve are also smaller than those of the LMS. These lower oscillations results in a more accurate estimation of the vector of coefficient $\mathbf{w}$ [PS08].



It is important to note than all the advantages of accuracy and speed of convergence of the RLS algorithm over the LMS are only respected in scenarios of high $E_S/N_0$. In case of $E_S/N_0$ ratios near or lower than 10 dB, the performance of both algorithms are equivalents. Such a behaviour is depicted on figure 3.18, where the $E_S/N_0$ is 11 dB. As noted from the RLS algorithm curve, the convergence is still faster than the LMS one, but the oscillation after 6000 iterations are equivalents.

Thus, the RLS algorithm has optimal performance in high $E_S/N_0$ scenarios, outperforming

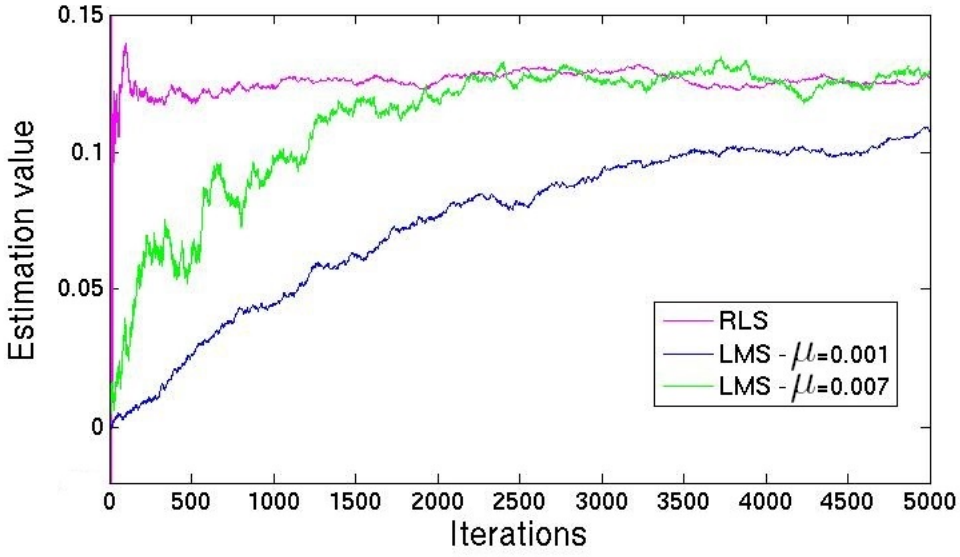



Figure 3.17: Comparison of the convergence and the speed between RLS and LMS algorithms for $E_S/N_0$=20 dB

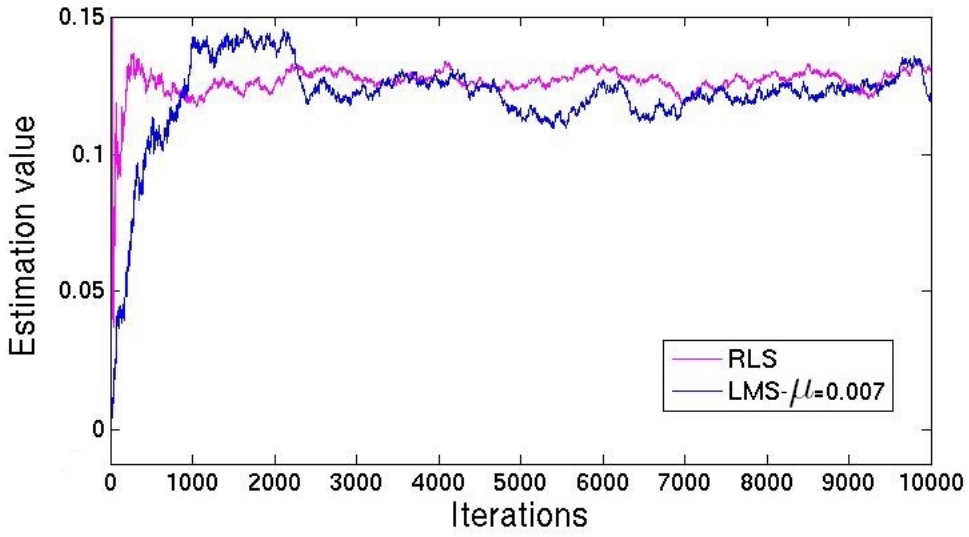



Figure 3.18: Comparison of the convergence and the speed between RLS and LMS algorithms for $E_S/N_0$=11 dB

the LMS algorithm. Nevertheless for the case of low $E_S/N_0$ scenarios, the performance of both the RLS and the LMS algorithms are similar in terms of precision.

### 3.4.5 Complexity analysis

The LMS algorithm is characterized by its low complexity. This complexity can be easily analyzed from table 3.1, which shows a brief description of the steps needed to implement the LMS algorithm.

As seen in table 3.1, the LMS algorithm is based on two steps: the computation of the estimation error $e_n$ and the computation of the refresh term $\mathbf{w}_n$. Both operations are carried out by $2L+1$ multiplications and $2L$ additions, where $L$ is the number of

Table 3.1: The LMS algorithm

| |
|---|
| Initialize $\bar{W}_n$ and $\mu$. |
| For each instant of time $n$, |
| Compute the estimation error $e_n$ as: |
| $e_n = y_n - \mathbf{x_n^T w_n}$. |
| Update the tap-weight vector: |
| $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \mathbf{x}_n e^{\mathbf{w_n}}$. |

Table 3.2: The RLS algorithm

| |
|---|
| Initialize $\mathbf{P}_n$, $\mathbf{w}_n$ and $\gamma$. |
| For each instant of time $n$, |
| Compute the gain vector $\bar{k}$ as: |
| $\mathbf{k}_n = \frac{\mathbf{P}_{n-1}\mathbf{x}_n}{\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{x}_n + \gamma}$. |
| Compute the a-priori estimation error as: |
| $e_{n/n-1} = y_n - \mathbf{w}_{n-1}^T\mathbf{x}_n$. |
| Update the tap-weight vector: |
| $\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{k}_n e^*_{n/n-1}$. |
| Update the inverse correlation matrix: |
| $\mathbf{P}_n = \frac{1}{\gamma}(\mathbf{P}_{n-1} - \mathbf{k}_n\mathbf{x}_n^T\mathbf{P}_{n-1})$ |

coefficients in $\mathbf{w}_n$.

The complexity of the RLS algorithm can be analyzed from table 3.2. This table shows a brief description of the steps needed to implement the RLS algorithm for a non-linear equalizer adaptation.

As seen in table 3.2, the RLS algorithm is carried out by two matrix multiplications and some vector multiplications. Such operations need a number of multiplications equal to $3L^2 + 5L + 1$ and $\frac{3}{2}L^2 + \frac{5}{2}L$ additions, where $L$ is the number of coefficients to estimate. Therefore, the RLS algorithm needs an additional number of operation proportional to $L^2$ compared to LMS algorithm.

As a conclusion, the RLS algorithm is recommended if the $E_S/N_0$ is high and if the complexity is not a prohibitive factor in the system to be adapted. As the performance of the LMS and the RLS are similar for low $E_S/N_0$, the LMS algorithm is recommended instead of the RLS for such a scenario.

## 3.5 EFFECTIVENESS OF THE POST-DISTORTION TECHNIQUES

This section evaluates the ability to mitigate the non-linear ISI of each post-distortion technique presented in section 3.3. Such an evaluation also considers the system complexity characterizing the post-distortion techniques. The ability to mitigate the non-linear ISI is stated in terms of the gain of $E_S/N_0$. The gain is defined as the difference (in dB) between the performance of a system impaired by a non-linear PA and implementing a compensation technique, and another system without compensation. The performance of a digital communication system is stated from different methods. The literature assesses the performance in terms of bit error rate (BER), symbol error rate (SER) or packet

error rate (PER) as a function of the $E_S/N_0$. Examples of BER curves are shown on figure 3.19. As seen on this figure, the gain of a compensated system with respect to the uncompensated non-linear PA case is defined as the difference in dB between these curves for a given BER (BER=$10^{-2}$ on figure 3.19). The concept of loss can be defined in the same way by comparing the compensated or non-linear PA cases with respect to the linear PA case.

The performance can be also presented as a function of the total degradation (TD), which is the result of the sum between the OBO and the increment of $E_S/N_0$ needed to keep a given BER or SER with respect to the linear PA case [GLL04]. The TD is expressed in dB. As an example, figure 3.20 presents a typical curve of TD as a function of the OBO. Figure 3.20 also shows the value of $\mathrm{TD}_{min}$, which characterizes the optimal system operation point.



### 3.5.1 Effectiveness comparison between the Volterra ZFE, the Volterra Canceller and the turbo canceller

This section evaluates the gain of $E_S/N_0$ of digital communication systems implementing different approaches of the Volterra ZFE, the Volterra Canceller and the turbo canceller with respect to the uncompensated system. The literature related to these post-distortion techniques presents several works implementing multiple standards with multiple digital modulations. In addition, the gain presented in those works is stated in terms of BER, SER or TD. Therefore, in order to compare efficiently the gain of the digital communication systems implementing each one of the post-distortion techniques, this section only considers the works implementing the particular case of 16-QAM and 16-APSK digital modulations. The conclusions obtained for such digital modulations can be then generalised for any other

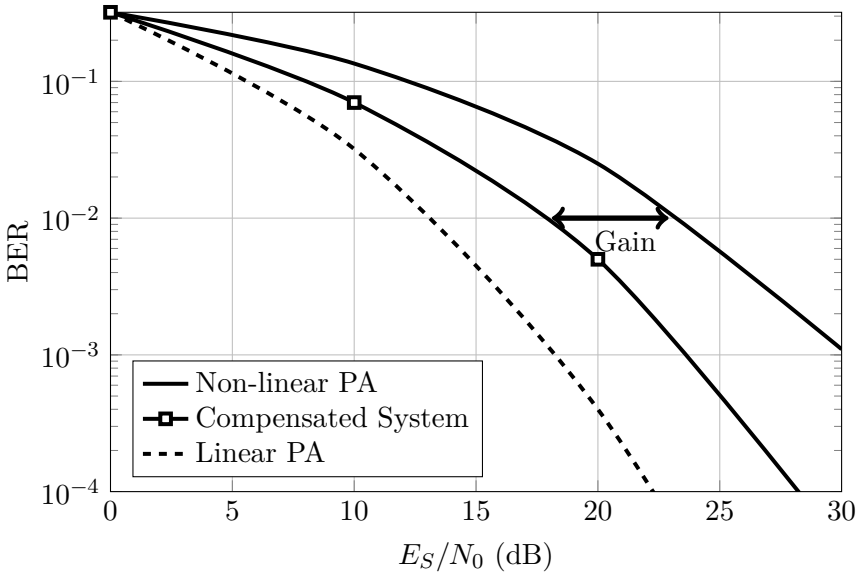



Figure 3.19: The BER curve as a function of the $E_S/N_0$

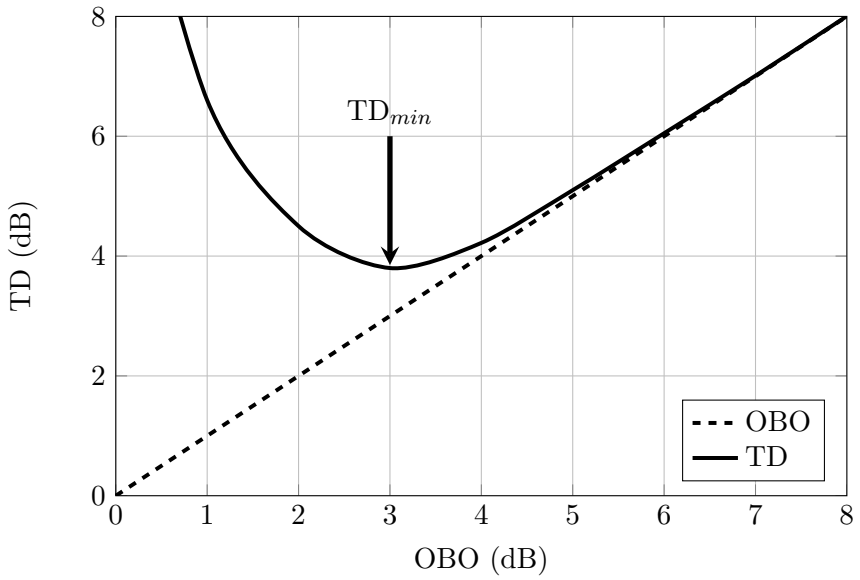



Figure 3.20: TD as a function of OBO

Table 3.3: Post-distortion techniques performance

| Technique | References | Digital Modulation | Roll-off | Gain [dB] |
|---|---|---|---|---|
| Volterra ZFE | [GLL04] | 16-QAM | 0.3 | 0.60 |
| | | 16-APSK | 0.3 | 0.25 |
| | [BS10] | 16-APSK | 0.25 | 0.8 |
| | | 16-QAM | 0.25 | 1.5 (RLS) |
| | | 16-QAM | 0.25 | 1.35 (LMS) |
| Volterra Canceller | [BS10] | 16-APSK | 0.25 | 1.2 |
| | | 16-QAM | 0.25 | 1.7 |
| | [Bur06] | 16-QAM | 0.2 | 0.68 |
| | [Bur06](Ideal VC) | 16-QAM | 0.2 | 1.57 |
| Turbo canceller | [BC05] | 16-QAM | 0.2 | 0.79 |

digital modulation. The most relevant informations characterizing the post-distortion techniques are summarized in table 3.3.

In [GLL04] and [BS10] the gain is assessed in terms of TD for a given BER=$10^{-3}$. In contrast, [BC05] and [Bur06] presents the performance in terms of BER curves as a function of the $E_S/N_0$. The IBO considered for [BC05] and [Bur06] is IBO=0 dB (worst

case). No information about the IBO is given in [GLL04] and [BS10].

As shown in table 3.3, the gain associated with the Volterra Canceller proposed by [Bur06] (0.68 dB) is bigger than the gain obtained with the Volterra ZFE considered in [GLL04] (0.60 dB). This difference of gain can be bigger if the system model [Bur06] and [GLL04] implemented the same roll-off factor. If 16-QAM digital modulation is considered, the turbo canceller seems to be the most efficient post-distortion technique. As stated in table 3.3, the difference of gain between the turbo canceller and the Volterra canceller proposed in [Bur06] is 0.11 dB, which is the 14% of the whole turbo canceller gain.

A particular case considering the Volterra Canceller and the Volterra ZFE is that proposed in [BS10]. This work is analyzed separately because the proposed satellite communication system is impaired by non-linear ISI and inter-carrier interference (the last impairment is not considered by the other works). As seen in table 3.3, for the case considering 16-APSK digital modulation, the gain of a Volterra Canceller is 1.2 dB while the gain of the Volterra ZFE is of 0.8 dB. Thus, the difference of gain is 0.4 dB, which is the 33% of the gain of the Volterra Canceller. With respect to the 16-QAM case, the difference of gain is lower (0.2 dB), which is the 12% of the gain of the Volterra Canceller.

The gain of the ideal Volterra Canceller proposed in [Bur06] represents the limit of gain of the Volterra Canceller, i.e. obtained when the non-linear ISI term is estimated from error-free a-priori mapped symbols $\hat{s}_n$. Obviously, this is an ideal performance.

As stated previously, a complete analysis of the effectiveness of a post-distortion technique can be done by analyzing the gain of such a technique with its system complexity. Then, as the system complexity associated with the Volterra Canceller and the Volterra ZFE are similar and as the Volterra Canceller reaches a bigger gain, it can be inferred that the Volterra Canceller is more efficient than the Volterra ZFE. The same analysis can be done between the turbo canceller and the Volterra Canceller. Even if the gain of the turbo canceller is bigger (14%), the implementation of this technique is more complex than the one of the Volterra Canceller. Such a complexity is added by the feed-back path of figure 3.11, which maps the extrinsic bits $llr_{e,j}$ into symbols $\hat{s}_{e,n}$ and computes the confidence factor. Another important complexity issue associated with the turbo canceller is the memory needed to train the canceller. This memory is proportional to the latency induced by the turbo decoder.

Hence, the post-distortion technique showing the better trade-off between system complexity and gain is the Volterra Canceller. In addition, the ideal Volterra Canceller presents the best gain among all the post-distortion techniques introduced in this state-of-the-art. Thus, the Volterra Canceller can be modified in order to reach a gain near that of the ideal case. Such an improvement is presented in chapter 4

### 3.5.2 Performance of the Neural Networks based post-distortion techniques

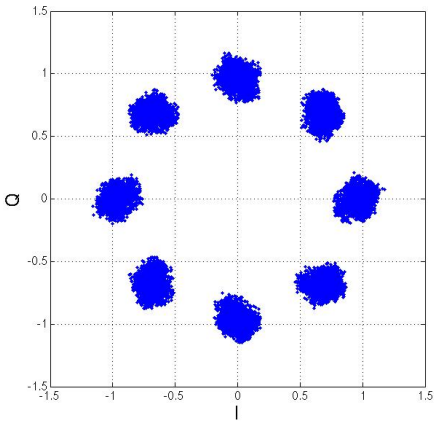
The state-of-the-art of Neural Network based post-distortion has been generally applied into 2-PAM or 4-QAM communications systems. Therefore, the comparison between the performance of this technique with the other post-distortion techniques described in this Chapter is meaningless.

In [PJ99], a Q-PSK based non-linear satellite system has been compensated with Neural Networks post-distortion, which results in a gain of 1.72 dB with respect to uncompensated case for a BER=$10^{-4}$. A QAM case with a different non-linear channel model is studied in [PPBP99], where the gain with respect to the uncompensated case is of 2.6 dB for a BER=$10^{-3}$ and $E_S/N_0$ close to 14 dB. Finally, [ZZ09] proposes a Neural Network based solution to compensate a 2 PAM based digital communication system impaired by a non-linear channel. In this case, the gain is 1.1 dB and the $E_S/N_0$ close to 14 dB.

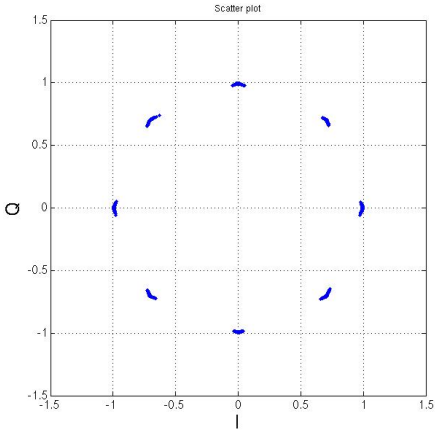


To analyze the behaviour of a Neural Network based post-distortion system, this section proposes to analyze qualitatively the performance of the method into a DVB-S2 system, which is the application case of this Thesis. Figure 3.21 shows six simulations representing both the constellation points received from the noisy channel, i.e. the $(s_i, s_j)$ on figure 3.12 and the constellation points at the output of the neural network. The simulations consider the following values of $E_S/N_0$: 35 dB, 25 dB and 15 dB. The HPA has an IBO=0 dB and the roll-off is 0.35. The neural networks considered in simulations are composed of 11 neurons in the hidden layer and two neurons in the output layer, and memory M=3. These parameters have been chosen by optimisation from Matlab simulations. The neural networks are adapted by means of a back-propagation algorithm.

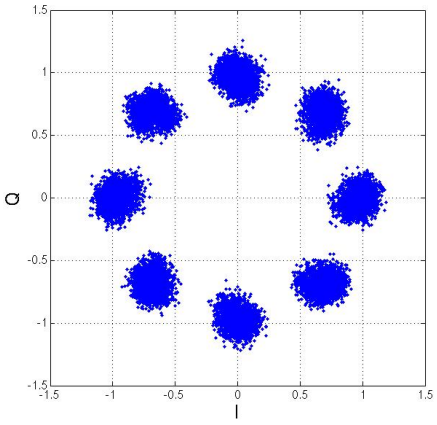
As seen on figure 3.21, the non-linear ISI can be eliminated for relatively high $E_S/N_0$, in this example 35 dB. However, when the power of noise is higher, the compensated constellation does not show any improvement. Hence, the results presented in this subsection justify the fact that neural networks can be only used for scenarios where $E_S/N_0$ is bigger than 20 dB.



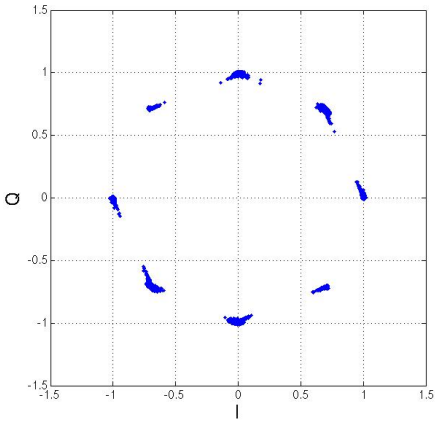

(a) Non-linear channel output, $E_S/N_0$=35 dB



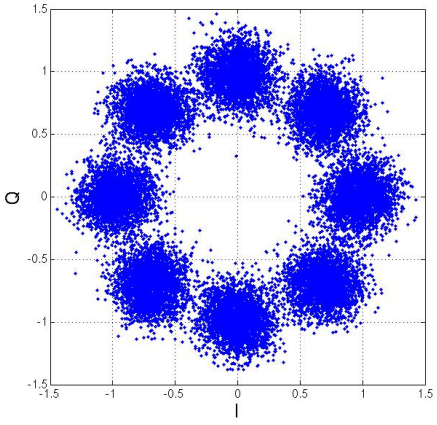

(b) Neural Network output, $E_S/N_0$=35 dB



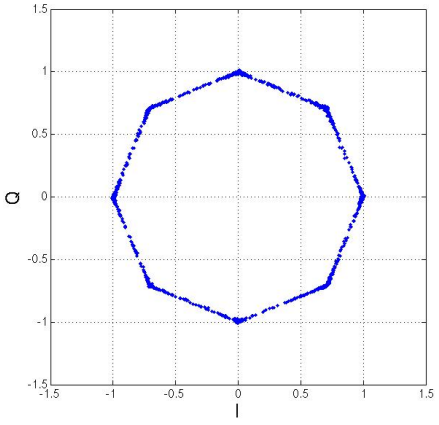

(c) Non-linear channel output, $E_S/N_0$=25 dB




(d) Neural Network output, $E_S/N_0$=25 dB




(e) Non-linear channel output, $E_S/N_0$=15 dB




(f) Neural Network output, $E_S/N_0$=15 dB

Figure 3.21: 8-PSK constellation under non-linear ISI channels: non-linear channel output vs. Neural Network output

## 3.6 LIMITATIONS OF POST-DISTORTION TECHNIQUES IN COMMERCIAL DVB-S2 RECEIVERS

This section focus on the limitations related to the implementation of each post-distortion technique presented in section 3.3 into a DVB-S2 receiver. The DVB-S2 receiver considered for this work is a commercial one. The design of a commercial receiver focuses on achieving the bigger gain of $E_S/N_0$ with the minimal system complexity. Therefore, any post-distortion system integrated in the commercial DVB-S2 receiver must respect both system constraints.

In [BS10], the Volterra ZFE has been implemented with 16-APSK and superior order digital modulations. Thus, the Volterra ZFE can be useful if it is implemented in not-mandatory modes of the DVB-S2 standard. The non-mandatory modes consider 16-APSK and 32-APSK digital modulations, which are not used in commercial receivers, hence, they are not considered in this work.



A DVB-S2 receiver working with mandatory cases, i.e. the digital modulations implemented by the receiver are Q-PSK and 8-PSK, achieves Quasi Error Free transmissions in $E_S/N_0$ conditions lower than 10 dB [DS06]. Moreover, the $E_S/N_0$ can be negative when Q-PSK digital modulation is used with code rate values 2/5, 1/3 and 2/5. Then, the poor performance of Volterra ZFEs in low $E_S/N_0$ conditions prevents their use in the context of a commercial DVB-S2 receiver.

The Volterra Canceller has not been implemented in the state-of-the-art to compensate for the non-linear ISI in communications systems based on 8-PSK or Q-PSK modulations. Nevertheless, as the Volterra Canceller is not based in the estimation of an inverse transfer function, this technique can be carried out even if the $E_S/N_0$ is that of commercial DVB-S2 receivers. This is possible because the non-linear ISI, $I_n$ in (3.5), is estimated by taking into account the noise associated with the received symbols.

The turbo canceller has shown better performance than the Volterra Canceller. The limitation of the turbo canceller into a commercial DVB-S2 receiver is firstly related to the system complexity required to carry out this technique, prohibitive for commercial applications. Secondly, the DVB-S2 standard is characterized by frames composed of 64800 bits (appendix A) and the latency associated to the decoding of such a frame. Thus, the memory needed to train the turbo canceller prevents the implementation of a this technique [CLLJ12].

The last technique analyzed is the Neural Network based post-distortion techniques. In terms of complexity, the high quantity of neurons needed to model the non-linear ISI channel prevents its implementation in commercial DVB-S2 receivers. In addition, the complexity of the Neural Networks required in the adaptation block grows exponentially with respect to the number of neurons considered. Moreover, as the post-distortion based on Neural Networks are only efficient for $E_S/N_0$ scenarios bigger than 20 dB, this technique can be only implemented in DVB-S2 receivers considering 16-APSK and 32-APSK digital modulations.

Hence, in the context of the compensation of the non-linear ISI for commercial DVB-S2 receivers, the Volterra Canceller is the state-of-the-art post-distortion technique showing the better trade-off between the gain in terms of $E_S/N_0$ and the system complexity. However, the gain a Volterra Canceller achieves is almost 50% lower than that of an ideal Volterra Canceller (table 3.3). Thus, it is worth trying to improve the $E_S/N_0$ gain of the Volterra Canceller.

## 3.7 Summary

This chapter has presented the state-of-the-art in non-linear ISI compensation techniques. The first section has introduced the Volterra series regression, widely used in the different compensation techniques.



As this work addresses receiver side techniques, pre-distortion techniques have been only briefly introduced. Nevertheless, such compensation techniques are efficient approaches able to eliminate both the non-linear ISI and the spectral spreading induced by a non-linear HPA.

Four state-of-the-art post-distortion techniques were introduced. The first one was the Volterra ZFE. It is based on the estimation of the inverse of the non-linear channel transfer function. The second one was the Volterra Canceller. It estimates and eliminates the non-linear ISI term from the received symbols. The third one also estimates the non-linear ISI but iteratively by using the feed-back provided by turbo decoder. Finally, the last post-distortion technique presented was the Neural Network based post-distortion technique.

As the post-distortion techniques need to be adapted to the non-linear channel, this chapter has also presented few training methods used to carry out each every post-distortion technique. Thus, the LS, the LMS and the RLS algorithms have been presented. In addition, the complexity and the performance of these algorithms have been analysed.

Taking as reference the results found in literature, the effectiveness of the Volterra ZFE, the Volterra Canceller and the turbo canceller have been compared. The criteria to assess the effectiveness are the gain of $E_S/N_0$ and the system complexity associated with each post-distortion technique. This comparison showed that the turbo canceller presents the biggest gain of $E_S/N_0$. However, the Volterra Canceller has shown a reduced system complexity and a gain near turbo canceller's one. Moreover, when the Volterra Canceller estimates the non-linear ISI term from ideal data, this technique outperforms the turbo canceller.

The performance of the Neural Network based post-distortion techniques have been analysed from computer simulations. These simulations consider the model of a DVB-S2 system, which is the application case of this work. As shown in the simulations, the Neural Networks can estimate and compensate for the non-linear ISI only for scenarios where $E_S/N_0$ is bigger than 20 dB.

Finally, this chapter has analysed which post-distortion technique is best suited for commercial DVB-S2 receivers. Such an analysis infers that the Volterra Canceller is the technique showing the best trade-off between gain of $E_S/N_0$ and system complexity.

3

# 4 Polynomial linearization for non-linear ISI compensation: a post-distortion approach

Different non-linear ISI compensation techniques have been described in chapter 3. The non-linear ZFE, the Volterra Canceller, the turbo canceller and the Neural Network based post-distorter were introduced as a state-of-the-art in post-distortion techniques. In the application case of a low-cost receiver, the Volterra Canceller seemed to be the solution showing better trade-off between performance and complexity. Moreover, when the non-linear ISI is estimated from error-free constellation points, the Volterra Canceller outperforms any compensation technique presented in the state-of-the-art.

This chapter presents a novel non-linear ISI compensation technique based on the linearization of the non-linear channel at the receiver side. Such a linearization is carried-out by means of an orthogonal polynomial regression. In terms of performance, the combination of a polynomial linearizer with a Volterra Canceller shows performance near that of an ideal Volterra Canceller.

In addition, this chapter presents two methods to adapt the polynomial linearizer in a blind mode. Thanks to these methods, the adaptation algorithms can be used independently of the standard implemented in the digital communication system.

This chapter is organised as follows. Section 4.1 introduces the polynomial linearizer implemented from two different families of orthogonal polynomials. A description of two blind adaptation techniques is presented in section 4.3. Section 4.4 analyses and compares the performance of the novel technique with the state-of-the-art. Finally, section 4.5 summarizes this chapter.

## 4.1 Polynomial regression for channel linearization

### 4.1.1 Generalities on post-distortion based linearizers

An efficient way to avoid the non-linear ISI associated with the received symbols is by linearizing the non-linear channel. Generally, the linearization of any non-linear system is carried out by multiplying the transfer function of such a system by its inverse function. Figure 4.1 presents the linearization of a satellite HPA. As shown on this figure, the linearization of the amplifier is done by combining the non-linear AM/AM transfer function of the HPA with its inverse function, resulting in a linear AM/AM transfer function.

As presented in section 3.2, pre-distortion techniques avoid the adverse effects induced by non-linear channels by linearizing the HPA at the transmitter side. The pre-distortion is performed before the PA.



As the linearization is based on the multiplication of both the PA and the linearizer transfer functions, the result of first applying the PA transfer function or the linearizer transfer function does not change the final result, meaning that the resulting transfer function is always linear. This implies that, instead of implementing the linearizer at the transmitter side as it is usually done [KS89], it can also be implemented on the receiver side as it is proposed in this work. The advantage of linearizing at transmitter is that the compensation for non-linear ISI is always guaranteed, which is interesting for digital

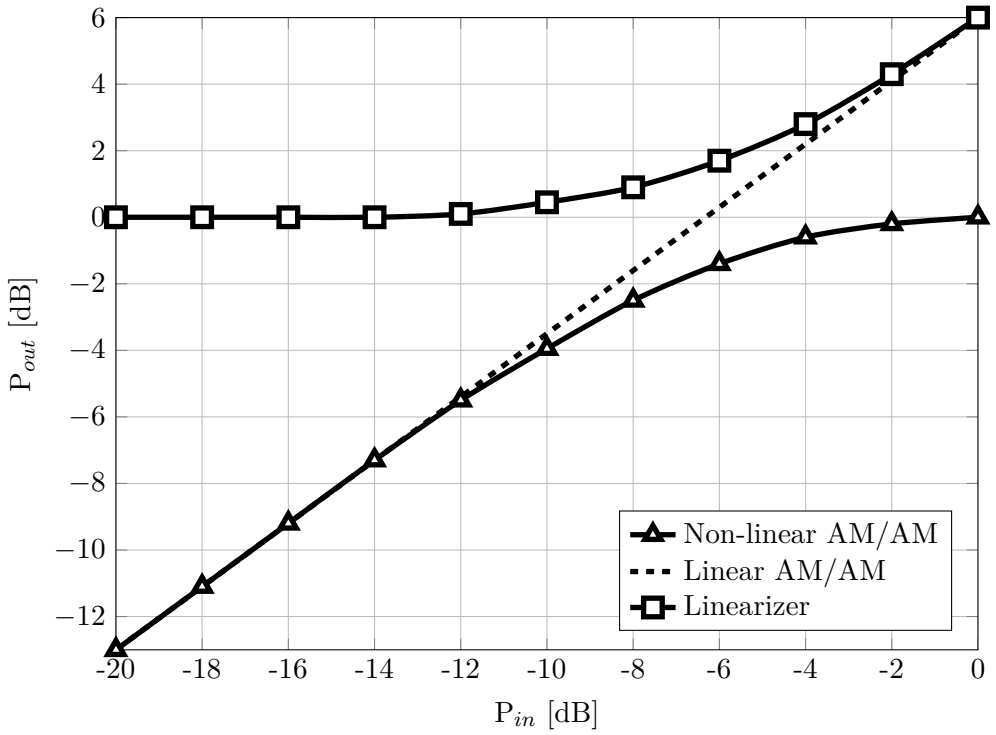



Figure 4.1: Linearization by appling the inverse of the HPA transfer function

communication system which transmitter does not consider pre-distortion as a mandatory characteristic. The disadvantage of a post-distortion linearizer is that it must deal with the Gaussian channel noise (AWGN). Such an impairment does not affect the linearizer implemented at transmitter. Therefore, the post-distortion linearizer must be adapted to the additive channel noise (AWGN).

The implementation of a post-distortion linearizer into a communication system is illustrated on figure 4.2. The linearizer (block LIN) is placed before the receiver digital matched filter because, as demonstrated in section 2.4.2, the memory effect of the non-linear ISI is induced by such a filter. Thus, compensating before filtering prevents the non-linear ISI.

On figure 4.2, the output of the linearizer is expressed as:

$$y'_t = Lin(y_t). \tag{4.1}$$

Where $Lin(\cdot)$ is a function characterizing block LIN. Such a function is an estimation of the inverse of the PA transfer function. A Taylor series expansion can model the inverse PA transfer function. As defined in the Taylor's theorem [Tay13], a function can be approximated around a given point $\alpha$ by means of a $T$ order polynomial. When applied to the computation of the inverse of the PA transfer function, the value of $\alpha$ must correspond to the PA back-off. Therefore, the Taylor regression is defined as:

$$\hat{Lin}(y_t) = \sum_{j=0}^{T} c_j (y_t - \alpha)^j. \tag{4.2}$$

The Taylor coefficient $c_i$ is defined as:

$$c_j = \frac{Lin^{(j)}(\alpha)}{j!}. \tag{4.3}$$

where $Lin^{(j)}(\cdot)$ is the $j^{th}$ derivate of $Lin(\cdot)$.

As described in (4.3), the estimation of the coefficients $c_k$ requires the knowledge of function $Lin(\cdot)$. As the post-distortion techniques are implemented at the receiver side, the

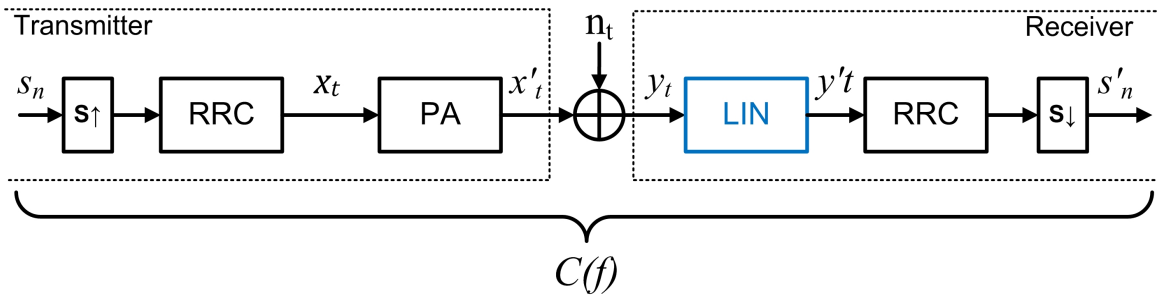



Figure 4.2: Post-distortion based linearization

non-linear PA transfer function and its inverse (function $Lin(\cdot)$) are unknown parameters. Therefore, the coefficients $c_j$ must be estimated. Actually, even if the PA transfer function was known, block LIN cannot be implemented only by applying the inverse of the PA transfer function. This is due to the presence of the Gaussian channel noise (AWGN). Therefore, the estimation of the $c_j$ is done by taking into account the down-link noise.

The limitation of linearizing a non-linear channel with a Taylor approximation is that the model of the inverse of the PA transfer function is done around the value of $\alpha$. Therefore, the implementation of this method can be only possible in systems for which peak-to-average power ratio (PAPR) is small, which is not the case of systems with high order digital modulations [BJ10]. Consequently, the model of the inverse of the PA transfer function must consider the wide dynamic of the PA.

In [RQZ04], a pre-distortion system is implemented by modelling the inverse of the PA transfer function from orthogonal polynomials regression. The results demonstrate that linearizing with an orthogonal polynomial regression is efficient in terms of stability, system complexity and performance. Since the linearizer can be applied after or before the PA, such a polynomial regression can be also applied at the receiver side, leading to a post-distortion polynomial linearizer.



### 4.1.2 Laguerre Polynomial Linearizer

Among all the families of orthogonal polynomials, the one optimally adapted to model non-linear systems is the family of Laguerre polynomials [FLA94]. For instance, the complex Laguerre polynomials from order 0 to 6 are shown in table 4.1.

Table 4.1: Laguerre polynomials

| |
|---|
| $L_0(y) = 1.$ |
| $L_1(y) = -y + 1.$ |
| $L_2(y) = \frac{1}{2}(yy^* - 4y + 2).$ |
| $L_3(y) = \frac{1}{6}(-y^2y^* + 9yy^* - 18y + 6).$ |
| $L_4(y) = \frac{1}{24}(y^2(y^*)^2 - 16y^2y^* + 72yy^* - 96y + 24).$ |
| $L_5(y) = \frac{1}{120}(-y^3(y^*)^2 + 25y^2(y^*)^2 - 200y^2y^* + 600yy^* - 600y + 120).$ |
| $L_6(y) = \frac{1}{720}(y^3(y^*)^3 - 36y^3(y^*)^2 + 450y^2(y^*)^2 - 2400y^2y^* + 5400yy^* - 4320y + 720).$ |

Such polynomials have been used in [CLLJ12] to carry out the linearization of the received signal. The Laguerre polynomial linearizer is implemented into a typical receiver, block LIN on figure 4.3. The output of the linearizer can be expressed as a function of the received signal $y_t$ as:

$$y'_t = \sum_{j=0}^{T} d_j L_j(y_n). \tag{4.4}$$

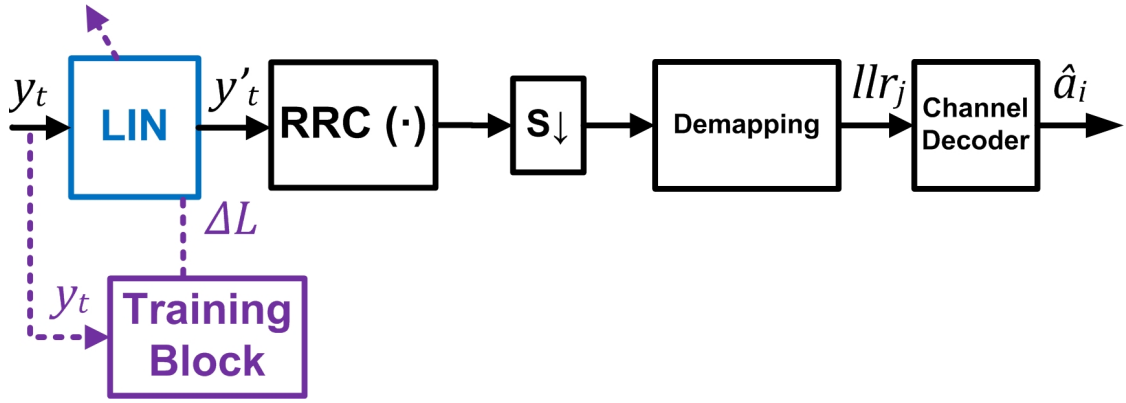



Figure 4.3: Polynomial linearizer implemented into a typical receiver architecture

In (4.4), $T$ is the order of the polynomial regression and $d_j$ is the regression coefficient multiplying the polynomial of order $j$. The set of $T+1$ coefficients characterizing the Laguerre polynomial linearizer are adapted to the non-linear channel by means of either the RLS or the LMS algorithms presented in section 3.4, which are implemented in the Training Block.



### 4.1.3 Hermite Polynomial Linearizer

As demonstrated in section 2.4, when the non-linearity characterizing a channel is solely induced by the PA, the received signal $y_t$ is a function of only the odd order combinations of the transmitted signal $x_t$.

As any even order combination of $x_t$ is uncorrelated with the received signal $y_t$, the regression fitting the inverse of the PA transfer function must be done only with odd order combinations of the received signal $y_t$. This concept can be used to optimise the polynomial linearizer. As seen in table 4.1, any Laguerre polynomial is composed of odd and even order powers of $y_t$. Thus, using an order $T$ Laguerre polynomial regression to fit the inverse of the PA transfer function requires all the Laguerre polynomials from order 0 to $T$. If $T = 3$ then four polynomials, of order 0, 1, 2 and 3, are required, and 4 coefficients regression $c_i$ must be estimated.

Any reduction in the number of polynomials required in the regression simplifies the linearizer. This simplification can be achieved by carrying out the regression from a different family of orthogonal polynomials. This family must be characterized by odd order polynomials independent of even order combinations, and even order polynomials independent of odd order combinations. Such a condition is respected by the Hermite polynomials. These polynomials have been implemented to model non-linear systems in [Tsi95] and [Ogu07]. For instance, the Hermite polynomials from order 0 to 6 are presented in table 4.2.

Table 4.2: Hermite polynomials

| |
|---|
| $H_0(y) = 1.$ |
| $H_1(y) = y.$ |
| $H_2(y) = 2yy^* - 1.$ |
| $H_3(y) = 2y^2y^* - 3y.$ |
| $H_4(y) = 4y^2(y^*)^2 - 12yy^* + 3.$ |
| $H_5(y) = 4y^3(y^*)^2 - 20y^2y^* + 15y.$ |
| $H_6(y) = 8y^3(y^*)^3 - 60y^2(y^*)^2 + 90yy^* - 120.$ |



Consequently, the Hermite polynomial linearizer based on an order $T$ Hermite polynomial regression can be defined as:

$$y'_t = \sum_{j=0}^{T} p_{2j+1} H_{2j+1}(y_t). \tag{4.5}$$

Where $p_{2j+1}$ is the Hermite coefficient multiplying the $H_{2j+1}$ polynomial. As in the case of the Laguerre polynomial linearizer, the coefficients are trained by means of LMS or RLS algorithms. The Hermite polynomial linearizer has been introduced in [CLLJ13]. As shown in [CLLJ13], when the linearization is done with Hermite polynomial regression, the reduction in the complexity is achieved in both the Linearizer block (LIN) and the Training Block. Therefore, it is interesting to analyze the complexity reduction resulting from the implementation of Hermite polynomials.

### 4.1.4 Comparison between the Laguerre and the Hermite polynomial linearizers

The system complexity of both the Laguerre polynomial linearizer and the the Hermite polynomial linearizer is determined on the one hand by the order of the polynomial regression, and on the other hand by the method used to adapt the linearizer to the non-linear channel.

The order of the polynomial regression is defined by the complexity allowed to the linearizer and by the $E_S/N_0$ level which characterizes the system where the linearizer is implemented. In fact, the level of the noise associated with $y_n$ grows inversely with respect to the $E_S/N_0$. Then, the value of the $T$ order polynomial (Laguerre or Hermite polynomials) varies geometrically with respect to the noise level. Therefore, in order to guaranteed the convergence of the linearizer, the high order polynomials ($T$ larger than 5) must be avoided in scenarios for which $E_S/N_0$ is lower than 15 dB. Such values have been determined from computer simulations. The simulations consider a DVB-S2 receiver, a non-linear HPA and a linearizer to compensate for the non-linear ISI.

The complexity of a Laguerre polynomial linearizer (excluding the Training Block) performing polynomial regressions of order 2 to 6 is summarized in table 4.3. The complexity is analyzed in terms of number of complex additions and complex multiplications needed to carry out the Laguerre polynomial linearization.

Table 4.3: Complexity of the Laguerre polynomial linearizer for different regression orders

| Order | Complex Additions | Complex multiplications |
|---|---|---|
| 2 | 5 | 4 |
| 3 | 9 | 6 |
| 4 | 14 | 8 |
| 5 | 20 | 10 |

The complexity required for the LIN block performing an order $T$ Hermite polynomial regression is presented in table 4.4.



Table 4.4: Complexity of the Hermite polynomial linearizer for different odd regression orders

| Order | Complex Additions | Complex multiplications |
|---|---|---|
| 3 | 2 | 4 |
| 5 | 5 | 7 |

Therefore, taking account of the comparison between table 4.3 and table 4.4, a reduction of 33% in the number of multipliers is achieved for an order 3 linearizer, and a reduction of almost 30% for the order 5. With respect to the number of addition, the difference is even larger. The reduction of additions is 60% and 75% for order 3 and 5 regressions respectively.

The simplification of the Training Block adapting each one of the polynomial linearizers is obtained from the reduction of the number of polynomials needed for an order $T$ polynomial regression. As example, while a Laguerre polynomial linearizer of order 5 needs to adapt 6 coefficients (order 0 to 5), the Hermite polynomial linearized only needs the adaptation of 3 coefficients (order 1, 3 and 5). As the number of complex multiplications and additions needed to carry out the adaptation methods presented in section 3.4 is a function of the number of coefficients to train, the reduction of half the number of coefficients leads to the simplification of the Training Block.

The complexity of the LMS and RLS based Training Block adapting a Laguerre and a Hermite polynomial linearizer is detailed in table 4.5. Again, the complexity is compared in terms of complex additions and multiplications, for the cases of polynomial regressions of order 3 and 5. The term $L$ represents the number of coefficients to adapt.

Table 4.5: Training Block complexity

| | LMS | | RLS | |
|---|---|---|---|---|
| | Cmplx. Mult. $(2L+1)$ | Cmplx. Add. $(2L)$ | Cmplx. Mult. $(3L^2+5L+1)$ | Cmplx. Add. $(\frac{3}{2}L^2+\frac{5}{2}L)$ |
| Laguerre Linearzer (T=3) | 9 | 8 | 69 | 34 |
| Laguerre Linearzer (T=5) | 13 | 12 | 139 | 69 |
| Hermite Linearzer (T=3) | 5 | 4 | 23 | 11 |
| Hermite Linearzer (T=5) | 7 | 6 | 43 | 21 |



As shown in table 4.5, the complexity of the LMS algorithm is set by a linear function of the number of coefficients to adapt. For a given polynomial regression order, the number of coefficients to adapt for an Hermite polynomial linearizer is half the number of coefficient of a Laguerre one. Then, the number of complex addition needed to adapt a Hermite linearizer is half the number needed by the Laguerre linearizer, and almost half the number of complex multipliers.

The case of implementing a RLS algorithm is different than the LMS case because it is characterized by a quadratic function of the number of taps to adapt ($L$). Then, the number of operations needed for the case of a Laguerre polynomial linearizer can be three (T=3) or several times larger than the operations needed in the Hermite polynomials case, depending on the order of the polynomial regression.

The complexity of the Training Blocks adapting the two linearizers introduced in this section is also compared on figure 4.4. This figure depicts the number of complex multiplications needed to adapt each linearizer, implementing both the LMS or the RLS algorithms. The regressions order considered range from 1 to 13. For the LMS case, training a Laguerre polynomial linearizer of order 5 requires the same number of complex multiplications than a Hermite polynomial linearizer of order 11. A resembling difference is observed when the RLS algorithm is implemented. Training a Laguerre polynomial linearizer of order 3 needs a number of multipliers near that needed to train an order 7 Hermite polynomial linearizer.

Therefore, using Hermite polynomials to perform a polynomial regression always leads to a complexity reduction. The order of the reduction is then determined by the adaptation method chosen for the Training Block.

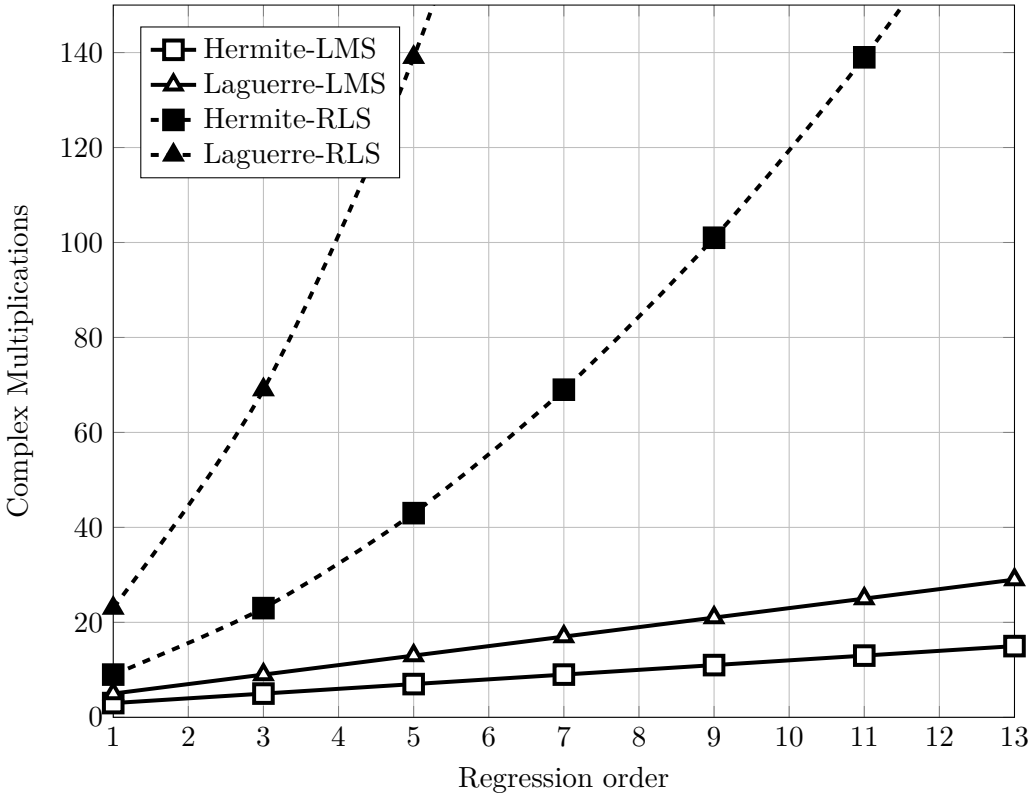



Figure 4.4: Number of complex multiplications vs. Polynomial regression order

It is also interesting to study the performance of a polynomial linearizer implemented as a post-distortion technique. Since the channel is affected by the down-link Gaussian channel noise (AWGN), the received signal is affected by both the PA non-linearity and the noise. Then, the estimation of the coefficients of the polynomial regression can be more or less biased, depending on the noise power. The minimization of the error function characterizing the LMS algorithm and the RLS algorithm must thus be redefined. The error for the case of the LMS algorithm adapting a Laguerre polynomial linearizer is now defined as:

$$J = E|x_t - {y'}_t|^2 = E|x_t - \sum_{j=0}^{T} d_j L_j(x'_t + n_t)|^2. \tag{4.6}$$

For the case of the RLS algorithm, the error is computed as:

$$J = |x_t - {y'}_t|^2 = |x_t - \sum_{j=0}^{T} d_j L_j(x'_t + n_t)|^2. \tag{4.7}$$

In (4.6) and (4.7), $n_t$ represents the Gaussian channel noise (AWGN). Then, as the minimization considers both the non-linear PA output signal $x'_t$ and the noise $n_t$, the precision in the coefficients estimation is also a function of the level of $n_t$. Any error in the estimation of the coefficients makes the channel non-linear, inducing a non-linear ISI in the received symbols.

## 4.2 Polynomial Linearizer combined with Volterra Canceller

The polynomial linearizer reduces the non-linear ISI associated with the received symbols. In order to eliminate the residual non-linear ISI, this work proposes to implement a Volterra Canceller after the linearization, [CLLJ12] and [CLLJ13]. This combined technique is one of the major contributions of this Thesis. The polynomial linearizer and the Volterra Canceller implemented into a receiver are illustrated on figure 4.5.

As the non-linear ISI associated with the linearized received symbols $s'_n$ is weak, the a-priori symbols $\hat{s}_n$ are mapped more precisely. This helps the Volterra Canceller working near the ideal performance presented in [BC05]. Moreover, a less complex canceller can be used since the non-linear ISI is of reduced order and memory.

The adaptation of the polynomial linearizer starts once the synchronization of the receiver is done. The adaptation of the Volterra Canceller starts after the convergence of the polynomial linearizer. The Volterra Canceller is trained, as shown in section 3.3, by means of either the LMS or the RLS algorithms. Once an initial adaptation is done, a periodic refresh of both the polynomial linearizer and the Volterra Canceller blocks is required to track any dynamic change.

One of the challenges related to the implementation of the adaptation methods is the need of training data. Many standards, such as the DVB-S2 one, consider pilots symbols but not as a mandatory case, i.e. the presence of pilots symbols is facultative. Thus, the existence of training data can be a prohibitive factor for the implementation of a post-distortion technique.

## 4.3 Blind adaptation techniques

This section deals with the communication systems which do not guarantee or do not consider at all the presence of training data for the adaptation of the post-distortion

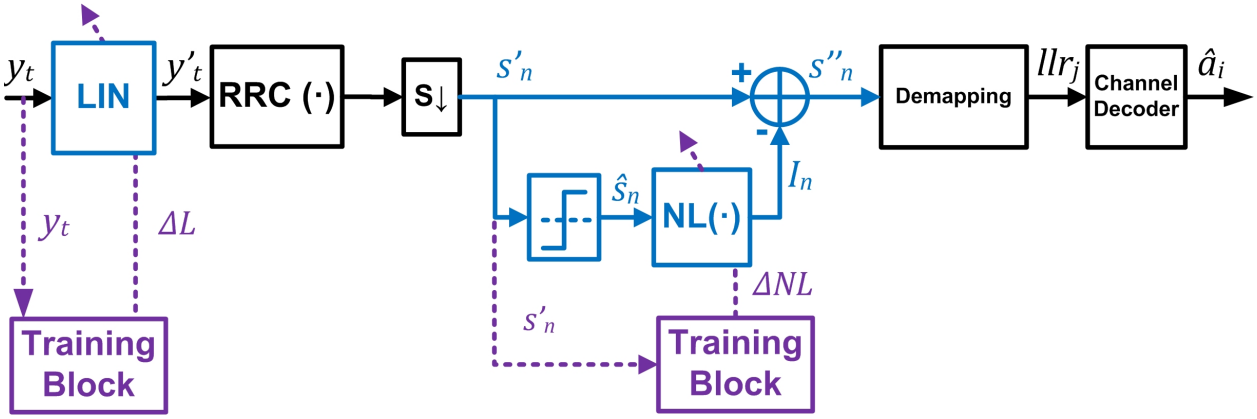



Figure 4.5: Polynomial linearizer with Volterra Canceller implemented into a typical receiver architecture

techniques. Thus, two techniques of self training data generation are proposed thereafter.

### 4.3.1 Channel decoder feed-back training

This subsection only focuses on digital communication systems considering channel coding. As shown in [PS08], the existence of errors in the output bits of a channel decoder can be easily verified. Such a verification is done by multiplying the bits with the parity-check matrix characterizing the code. Then, if an error-free estimation of the transmitted bits can be guaranteed, these bits can be used as the training data needed to perform the adaptation.

This concept has been used in [KHCO10] and [CLLJ12] to feed the Training Blocks with training data. In both cases the channel codes were LDPC codes and the digital communication system was the DVB-S2. In [CLLJ12], the output bits of the channel decoder are fed-backed toward the Training Block as shown on figure 4.6.

Figure 4.6 shows a receiver implementing the combined technique of polynomial linearizer and Volterra Canceller. The output of the Volterra Canceller $s''_n$ is demapped to produce the $llr_i$s. The $llr_i$s are used by the channel decoder to produce bits $\hat{a}_i$. The training symbols $p_n$ are generated by remapping the decoded bits $\hat{a}_i$ onto constellation points. Finally, the constellation points $p_n$, which are a replica of the transmitted symbols $s_n$, are used to adapt the Volterra Canceller.

The training data needed to adapt the polynomial linearizer are also obtained from the channel decoder output. After mapping, symbols $p_n$ are up-sampled and filtered by the digital matched filter. As the resulting signal $z_t$ is a replica of $x_t$ (the input of the non-linear PA), $z_t$ can be used as the data needed by the Training Block adapting the Hermite polynomial linearizer.

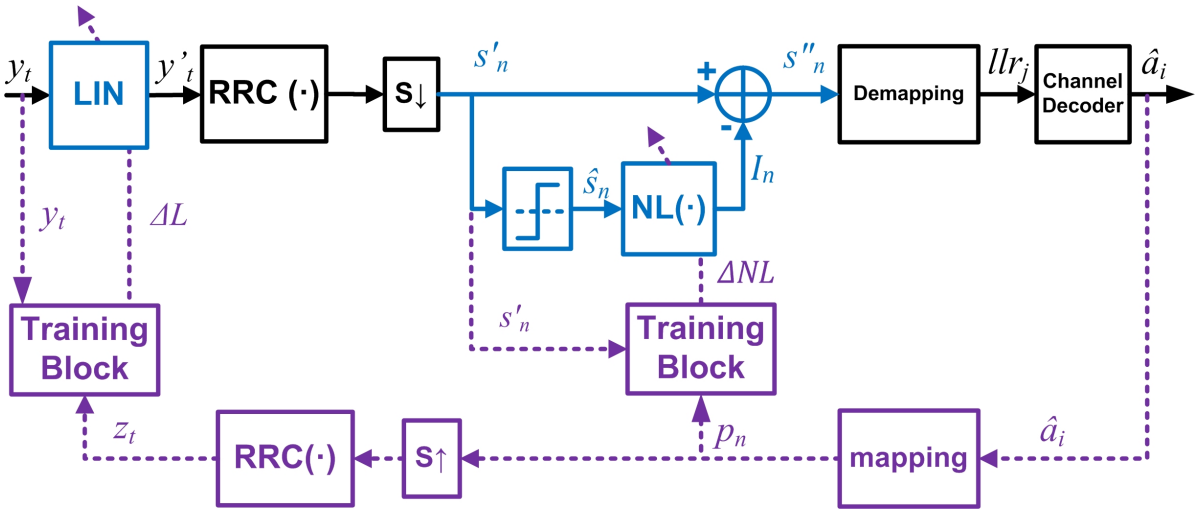



Figure 4.6: Generation of training data from the channel decoder output

A weakness associated with this method is that it can be only implemented into encoded communication systems. Another issue shown by the generation of training data from the channel decoder output is the complexity demanded. The first complexity issue is detected in the direct mapping of bits $\hat{a}_i$ onto constellation points $p_i$ shown on figure 4.6. In fact, such a mapping can only be done if the codes implemented are systematics codes. Such codes are characterized by having a codeword composed of two blocks. The first block consists of a set of unencoded transmitted bits, while the second block is composed of the redundancy [PS08]. In the case of non-systematics codes, a channel encoder must be added into the feed-back path to generate symbols $p_n$. Moreover, as many communications systems also considers symbol and/or bit interleaving, if an interleaving is performed in the receiver, the feed-back path also needs to include the interleaving blocks. Therefore, if the receiver considered is a commercial/low-cost one, the implementation of an extra interleaver or an extra encoder can prevent the use of this method.



The addition of any extra block into the feed-back path induces an extra delay between the inputs of the Training Block. As shown in section 3.3.2, the Training Block of a Volterra Canceller works by computing the error between $s'_n$ and $p_n$:

$$e_n = |s'_n - \sum_{i=-M}^{M} b_i p_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} b_{i,j,k} p_{n+i} p_{n+j} p^*_{n+k}|. \tag{4.8}$$

Thus, due to the existence of the delay between $s'_n$ and $p_n$, a storage of the values of $s'_n$ must be done. The amount of values of $s'_n$ to be stored is proportional to the delay of the feed-back path. Therefore, the larger the number of blocks added in the feed-back path is, the larger the size of the memory needed is.

The amount of memory needed to carry out the adaptation of the compensation systems can be even larger if the channel codes are based on turbo codes. As mentioned in section 3.3.3, a turbo decoder works iteratively. The number of iterations executed by the decoder varies depending on the impairments associated with the received signal, the level of noise and the codes implemented. As an example, the DVB-S2 standard defines the systems performance for a maximum number of iterations executed by the turbo decoder (LDPC decoder) equal to 50. Then, the large number of iteration leads to delays growing exponentially with the number of iterations. Therefore, the memory needed for the Training Block grows likewise.

### 4.3.2 Direct decision feed-back training

As mentioned in the precedent subsection, the additional memory required by the channel decoder feed-back training can prevent its implementation. However, it is possible to carry out a different blind adaptation by means of a direct decision method. The principle of a this method is to get the training data directly from the compensated symbols. Therefore, the application of this method avoids the delay induced by the channel decoder and the other blocks added to the feed-back path.

The direct decision method has been proposed in [PS08] and [TTD00] to adapt equalizers independently of the application case. The implementation of this technique into a

receiver including a polynomial linearizer with a Volterra Canceller is one of the major contribution of this Thesis. The combined post-distortion technique performing direct decision feed-back training is illustrated on figure 4.7.

As seen on figure 4.7, the output symbols of the Volterra Canceller are mapped by the Direct Decision block onto estimated symbols $\hat{p}_n$. These symbols are the estimation of the symbols $p_n$ obtained with the channel decoder feed-back training. The simplest method of estimation of $\hat{p}_n$ is the hard-decision mapping, which maps the output of the Volterra Canceller onto minimum distance constellation points. Otherwise, to obtain better estimations of $\hat{p}_n$, other direct decision methods are presented in [PS08]. Among the different methods, the most widely used one is the Godard method. This method estimates the transmitted symbols from the output of a Volterra Canceller as follows:

$$DD(s''_n) = \frac{s''_n}{\tilde{s}_n}(|\tilde{s}_n| + K|s''_n| - |s''_n|^3). \tag{4.9}$$

Where the function $DD(\cdot)$ represents the Godard function. The value of $\tilde{s}_n$ is the a-posteriori mapping of $s''_n$ onto minimum distance constellation points. For the case described on figure 4.7, $\tilde{s}_n$ corresponds to $\hat{p}_n$. Finally, factor $K$ is defined as:

$$K = \frac{E|s_n|^4}{E|s_n|^2}. \tag{4.10}$$

Once the training symbols $\hat{p}_n$ are computed, they can be used to train the Volterra Canceller. The training signal $\hat{z}_t$ is obtained by applying up-sampling and filtering, as done for the channel decoder feed-back training case.

In terms of timing, there are only two blocks inducing an additional delay. Such blocks are the Direct Decision block and the digital matched filter. Then, the latency induced by the channel decoder and any other block needed to carry out the channel decoder feed-back training is avoided. Therefore, the memory required for the implementation of

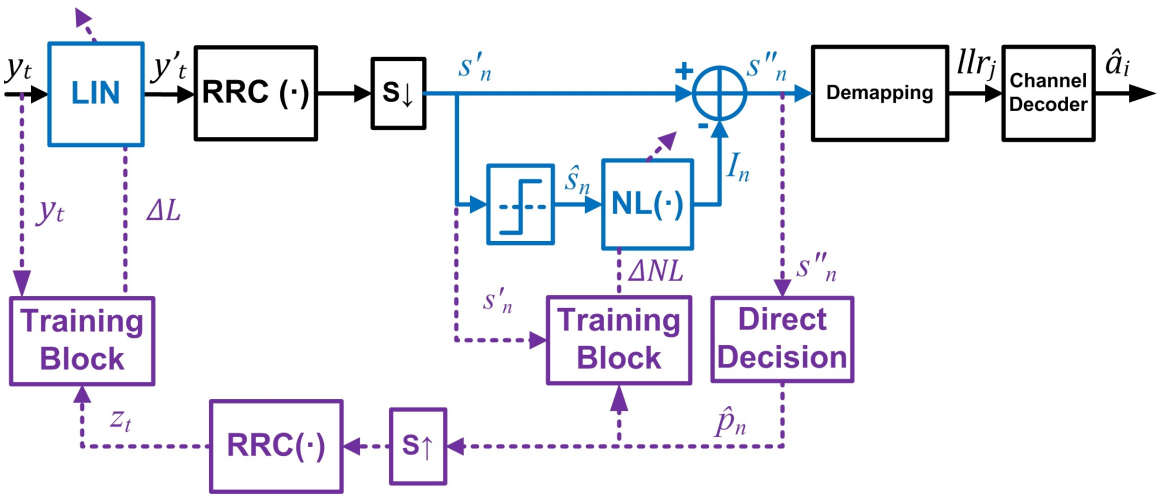



Figure 4.7: Generation of training data from direct decision mapping

this method is reduced.

## 4.4 Simulations results

The performance of the polynomial linearizer with Volterra Canceller and the comparison of this technique with the stand-alone Volterra Canceller and the Volterra ZFE are analyzed in this section. The performance of each post-distortion technique are assessed from computer simulations. For the simulations, the different compensation techniques have been modelled in Matlab and then integrated into the Matlab model of a typical DVB-S2 receiver.

The DVB-S2 system considers 8-PSK digital modulation (mandatory case) and digital RRC filters with a roll-off factor equal to 0.35. The HPA has been modelled from a Saleh's model. The Saleh's model parameters were obtained from curve fitting simulations based on the DVB-S2 standard input-output power relationship presented in [DS06]. The HPA IBO considered in simulations is IBO=0 dB, which is the worst case.



The combined non-linear ISI compensator based on Laguerre Polynomial linearizer and Volterra Canceller implements an order 3 polynomial regression, i.e. the Laguerre polynomials considered are of order 0, 1, 2 and 3. The Volterra Canceller performs ten multidimensional combinations of order 1 and 3 and memory M=3, [CLLJ12]. The adaptation of the four coefficients of the Laguerre polynomial linearizer and the ten coefficients of the Volterra Canceller is done from a RLS algorithm. A simplified linearizer is implemented using order 1 and 3 Hermit polynomials to yield best performance [CLLJ13]. This linearizer and its combined Volterra Canceller characterized by ten multidimensional combinations of order 1 and 3 and M=3, are both trained from the LMS algorithm. The RLS and LMS Training Blocks both require 4000 training symbols each to obtain the Laguerre/Hermite polynomial linearizer's and the Volterra Canceller's coefficients vectors. The stand-alone Volterra Canceller trained from a RLS algorithm is composed of twelve multidimensional convolutions of order 1 and 3 and M=3. Finally, the Volterra ZFE considers eighteen multidimensional combinations of order 1 and 3, memory M=3 and trained with a RLS algorithm. For each case, the training data is obtained with the direct decision feed-back training introduced in section 4.3.2.

The performance of the non-linear ISI compensation techniques are assessed using the average number of LDPC iterations (as defined in [DS06]) needed to decode with no error the transmitted message, for a given $E_S/N_0$. Matlab simulations have demonstrated that an accurate value of the average number of LDPC iteration can be obtained from 10000 LDPC decoder executions. When compared to usual assessment, such as BER, SER or PER, this metric offers the advantage of saving considerable simulation times [CLLJ12] and [CLLJ13]. The reduction of the simulation times can be exemplified taking as reference the DVB-S2 standard. As described in [DS06], the performance of a DVB-S2 receiver are stated in terms of PER vs $E_b/N_0$ for a quasi error free (QEF) transmission, i.e. PER=$10^{-7}$. The packets are composed of 188 bytes. A PER=$10^{-7}$ means that there is one packet affected by errors among 10,000,000 received packets. Then, in order guarantee

an accurate PER, it is necessary to detect a few packets with error. Thus, taking 50 as the number of packets with error detected, the number of bits needed to compute the performance of the DVB-S2 receiver is equal to:

$$Nbit = (188 \times 8 \times (2 - CR_{BCH})) \times (2 - CR_{LDPC}) \times 10,000,000 \times 50. \quad (4.11)$$

Where $CR_{LDPC}$ and $CR_{BCH}$ are the LDPC and BCH code rates detailed in appendix A. For calculus simplification, (4.11) does not considers the header and the pilots characterizing a DVB-S2 stream (appendix A). Considering a $CR_{LDPC} = 8/9$ for which the $CR_{BCH} = 0.997$, the value of $Nbit$ is larger than $8.3\text{x}10^{11}$. This quantity of bits represents about $1.3\text{x}10^{7}$ executions of the LPDC decoder. Thus the simulation time is divided by 1300. This difference in the simulation time is due to the PER=$10^{-7}$ demanded by the standard, i.e. the simulation time is only divided by 13 for PER=$10^{-5}$.

In order to compare the effectiveness of the different non-linear ISI compensation techniques, the performance for a DVB-S2 system with a code rate equal to 8/9 are shown on figure 4.8 [CLLJ12]. All the values of $E_S/N_0$ characterizing each curve of figure 4.8 must guarantee the convergence of the LDPC decoder. Curve "LPL+VC" presents the performance of the DVB-S2 receiver compensated with the Laguerre polynomial linearizer and the Volterra Canceller. Curve "HPL+VC" illustrates the performance of the linearizer based on Hermite polynomials and trained with a LMS algorithm. This figure also shows the results obtained in the ideal case, i.e. the HPA has a linear behaviour (curve "Linear HPA"). Compensating the non-linear ISI with a stand-alone Volterra Canceller is also considered (curve "VC"). The case of the ideal Volterra Canceller introduced in [BC02] is shown as curve "Ideal VC". The performance of a Volterra ZFE is depicted. Finally, performance for a system with uncompensated non-linear ISI are also shown (curve "Non-linear HPA").

Figure 4.8 shows that the combined technique of polynomial linearizer and Volterra Canceller presents performance close to the ideal Volterra Canceller. For any given average number of LDPC iterations, the performance of the systems "HPL+VC" and "LPL+VC" are almost similar. As explained before, this is due to the fact that the RLS and the LMS algorithms have equal performance in scenarios with values of $E_S/N_o$ near or lower than 10 dB, and both polynomial regressions have the same order. By analyzing figure 4.8, it can be inferred that the combined technique outperforms the stand-alone Volterra Canceller and the Volterra ZFE. For a number of average iterations equals to 23, the gain of the Volterra ZFE with respect to the uncompensated case is 0.18 dB. The Gain of the stand-alone Volterra Canceller is 0.28 dB. A gain of 0.42 dB is achieved by the Laguerre polynomial linearizer with Volterra Canceller (RLS training). The Hermite polynomial linearizer with Volterra Canceller attains a gain equal to 0.43 dB. Finally, the gain of an ideal Volterra Canceller is 0.465 dB. The loss between the uncompensated non-linear case and the ideal case is 0.72 dB.

In order to compare the performance of the different compensation techniques working with others levels of $E_S/N_0$, figures 4.9, 4.10, 4.11, 4.12 and 4.13 show the performance for the cases considering LDPC codes rates equal to 9/10, 5/6, 4/5, 3/4 and 2/3.

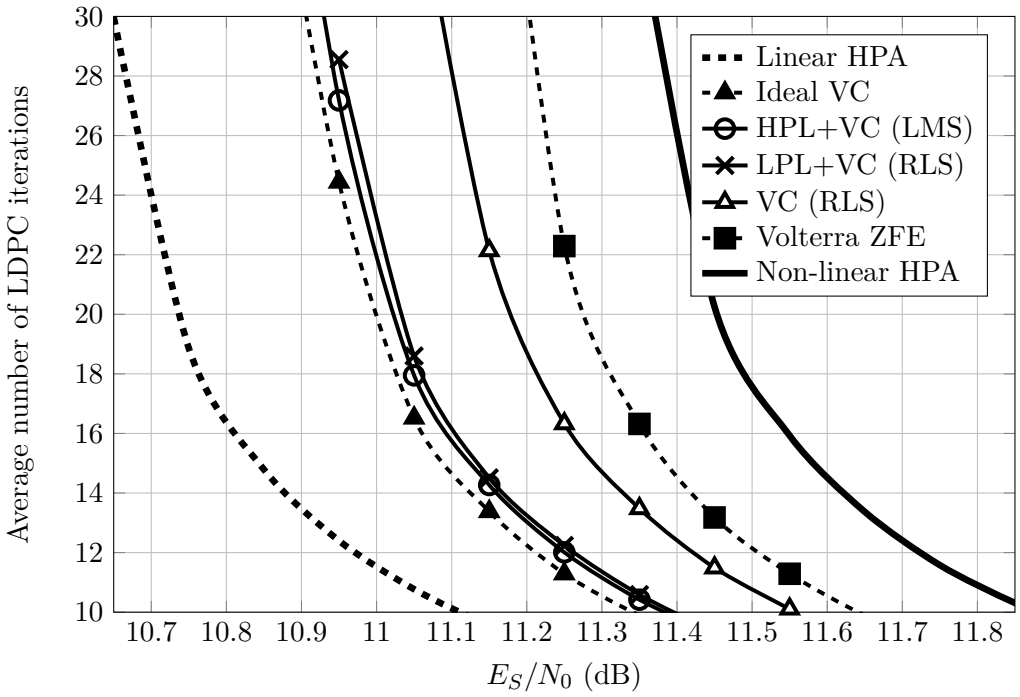



Figure 4.8: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=8/9

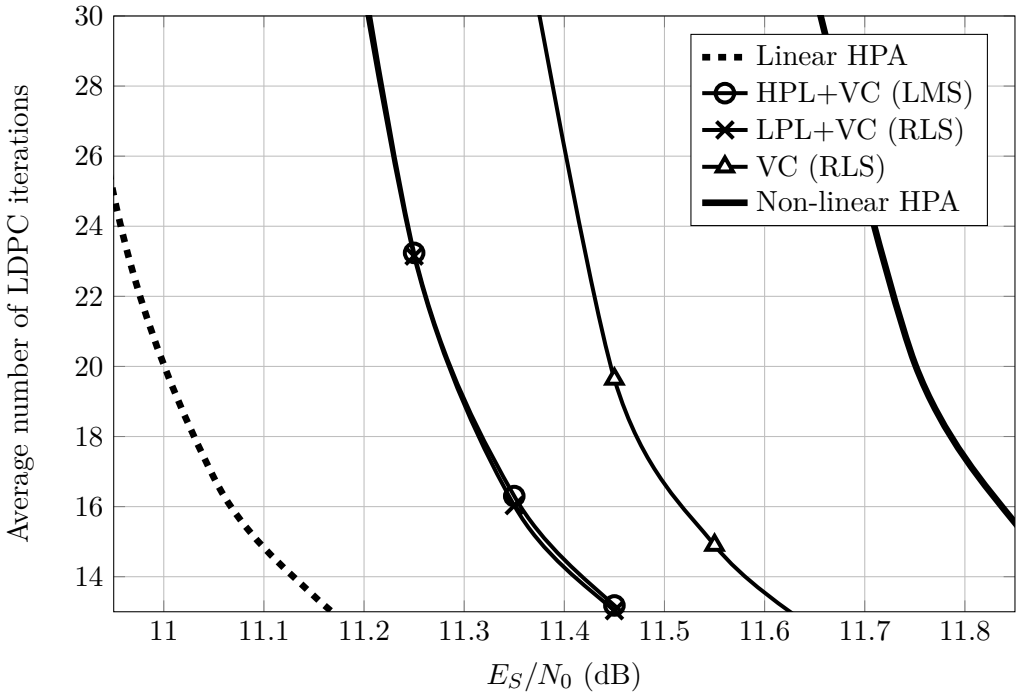



Figure 4.9: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=9/10

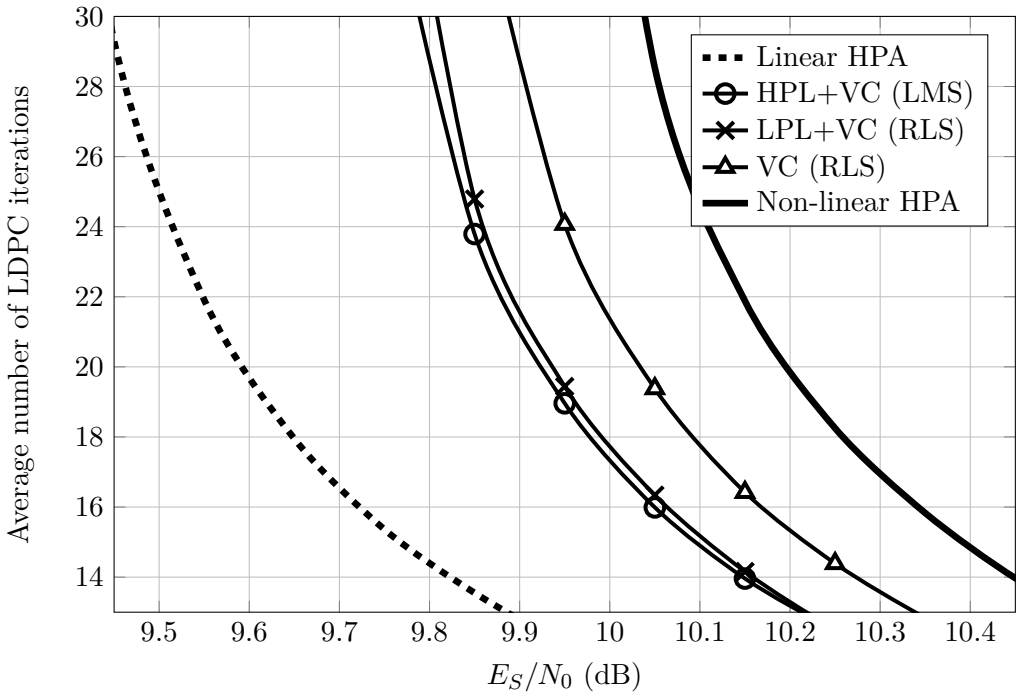



Figure 4.10: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=5/6

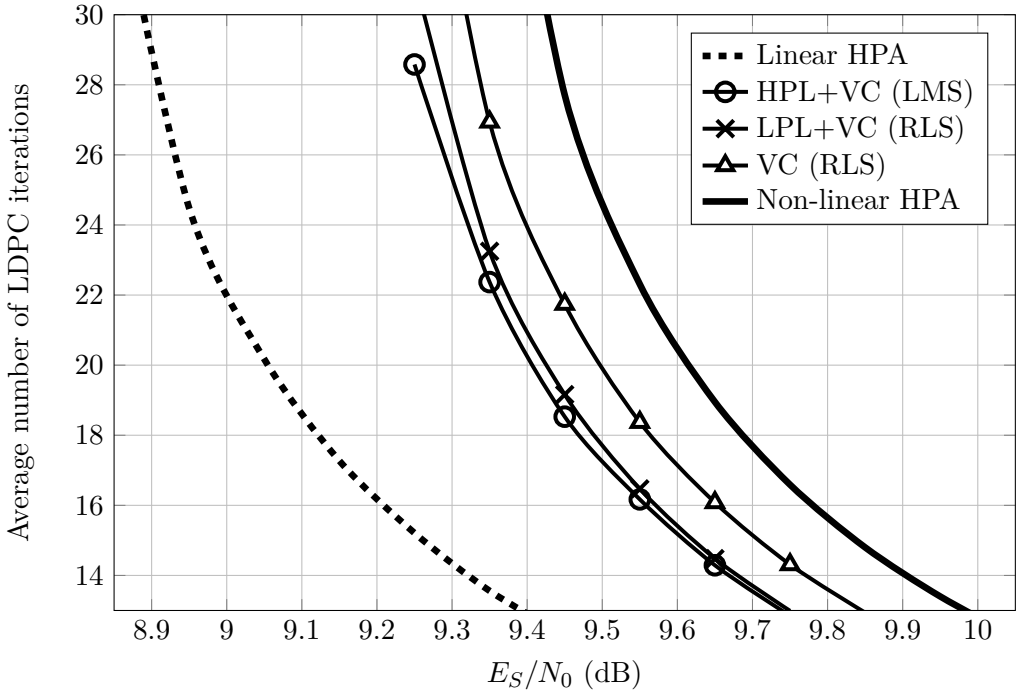



Figure 4.11: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=4/5

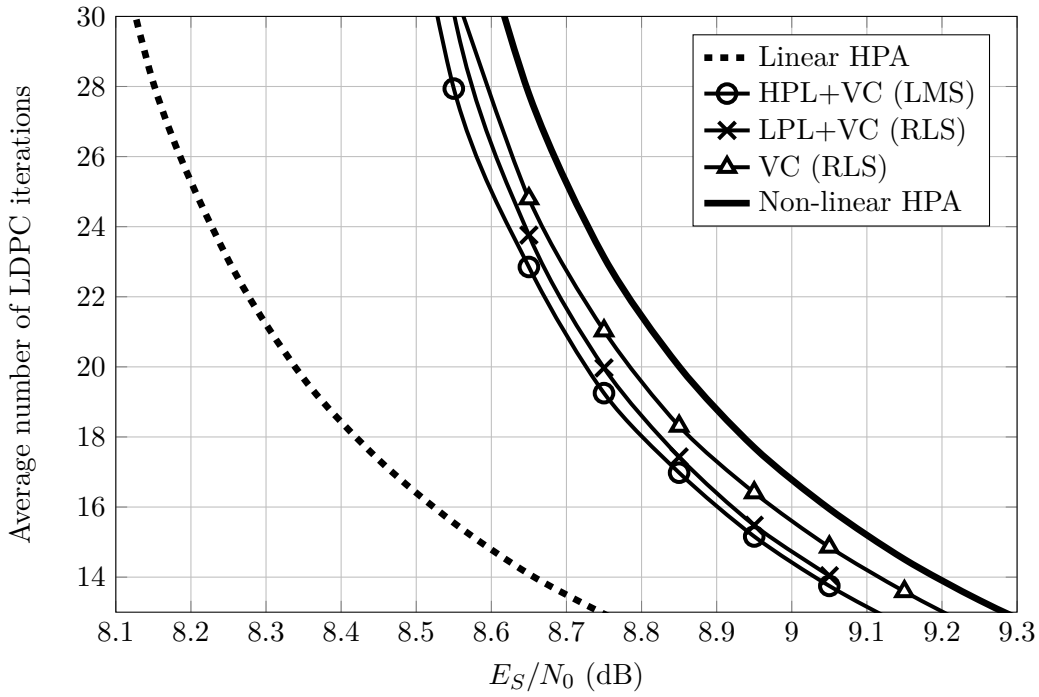



Figure 4.12: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=3/4

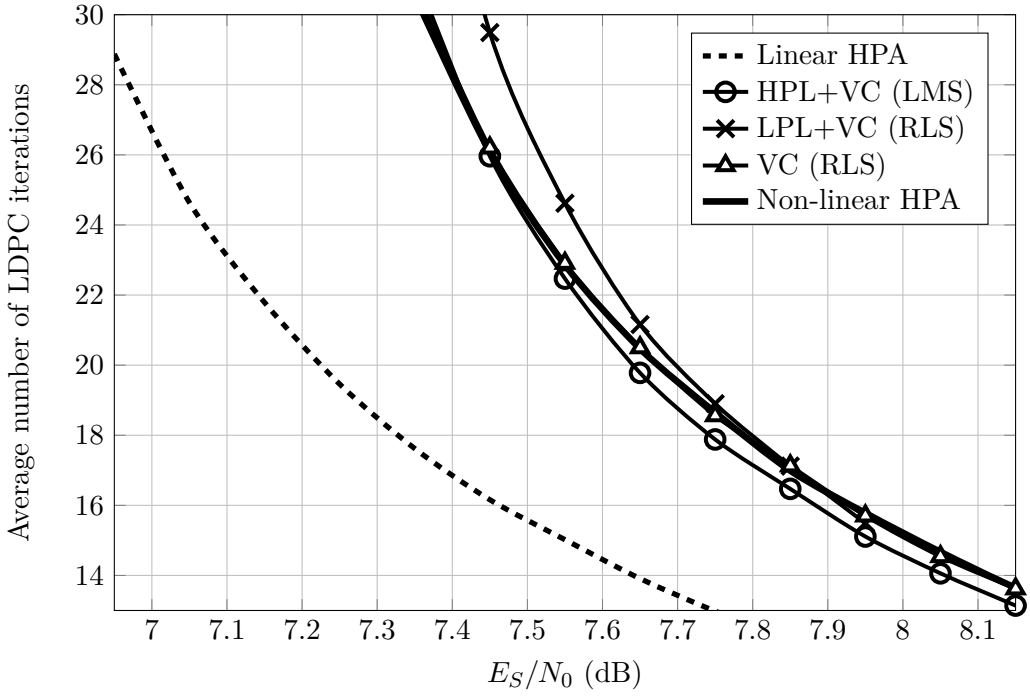



Figure 4.13: Performance of the DVB-S2 receiver: Average number of LDPC iterations for CR=2/3

Several conclusions can be drawn from the analysis of figures 4.9, 4.8, 4.10, 4.11, 4.12 and 4.13. The first conclusion is related to the correlation between the code rate and the degradation of the receiver's performance. As seen on figure 4.9, the degradation of an uncompensated receiver (curve non-linear HPA) with respect to the linear HPA case is of 0.76 dB (CR=9/10). In contrast, the minimal degradation is found for the curve shown on figure 4.13, with a value of 0.44 dB. Therefore, the lower the code rate value is, the lower the degradation. As the degradation is reduced, then the gain obtained from a post-distortion technique is also reduced. The gain of the Volterra Canceller is also degraded by the low $E_S/N_0$ associated with small code rates. It is clearly seen on figures 4.12 and 4.13 (CR=3/4 and 2/3). This degradation is induced by the a-priori mapping which accuracy is a function of the $E_S/N_0$.

Another important conclusion is that the Hermite polynomial linearizer adapted with an LMS method presents better performance than the the version based on Laguerre polynomials and RLS adaptation. This fact is valid for any code rate value [CLLJ13].



In addition, the Hermite polynomial linearizer combined with a Volterra Canceller outperforms the stand-alone Volterra Canceller in terms of average number of LDPC iterations for all the code rates except for the case of CR=2/3, for which both have similar performance. Such degradations in the performance of the polynomial linearizer are due to the small $E_S/N_0$ characterizing the limits of the code rate considered. This behaviour is explained by the concept introduced in section 4.1.2. In fact, in order to assure a minimal error, the linearizer is adapted by means of a trade-off between the non-linear channel and the noise level.

An analysis of the behaviour of each non-linear ISI compensation technique for different values of code rates is depicted simultaneously on figure 4.14. Indeed, this figure shows the degradation in dB of the different compensation techniques and the uncompensated non-linear ISI case with respect to the Linear HPA case. Each simulation point of figure 4.14 is obtained for an average of 23 LDPC iterations. As seen on figure 4.14, the performance in terms of degradation shown on curve "HPL+VC" are similar to that of "LPL+VC" for high code rates (5/6, 8/9 and 9/10). Combined with a Volterra Canceller, the linearizer implementing Hermite polynomials outperforms the one implementing the Laguerre polynomial linearizer for lower code rates. This performance improvement have two origins associated with the training algorithm implemented and the number of polynomials considered by the linearizer. As stated in chapter 2, the performance of the RLS and the LMS algorithms are similar at $E_S/N_0$ near 10 dB, i.e. no performance degradation are induced by the adaptation method. With respect to the number of polynomials, the accuracy and the convergence speed of the training methods are functions of the number of coefficients to train (the number of coefficients is equal to the number of polynomials). Thus, the less-complex architecture of the Hermite polynomial linearizer improves the estimation of the non-linear channel. This implies that a reduction in the degradation of the receiver performance.

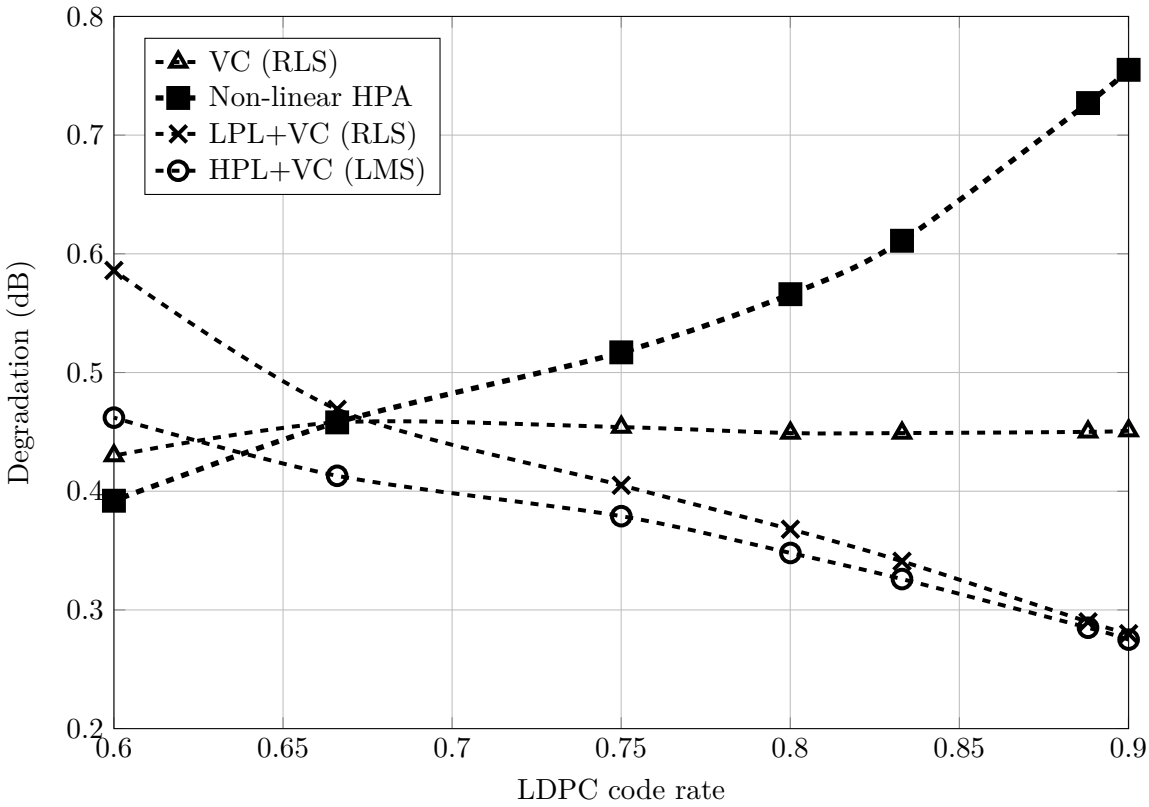



Figure 4.14: DVB-S2 receiver degradation with respect to linear HPA case vs. LDPC code rate

## 4.5 Summary

This chapter has introduced the concept of post-distortion linearization to avoid the non-linear ISI. The linearizer based on orthogonal polynomial regression can efficiently estimate the inverse of the non-linear PA transfer function. Thus, two different linearizers implemented from Laguerre and Hermite polynomials regressions have been introduced. The linearizer showing the lowest system complexity is the Hermite polynomial linearizer.

This chapter has also proposed a combined technique where a Hermite polynomial linearizer works together with a Volterra Canceller. The Hermite polynomial linearizer reduces the memory and the order of the non-linear ISI. This helps the Volterra Canceller working near ideal performance.

In addition, this chapter has proposed two techniques of self-training data generation. Thanks to the implementation of those techniques, the Volterra Canceller combined with a Hermite polynomial linearizer can be integrated into any digital communication systems without independently of the application case.

Finally, this chapter has compared the performance of the technique based on Hermite

polynomial linearizer combined with a Volterra Canceller with others techniques proposed in the state-of-the-art. The performance have been obtained from computer simulations of a DVB-S2 system. This chapter has also presented a novel metric determining the performance of the post-distortion techniques. This metric assesses the performance by means of the average number of LDPC iterations needed to decode with no error the transmitted data. This significantly reduces the simulation time compared to traditional metrics such as BER. The simulation results have shown that the proposed technique outperforms the stand-alone Volterra Canceller and the Volterra ZFE, showing performance near that of the ideal Volterra Canceller. It has been also shown that the combined technique carried out from Hermite polynomial regression outperforms the technique based on Laguerre polynomials regression, instead of the strong complexity reduction.

4

# 5 Implementation of the Polynomial Linearizer with Volterra Canceller

The innovative non-linear ISI compensation technique based on the combination of a polynomial linearizer and a Volterra Canceller has been presented in chapter 4. The combined Volterra Canceller has been modelled with the architecture described in section 3.3.2. The linearizer implementing the Hermite polynomial regression shows better trade-off between system complexity and gain of $E_S/N_0$ than others post-distortion methods. Computer simulations of a DVB-S2 system impaired by HPA non-linearity have validated that the combined technique outperforms the state-of-the-art in post-distortion techniques.

This chapter presents the implementation of the combined technique, designed for commercial DVB-S2 receivers. Therefore, the digital designs of both the Volterra Canceller and the polynomial linearizer are focused on the reduction of the system complexity. The polynomial linearizer is implemented from Hermite polynomial regression. Hence, the Volterra Canceller is optimised for a 8-PSK digital modulation, which is the mandatory case for the standard. Both the Hermite polynomial linearizer and the Volterra Canceller are trained from an LMS algorithm.

The chapter is organised as follows. The first section presents a modified architecture for the Volterra Canceller adapted to 8-PSK digital modulation. This section also shows the implementation of the Volterra Canceller. The architecture and implementation of the Hermite polynomial linearizer are described in section 5.2. Finally, section 5.3 summarizes this chapter.

## 5.1 Architecture and implementation of the Volterra Canceller

The architecture of the Volterra Canceller presented in the literature has been introduced in section 3.3.2. Such an architecture, flexible and adapted to any digital modulation, is described on figure 3.9 and in (3.5). For the sake of clarit, (3.5) is reintroduced in this section:

$$I_n = \sum_{i=-M}^{M} b_i \hat{s}_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} b_{i,j,k} \hat{s}_{n+i} \hat{s}_{n+j} \hat{s}^*_{n+k} - C_n. \tag{5.1}$$

Where $I_n$ represents the non-linear ISI, $M$ is the memory order, $b_i$s are the Volterra coefficients, $\hat{s}_n$ is the a-priori mapping estimation of the received symbols and $C_n$ is the centroid value corresponding to $s'_n$, the received symbol. However, the architecture of a typical Volterra Canceller is not optimal in terms of complexity for constant envelope modulations, such as N-PSK. Indeed, when an 8-PSK modulation is used, the computation and subtraction of centroids are too complex to be implemented as such in the Volterra Canceller and do not add significant performance gain. Hence, it is convenient to avoid any operation with the centroid values. In fact, the centroid corresponding to $s'_n$ can be omitted in (5.1) if the term $b_0 \hat{s}_n$ is not included in the Volterra series regression. This simplification has been validated by SystemC simulations. In the simulations, two Volterra Cancellers (one implementing the simplified architecture and the other considering the centroid values) have compensated for the non-linear ISI into a DVB-S2 receiver. The results in terms of BER do not infer a performance degradation for a Volterra Canceller omitting the centroid values.

In consequence, the Volterra Canceller's architecture implemented for receivers based on 8-PSK digital modulation presents some changes with respect to the typical architecture presented on figure 3.9, as illustrated on figure 5.1.

As seen on figure 5.1, the modified architecture of the Volterra Canceller is composed of three blocks: a mapping and shift-register block, block NL($\cdot$) and a LMS training block. The mapping block carries out the a-priori mapping of the received symbols onto minimum distance constellation points. After mapping, a shift-register delays the incoming symbols in order to get the memory necessary to compute the non-linear ISI term $I_n$. Block NL($\cdot$) estimates the values of $I_n$ and and $\check{s}_n$, both obtained from the mapped symbols $\hat{s}_i$ and the vector of Volterra coefficients $\mathbf{vt_m}$, where $m$ is the instant (different from $n$) of the estimation. The LMS training block adapts vector $\mathbf{vt_m}$ from the vector whose elements are the Volterra combinations $\mathbf{vc_n}$ (computed by block NL($\cdot$)) and the estimation error $e_n$.

The architecture, the functionality and the digital design of each block of the architecture detailed on figure 5.1 are described in the next sections.

### 5.1.1 The Mapping block and the shift-register

This block executes two different tasks. Firstly, an a-priori-mapping of the received symbols is estimated. Secondly, thanks to a shift-register, this block save the $M$ last

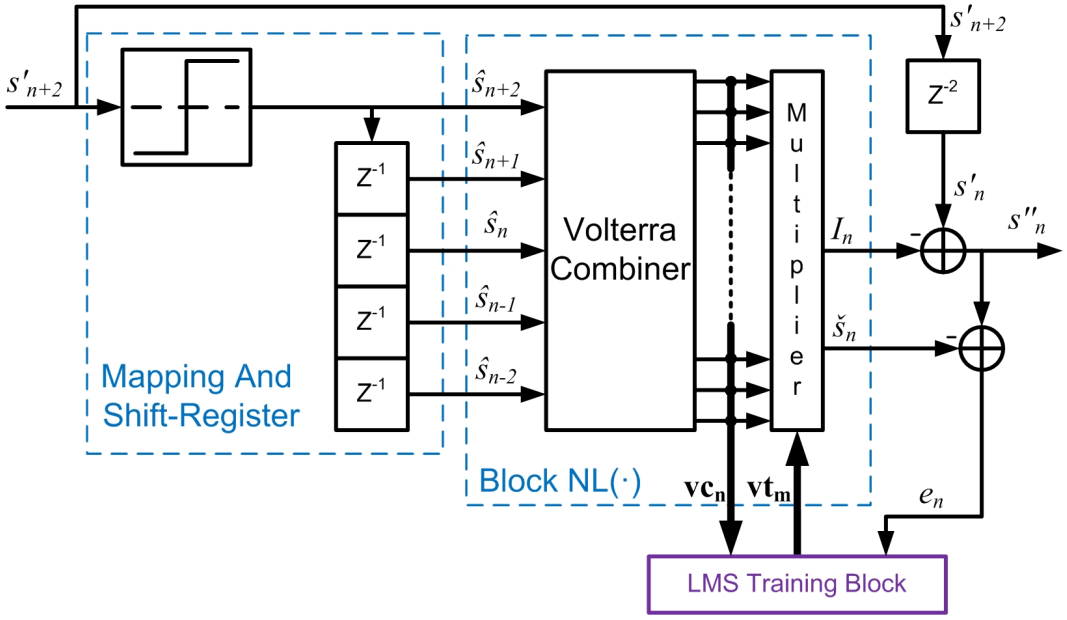



Figure 5.1: Blocks constituting the Volterra Canceller implementation

a-priori estimations needed to estimate the term $I_n$.

To carry out the a-priori mapping it is necessary to define the type of constellation implemented and then the value associated with each constellation point. The values of each one of the eight 8-PSK constellation points are defined on figure 5.2. Each symbol is encoded on three bits ($\mathbf{o_0}$=000, $\mathbf{o_1}$=001, $\cdots$, $\mathbf{o_7}$=111).

The mapping of the received symbols onto constellation points can be done in two different manners. The first method is carried out from the comparison between the phase of the received symbol and the phase of the constellation points surrounding the received symbols. To illustrate this method an example is taken. The example assumes that a received symbol $s'_n$ is located in the first quarter, figure 5.3.

The considered received symbol $s'_n$ with phase $\theta_{s'}$ is located between constellation points $\mathbf{o_7}$ and $\mathbf{o_0}$. To compare the phases values between $s'_n$ and the adjacent constellation points it is necessary to define three regions in the quarter, as shown on figure 5.3. Such regions are bounded by angles $\theta_1$ and $\theta_2$. The values of both angles are computed as half the phase between two consecutive constellation points. The values of $\theta_1$ and $\theta_2$ are:

$$\theta_1 = \frac{\angle(\mathbf{o_1}) - \angle(\mathbf{o_0})}{2} + \angle(\mathbf{o_0}) = 67.5°. \tag{5.2}$$

$$\theta_2 = \frac{\angle(\mathbf{o_0}) - \angle(\mathbf{o_7})}{2} + \angle(\mathbf{o_7}) = 22.5°. \tag{5.3}$$

Where $\angle(\mathbf{o_1})$ represents the angle of constellation point $\mathbf{o_1}$.

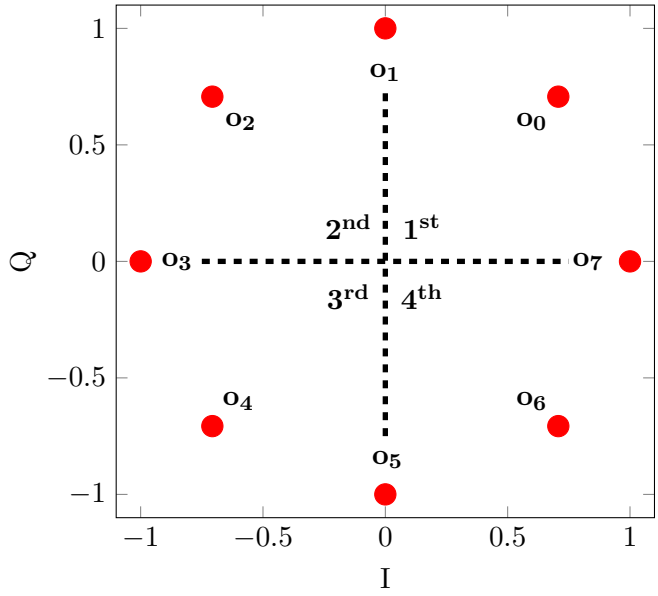



Figure 5.2: 8-PSK constellation for the Volterra Canceller

Once the regions of the quarter are defined, the mapping is done by comparing the tangent value of the angles defining the regions with the tangent of the phase of $s'_n$. The tangent values of $\theta_1$ and $\theta_2$ are:

$$\tan(\theta_1) = \tan(67.5^\circ) = 2.414. \tag{5.4}$$

$$\tan(\theta_2) = \tan(22.5^\circ) = 0.414. \tag{5.5}$$

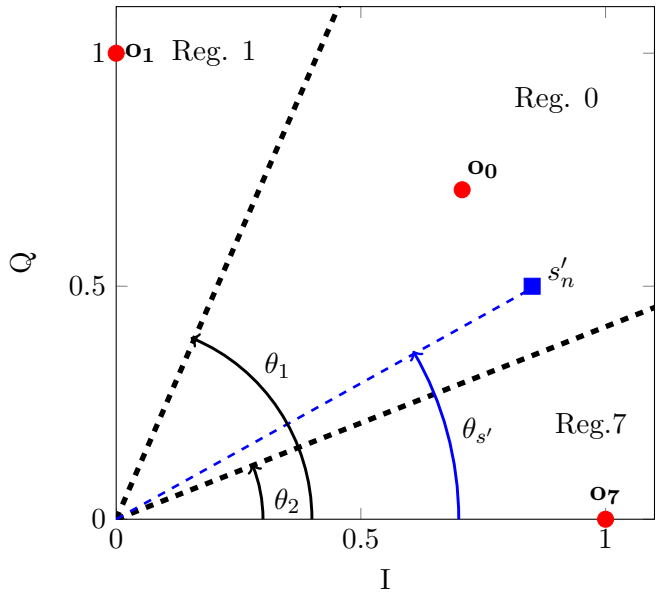



Figure 5.3: A-priori mapping from phase comparison

And the tangent of the phase of $s'_n$ is defined as:

$$\tan(\theta_{s'}) = \frac{s'_{Q,n}}{s'_{I,n}}. \tag{5.6}$$

Where $s'_{I,n}$ and $s'_{Q,n}$ are the real and imaginary parts of $s'_n$. Each part is quantified on $\Gamma_a$ bits.

After defining the values of the tangents, the decision of the a-priori mapping of $s'_n$ onto its respective constellation point is done as shown in table 5.1.

Table 5.1: Criterium of decision for a-priori mapping from phase comparison

| Condition | Decision |
|---|---|
| $\frac{s'_{Q,n}}{s'_{I,n}} < 0.414$ | $\hat{s}_n = \mathbf{o_7}$ |
| $0.414 \leq \frac{s'_{Q,n}}{s'_{I,n}} \leq 2.414$ | $\hat{s}_n = \mathbf{o_0}$ |
| $\frac{s'_{Q,n}}{s'_{I,n}} > 2.414$ | $\hat{s}_n = \mathbf{o_1}$ |

As seen in table 5.1, the a-priori mapping based on phase comparison requires a $\Gamma_a$-bit division between two values ($s'_{Q,n}$ and $s'_{I,n}$). Such a divider is a complex block to avoid if the design is focused on the minimization of the complexity. The divider can be replaced by adder blocks if the a-priori mapping is done by comparing the error between $s'_n$ and each adjacent constellation point instead of the phase comparison. This error is determined separately for axis I and axis Q, as illustrated on figure 5.4.

On figure 5.4, the computation of the error is done with respect to $\mathbf{o_0}$ and $\mathbf{o_7}$ points. In consequence, the errors computed are:

$$E0_I = s'_{I,n} - \mathbf{o_0}_I. \tag{5.7}$$

$$E7_I = s'_{I,n} - \mathbf{o_7}_I. \tag{5.8}$$

$$E0_Q = s'_{Q,n} - \mathbf{o_0}_Q. \tag{5.9}$$

$$E7_Q = s'_{Q,n} - \mathbf{o_7}_Q. \tag{5.10}$$

Then, the a-priori mapping estimation of $\hat{s}_n$ is done as detailed in table 5.2.

As stated in (5.7), (5.8), (5.9) and (5.10), the a-priori mapping method based on distance error requires only four $\Gamma_a$-bit subtraction. This method is less complex than that based on phase comparison, which requires a $\Gamma_a$-bit division, carried out from $\Gamma_a$-1 $\Gamma_a$-bit additions. Instead of its reduced complexity, the a-priori mapping based on distance comparison shows similar bahaviour as the method based on phase comparison. This has been verified from SystemC/VHDL simulations. Therefore, the technique based on

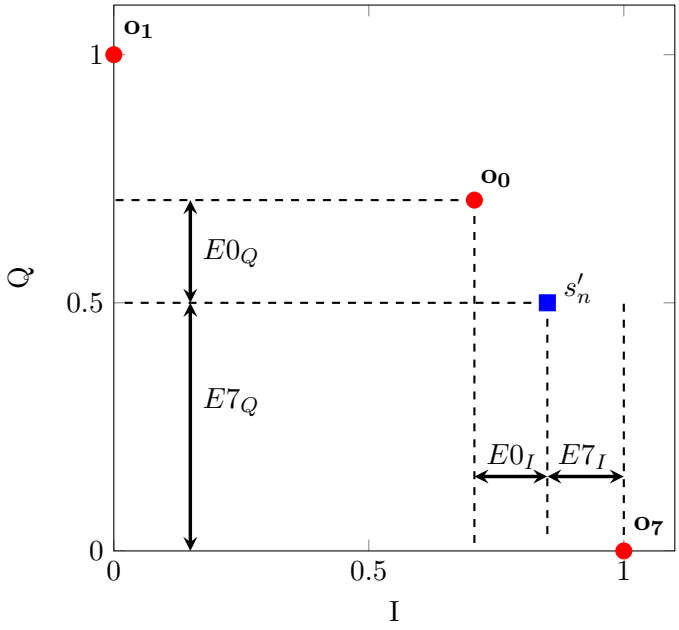



Figure 5.4: A-priori mapping from comparison of distances

Table 5.2: A-priori mapping decision from distance comparison

| Condition | Decision |
|---|---|
| $E0_I < E7_I, E0_Q < E7_Q$ | $\hat{s}_n = \mathbf{o_0}$ |
| $E0_I \geq E7_I, E0_Q \geq E7_Q$ | $\hat{s}_n = \mathbf{o_7}$ |
| $E0_I < E7_I, E0_Q \geq E7_Q, E0_I < E7_Q$ | $\hat{s}_n = \mathbf{o_0}$ |
| $E0_I < E7_I, E0_Q \geq E7_Q, E0_I \geq E7_Q$ | $\hat{s}_n = \mathbf{o_7}$ |
| $E0_I \geq E7_I, E0_Q < E7_Q, E0_Q < E7_I$ | $\hat{s}_n = \mathbf{o_0}$ |
| $E0_I \geq E7_I, E0_Q < E7_Q, E0_Q \geq E7_I$ | $\hat{s}_n = \mathbf{o_7}$ |

distance comparison is the one implemented in this work.

The number of shift blocks required in the shift register is proportional to the memory of the Volterra Canceller. Consequently, if the combinations of the Volterra Canceller are computed from symbols in the interval $[\hat{s}_{n-M}\text{-}\hat{s}_{n+M}]$, the number of shift blocks needed is $2M$. In the example of figure 5.1, the value of $M$ is two, which leads to four shift registers. The complexity of each shift-block depends on the number of bits quantifying $\hat{s}_{n-i}$. In the case of 8-PSK constellations, the symbols are encoded with three bits. Therefore, each shift-block is composed of three flip-flops.

### 5.1.2 The Volterra Combiner

Placed inside block $NL(\cdot)$ of the Volterra Canceller, figure 5.5, the task of the Volterra combiner is to compute the multidimensional combinations between the shifted symbols $\hat{s}_{n+i}$. As described in (5.1), this implementation only considers combinations of order one and three. The combinations of order five or higher do not improve the system performance. This has been determined with the methodology used to justify the omission of centroid values. Hence, order five or higher combinations are not taken into account in the design of the Volterra Combiner.

The Volterra Combiner is made of the Direct Mapping block and the Combination block. These blocks are implemented as shown on figure 5.6. On figure 5.6, the output of the two kinds of blocks composes the Volterra Combination vector $\mathbf{vc_n}$, which is defined as:

$$\mathbf{vc_n} = [vc_{0,n}, vc_{1,n}, vc_{2,n}, \cdots, vc_{N-2,n}, vc_{N-1,n}]^T. \tag{5.11}$$

The Direct mapping block determines the order one combinations of $\mathbf{vc_n}$. These combinations are used to compute the first term (order 1) of the Volterra series regression presented in (5.1). The Direct Mapping block encodes the constellation points $\hat{s}_{n+i}$ quantified on three bits into a complex number which parts, real and imaginary, are quantified on $\Gamma$ bits. This direct mapping is easily done from two look-up tables (LUT), one for the real values ($\text{LUT}_I$) and one for the complex value ($\text{LUT}_Q$). Since $\hat{s}_{n+i}$ can only take eight possibles values, each LUT is composed of eight elements. The address for each element of the LUT is the value of $\hat{s}_{n+i}$. Thus, if the a-priori mapped symbol is $\hat{s}_{n+i} =' 010'$, the real value of $vc_{i,n}$ is $\text{LUT}_I$ @ $'010'$ and the complex value is $\text{LUT}_Q$ @ $'010'$.

The Combination Block computes the order three combinations between the symbols $\hat{s}_{n+i}$s. Indeed, the output of each combination block corresponds to one of the combinations

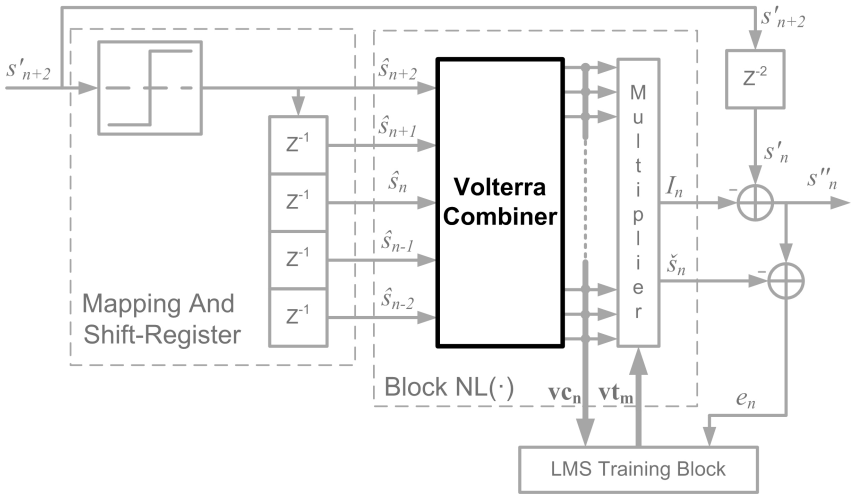



Figure 5.5: The Volterra Combiner into the Volterra Canceller

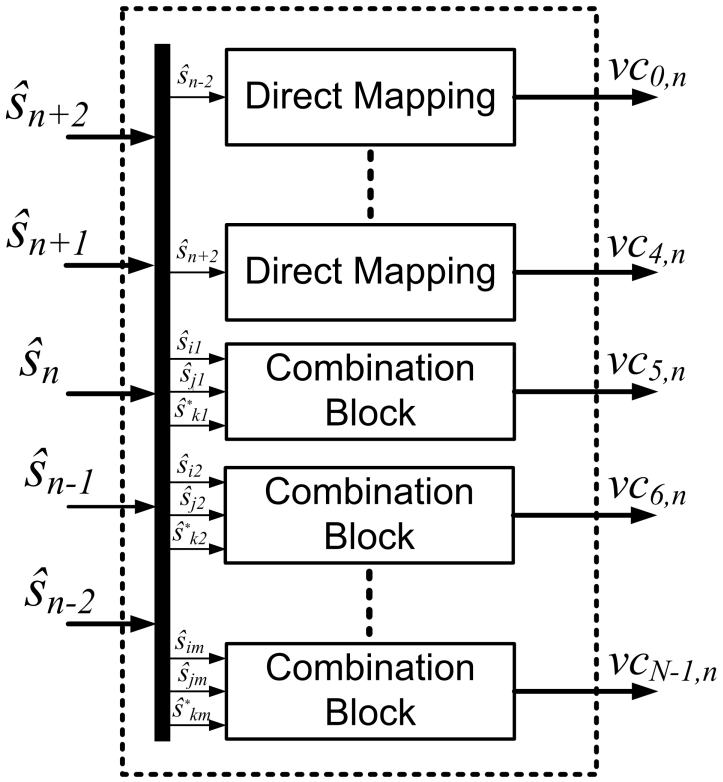



Figure 5.6: Volterra combiner block diagram

of the second term of (5.1), i.e.:

$$\sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M} b_{i,j,k}\hat{s}_{n+i}\hat{s}_{n+j}\hat{s}^*_{n+k}. \tag{5.12}$$

A possible implementation of the Combination Block results from firstly encoding each $\hat{s}_{n+i}$ with a direct mapping block, and then multiplying this complex value. If the combination is done this way, each combination block must be composed of three Direct Mapping blocks and two complex multipliers. A complex multiplier is composed of three $\Gamma$-bit multipliers. Thus, the complexity of the Volterra Combiner grows exponentially with the number of Combination Blocks considered.

An efficient way to avoid the complexity of the complex multipliers is by computing the order three combinations from a LUT. As the $\hat{s}_{n+i}$s can only take eight different values ($\mathbf{o_0}$ to $\mathbf{o_7}$), the size of each LUT containing the order three combination is limited. Such a size can be computed from a combination with the repetition lemma [Epp10]. This lemma states that for any group composed of $m$ different elements, if an element is taken $n$ different times, the number of possible combinations with repetition is equal to:

$$CR^n_m = \frac{(m+n-1)!}{n!(m-1)!}. \tag{5.13}$$

Where $m$ is the number of constellation points and $n$ is the order of the combination. In the present case, $m = 8$ and $n = 3$ yielding a value of 120 for $CR^n_m$. Therefore, the LUT

is composed of 120 complex values representing all the possibles results of the order three combinations, (5.12). Indeed, the value of each LUT's element is defined as:

$$LUT\ @\ i = \mathbf{o_A} \times \mathbf{o_B} \times \mathbf{o_C}.$$

Values $\mathbf{o_A}$, $\mathbf{o_B}$ and $\mathbf{o_C}$ represents the values carried by $\hat{s}_i$, $\hat{s}_j$ and $\hat{s}_k$. The three values are reordered to satisfy:

$$\mathbf{o_A} \leq \mathbf{o_B} \leq \mathbf{o_C}.$$

The 120 values representing the possible results of order three combinations are arranged in the LUT of figure 5.7 in the order described as follows:

$$\begin{aligned}
LUT\ @\ 0 &= \mathbf{o_0} \times \mathbf{o_0} \times \mathbf{o_0}.\\
LUT\ @\ 1 &= \mathbf{o_0} \times \mathbf{o_0} \times \mathbf{o_1}.\\
LUT\ @\ 2 &= \mathbf{o_0} \times \mathbf{o_0} \times \mathbf{o_2}.\\
&\vdots\\
LUT\ @\ 7 &= \mathbf{o_0} \times \mathbf{o_0} \times \mathbf{o_7}.\\
LUT\ @\ 8 &= \mathbf{o_0} \times \mathbf{o_1} \times \mathbf{o_1}.\\
LUT\ @\ 9 &= \mathbf{o_0} \times \mathbf{o_1} \times \mathbf{o_2}.\\
&\vdots\\
LUT\ @\ 118 &= \mathbf{o_6} \times \mathbf{o_7} \times \mathbf{o_7}.
\end{aligned}$$

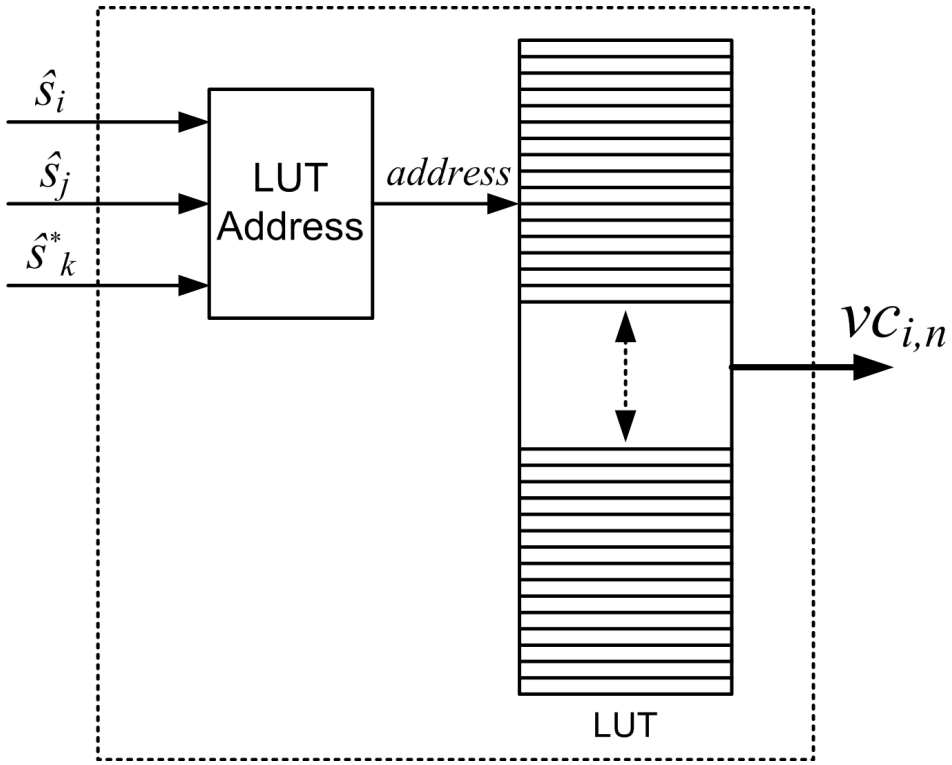



Figure 5.7: The Combination Block

$$LUT\ @\ 119 = \mathbf{o_7} \times \mathbf{o_7} \times \mathbf{o_7}.$$

As shown on figure 5.7, the LUT pointer is the value of *address*, which ranges from 0 to 119. The value of *address* is determined by the LUT Address block, whose inputs are the three constellation points $(\hat{s}_i, \hat{s}_j, \hat{s}_k)$. This block determines the value of *address* by means of the following equation:

$$address(\mathbf{o_A}, \mathbf{o_B}, \mathbf{o_C}) = \Upsilon_1(\mathbf{o_A}) + \Upsilon_2(\mathbf{o_B}) - \Upsilon_2(\mathbf{o_A}) + \Upsilon_3(\mathbf{o_C}) - \Upsilon_3(\mathbf{o_B}). \quad (5.14)$$

Where functions $\Upsilon_1, \Upsilon_3$ and $\Upsilon_3$ are three discrete functions described in table 5.3. The principle of (5.14) is to determine *address* in three steps. The first one is a coarse computation done by $\Upsilon_1$. The second and the third steps consist of fine computations done by functions $\Upsilon_2$ and $\Upsilon_3$. An example is given for $(\hat{s}_j, \hat{s}_k, \hat{s}_l)$ carrying the constellation points ($\mathbf{o_0}$,$\mathbf{o_1}$,$\mathbf{o_2}$). Using (5.14) and table 5.3, *address*($\mathbf{o_A}$,$\mathbf{o_B}$,$\mathbf{o_C}$) is;

$$address(\mathbf{o_0}, \mathbf{o_1}, \mathbf{o_2}) = \Upsilon_1(\mathbf{o_0}) + \Upsilon_2(\mathbf{o_1}) - \Upsilon_2(\mathbf{o_0}) + \Upsilon_3(\mathbf{o_2}) - \Upsilon_3(\mathbf{o_1}) = 9. \quad (5.15)$$

Hence, $LUT\ @\ 9 = \mathbf{o_0} \times \mathbf{o_1} \times \mathbf{o_2}$.

Table 5.3: Table for functions $\Upsilon_1, \Upsilon_3$ and $\Upsilon_3$

| $\mathbf{o_i}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $\Upsilon_1(\mathbf{o_i})$ | 0 | 36 | 64 | 85 | 100 | 110 | 116 | 119 |
| $\Upsilon_2(\mathbf{o_i})$ | 0 | 8 | 15 | 21 | 26 | 30 | 33 | 35 |
| $\Upsilon_3(\mathbf{o_i})$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

As the value of *address* is bounded by the interval [0-119], it is quantified on 7 bits.

The implementation of the Volterra Canceller presented in this work considers five Direct Mapping blocks and and nine Combination Blocks. Then, the length of the output vector $\mathbf{vc_n}$ is 14. Each element of $\mathbf{vc_n}$ is quantified on eight bits for the real and eight bits for the imaginary part. The number of Combination Blocks and the combinations constituting each block have been chosen in order to lead to a Volterra Canceller with good trade-off between system complexity and performance in terms of gain of $E_S/N_0$. This has been done from VHDL/SystemC simulations of a DVB-S2 system. The combinations characterizing each element of $\mathbf{vc_n}$ are described in table 5.4.

Once the Volterra Combinations vector $\mathbf{vc_n}$ is computed, it is sent to the Multiplier block.

Table 5.4: Volterra combinations considered in the implementation of the Volterra Canceller

| $vc_{i,n}$ | $\hat{s}_i$ | $vc_{i,n}$ | $\hat{s}_j\hat{s}_k\hat{s}_l^*$ | $vc_{i,n}$ | $\hat{s}_j\hat{s}_k\hat{s}_l^*$ |
|---|---|---|---|---|---|
| $vc_{0,n}$ | $\hat{s}_n$ | $vc_{5,n}$ | $\hat{s}_{n+1}\hat{s}_{n-1}\hat{s}_n^*$ | $vc_{10,n}$ | $\hat{s}_{n+1}\hat{s}_n\hat{s}_{n+1}^*$ |
| $vc_{1,n}$ | $\hat{s}_{n-2}$ | $vc_{6,n}$ | $\hat{s}_{n+1}\hat{s}_n\hat{s}_{n-1}^*$ | $vc_{11,n}$ | $\hat{s}_n\hat{s}_{n+1}\hat{s}_n^*$ |
| $vc_{2,n}$ | $\hat{s}_{n-1}$ | $vc_{7,n}$ | $\hat{s}_n\hat{s}_{n-1}\hat{s}_{n+1}^*$ | $vc_{12,n}$ | $\hat{s}_{n-1}\hat{s}_{n-2}\hat{s}_{n-1}^*$ |
| $vc_{3,n}$ | $\hat{s}_{n+1}$ | $vc_{8,n}$ | $\hat{s}_{n-1}\hat{s}_n\hat{s}_{n-1}^*$ | $vc_{13,n}$ | $\hat{s}_{n+1}\hat{s}_{n+2}\hat{s}_{n+1}^*$ |
| $vc_{4,n}$ | $\hat{s}_{n+2}$ | $vc_{9,n}$ | $\hat{s}_n\hat{s}_{n-1}\hat{s}_n^*$ | | |

### 5.1.3 The Multiplier block

As shown on figure 5.8, the Multiplier block is placed inside block $NL(\cdot)$ of the Volterra Canceller. The Multiplier Block computes two values. The first value is the non-linear ISI term $I_n$ associated with the received symbol $s'_n$. The second one is the estimation of $s'_n$ without non-linear ISI, noted as $\check{s}_n$. Both $I_n$ and $\check{s}_n$ values are obtained from the Volterra Combinations vector ($\mathbf{vc_n}$) and the set of Volterra coefficients ($\mathbf{vt_m}$), named Volterra taps vector.

In the precedent subsection, the Volterra Combiner was implemented by replacing complex multipliers by LUT tables. Such an implementation is possible because each $\hat{s}_i$ can only take eight different values. The Multiplier block multiplies vectors $\mathbf{vc_n}$ (row-vector) and $\mathbf{vt_m^H}$ (column-vector), which results in a sum of products. Each element of vector $\mathbf{vc_n}$ can take 120 different values. The elements of vector $\mathbf{vt_m}$ are quantified on $K$ bits and ranges from $-2^{K-1}$ to $2^{K-1}-1$. Thus, the high number of possible combinations for the multiplication between elements of $\mathbf{vc_n}$ and $\mathbf{vt_m}$ prevents the use of a LUT for the multiplier block.

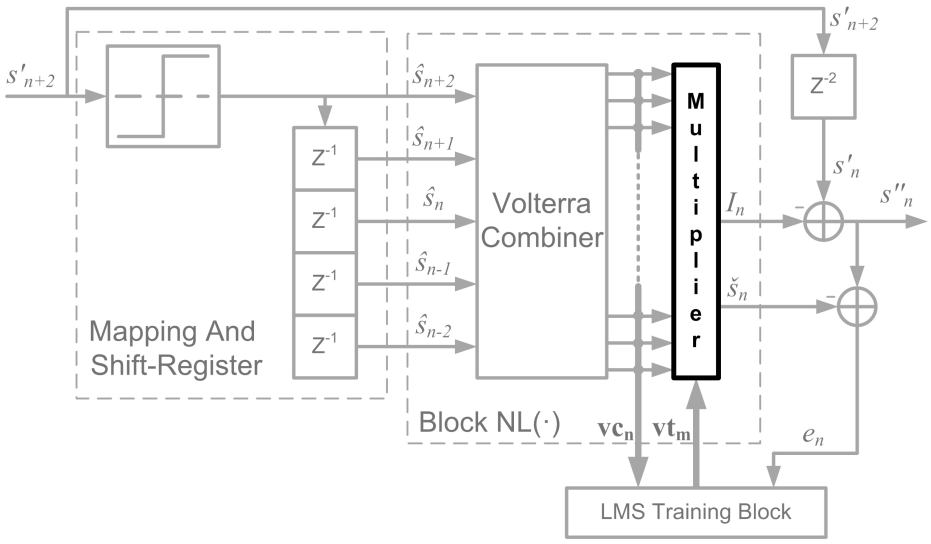



Figure 5.8: The Multiplier block into the Volterra Canceller

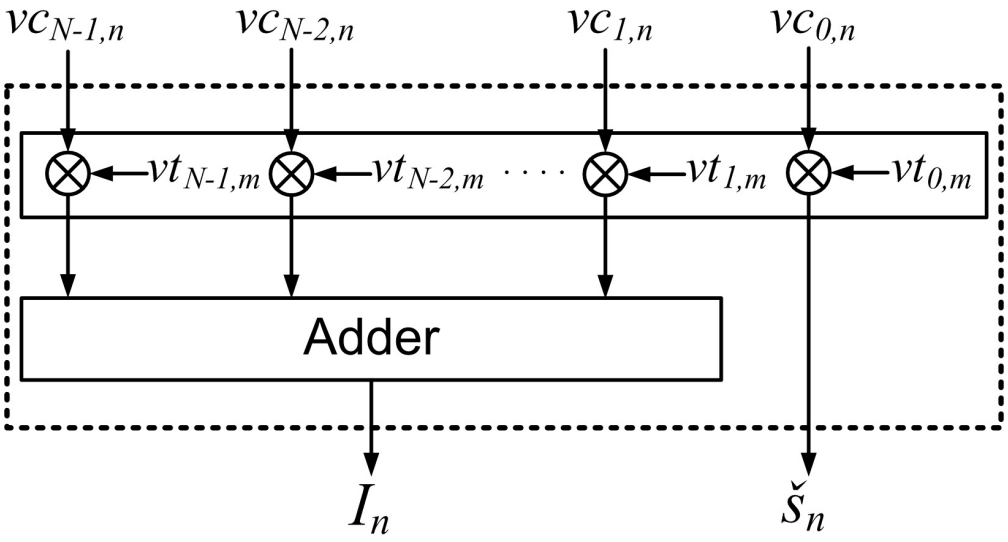



Figure 5.9: The Multiplier block

Hence, the Multiplier block is implemented from actual complex multipliers. The architecture of such a block is depicted on figure 5.9. As observed on this figure, the non-linear ISI $I_n$ can be also obtained as:

$$I_n = \sum_{i=1}^{13} vc_{i,n} vt^*_{i,m}. \tag{5.16}$$

The estimation of $s'_n$ without non-linear ISI is determined from the following multiplication:

$$\check{s}_n = vc_{0,n} vt^*_{0,m} = \hat{s}_n vt^*_{0,m}. \tag{5.17}$$

A standard complex multiplier is composed of four single multipliers and two single additions. The complexity of each complex multiplier is reduced by implementing the method proposed by [OVS94]. In fact, the multiplication of two complex values $A$ and $B$ can be carried out by only three single multiplications by computing the real part $C_Q$ and the imaginary part $C_Q$ as follows:

$$C_I = (A_I - A_Q)B_Q + (B_I - B_Q)A_I. \tag{5.18}$$

$$C_Q = (A_I - A_Q)B_Q + (B_I + B_Q)A_Q. \tag{5.19}$$

Where $A_I$ and $A_Q$ are the real and the imaginary part of $A$ and $B_I$ and $B_Q$ are them of $B$. As seen in (5.18) and (5.19), the first term of the right side is the same for both the computation of $C_I$ and $C_Q$. Then, the number of multiplication needed is three and the additions needed are five.

### 5.1.4 The LMS Training Block

The LMS Training Block adapts off-line the set of Volterra coefficients $\mathbf{vt_m}$, figure 5.10. As explained in section 3.4.2, the LMS algorithm is basically carried out in the two steps detailed in table 3.1. The first step is to determine the estimation error ($e_n$), defined as:

$$e_n = s'_n - \tilde{s}_n = s'_n - \mathbf{vc_n vt_m^H}. \tag{5.20}$$

Where $\tilde{s}_n$ is the estimation resulting from the Volterra regression of $s'_n$. The product between vectors $\mathbf{vc_n}$ and $\mathbf{vt_m^H}$ requires several complex multiplier and adder blocks. However, the implementation of these blocks can be avoided, reducing the system complexity. With $\tilde{s}_n$ equals to:

$$\tilde{s}_n = \check{s}_n + I_n, \tag{5.21}$$

and the output of the Volterra Canceller defined as:

$$s''_n = s'_n - I_n, \tag{5.22}$$

the error $e_n$ can be computed directly from the output of the Volterra Canceller and $\check{s}_n$ as:

$$e_n = s''_n - \check{s}_n. \tag{5.23}$$

Therefore in (5.23), all the multiplications and additions of (5.20) are replaced a single addition.

Consequently, the implementation of the LMS training block only considers the second step of the LMS algorithm. This step computes the refresh value for vector $\mathbf{vt_m}$ as follows:

$$\mathbf{vt_{m+1}} = \mathbf{vt_m} + \mu e_n^* \mathbf{vc_n}. \tag{5.24}$$

As stated in section 3.4.2, the refresh value (second term in (5.24)) represents a scaled version of the gradient. The scale factor is $\mu$, then the gradient is defined by $e_n^* \mathbf{vc_n}$.

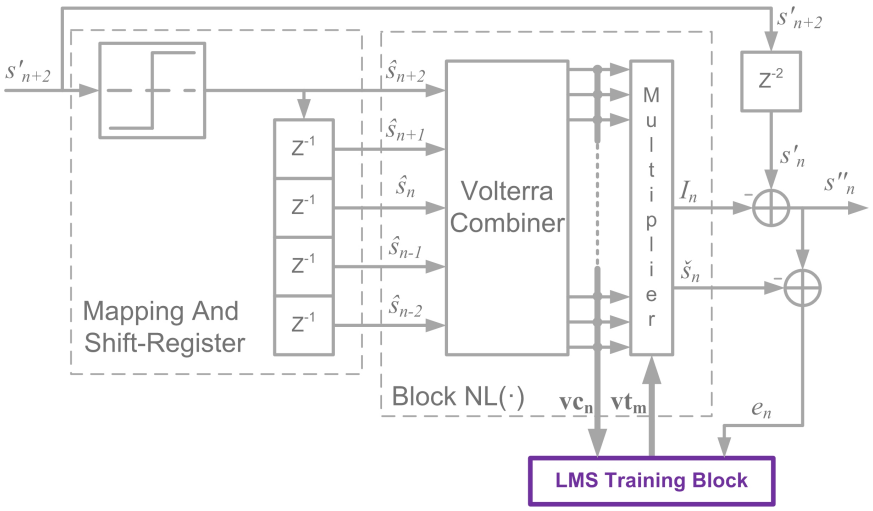



Figure 5.10: The LMS Training block into the Volterra Canceller

The quantification of the elements of $\mathbf{vt_m}$ has been determined by analyzing table 5.4. As seen in this table, element $vc_0$ contains the value of $\hat{s}_n$. Therefore, the correlation between each value of $\mathbf{vc_n}$ and $s'_n$ is maximal for $vc_0$. Hence, $vc_0$ needs more quantification bits than the other elements, i.e. a larger dynamic. This correlation has been verified from Matlab simulations of a DVB-S2 system, taking as reference values of IBO from 0 to 10 dB and the three possible values of roll-off (0.2, 0.25 and 0.35). The best trade-off between the quantification of $\mathbf{vt_m}$ and the gain of the Volterra Canceller has been determined from SystemC/VHDL simulations. These simulations consider the same criteria used in the design of the previously introduced blocks. From this analysis, the coefficient multiplying $vc_0$, i.e. $vt_0$, is quantified on 9 bits whereas the remaining elements of $\mathbf{vt_m}$ are only quantified on 5 bits.

Once the quantification is defined, it is necessary to analyze the convergence of the LMS implementation. Considering a coefficient of $\mathbf{vt_m}$ quantified on 5 bits and two-complement, the possibles values for this coefficient are in the interval:

$$-16 \leq vt_i < 15.$$

Consequently, the minimal value of the refresh term, i.e. "00001", is the 6% of the maximum value of $vt_i$. With such a refresh term the convergence of the LMS algorithm cannot be guaranteed, as seen in section 3.4. Hence, quantifying with 5 bits is efficient in terms of gain of $E_S/N_0$ but not in terms of convergence. So, in order to avoid

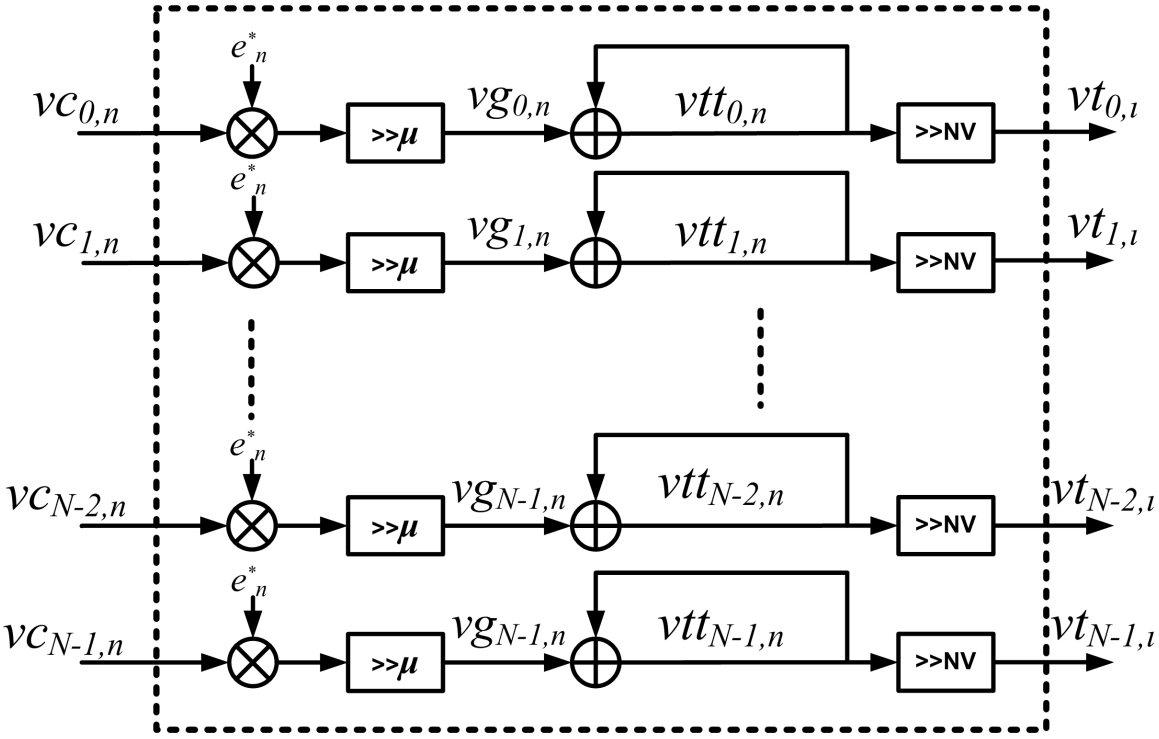



Figure 5.11: LMS Training Block for the Volterra Canceller ("«" (shift left),"»" (shift right))

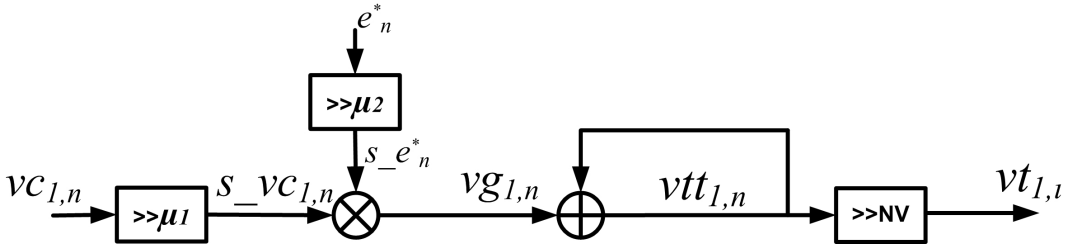



Figure 5.12: LMS Training Block for the Volterra Canceller ("«" (shift left),"»" (shift right))

any divergence in the Training Block, the dynamic of $\mathbf{vt_m}$ is artificially enlarged by an accumulated value $vtt_i$. Then, a shift operation of NV bits brings back each elements of $\mathbf{vt_m}$ to a dynamic of 5 bits. This original architecture is presented on figure 5.11.

The complexity associated with the multiplications between $e_n^*$ and each element of $\mathbf{vc_n}$ can be reduced. It is done by shifting by $\mu$ bits before multiplying. Then, the gradient is obtained by multiplying together the shifted values $s\_vc_{i,n}$ and $s\_e_n^*$, figure 5.12 . The optimal quantification of $s\_vc_{i,n}$ is done on 4 bits and $e_n^*$ is quantified on 5 bits. This quantification has been obtained from SystemC/VHDL simulation considering the original quantification and the simplified one. No BER degradation is induced by the proposed simplification. The multiplication of both values is carried out by means of three 5-bit additions. Then, the reduction in the complexity is achieved with respect to the multiplier of two values quantified on 8 bits, implemented as seven 8-bit additions.

## 5.2 Architecture and Implementation of the Hermite Polynomial Linearizer

This section describes the implementation of the Hermite polynomial linearizer introduced in section 4.1.3. As in the case of the Volterra Canceller, the design of the Hermite linearizer is focused on the minimization of the system complexity.

The architecture of the Hermite polynomial linearizer implemented is illustrated on figure 5.13. The linearizer is composed of three blocks: The Hermite Polynomial Expansion block, the Multiplier block and the LMS Training Block. The Hermite Polynomial Expansion block computes from the input signal $y_t$ the values of the Hermite polynomials of order one ($hc_{1,t}$) and three ($hc_{3,t}$). Both $hc_{1,t}$ and $hc_{3,t}$ form vector $\mathbf{hc_t}$. The Multiplier computes the linearized signal $y'_n$ from $\mathbf{hc_t}$ and $\mathbf{ht}_\iota$. Vector $\mathbf{ht}_\iota$ is the set of coefficients characterizing the polynomial regression. The elements of $\mathbf{ht}_\iota$ are estimated off-line by the LMS Training Block.

The next subsections describes with details the digital design of each block composing the Hermite polynomial linearizer.

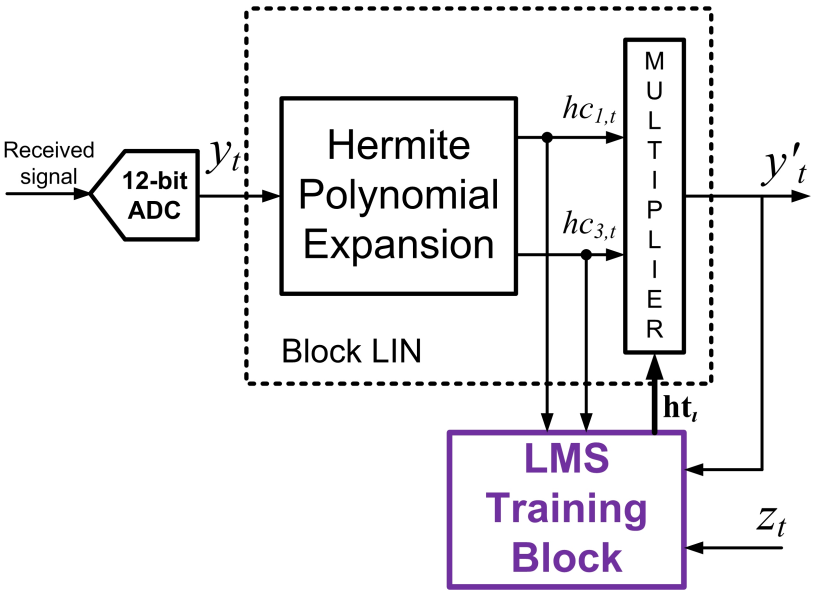



Figure 5.13: Blocks constituting the implementation of the Hermite polynomial linearizer

### 5.2.1 The Hermite Polynomial Expansion block and the Multiplier block

The Hermite polynomials considered in the regression of order one and three are:

$$H_{1,t}(y_t) = y_t. \tag{5.25}$$

$$H_{3,t}(y_t) = \frac{1}{3}(2y_t^2 y_t^* - 3y_t = 2y_t|y_t|^2 - 3y_t). \tag{5.26}$$

In the case of a floating-point implementation for a DVB-S2 system based on 8-PSK digital modulations, the values of $|y_t|$ are near one. This is because PSK modulations are characterized by a constant envelop transmission. Then, the value of $|y_t|^2$ and $y_t|y_t|^2$ are also near one. This ideal case is not respected in the digital implementations characterizing the DVB-S2 receiver used in this work. For the implementation presented in this chapter $y_t$ is quantified on 8 bits. Then, the possible values of $y_t$ are within the interval:

$$-128 \leq y_t < 127.$$

Thus, as the value of $|y_t|$ is larger than one, the value of $|y_t|^2$ grows quadratically with respect to the value of $|y_t|$. Such a difference is even larger if $y_t|y_t|^2$ is compared with $|y_t|$.

A normalization can be performed to have the same behaviour as the floating point implementation. The implementation of this work is carried out by taking as input of the Hermite polynomials the normalised value $\bar{y}_t$:

$$\bar{y}_t = \frac{y_t}{\sigma_{y_t}}, \tag{5.27}$$

where $\sigma_{y_t}$ is the standard deviation of $y_t$. The value of $\sigma_{y_t}$ is arbitrarily chosen as the quantified fixed point value "01100100" or 100 in signed decimal value. This value helps

later simplify the linearizer. The value of $\sigma_{y_t}$ is experimentally set to 100 by the analogue blocks placed before the ADC. Hence, the average value of the powers of $y_t$ and $y_t|y_t|^2$ is equal to one. Then, the Hermite polynomials of order one and three are redefined as as:

$$\hat{H}_{1,t}(y_t) = \frac{y_t}{\sigma_{y_t}} = \frac{y_t}{100}. \tag{5.28}$$

$$\hat{H}_{3,t}(y_t) = \frac{1}{3}(\frac{2y_t^2 y_t^*}{\sigma_{y_t}^3} - 3\frac{y_t}{\sigma_{y_t}}) = \frac{1}{3}(2\frac{y_t|y_t|^2}{1000000} - 3\frac{y_t}{100}). \tag{5.29}$$

To simplify further, the divisions by 100 and by 1000000 in (5.28) and (5.29) must be approximated by a power of two division or multiplication, in order to be replaced by shift blocks. This is achieved by multiplying (5.28) and (5.29) by 1500, yielding:

$$\hat{H}_{1,t}(y_t) = 15y_t. \tag{5.30}$$

$$\hat{H}_{3,t}(y_t) = \frac{y_t|y_t|^2}{1000} - 15y_t. \tag{5.31}$$

The division by 1000 is approximated by a division by 1024, which corresponds to a 10-bit right-shift operation. In the same way, the multiplication by 15 is approximated as a multiplication by 16, which can be done with a 4-bit left-shift operation. This approximation does not induce a performance degradation when it has been tested in SystemC/VHDL simulations. The final architecture of the Hermite Polynomial expansion block implementing shift operations is illustrated on figure 5.14. On this figure, the output values $hc_{1,t}$ and $hc_{3,t}$ representing the polynomial expansion are quantified on 12 bits. Then, the value of the compensated signal $y'_t$ is obtained as:

$$y'_t = \mathbf{hc_t ht_\iota^H} = hc_{1,t}ht^*_{1,\iota} + hc_{3,t}ht^*_{3,\iota}. \tag{5.32}$$

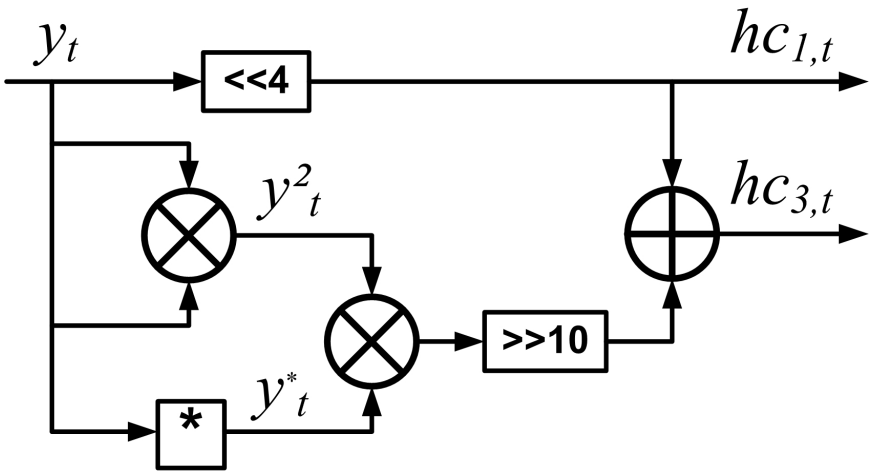



Figure 5.14: The Hermite Polynomial Expansion block

Where $\mathbf{ht}_\iota$ is the vector containing the coefficients of the Hermite polynomial regression. Each element of $\mathbf{ht}_\iota$ and the compensated signal $y'_t$ are quantified on 8 bits. The complex multiplications between $hc_{i,t}$ and $ht_{i,\iota}$ is done with the multiplier described in section 5.1.3.

### 5.2.2 The LMS Training Block

The LMS Training Block adapts off-line the set of Hermite polynomial regression coefficients $\mathbf{ht}_\iota$, figure 5.15.

As in the case of the Volterra Canceller, the LMS algorithm is executed in two steps. The first step computes the estimation error, which is defined as:

$$e_t = z_t - y'_t = z_t - \mathbf{hc_t ht_\iota^H}. \tag{5.33}$$

Where $z_t$ is the training signal, which is a replica of the PA input. In this implementation, the signal $z_t$ is obtained from the output of the Volterra Canceller by performing the blind estimation technique introduced in section 4.3.2.

Again, the second step is implemented as done for the Volterra Canceller training, figure 5.16. The gradient vector $\mathbf{hg_t}$ is obtained by multiplying the shifted version of vector $\mathbf{hc_t}$ and error $e_n$. The vector $\mathbf{s\_hc_t}$ is quantified on 2 bits and the error is quantified on 8 bits. The resulting gradient is quantified on 9 bits. This gradient is then added to the accumulated value $\mathbf{htt_t}$. Each element of the vector is quantified on 32 bits. Such a high number of bits is necessary to insure the LMS algorithm convergence. Vector $\mathbf{ht}_\iota$ is finally

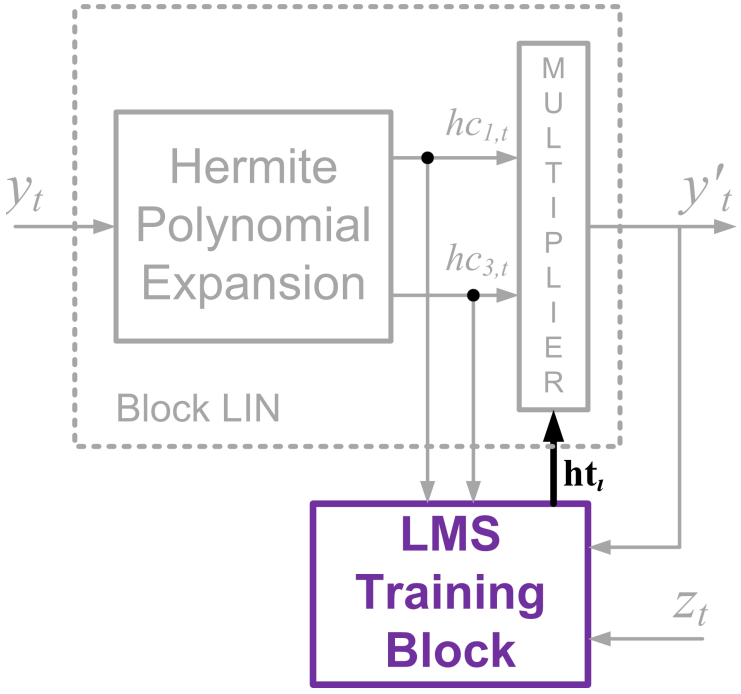



Figure 5.15: LMS Training Block into the Hermite polynomial linearizer

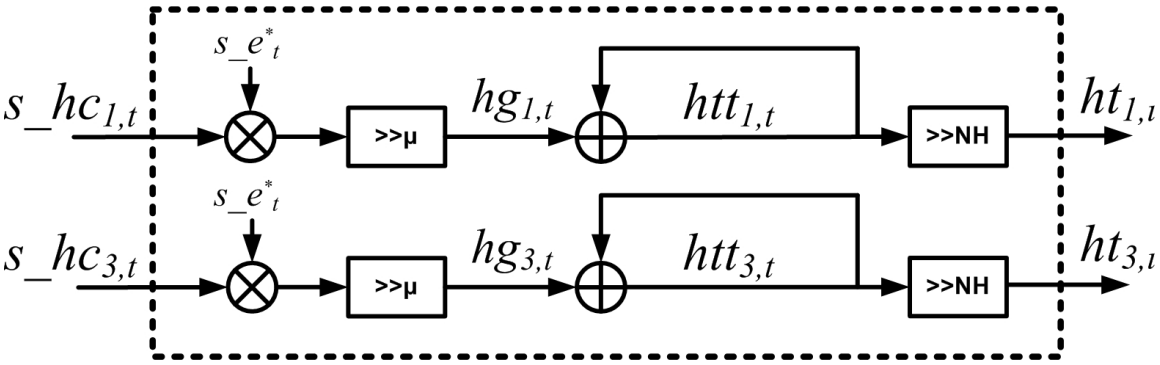



Figure 5.16: LMS Training Block for the Hermite polynomial linearizer

obtained by rounding vector $\mathbf{htt_t}$. The value of NH is 8.

## 5.3 Summary



This chapter has proposed an innovative low-complex digital design for the Volterra Canceller and the Hermite polynomial linearizer, optimised to work within DVB-S2 receivers.

The digital design of the Volterra Canceller has been implemented from four blocks. The first block consists of the mapping block with shift register. This chapter has proposed two different architectures of the mapping block, the first one based on phase comparison and the second one based on distances comparison. As the second architecture is less complex and has similar behaviour, the Mapping Block implemented in the Volterra Canceller is the one based on distances comparison. The second block is the Combination Block, which is implemented by means of LUTs to avoid the complexity of several complex multipliers. The Combination Block block has been optimised in terms of system complexity and performance by determining the optimal Volterra Combinations used in the Volterra regression. The third block is the Multiplier Block, which carries out single complex multiplications from only 3 multipliers. The fourth block is the LMS Training block, which trains the Volterra Canceller. The complexity of this block has been reduced thanks to the combination of the output of the multiplier block and the LMS Training block. Hence, the LMS Training block implements only half of the LMS algorithm.

The design of the Hermite polynomial linearizer has been implemented from three blocks. The Hermite Polynomial Expansion block computes the values of the Hermite polynomial from the input signal. The design this block is implemented from shift operations, avoiding complex multipliers. The second and third blocks are the Multiplier block and the LMS Training block, whose architectures are similar to thats of the Volterra Canceller.

Both the Volterra Canceller and the Hermite polynomial linearizer architectures have been implemented in VHDL. The FPGA integration and measurement results are presented in Chapter 6.

5

# 6

# Measurements

This chapter presents the implementation and the measurement results of a complete DVB-S2 transmission chain, composed of a transmitter, a Gaussian noise channel (AWGN) and a DVB-S2 receiver. The transmitter and the channel are implemented from specific devices emulating a DVB-S2 transmitter without HPA and the noise. The HPA and the receiver are implemented into an FPGA. The HPA is approximated by the model proposed in [DS06]. The integrations into a DVB-S2 receiver of the stand-alone Volterra Canceller and then of the technique combining a Volterra Canceller with a linearizer are also studied. All the measured results of performance presented in this chapter are expressed in terms of the average number of LDPC iterations as a function of the $E_S/N_0$. Measurements are done from three different configurations of the DVB-S2 receiver, one no considering post-distortion, the second compensating with a stand alone Volterra Canceller and the last one implementing the method proposed in this Thesis. Then, measured results are compared with the simulation results of section 4.4.

Chapter 6 is organised as follows. Section 6.1 describes the transmission chain Test-Bench. Section 6.2 introduces the different configurations for the DVB-S2 receiver. Section 6.3 presents and analyses the measurement results. Finally section 6.4 summarizes this chapter.

## 6.1 Test bench

The test-bench described in this chapter is based on the DVB-S2 transmission chain of figure 6.1. This chain reproduces the scenario used in computer simulations of chapter 4. The transmitter generates a DVB-S2 signal $x_t$ which is impaired by a non-linear HPA. This signal is then impaired by Gaussian channel noise (AWGN). The DVB-S2 receiver processes the received signal $y_t$. Both the DVB-S2 transmitter and the Gaussian channel noise are emulated with specific devices. A behavioural model of the HPA is implemented on an FPGA, figure 6.2. This requires the output of the DVB-S2 transmitter to be digitized before being processed by the HPA. To add the channel noise, the HPA output is ditigal-to-analogue converted. The DVB-S2 receiver implemented on the same FPGA as the HPA digitizes the the noisy non-linear signal by means of ADC-2.

The Test-Bench emulating the DVB-S2 transmitter, the HPA, the Gaussian channel noise (AWGN) and the DVB-S2 receiver is illustrated on figure 6.3. The SFU is a TV signal generator emulating the DVB-S2 transmitter. The AWGN generates the channel noise. The $E_S/N_0$ level is determined with a spectrum analyser. The FPGA board is connected to a computer to assess the receiver's performance.

6

## 6.2 Receiver implementations

The receiver configurations used to measure the gain of $E_S/N_0$ characterizing the post-distortion techniques of Chapter 5 are the topic of this section. As done for computer simulations, the receiver's performance are stated in terms of the average number of LDPC iterations for a giver $E_S/N_0$. Four different scenarios are defined for the DVB-S2 receiver. The first one considers a DVB-S2 transmitter with linear HPA, i.e. the transmission is not


Figure 6.1: DVB-S2 transmission chain

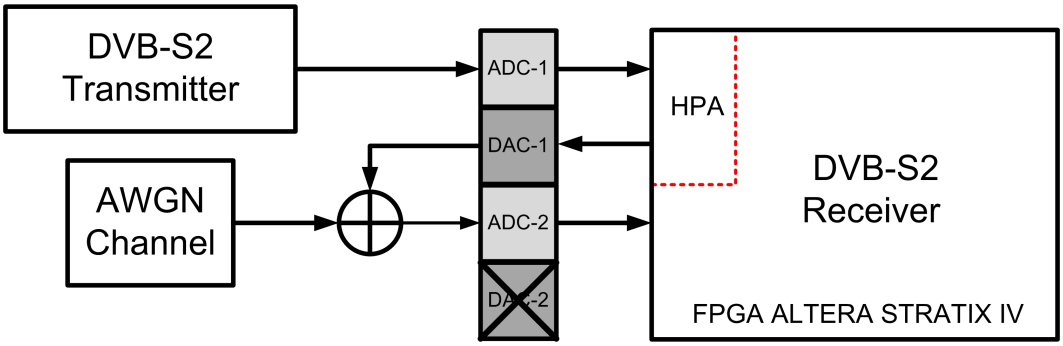



Figure 6.2: Data conversion connections

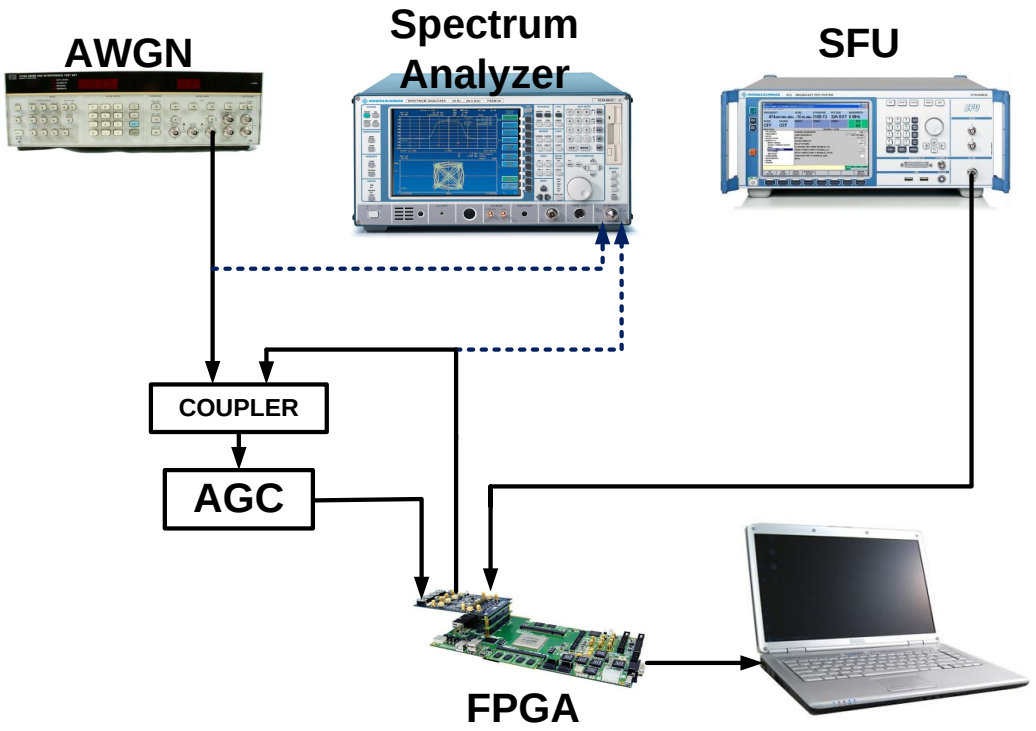



Figure 6.3: Test-Bench

impaired by non-linear ISI, leading to ideal performance. The transmitter of the second scenario is affected by the HPA non-linearity and its receiver does not compensate for the non-linear ISI (worst case). The third scenario considers non-linear HPA and a DVB-S2 receiver with a stand-alone Volterra Canceller. Also impaired by a non-linear HPA at the transmitter, the fourth scenario combines the Volterra Canceller and the Hermite polynomial linearizer presented in Chapter 5.

The following subsections detail the different configurations of the DVB-S2 receiver of figure 6.1 for each of the four possible scenarios.

### 6.2.1 DVB-S2 receiver architecture without non-linear ISI compensation

The first and the second measurement scenarios consider a DVB-S2 receiver without compensation for the non-linear ISI. This subsection describes the initial implementation of the the DVB-S2 receiver designed by NXP Semiconductors, figure 6.4.

After the analog-to-digital conversion, the signal is translated to 0 Hz by the down-conversion block before being demodulated. Performed from down-sampling, the down-conversion block degrades the $E_S/N_0$. As postulated in [TI06], the processing degradation induced by down-sampling digital signals is defined as:

$$PD = 10log_{10}(\frac{f_{samp}}{2\Delta B}). \tag{6.1}$$

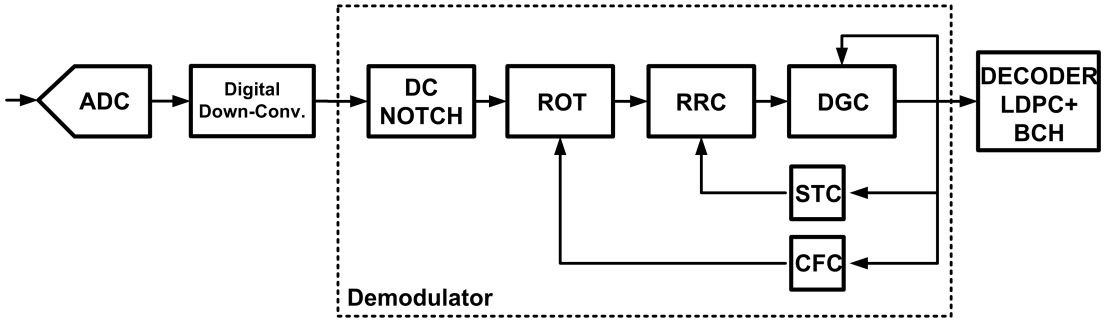



Figure 6.4: Blocks of the DVB-S2 receiver implementation

Where $f_{samp}$ represents the down-sampling frequency and $\Delta B$ is the band-width of the input signal. Any DC component affecting the base-band input signal is filtered by a notch at 0 Hz. The filtered signal is then frequency rotated (block ROT) to correct the rotation of the PSK constellation caused by any frequency error. The frequency error is estimated by block CFC (carrier frequency correction). The digital matched filter, block RRC generates the constellation points. This block also applies a sampling time correction, which is estimated by block STC (sampling time correction). A Digital Gain Control block (DGC) matchs the magnitude of the received symbols to the dynamic of the demodulator. Then, the symbols are de-mapped and decoded (LDPC and BCH decoder).

### 6.2.2 Integration of the Volterra Canceller into the DVB-S2 receiver

The third measurement scenario considers the integration of a stand-alone Volterra Canceller into the digital part of the DVB-S2 receiver. Since the Volterra Canceller, block VC, has as input the received symbols, this block is placed after block DGC, figure 6.5.

Note that blocks DGC, STC and CFC are not connected as in the design of figure 6.4 but rather to the VC block output. Then, DGC, STC and CFC blocks work with a compensated non-linear ISI input, improving its convergence in terms of time and precision. This has been verified with SystemC/VHDL simulations. Simulations have also shown that this configuration does not degrade the convergence if the transmission is

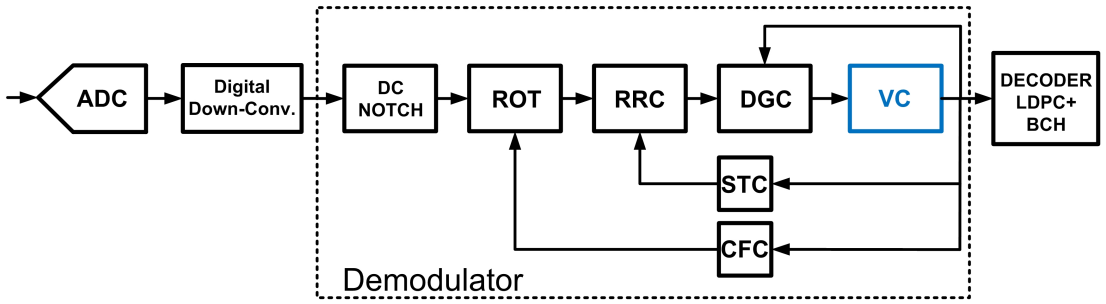



Figure 6.5: Integration of the Volterra Canceller into the DVB-S2 receiver

not impaired by the non-linear ISI. This is an additional contribution of this work.

### 6.2.3 Integration of the Hermite Polynomial Linearizer into the DVB-S2 receiver

The scenario including the Hermite polynomial linearizer combined with the Volterra Canceller is described in this subsection. As presented in section 4.1, the polynomial linearizer should be placed before the receiver digital matched filter. Therefore, in the context of the DVB-S2 receiver implemented in this work, the Hermite polynomial linearizer should be placed just before block RRC. However, this implementation places the Hermite polynomial linearizer before the ROT block, figure 6.6. This is done in order to correct the frequency error before estimates and compensates for the non-linearity, avoiding instability in the implementation. The HPL block is by-passed until the synchronization is done. Once the receiver is synchronized, the Hermite polynomial linearizer is adapted and then activated.

The training data necessary to adapt the Hermite polynomial linearizer is obtained with the direct decision feed-back training technique presented in section 4.3.2. The implementation of this technique into a DVB-S2 demodulator is shown also on figure 6.6. The output symbols of the Volterra Canceller are a-priori mapped onto constellation points. Those points are up-sampled and filtered by the RRC block. The coefficients of this block are obtained from the RRC block placed at the demodulator path. The resulting signal is then rotated by a ROT block. The phase of rotation is the negative value of the phase sent from the ROT block of the demodulator path. The output of the ROT block of the feed-back path is the training data needed to carry out the LMS algorithm of the Hermite polynomial linearizer.

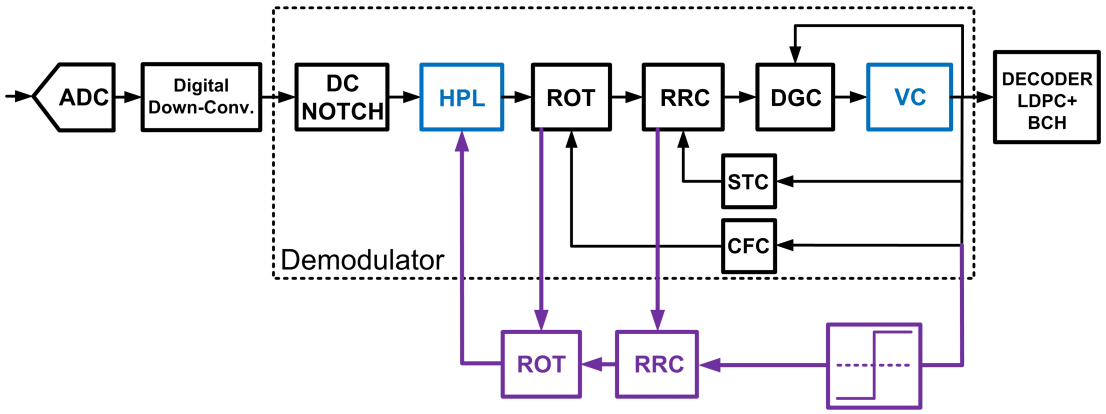



Figure 6.6: Integration of the Hermite polynomial linearizer and the direct decision feed-back training block into the DVB-S2 receiver

### 6.2.4 FPGA implementation of the DVB-S2 receiver

This sub-section describes how the FPGA resources are used for the different blocks constituting the DVB-S2 receiver and the HPA, figure 6.7. The FPGA resources are stated in terms of adaptive module logic (ALM) units needed for the implementation of each block constituting the DVB-S2 receiver [Alt11]. As observed on figure 6.7, the demodulator considered in the fourth measurement scenario, i.e. including the combined non-linear ISI compensation technique uses only 19% of the resources. The most complex block is the decoder (LDPC + BCH), which requires almost the 60% of the resources of the FPGA. The HPA only demands for the 0.002% of the FPGA resources. FInally, label "others" makes reference to several blocks used to command and test the DVB-S2 receiver. These blocks occupy 22 % of the FPGA.

Figure 6.8 describes the resources allocation of the DVB-S2 demodulator of figure 6.6. The most complex block of the combined non-linear ISI compensation technique is the Volterra Canceller, which requires 19% of the demodulator resources. This block occupies more than twice the resources of the Hermite polynomial linearizer combined with the feed-back Training block (8% of the resources). Nevertheless, the complexity of the Volterra Canceller is not critical. This is demonstrated by comparing the complexity of the Volterra Canceller and the RRC block. The Volterra Canceller needs 3252 ALM units while the RRC block implements 2144 ALM units, i.e. the complexity is of the same order. The coefficients of a RRC filter are predefined for three different values of roll-off. Hence, the difference of almost 1100 ALM with respect to the Volterra Canceller is due to the absence of the coefficient estimation in the RRC block.

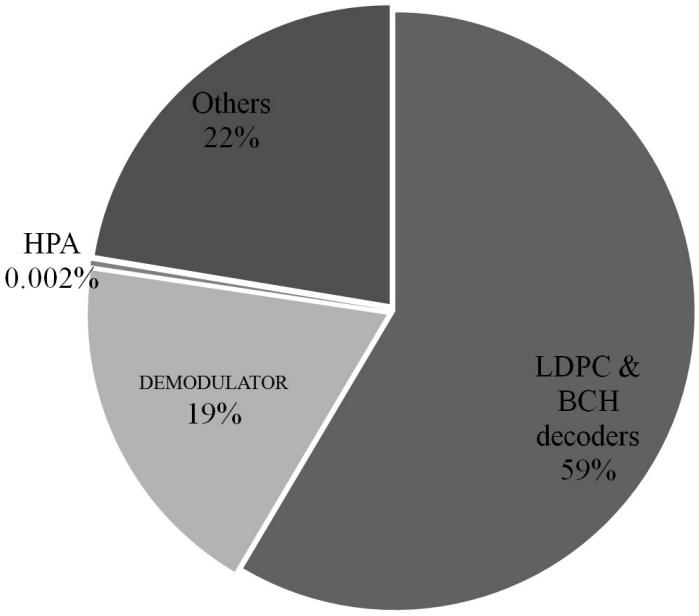



Figure 6.7: FPGA resources allocation

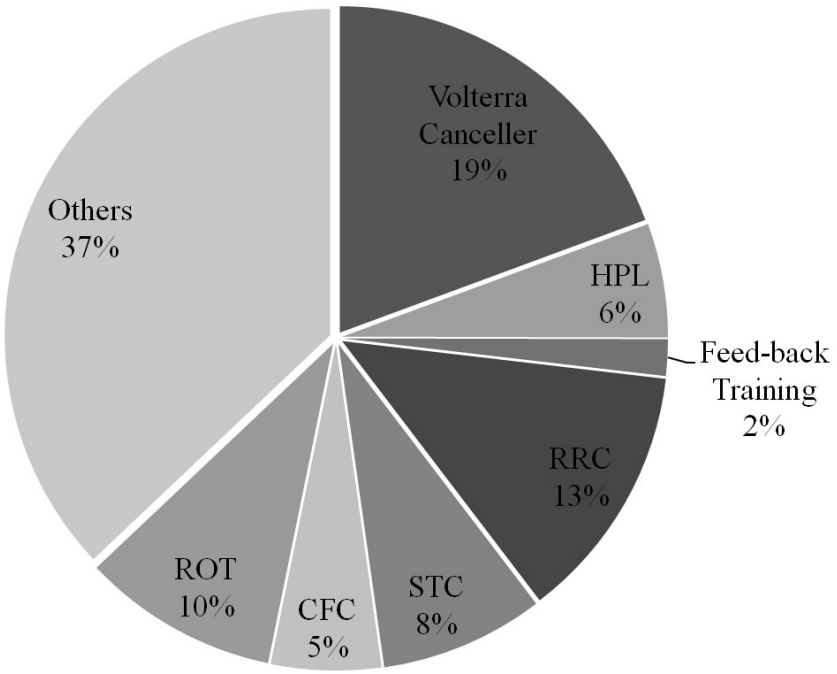



Figure 6.8: Demodulator resources allocation

## 6.3 MEASUREMENTS

### 6.3.1 Actual Test-Bench implementation

The actual implementation of the Test-Bench emulating the DVB-S2 transmitter used to determine the performance of the DVB-S2 receiver is illustrated on figure 6.3. The DVB-S2 transmitter is emulated by a TV signal generator Rhode & Schwarz SFU - Broadcast Test System [SFU05]. The SFU is able to modulate a signal carrier of frequency in the range 100 kHz to 3 GHz with a DVB-S2 base-band signal. The SFU can also set the output power, the output bandwidth $\Delta B$, the roll-off, the digital modulation (Q-PSK, 8PSK, 16-APSK and 32-APSK) and the LDPC code-rate. For the DVB-S2 chain considered in this work, the signal carrier frequency is $f_0 = 12.5$ MHz. The output bandwidth is 3 MHz and the roll-of is 0.35 (mandatory case for the DVB-S2 standard). The digital modulation is 8-PSK.

The HPA block approximates the HPA characteristics proposed in [DS06]. Such characteristics have been implemented into an FPGA. The FPGA core used is an Altera STRATIX IV one [Alt11] integrated into a TerAsic DE4 board [Ter10]. The board works together with the ADC-DAC board TerAsic Data Conversion HSMC, composed of two 14-bit ADC (ADC-1 and ADC-2) and two 14-bit DAC [AD08], figure 6.2 . The sampling frequency for the ADC-1 is 50 MHz, which is more than twice the maximal frequency ($f_{max}$) of the input signal ($f_{max} = 14$ MHz for $\Delta B = 3$ MHz). The output of the HPA block is then converted by the DAC-1 onto an analogue non-linear signal. As stated in section 2.4.1, the non-linear HPA output signal presents spectral components for all the

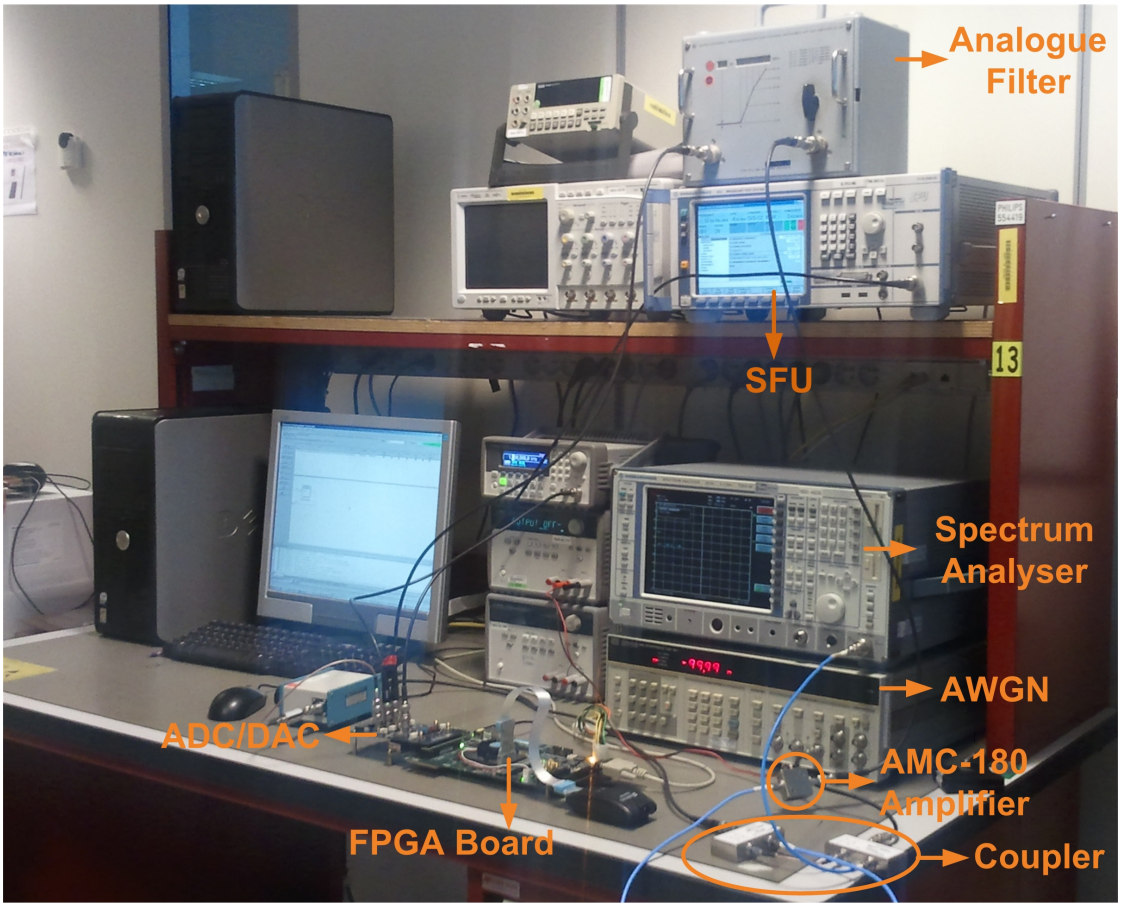



Figure 6.9: Actual implementation of the Test-Bench

multiples of the carrier frequency. Thus, in order to avoid aliasing, the sampling frequency of DAC-1 must be higher as that of ADC-1. Considering IBO = 0 dB, the power spectral difference between the component surrounding $f_0$ and $8f_0$ is larger than 50 dB. Then, the maximal frequency considered for the HPA output signal is $7f_0$ or 87.5 MHz. The sampling frequency for DAC-1 is 200 MHz. The components of the HPA output signal with frequency higher than $f_0$ are then analogically filtered out. The back-off of the HPA, which is set to 0 dB, is determined by the SFU output power.

The Gaussian channel noise (AWGN) signal is generated by a Hewlett-Packard 3708A Noise and Interference Test Set [AWG90]. The output Gaussian noise power is set with a step of 0.1 dB/Hz. The method to determine the $E_S/N_0$ level is based on comparing the channel power of the HPA output signal with the channel power of the noise signal. The channel power is measured by integrating the power into the transmission band. Such an integration is done by a spectrum analyzer Rhode & Schwarz FSEA 30.

The output signal of block HPA and the Gaussian channel noise (AWGN) are added by means of a coupler HH-108 Tyco Electronics [Cou90]. The resulting signal is then amplified by a Voltage controlled amplifier AMC-180. This amplifier is used as an automatic gain

control (AGC). The goal of the amplifier is to adapt the signal level to the dynamic range of ADC-2. Before the analogue-to-digital conversion, the amplified signal is analogically filtered with a cut-off frequency of 20 MHz.

### 6.3.2 HPA implementation test

The output spectrum of the non-linear HPA FPGA implementation after analogue filtering is shown on figure 6.10. The spectral is centered at frequency $f_0$ (12.5 MHz). As mentioned previously, the IBO considered is 0 dB. The spectrum illustrated on figure 6.10 almost matches the simulated spectrum on figure 2.16c. The difference in dB between the order one and the order three combination components in both figures is 17 dB. The order five combination on figure 6.10 is 33 dB below the order one, while this difference on figure 2.16c is 30 dB.

### 6.3.3 Performance measurement

The performance of the DVB-S2 receiver is determined from the number of iterations of the LDPC decoder, which is stored into a register of the FPGA, as a function of the

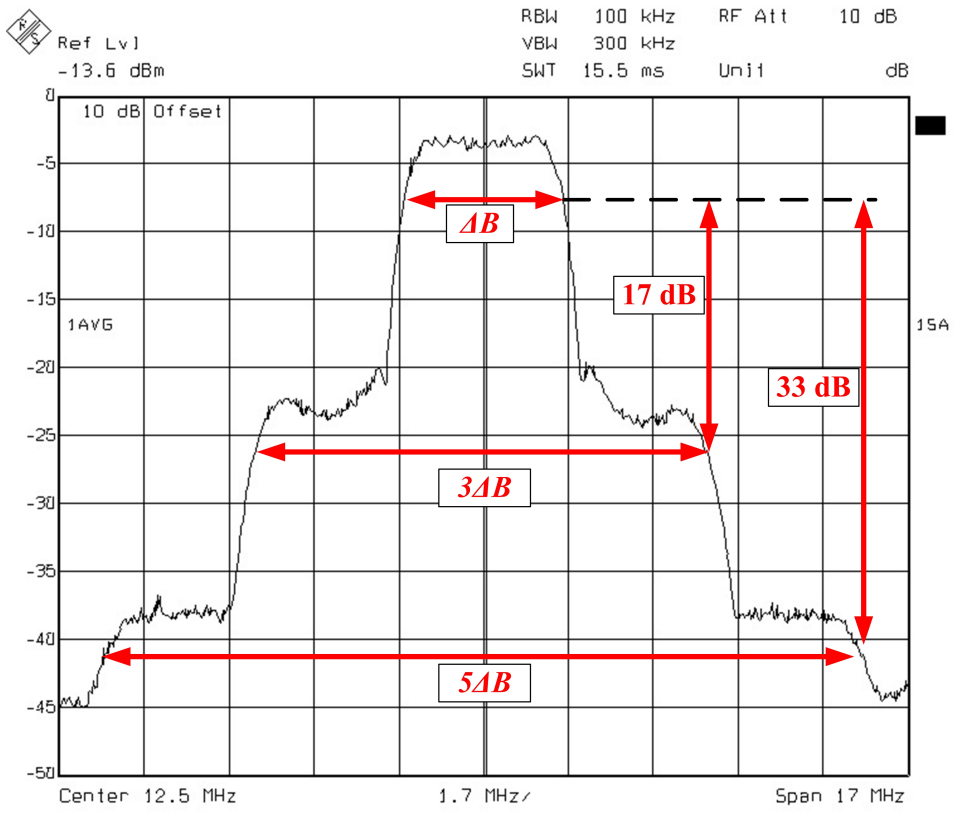



Figure 6.10: HPA output spectrum for IBO=0 dB

$E_S/N_0$. The value of this register is then used to compute the average number of LDPC iterations needed to determine the receiver's performance. The average is determined from 10240 LDPC decodings.

The receiver performance are analysed for the three different configurations described in section 6.2. The first configuration is that of a DVB-S2 receiver without post-distortion. The measured performance considers the cases with and without HPA non-linearity, i.e. the ideal and the worst cases, figure 6.11. Equivalent simulation results are also included. As observed on this figure, there is an offset of $E_S/N_0$ between measured performance and simulated performance. As mentioned previously, this degradation is mostly induced by the digital down-conversion block. A residual degradation of $E_S/N_0$ is due to the inaccuracies associated with digital implementation, such as quantification and the design simplifications, which are not included in simulation models. The Test-Bench considers $f_{samp}$ = 25 MHz, which is half the ADC frequency, and $\Delta_B$=20 MHz. Then, by combining these parameters with (6.1), the processing degradation is 2.04 dB. From now on, this offset is taken into account in the next measurement results. Once the offset is corrected, the measured results are correlated with the simulation results, 6.12. The residual degradation of $E_S/N_0$ is 0.09 dB.



The curves depicted on figure 6.12, representing the ideal case (linear HPA) and the worst case (non-linear uncompensated case) bound the receiver's performance. The degradation for simulation cases (0.76 dB) is lower to measured case (0.85 dB). This loss of 0.09 dB is probably due to the additional interference induced by the analogue filter, which is

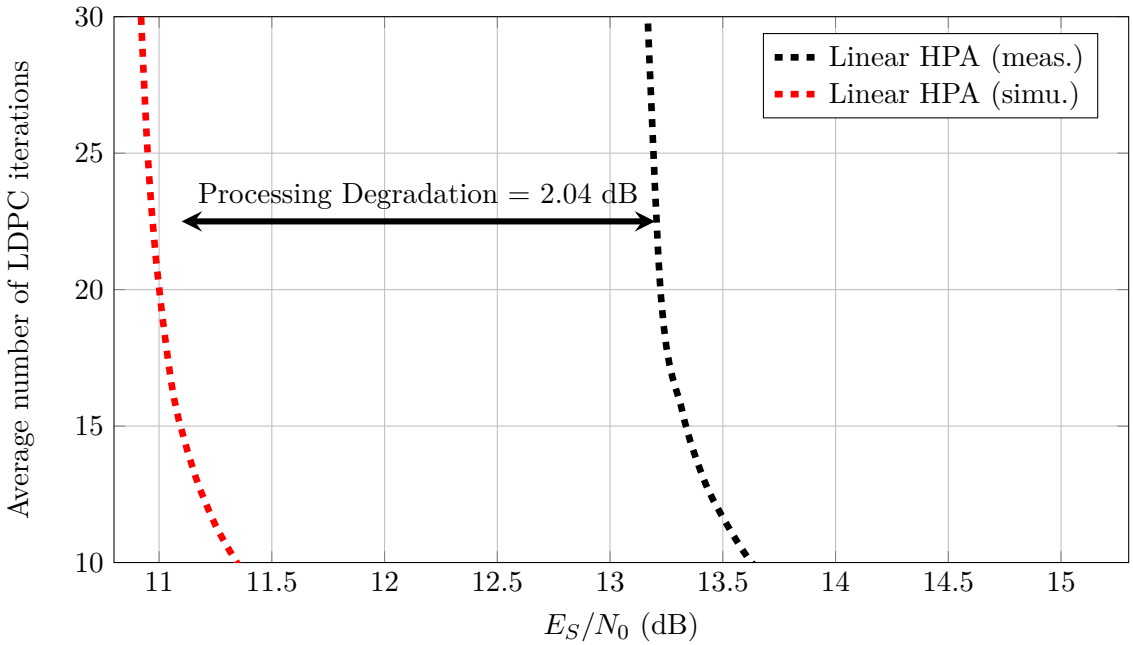



Figure 6.11: Processing degradation induced by the down-conversion block for the Linear HPA case. CR=9/10.

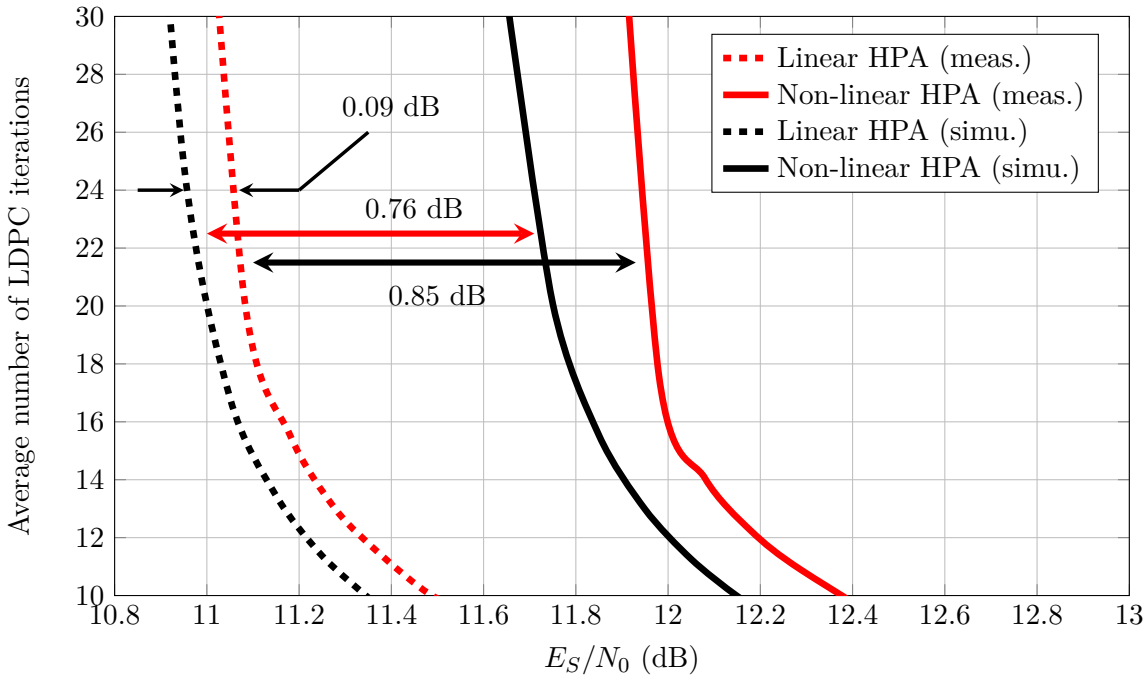



Figure 6.12: Performance comparison between measurements and simulation results. Definitions of performance bounds. CR=9/10. $E_S/N_0$ offset corrected.

not considered in the simulation model. Figure 6.13 illustrates the measured performance of a receiver implementing a stand-alone Volterra Canceller. This figure also shows the simulated performance. It is seen that the measured performance present a lower gain than simulated. The difference is 0.05 dB for 22 average LDPC iterations (0.29 dB for simulations vs. 0.25 dB for measurements). Again, this is due to inaccuracies of the digital design.

The last case considered for CR=9/10 is that of a receiver implementing a Hermite polynomial linearizer combined with a Volterra Canceller. The measured and simulated performance for this receiver configuration are presented on figure 6.14. Again, the measured gain is 0.08 dB lower than simulated. Nevertheless, the improvement in terms of gain due to the combined technique matches that simulated, figure 6.15. The gain of HPL+VC (sim.) case with respect to VC (sim.) case is 0.16 dB while the gain of HPL+VC (meas.) with respect to VC (meas.) is 0.14 dB. However, a gain of 0.14 dB for HPL+VC (meas.) case can be interpreted as an improvement of 60% of the performance of VC (meas.) case. This improvement is bigger than that of HPL+VC (sim.) case with respect to VC (sim.) case, which is 55%.

Additional measurements are done and compared with simulations for LDPC code rates 8/9, 5/6, 3/4 and 2/3. They are depicted on figures 6.16, 6.17, 6.18 and 6.19 respectively.

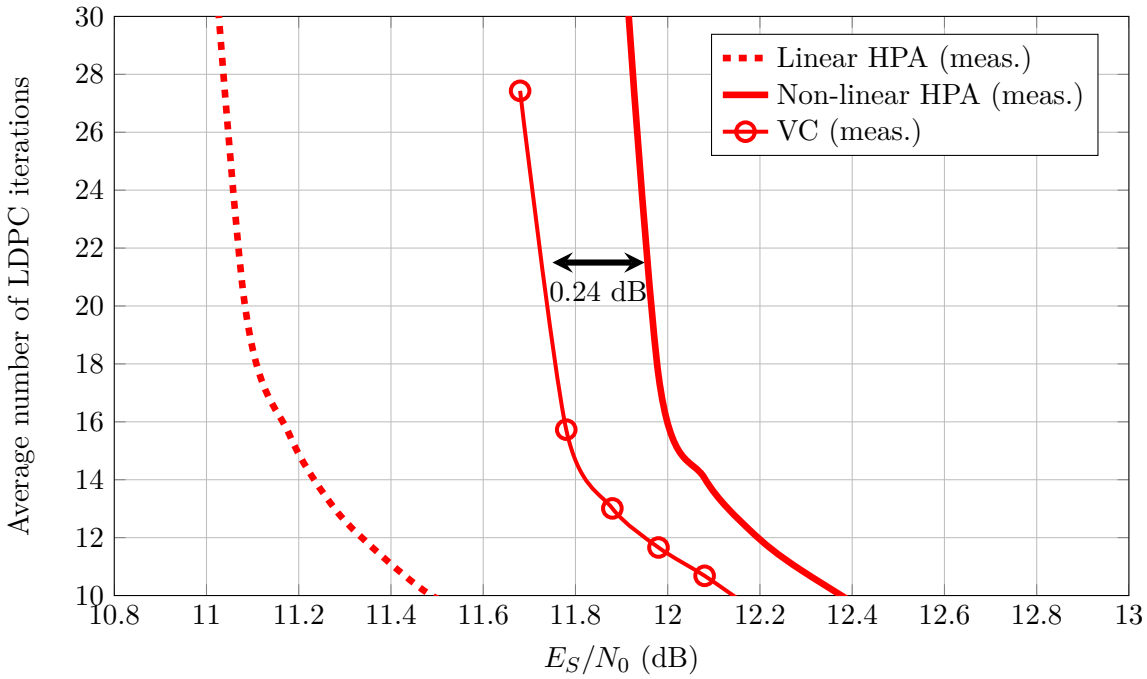



Figure 6.13: Performance comparison for measured and simulated receivers implementing a Volterra Canceller (VC). CR=9/10.

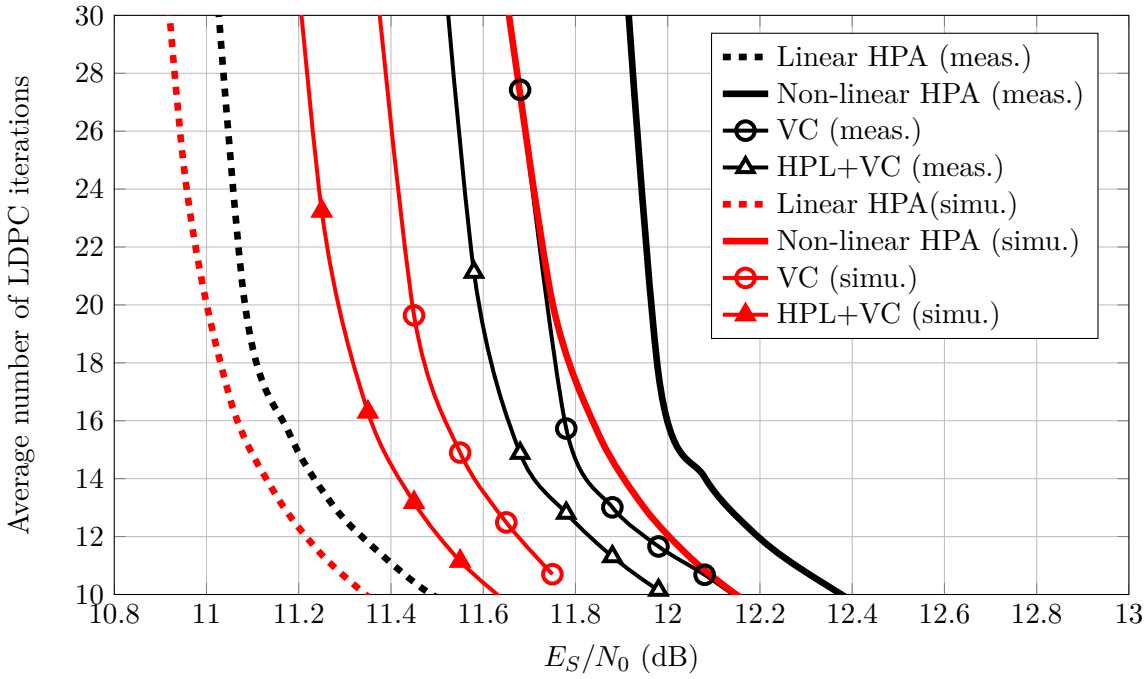



Figure 6.15: Performance measurements vs. simulation results: Average number of LDPC iterations for CR=9/10

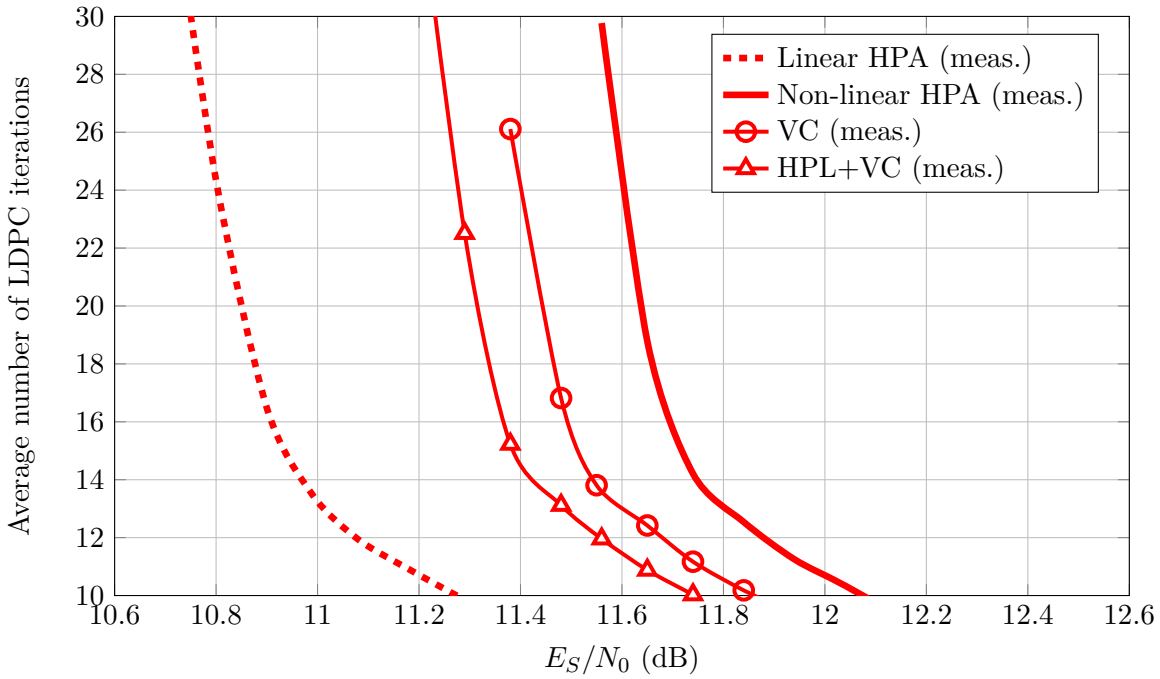



Figure 6.16: Performance measurements vs. simulation results: Average number of LDPC iterations for CR=8/9

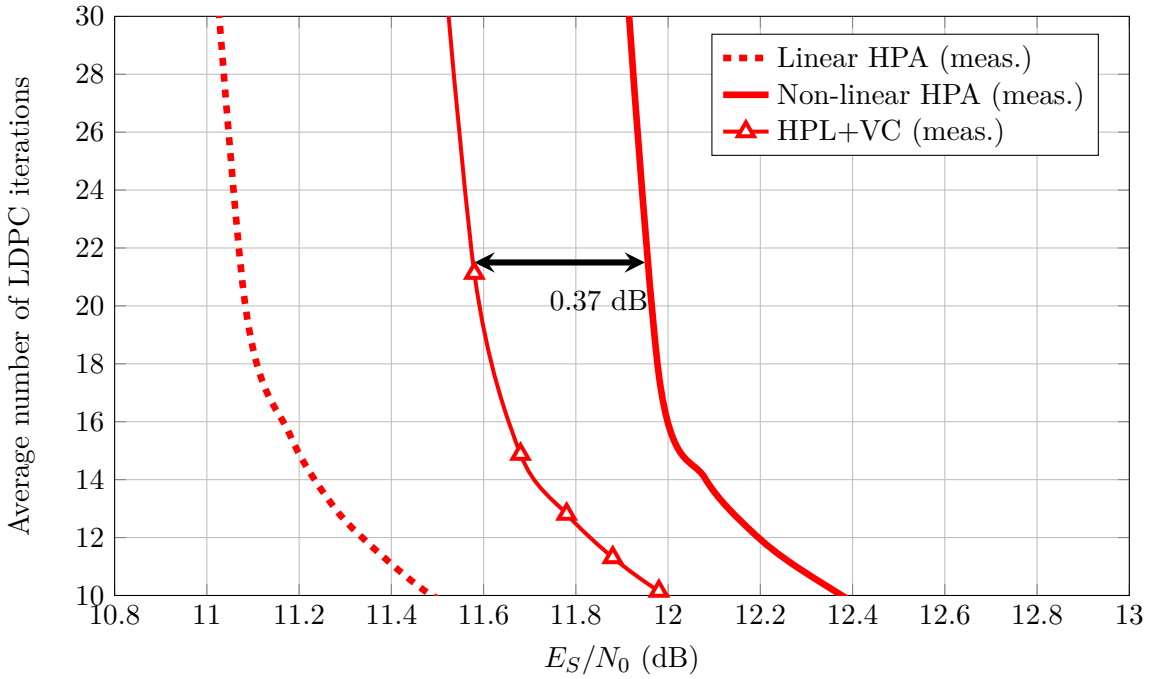



Figure 6.14: Performance comparison for measured and simulated receivers combining a Hermite polynomial linearizer with a Volterra Canceller (HPL+VC). CR=9/10.

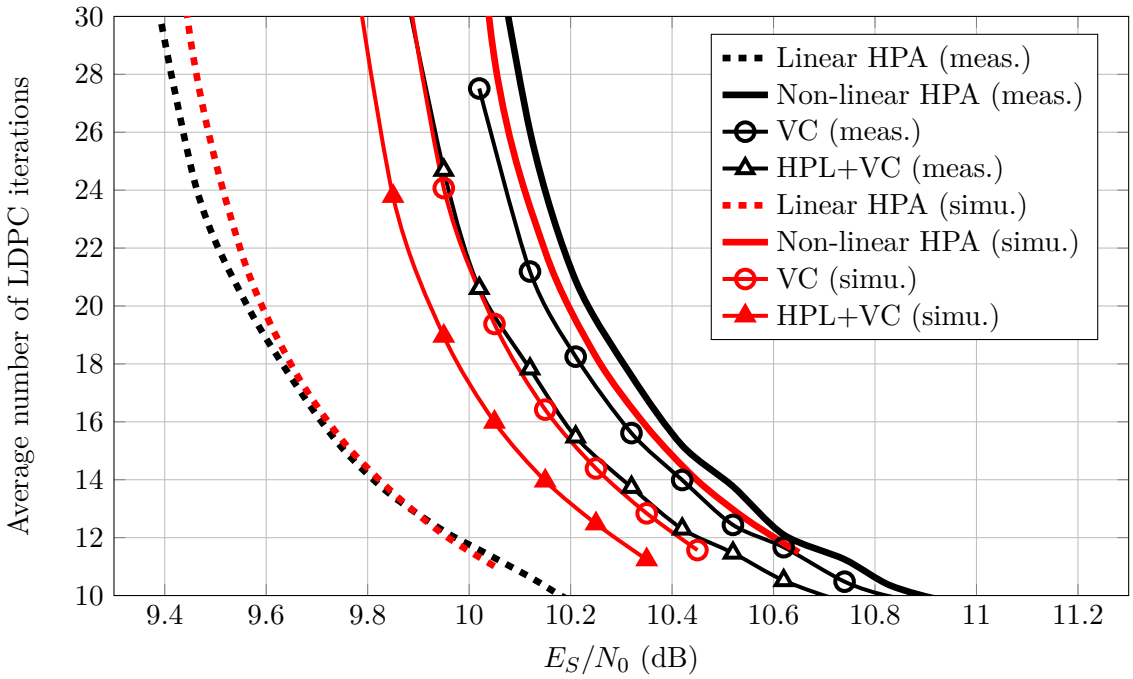



Figure 6.17: Performance measurements vs. simulation results: Average number of LDPC iterations for CR=5/6

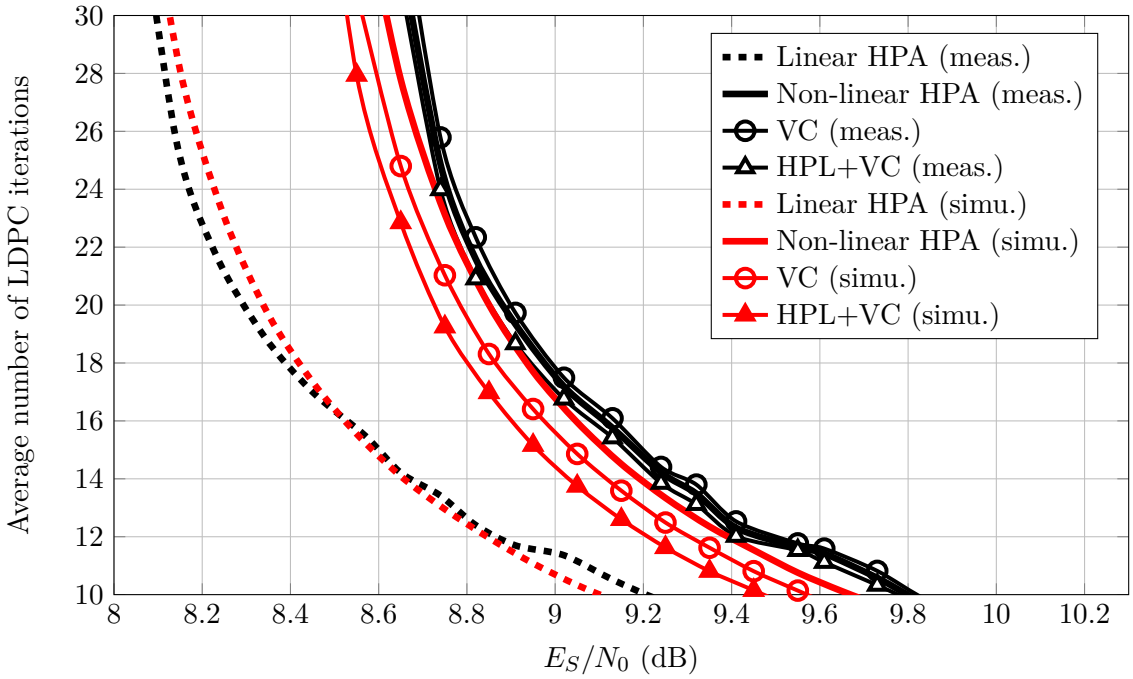



Figure 6.18: Performance measurements vs. simulation results: Average number of LDPC iterations for CR=3/4

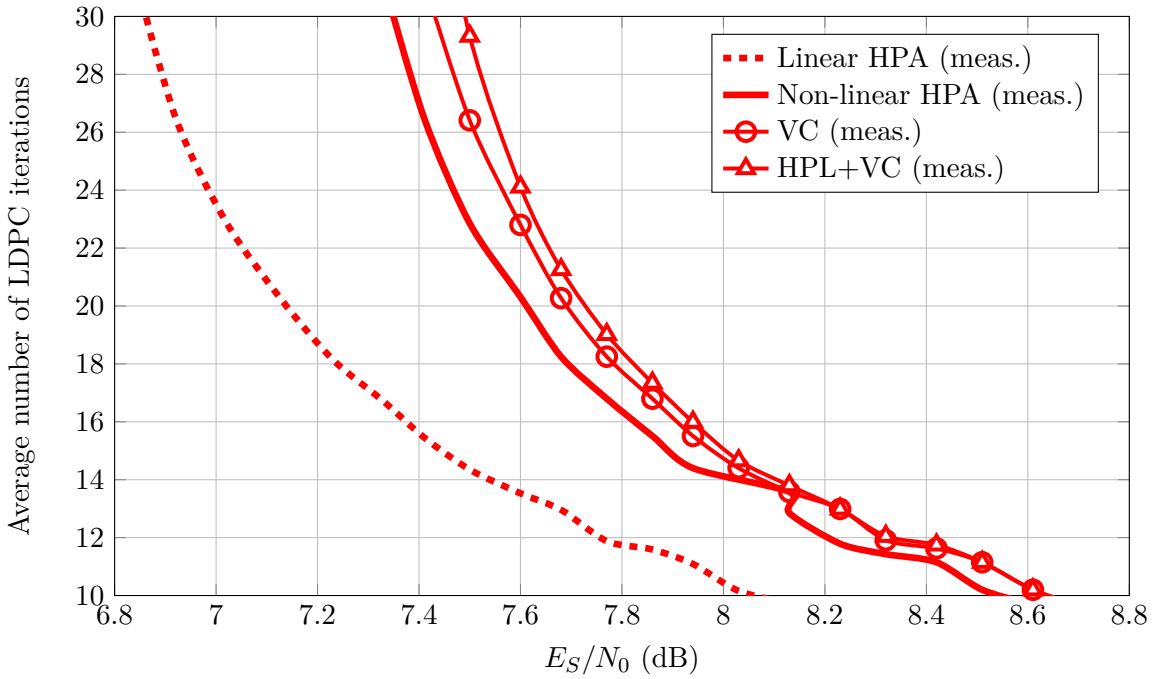



Figure 6.19: Performance measurements vs. simulation results: Average number of LDPC iterations for CR=2/3

The conclusions obtained from the analysis of figures 6.15, 6.16, 6.17, 6.18 and 6.19 are assessed from an average number of LDPC iterations equals to 22. The gains obtained from simulations and measurements for the Linear HPA , VC and HPL+VC cases with respect to the non-linear HPA case without compensation at the receiver are summarized in table 6.1. All the gain values as expressed in dB.

Table 6.1: Gain of $E_S/N_0$ [dB] with respect to uncompensated non-linear case for 22 average LDPC iterations

| *Case* / *CodeRate* | Linear HPA (meas.) | Linear HPA (simu.) | VC (meas.) | VC (simu.) | HPL+VC (meas.) | HPL+VC (simu.) |
|---|---|---|---|---|---|---|
| 2/3 | 0.48 | 0.43 | -0.09 | 0 | -0.13 | 0.01 |
| 3/4 | 0.57 | 0.51 | -0.02 | 0.06 | 0.02 | 0.11 |
| 5/6 | 0.67 | 0.6 | 0.08 | 0.16 | 0.19 | 0.27 |
| 8/9 | 0.81 | 0.74 | 0.22 | 0.28 | 0.34 | 0.43 |
| 9/10 | 0.85 | 0.76 | 0.24 | 0.29 | 0.37 | 0.45 |

As seen in table 6.1, the correlation between measured and simulated results is valid for all the code rates considered. As already noted from simulations results in chapter 4, the Volterra Canceller and the combined technique degrades the receiver performance for code rate 2/3. This is induced by the Volterra Canceller, which estimates the non-linear ISI

term from highly noisy received symbols, i.e. the a-priori mapping is inaccurate. Then, for CR =2/3 the post-distortion technique should be by-passed.

### 6.3.4 Effectiveness of the Volterra Canceller combined with a Hermite polynomial linearizer

In mode of conclusion, this sub-section analyses effectiveness of the combination of a Volterra Canceller with a Hermite polynomial linearizer. As mentioned previously, the effectiveness is stated from two variables: the performance and the complexity. As illustrated on figure 6.8, the implementation of the Volterra Canceller occupies 19% of the FPGA resources while the Hermite polynomial linearizer with the feed-back training only occupy 8%. The resources occupied for a Hermite polynomial linearizer combined with a stand-alone Volterra Canceller are 42% bigger than the resources occupied for the stand-alone Volterra Canceller. Nevertheless, performing Hermite polynomial linearization before the Volterra Canceller improves the performance of the Volterra Canceller by 60% for CR=9/10. This gain is large enough to justify a 42% of additional complexity.



Another important conclusion related to the effectiveness of the post-distortion techniques is obtained by analyzing the additional complexity and the reduction in the average number of LDPC iterations. As shown on figure 6.15, for an $E_S/N_0$ near to 11.7 dB, the average number of iterations is 27 for VC (meas.) case and 15 for the HPL+VC (meas.) case. Then, the average number of LDPC iterations can be reduced in 48% with 42% of additional complexity. An analysis based on the power consumption can be done further. Reducing the number of LDPC iterations helps to save the power of the 59% of the FPGA resources used by the decoder (see figure 6.7). In contrast, the Hermite polynomial linearizer consumes 8% of the resources of the demodulator, i.e. 1.5% of the FPGA resources. Therefore, performing Hermite polynomial linearization is also efficient in terms of power consumptions.

## 6.4 Summary

The performance measurements of a DVB-S2 receiver integrated into a DVB-S2 transmission chain has been the topic of this chapter.

The transmission chain has been implemented with a Test-Bench emulating a DVB-S2 system. The transmitted DVB-S2 signal is generated with a digital TV generator. The non-linear HPA has been implemented into an FPGA by approximation of the HPA characteristics suggested by the DVB-S2 standard. The Gaussian channel noise (AWGN) is generated by a noise generator device. The DVB-S2 receiver is implemented in an FPGA.

As done for computer simulations, the performance of the DVB-S2 receiver have been stated in terms of average LDPC iterations as a function function of the $E_S/N_0$. Four measurement scenarios have been proposed. The first one implements a transmitter with linear HPA. The second, third and fourth scenarios are impaired by HPA non-linearity. The second scenario does not perform non-linear ISI compensation at receiver. The third

scenario implements a stand alone Volterra canceller. Finally, the fourth scenario considers the combination of a Hermite polynomial linearizer with a Volterra Canceller introduced in chapter 5.

The analysis of the FPGA resources allocation shows that the the Volterra Canceller demands for the 19% of the resources of the demodulator while the Hermite polynomial linearizer only needs for the 8%.

The performance measurements have been obtained for different values of LDPC code rate. The measured results are correlated with simulation results. A lower gain is stated for the measurement results of the sand-alone Volterra Canceller and the combined technique with respect to simulation results. However, in terms of effectiveness, the gain obtained by the proposed Hermite polynomial linearizer is large enough to justify the additional complexity.

6

# 7 Conclusions and Perspectives

## 7.1 Conclusions

Digital communication systems are degraded by different types of interferences. Such interferences are mainly induced by analogue blocks mismatches. The main source of non linearity is the Power Amplifier. This is due to power saving requirements necessitating the PA to work in saturation. This Thesis has presented a study of different compensation techniques for the adverses effects induced by non-linear PAs. These compensation techniques can be associated with two families: the pre-distortion and the post-distortion. The pre-distortion family is implemented at transmitter side while the post-distortion is a receiver side technique. As this thesis addresses receiver's architectures, the compensation techniques proposed in this work belong to the post-distortion family. The application case of this work has been the satellite video standard DVB-S2 based on 8-PSK digital modulations and LDPC codes. Thus, the digital design of the post-distortion technique proposed in this work has been optimised for the DVB-S2 standard.

In Chapter 2 a study of typical digital communication systems showed that any perturbation induced by the analogue components results in an undesirable ISI. An important perturbation existing in most of digital communication systems is the non-linear behaviour. The non-linearity is a function of the power efficiency required for each analogue block. The most critical analogue block in terms of power efficiency and the most significant source of non-linearity inside the digital communication system is the PA. Therefore, chapter 2 has introduced the most relevant PA definitions and different models for different types of PA. After, the adverse mismatches associated with the PA non-linearity have been presented. The first mismatch studied has been the spectral spreading, which regrowths the bandwidth of the transmitted signal. The spectral spreading has been characterized from a polynomial model, showing that such a transmitter impairment is a function of the odd order combinations of the transmitted signal and the back-off associated with the PA. The second mismatch associated with the PA non-linearity is the non-linear ISI. The non-linear ISI has been characterized from a base-band satellite system model, showing that the degradation induced by such a mismatch is a function of the back-off as well as the roll-off associated with the digital matched filters at transmitter and receiver sides. Chapter 2 also gave the computer simulations results of a DVB-S2 system implementing a non-linear HPA. These results have confirmed the characterization of the spectral spreading and the non-linear ISI.

Chapter 3 has presented an overview of compensation techniques for non-linear PA impairments. After introduces the widely used Volterra series regression, a brief description of pre-distortion techniques has been done. Such techniques are able to compensate both the spectral spreading and the non-linear ISI. Then, chapter 3 has introduced four different approaches of post-distortion techniques: the Volterra ZFE, the Volterra Canceller, the turbo canceller and finally the Neural Network based post-distortion techniques. The Volterra ZFE mitigates the non-linear ISI by applying the inverse of the non-linear channel transfer function to the received symbols. Such an inverse function is approximated with a Volterra series regression. The Volterra Canceller estimates and eliminates the non-linear ISI term associated with the received symbols. Again, a Volterra series regression is used to estimate the non-linear ISI term. The turbo canceller works together with a turbo decoder, eliminating iteratively the non-linear ISI from the received symbols. Finally, the Neural Network based post-distortion technique implements a multi-layer perceptron architecture to compute the inverse of the non-linear channel transfer function. Chapter 3 has also presented three adaptation algorithms used to train the different post-distortion techniques. The algorithms introduced are the (LS) least-square algorithm, the least-means square (LMS) algorithm and the recursive least-square (RLS) algorithm. It has been demonstrated that the RLS algorithm has the best convergence (time and precision) under high $E_S/N_0$ scenarios. Nevertheless, for low values of $E_S/N_0$ the LMS algorithm is as performer as the RLS algorithm. Therefore, as the LMS algorithm presents the minimal system complexity, this algorithm is chosen to train post-distortion techniques in low $E_S/N_0$ scenarios. The remaining of chapter 3 has shown the comparison of the performance of each post-distortion technique introduced in this chapter. These performance have been obtained from the results proposed in the literature. The better performance are reached by the turbo canceller but the Volterra canceller shows performance near the turbo canceller's performance and a reduced system complexity. In addition, it has been shown that a Volterra Canceller estimating the non-linear ISI term from ideal information outperform the turbo canceller.

Chapter 4 has introduced the major contribution of this thesis: the polynomial linearizer implemented as a post-distortion technique. The polynomial linearizer is based on the estimation of the inverse of the non-linear PA transfer function by means of an orthogonal polynomial regression. The first orthogonal polynomials used to carry-out the regression have been the Laguerre polynomials, which are optimal to approximate non-linear systems. The second polynomials have been the Hermite polynomials, which can be implemented with a reduced system complexity. Chapter 4 has also proposed a combined technique based on a polynomial linearizer working together with a Volterra Canceller. Such a combination helps the Volterra Canceller to work near ideal performance. Another contribution of this chapter has been the introduction of two self-training data generation methods. The implementation of these methods allows the integration of the Volterra Canceller combined with a polynomial linearizer into a digital communication system independently of the application case. The first one generates the training data from the output channel decoder bits. The second method estimates the training data from the Volterra Canceller output. The remaining of chapter 4 has presented a performance comparison between the novel combined technique and the other non-linear ISI compensation techniques proposed in the state-of-the-art. Such a comparison has been obtained from computer simulations

of a DVB-S2 system. The metric used to asses the performance is based on an original method comparing the average number of LDPC iterations needed to decode with no error the transmitted data. The results confirm that the combination of a Volterra Canceller with a Hermite polynomial linearizer outperforms the stand-alone Volterra Canceller, the Volterra ZFE and the Laguerre polynomial combined with a Volterra Canceller.

An innovative digital implementation for the Volterra Canceller and the Hermite polynomial linearizer has been proposed in chapter 5. Such an implementation is designed to work into commercial DVB-S2 receivers. Therefore, the digital design of the combined technique is focused on the minimization of the system complexity. For instance, the mapping block implemented is carried out from the comparison of the distances between the received symbol and the 8-PSK constellation points. The Volterra Combiner avoids the implementation of several complex multipliers needed to carry out the order three combinations of the Volterra regression by using a LUT. The multiplier block is smartly designed in order to use a minimum number of complex multipliers. Finally, the LMS Training Block has been combined with the Multiplier Block output. It computes only half the LMS algorithm. The digital design of the Hermite polynomial linearizer has been implemented with the same criteria as for the Volterra Canceller. The Hermite Polynomial Expansion block computes the values of the Hermite polynomial implementing shift operations to avoid complex multipliers.



Chapter 6 has presented the measurements of the performance of a DVB-S2 receiver. Three configurations were successfully implemented in this receiver. The first configuration does not compensate for the non-linear ISI. The second configuration considers a stand-alone Volterra Canceller. The final configuration combines a Hermite polynomial linearizer with a Volterra Canceller. It has been shown that the complexity of the combined technique is equivalent to the complexity of others blocks of the DVB-S2 receiver. Thus, the goal of low complexity is achieved. The DVB-S2 receiver has been implemented into an FPGA and integrated into a Test-Bench emulating a DVB-S2 transmission chain. Measurement results have shown a small gain degradation with respect to simulation results. However, the relative gain, i.e. the gain of a Hermite polynomial linearizer combined with a Volterra Canceller with respect to the stand-alone Volterra Canceller, is highly correlated to that obtained in simulations. Moreover, the trade-off between the added system complexity and the gain obtained in the performance suggests that the combined technique is effective.

## 7.2 PERSPECTIVES

This section proposes few perspectives for this Thesis:

### 7.2.1 Multi-standard Hermite polynomial linearizer combined with Volterra Canceller

Chapter 5 has presented the architecture and the digital design of a Hermite polynomial linearizer combined with a Volterra Canceller. This combined compensation system is carried out with minimal system complexity for the application case of commercial DVB-S2 receivers (considering 8-PSK digital modulation). However, this digital design is not compatible with a DVB-S2 receiver considering 16-APSK and 32-APSK digital modulations or with other application cases such as terrestrials or cable TV standards (considering N-QAM digital modulations). Therefore, such an architecture able to work independently of the digital modulation implemented would be a major improvement for the combined technique presented in this work. Nevertheless, a multi-digital modulation architecture prevents the implementation of most of the simplifications used in the design of the combined technique.

7

### 7.2.2 Polynomial linearizer compensating for non-linear multi-path channels

The application case of this work was based on a satellite TV standard, considering a non-linear HPA. There are others standards working with non-linear PAs and also impaired by other mismatches such as a multi-path channel. Hence, this subsection deals with the implementation of a polynomial linearizer into a non-linear multi-path channel. As introduced in section 4.1, the Hermite polynomial linearizer is implemented from the following regression

$$y'_t = \sum_{j=0}^{T} p_{2j+1} H_{2j+1}(y_t) = \sum_{j=0}^{T} p_{2j+1} H_{2j+1}(x'_t + n_t). \tag{7.1}$$

Where $y_t$ is the noisy non-linear received signal, $y'_t$ is the linearized signal, $x'_t$ is the PA output, $n_t$ is the Gaussian noise, $T$ is the order of the polynomial regression and $p_{2j+1}$ is the Hermite coefficient multiplying the $H_{2j+1}$ polynomial. Such a linearizer is adapted by minimizing the error function defined as:

$$J = E|x_t - y'_t|^2 = E|x_t - \sum_{j=0}^{T} p_{2j+1} H_{2j+1}(y_t)|^2. \tag{7.2}$$

Where $x_t$ is the training data, which is a replica of the PA input. In the case of a non-linear multi-path channel, the received signal becomes:

$$y_t = \sum_{j=-R}^{R} \alpha_j x'_{t-j} + n_t. \tag{7.3}$$

Where $R$ is the memory of the multi-path channel and $\alpha_j$ is the weight associated with the memory order $j$. Therefore, in order to linearize the multi-path non-linear channel,

the Hermite polynomial linearizer must not only considers a regression as a function of $y_t$ but as a function of the delayed versions of $y_t$, leading to a Hermite non-linear equalizer.

This compensation technique is a generalized version of the Hermite polynomial linearizer.

### 7.2.3 Implementation of smart adaptation methods

The LMS and RLS training methods considered in this Thesis where implemented with fixed step ($\mu$ for LMS and $\gamma$ for RLS). This subsection considers three methods to optimise in terms of convergence and precision the algorithms implemented in the training block of the Volterra Canceller and the Hermite polynomial linearizer.

The convergence and the precision of the LMS algorithm depends on the step value $\mu$ (section 3.4). The optimal value of $\mu$ is in fact a function of the $E_S/N_0$ scenario of the target system to train. Thus, a way to optimise the LMS algorithm is by implementing variable LMS step algorithms as presented in [CO12] and [YZWY10]. It is shown that implementing these algorithms enhances the convergence time and the precision of the algorithm. Thus, as the period of training is reduced, the power consumption can be reduced instead of the additional complexity added by the estimation of the optimal step. A similar study is done in [Alb12] to optimise the RLS algorithm. Such a work addresses a method to estimate the optimal forget factor $\gamma$ associated with the RLS algorithm. Again, this method increases the system complexity but improves the performance of the RLS algorithm.

The implementation of the Volterra Canceller proposed in this Thesis only considers a limited number of order one and order three combinations. Such combinations has been determined in order to match optimally with the HPA implemented in this work. Nevertheless, it would be appropriate to carry out a Volterra Canceller considering a flexible number of linear and non-linear combinations. Thus, if the memory and the level of the non-linearity is weak, the Volterra Canceller can be carried out from a reduced number of combinations. In contrast, for high levels of non-linear ISI, the number and the order of combinations can grow. Such a flexibility can be reached by using the smart adaptation algorithm proposed in [BKT10], the SPARSE RLS algorithm. Thus, considering a Volterra Canceller which can be carried out from $B$ linear and non-linear combinations, the SPARSE RLS algorithm choses a set of $\beta$ combinations ($\beta \leq B$) to carry out the estimation of the non-linear ISI term. The set of $\beta$ combination is chosen with a criterion of optimisation associated with a tolerated error. The implementation of the SPARSE RLS algorithm does not help to reduces the system complexity but offers an additional way of adaptation to the system. Then, better performance and reduced power consumption can be reached.

### 7.2.4 Implementation of different a-priori mapping methods

The a-priori mapping block has been implemented into the Volterra Canceller and into the direct decision feed-back training method proposed in section 4.4. In the case of the Volterra Canceller, the a-priori mapping determines the precision of the estimation of the

non-linear ISI term. With respect to the direct decision feed-back training method, the a-priori mapping estimates the training data used to adapt the Volterra Canceller and the Hermite polynomial linearizer to the non-linear channel. Therefore, the a-priori mapping is a critical block defining the performance of the whole combined technique proposed in this work.

This work has considered an a-priori mapping method based on the computation of the minimal distance of a received symbol with respect to the constellation points. This a-priori mapping method has been chosen because of its low system complexity, which is the main target for commercial receivers. However, [PS08] proposes others a-priori mapping methods showing better precision but higher system complexity than the method implemented in this work. Therefore, it would be interesting to study the improvement in the performance and the extra system complexity added by others a-priori mapping methods.

### 7.2.5 ASIC implementation



The last suggestion for a future work complementing this Thesis is the application specific for integrated circuit (ASIC) conception of the DVB-S2 receiver. This conception must include the Volterra Canceller and the Hermite polynomial linearizer. The ASIC implementation can help determine the actual system complexity added for the compensation system. The circuit can also help compare accurately the difference of the system complexity with respect to others post-distortion techniques. Another advantage of the ASIC conception is the possibility of measuring actual values of power consumption, which is another important factor in commercial receivers.

# A

# The DVB-S2 standard

The application case considered in this Thesis is a digital viedeo satellite system based on the DVB-S2 standard. Such a standard has been implemented in both the system model used during the computer simulations and the actual test-bench where a DVB-S2 receiver was implanted into an FPGA.

The present appendix summarizes the most relevant characteristics of the DVB-S2 standard introduced in [DS06]. This appendix is organised as follows. Section A.1 presents the generalities of the DVB-S2 standard. Section A.2 details the DVB-S2 stream. The LDPC and BCH codes used in the standard as well as the bit interleaving are presented in section A.3. Section A.4 describes all the digital modulations considered in the standard. Section A.5 introduces the DVB-S2 framing. Section A.6 describes the base-band filtering based on root-raised cosine filters. Finally A.7 summarizes this appendix.

## A.1 DVB-S2 STANDARD GENERALITIES

The blocks characterizing the digital base-band part of DVB-S2 transmitter are presented on figure A.1. The input data stream is the binary information to be transmitted. The DVB-S2 considers various stream formats. Such a stream is then processed by the stream adaptation block, which provides padding to complete the frame of constant length and then scrambling. The adapted stream is then encoded (BCH and LDPC encoding) and interleaved. The encoded stream is then mapped onto constellations points, and then processed by a framing block, which add the necessary header and pilot information used during the synchronization at the receiver side. Further, the transmitter applies scrambling to the constellation points. Finally, the constellation points are digitally filtered by means of a root raised cosine digital matched filter.

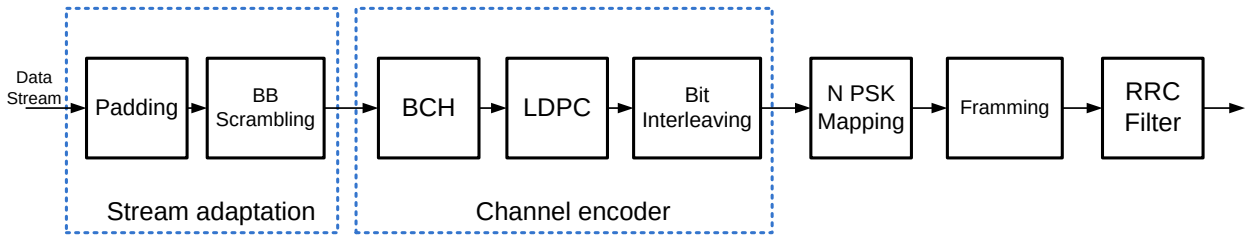



Figure A.1: Bock diagram for a DVB-S2 transmitter

## A.2 DVB-S2 STREAM ADAPTATION

The stream adaptation processes the input data stream in order to provide a complete constant length base-band frame (BB-Frame). Indeed, the stream adaptation is based firstly on applying a padding block. Such a padding block completes with zero bits the the input data stream, resulting in a constant stream composed of $n_{BB-Frame}$ bits.

After padding, the complete BB-frame is randomized by a scrambling block. Such a randomization is carried out from a pseudo random binary sequence (PRBS) polynomial.

## A.3 CHANNEL CODING

The channel coding characterizing the DVB-S2 standard is based on an outer coding (BCH cades), an inner coding (LDPC codes) and finally a bit interleaving block. The BB-frame with length $n_{BB-Frame}$ is firstly processed by the systematic BCH encoder. Thus, the output stream is the BB-frame with the BCH parity check bits BCH-R. The length of BCH-R is $n_{BCH-R}$. Then, the BB-Frame and the BCH-R frame are encoded by the systematic LDPC block. Such a block appends the LDPC parity check bits LDPC-R of length $n_{LDPC-R}$. The LDPC code rates (CR) considered for the DVB-S2 standard and the digital modulation including such code rates are shown in table A.1. The LDPC encoder output stream is detailed on figure A.2.



Table A.1: LDPC code-rate and digital modulation modes for the DVB-S2 standard

| *Modulation* / *LDPCCR* | Q-PSK | 8-PSK | 16-APSK | 32-APSK |
|---|---|---|---|---|
| 1/4 | √ | | | |
| 1/3 | √ | | | |
| 2/5 | √ | | | |
| 1/2 | √ | | | |
| 3/5 | √ | √ | | |
| 2/3 | √ | √ | √ | |
| 3/4 | √ | √ | √ | √ |
| 4/5 | √ | | √ | √ |
| 5/6 | √ | √ | √ | √ |
| 8/9 | √ | √ | √ | √ |
| 9/10 | √ | √ | √ | √ |

On figure A.2, the value of $n_{LDPC}$ is $n_{LDPC} = 64800$ for normal frames and $n_{LDPC} = 16200$ for short frames. After applying the input and output encoders, the LDPC output stream is bit interleaved by a block interleaved. The interleaved stream is known as

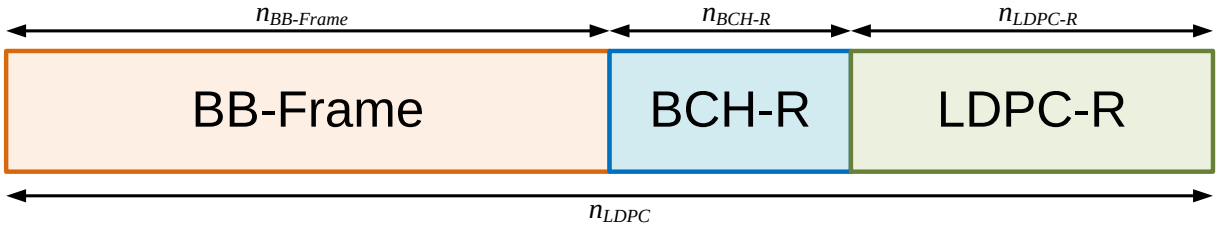



Figure A.2: Description of the LDPC output stream

FEC-Frame with length $n_{LDPC}$. The configuration of the interleaving depends on the digital modulation considered by the DVB-S2 system. The case of an interleaving configured for 8-PSK digital modulation is detailed on figure A.3.

As seen on figure A.3, the LDPC output stream is serially written into three columns and serially read out row-wise [DS06]. The most significant bit (MSB) of the LDPC output stream is read out first. The number of columns for a 16-APSK digital modulation is four and five for a 32-APSK digital modulation. The bit interleaving is not considered when

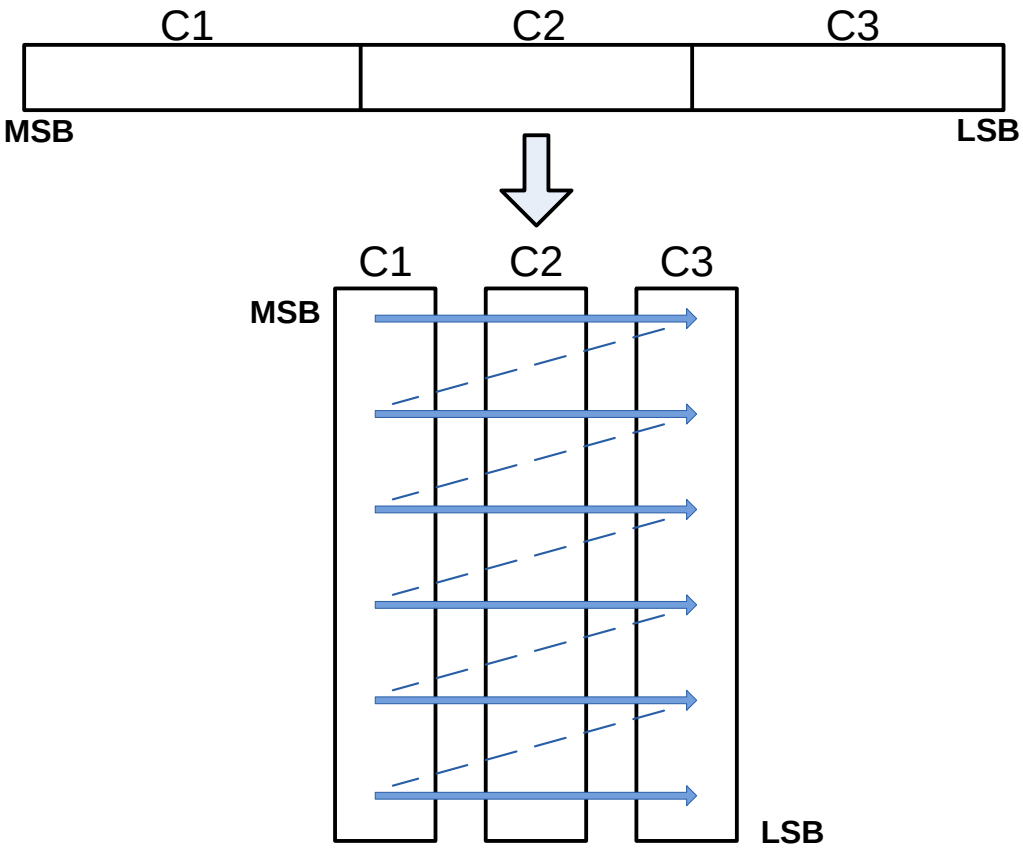



Figure A.3: Bit interleaving

the Q-PSK digital modulation is implemented.

## A.4 Digital modulation

As presented before, the DVB-S2 standard considers 4 different configuration of PSK digital modulations. Such configurations are Q-PSK, 8-PSK, 16-APSK and 32-APSK. The mandatory cases for the DVB-S2 receiver are the Q-PSK and the 8-PSK, while the 16-APSK and the 32-APSK are considered in professional applications.

The constellation characterizing the Q-PSK digital modulation is depicted on figure A.4. Such a constellation is composed of four Grey encoded points. The phase for the point encoded with the sequence "00" is $\pi/4$. The phase between two consecutive constellation points is $\pi/2$. The energy per constellation point is equal to one.

Figure A.5 depicts the 8-PSK constellation considered in the DVB-S2 standard. Such a constellation employs conventional Grey coding. The phase characterizing the constellation point encoded with "000" is $\pi/4$ and the phase between two consecutive points is also $\pi/4$. As in the case of Q-PSK, the energy per constellation point is equal to one.

The constellation points characterizing a 16-APSK digital modulation are shown on figure A.6. Such a constellation is composed of two concentric rings of constellation points. The ring of radius $\rho_1$ is composed of 12 grey encoded points while the ring of radius $\rho_2$ contains four Grey encoded points. The values of $\rho_1$ and $\rho_2$ must respect the following equation:

$$4\rho_1^2 + 12\rho_2^2 = 16. \tag{A.1}$$

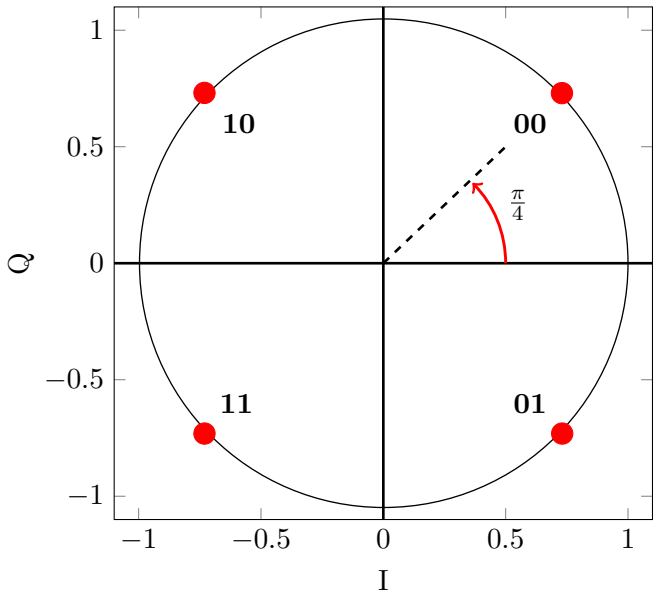



Figure A.4: Bit mapping for Q-PSK constellation

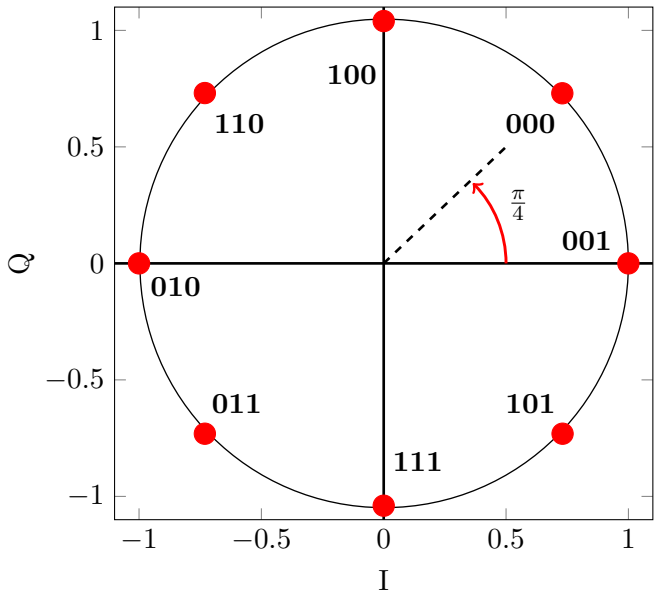



Figure A.5: Bit mapping for 8-PSK constellation



If the condition of (A.1) is respected, the average signal energy is equal to one.

As seen on figure A.6, the symbol of the inner ring encoded with "1100" has a phase equal to $\pi/4$. The phase between two consecutive constellation points of the inner ring is of $\pi/2$. For the outer ring, the phase of the symbol encoded with the binary word "0100" is $\pi/12$ and the phase between two consecutive points is $\pi/6$.

The constellation characterizing a 32-APSK constellation is not illustrated on this appendix. Such a constellation is composed of three rings, the inner one having four points, the second one 12 points and the outer one 16 points.

## A.5 DVB-S2 FRAMING

The DVB-S2 framing is based on the insertion of a header and pilots, and then applying physical layer scrambling in order to randomize the transmitted symbols (N-PSK/APSK constellation points with header and pilots). The DVB-S2 physical layer frame is described on figure A.7. Such a frame is composed of a header, several slots of 90 N-PSK/APSK symbols each and pilots blocks.

The DVB-S2 header is composed of 90 symbols. The first 26 symbols codes the value $18D2E82_{HEX}$, which characterizes the start of frame (SOF). The remaining symbols codes the physical layer signalling (PLS), which identify the digital modulation, code rate and the kind of frames (short or normal) characterizing the transmission.

The pilots block is composed of 36 Q-PSK symbols identified by the complex value

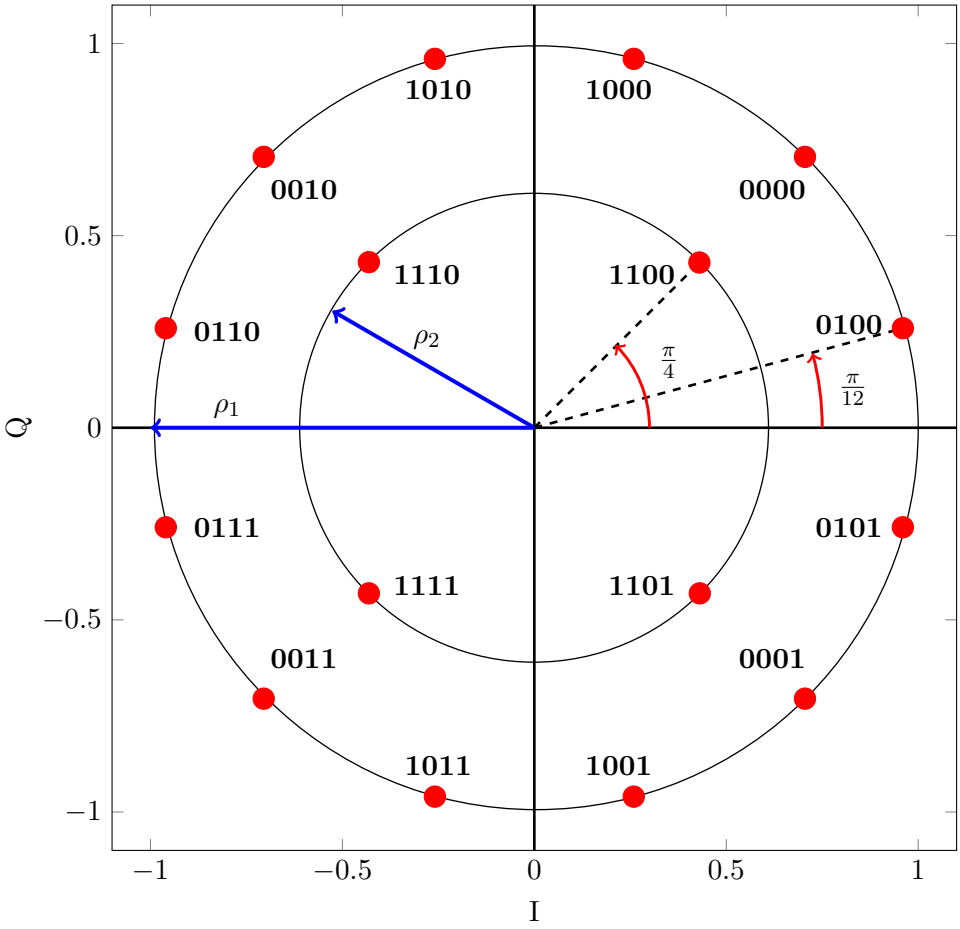



Figure A.6: Bit mapping for 16-APSK constellation

$(1/\sqrt{2}, 1/\sqrt{2})$. The first pilot block is inserted 16 slots after the header. The remaining of the pilot blocks are separated by 16 slots.

The scrambling block processes the physical layer frame, excluding the header. Such a randomization is carried out by multiplying the symbols by a complex randomization sequence.

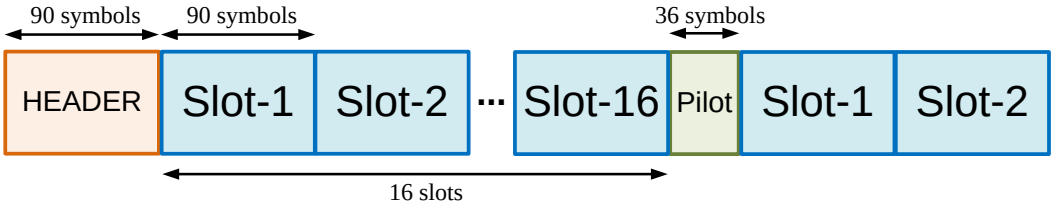



Figure A.7: Physical layer DVB-S2 frame

## A.6 Baseband filtering

The base-band filtering considered by the DVB-S2 standard is based on root raised cosine filtering. The roll-off factor ($\alpha$) takes the values 0.35, 0.25 or 0.2. The theoretical function defining such a filter is:

$$H_f = \begin{cases} 1 & \text{for} |f| < f_N(1-\alpha) \\ \left\{ \frac{1}{2} + \frac{1}{2} sin\left[ \frac{\pi}{2f_N} \left( \frac{f_N - |f|}{\alpha} \right) \right] \right\}^{1/2} & \text{for} f_n(1-\alpha) \leq |f| \leq f_n(1+\alpha) \\ 0 & \text{for} |f| > f_n(1+\alpha) \end{cases} \quad \text{(A.2)}$$

## A.7 Summary

This appendix has introduced the most relevant blocks characterizing the DVB-S2 standard. The description of the standard has been made from a DVB-S2 transmitter which blocks are grouped on five blocks: the DVB-S2 stream adaptation, the channel encoder, the digital modulator, the DVB-S2 framing block and the base-band filter.

A

# B

# Acronyms

| | |
|---|---|
| ADC | Analogue-to-digital converter |
| ALM | Adaptive module logic |
| AM/AM | Amplitude-to-amplitude transfer function |
| AM/PM | Amplitude-to-phase transfer function |
| APSK | Amplitude phase-shift keying |
| ASIC | Application specific for integrated circuit |
| AWGN | Additive white Gaussian noise |
| BER | Bit-error rate |
| CFC | Carrier frequency correction |
| CR | Code rate |
| DAC | Digital-to-analogue converter |
| DC | Direct current |
| DGC | Digital gain control |
| DNL | Differential non-linearity |
| DVB-S2 | Digital Video Broadcasting - Satellite 2 |
| FPGA | Field-programmable gate array |
| HPA | High power amplifier |
| HPL | Hermite polynomial linearizer |
| IBO | Input back-off |
| INL | Integral non-linearity |
| ISI | Inter-symbol interference |
| LDPC | Low-density parity check |
| LIN | Linearizer block |
| LLR | Log-likelihood ratio |
| LMS | Least-means squares |
| LNA | Low-noise amplifier |
| LO | Local oscillator |
| LPL | Laguerre polynomial linearizer |
| LS | Least-squares |
| LSB | Least-significant bit |
| LUT | Look-up table |
| MLP | Multi-layer perceptron |
| MSE | Means square error |
| OBO | Output back-off |

| | |
|---|---|
| PA | Power amplifier |
| PAPR | Peak-to-average power ratio |
| PSK | Phase-shift keying |
| QAM | Quadrature amplitude modulation |
| RF | Radio frequancy |
| RLS | Recursive least-squares |
| ROT | rotator block |
| RRC | Root-raised cosine |
| SER | Symbol-error rate |
| SSPA | Solid-state power amplifier |
| STC | Sampling time correction |
| TD | Total degradation |
| TV | Television |
| TWTA | Traveling-wave tube amplifier |
| V-ZFE | Volterra zero-forcing equalizer |
| VC | Volterra Canceller |
| VHDL | Very-high-speed integrated circuits hardware description language |
| ZFE | Zero-forcing equalizer |

# C

# Notations

| | |
|---|---|
| $\angle(\cdot)$ | Function determining the phase of a complex value |
| $\vec{\nabla} J$ | Vector gradient of the error cost function |
| $\Re(\cdot)$ | Real part of a complex value |
| $\alpha_i$ | i$^t h$ ISI coefficient |
| $\gamma$ | RLS forgetting factor |
| $\Delta B$ | Bandwidth |
| $\zeta$ | PA polynomial model coefficient |
| $\eta$ | PA efficiency |
| $\rho$ | Curve fitting parameter for Rapp PA model |
| $\nu$ | Amplifier gain |
| $\lambda_i$ | Auto-correlation matrix eigenvalue |
| $\mu$ | LMS step |
| $\mu_{max}$ | Maximal value for LMS step guaranteeing convergence |
| $\mathbf{\Phi}$ | Auto-correlation matrix |
| $a_i$ | Transmitted bit |
| $\hat{a}_i$ | Received bit |
| $\mathbf{A}$ | Delayed input matrix |
| $Am(\cdot)$ | AM/AM function |
| $Am_S(\cdot)$ | AM/AM function for Saleh model |
| $Am_R(\cdot)$ | AM/AM function for Rapp model |
| $c_i$ | Transmitted encoded bit |
| $C_n$ | Centroid value |
| $DD(\cdot)$ | Godard function |
| $E(\cdot)$ | Average value function |
| $e_n$ | Estimation error |
| $e_n^2$ | Square error |
| $f_n$ | Frequency normalised with respect to the symbol rate |
| $f_s$ | Symbol rate |
| $g_i$ | Gain Saleh function parameter |
| $\mathbf{hc_i}$ | Hermite convolutions vector |
| $\mathbf{hg_i}$ | Hermite gradient vector |
| $H_i(\cdot)$ | Hermite polynomial |
| $h_{i,n}$ | Hidden layer neuron output |
| $HLF(\cdot)$ | Hidden layer neuron non-linear function |

| | |
|---|---|
| $\mathbf{ht_i}$ | Hermite taps vector |
| $\mathbf{htt_i}$ | Accumulator vector for Hermite taps |
| $I_n$ | Non-linear ISI |
| $J_n$ | Error cost function |
| $\mathbf{k_n}$ | RLS gain vector |
| $L_i$ | Laguerre polynomial |
| $Lin(\cdot)$ | Linearizer transfer function |
| $llr_i$ | LLR value for the i$^{th}$ received bit |
| $n_t$ | Gaussian channel noise (AWGN) |
| $nn_n$ | Linear combination of $n_t$ |
| $OLF(\cdot)$ | Output layer neuron linear function |
| $P_{DC}$ | Power supply |
| $Ph(\cdot)$ | AM/PM function |
| $Ph_s(\cdot)$ | AM/PM function for Saleh model |
| $p_i$ | Added input power |
| $p_{in}$ | Input signal power |
| $\bar{p_{in}}$ | Means value of the input signal power |
| $\hat{p_{in}}$ | Normalised input signal power |
| $p_{in,sat}$ | Maximal signal input power |
| $p_n$ | Pre-distorted symbol |
| $\hat{p}_n$ | Training symbol |
| $\mathbf{P_n}$ | Inverse auto-correlation matrix |
| $p_o$ | Added output power |
| $p_{out}$ | Output signal power |
| $\bar{p_{out}}$ | Means value of the output signal power |
| $\hat{p_{out}}$ | Normalised output signal power |
| $p_{out,sat}$ | Maximal output signal power |
| $r_i$ | RRC filter coefficient |
| $R$ | roll-off value |
| $RXM(\cdot)$ | Transfer function for receiver side analogue components |
| $s_n$ | Transmitted symbol |
| $\hat{s}_n$ | A-priori mapping of the received symbol |
| $s'_n$ | Received symbol |
| $s''_n$ | Compensated symbol |
| $\check{s}_n$ | Estimation of $s'_n$ without non-linear ISI |
| $s_{I,n}$ | Real part of the transmitted symbol |
| $s_{Q,n}$ | Complex part of the transmitted symbol |
| $TXM(\cdot)$ | Transfer function for transmitter side analogue components |
| $\mathbf{vc_i}$ | Volterra convolutions vector |
| $\mathbf{vg_i}$ | Volterra gradient vector |
| $\mathbf{vt_i}$ | Volterra taps vector |
| $\mathbf{vtt_i}$ | Accumulator vector for Volterra taps |
| $\mathbf{w}$ | Coefficient vector for LS, LMS or RLS estimation |
| $w_{i,j}$ | Weigth matrix element for MLP |

| | |
|---|---|
| $x_i$ | Transmitted signal |
| $x'_i$ | Transmitted signal after PA |
| $x_{I,i}$ | Real part of the transmitted signal |
| $x_{Q,i}$ | Complex part of the transmitted signal |
| $x_{pb}$ | Pass-band transmitted signal |
| $\mathbf{x}$ | Input data vector |
| $y_t$ | Received signal |
| $y'_t$ | Linearized signal |
| $\mathbf{y}$ | Training data vector |
| $\mathbf{z}$ | Cross-correlation vector |

C

C

# Bibliography


[AD08] ADC-DAC. *Data Conversion HSMC Reference manual*. Altera Company, 2008.

[AGA06] A. Aghasi, A. Ghorbani, and H. Amindavar. Polynomial based predistortion for solid state power amplifier nonlinearity compensation. In *Circuits and Systems, 2006 IEEE North-East Workshop on*, page 181–184, 2006.

[Alb12] F. Albu. Improved variable forgetting factor recursive least square algorithm. In *Control Automation Robotics Vision (ICARCV), 2012 12th International Conference on*, pages 1789–1793, 2012.

[Alt11] Altera. *STRATIX IV Device Handbook Volume 1*. Altera Company, 2011.

[ARB+08] D. Ampeliotis, A.A. Rontogiannis, K. Berberidis, M. Papaleo, and G.E. Corazza. Turbo equalization of non-linear satellite channels using soft interference cancellation. In *Advanced Satellite Mobile Systems, 2008. ASMS 2008. 4th*, pages 289 –292, August 2008.

[AWG90] AWGN. *Noise and interference Test Set - Model 3708A*. Hewlett-Packard, 1990.

[BB83] S. Benedetto and E. Biglieri. Nonlinear equalization of digital satellite channels. *Selected Areas in Communications, IEEE Journal on*, 1(1):57–62, 1983.

[BC02] C.E. Burnet and W.G. Cowley. Intersymbol interference cancellation for 16QAM transmission through nonlinear channels. In *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, pages 322–326, 2002.

[BC05] C.E. Burnet and W.G. Cowley. Performance analysis of turbo equalization for nonlinear channels. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 2026 –2030, September 2005.

[BE03] A. Behravan and T. Eriksson. Baseband compensation techniques for bandpass nonlinearities [RF front-ends]. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 1, page 279–283, 2003.

[Bei11] Bassel F. Beidas. Intermodulation distortion in multicarrier satellite systems: analysis and turbo volterra equalization. *Communications, IEEE Transactions on*, 59(6):1580–1590, 2011.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Technical Program,*

*Conference Record, IEEE International Conference on Communications, 1993. ICC '93 Geneva*, volume 2, pages 1064–1070 vol.2, 1993.

[BJ10] I. Baig and V. Jeoti. Papr reduction in ofdm systems: Zadoff-chu matrix transform based pre/post-coding techniques. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on*, pages 373–377, 2010.

[BKT10] B. Babadi, N. Kalouptsidis, and Vahid Tarokh. Sparls: The sparse rls algorithm. *Signal Processing, IEEE Transactions on*, 58(8):4013–4025, 2010.

[BM10] S. Boumard and A. Mammela. The effect of power control on the average power amplifier efficiency. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 629–633, 2010.

[BS10] B. Beidas and R. Seshadri. Analysis and compensation for nonlinear interference of two high-order modulation carriers over satellite link. *IEEE Transactions on Communications*, 58(6):1824–1833, June 2010.

[Bur06] Craig Edward Burnet. *Mitigation of Adverse Effects in the Nonlinear Satellite Channel for 16QAM Transmission.* PhD thesis, University of South Australia, 2006.

[BWM+10] C. Baylis, L. Wang, M. Moldovan, J. Martin, H. Miller, L. Cohen, and J. De Graaf. Designing for spectral conformity: Issues in power amplifier design. In *Waveform Diversity and Design Conference (WDD), 2010 International*, pages 000220–000223, 2010.

[CLLJ12] E. Cabanillas, D. Lohy, C. Lahuec, and M. Jézéquel. Compensating the high power amplifier nonlinearity for a DVB-S2 system. *IEEE NEWCAS 2012*, page 1–4, 2012.

[CLLJ13] E. Cabanillas, D. Lohy, C. Lahuec, and M. Jézéquel. Efficient nisi compensation technique for a low-cost satellite video receiver. *IEEE ICC 2013*, pages 4157–4160, 2013.

[CO12] F. Cakir and A. Ozen. A novel variable step size lms algorithm employing cross correlation between channel output and error signal. In *Signal Processing and Communications Applications Conference (SIU), 2012 20th*, pages 1–4, 2012.

[Cou90] Coupler. *HH-108 High-Power Hybrid Junction 200 KHz - 35 MHz.* Tyco-Electronics, 1990.

[DJB95] C. Douillard, M. Jezequel, and C. Berrou. Iterative correction of intersymbol interference: turbo equalization. *ETT*, 6(5):507–511, 1995.

[DS06] DVB-S2. Digital video broadcasting (dvb); second generation framing structure, channel coding and modulation systems for broadcasting,

interactive services, news gathering and other broadband satellite applications. 2006.

[Epp10] S.S. Epp. *Discrete Mathematics with Applications*. Brooks/Cole, 2010.

[FB08] B. Farhang-Boroujeny. A square-root nyquist (m) filter design for digital communication systems. *Signal Processing, IEEE Transactions on*, 56(5):2127–2132, 2008.

[FJBQ10] Zhang Jian Feng, Cheng Jian, Liu Bo, and Yue Qiang. Performance comparison of phase modulated system based on dsss with hpa nonlinearity. In *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, pages 13–17, 2010.

[FLA94] Z. Fejzo and H. Lev-Ari. Adaptive modular laguerre non-linear model. In *, 1994 Sixth IEEE Digital Signal Processing Workshop, 1994*, pages 249–252. IEEE, October 1994.

[FNCS04] Md.G.C. Flores, M. Negreiros, L. Carro, and A.A. Susin. Inl and dnl estimation based on noise for adc test. *Instrumentation and Measurement, IEEE Transactions on*, 53(5):1391–1395, 2004.

[GLL97] A Glavieux, C Laot, and J. Labat. Turbo equalization over a frequency selective channel. *Proc. Int. Symp. on Turbo Codes, Brest, France*, pages 96–102, 1997.

[GLL04] L. Giugno, M. Luise, and V. Lottici. Adaptive pre- and post-compensation of nonlinear distortions for high-level data modulations. *IEEE Transactions on Wireless Communications*, 3(5):1490–1495, September 2004.

[GMC06] P.L. Gilabert, G. Montoro, and A. Cesari. A recursive digital predistorter for linearizing RF power amplifiers with memory effects. In *Microwave Conference, 2006. APMC 2006. Asia-Pacific*, pages 1040 –1043, December 2006.

[GMH+09] J. Goodman, B. Miller, M. Herman, M. Vai, and P. Monticciolo. Extending the dynamic range of RF receivers using nonlinear equalization. In *Waveform Diversity and Design Conference, 2009 International*, page 224–228, 2009.

[GWCW07] F.H. Gregorio, S. Werner, J.E. Cousseau, and R. Wichman. Broadband powerr amplifier distortion cancellation with model estimation inthe receiver. In *Signal Processing Advances in Wireless Communications, 2007. SPAWC 2007. IEEE 8th Workshop on*, page 1–5, 2007.

[Hay96] Simon S. Haykin. *Adaptive filter theory*. Prentice-Hall information and system sciences series. J. Wiley & Sons, third edition, 1996.

[HB08] F. Horlin and A. Bourdoux. *Digital compensation for analog front-ends: a new approach to wireless transceiver design*. J. Wiley & Sons, 2008.

[JP10] Jiang Jin and Juan Peng. Blind identification and digital compensation of memory nonlinearities for multiband systems. pages 7–11. IEEE, April 2010.

[JSS02] D. Jianping, N. Sundararajan, and P. Saratchandran. Communication channel equalization using complex-valued minimal radial basis function neural networks. 13(3):687–696, May 2002.

[KHCO10] P. Kim, J. Han, D.I. Chang, and D.G. Oh. Efficient channel equalization technique for DVB-S2 standard. In *Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc), 2010 5th*, page 208–212, 2010.

[KS89] G. Karam and H. Sari. Analysis of predistortion, equalization, and ISI cancellation techniques in digital radio systems with nonlinear transmit amplifiers. *Communications, IEEE Transactions on*, 37(12):1245–1253, 1989.

[LA08] K.P. Liolis and N.S. Alagha. On 64-apsk constellation design optimization. In *Signal Processing for Space Communications, 2008, SPSC 2008, 10th International Workshop on*, pages 1–7, 2008.

[LF12] D. N. Liu and M. P. Fitz. Iterative equalization in non-linear satellite channels. In *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*, page 220–224, 2012.

[MMK+06] 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):3852–3860, October 2006.

[MZ09] X. Mu and Y. Zhang. A modified gradient-based backpropagation training method for neural networks. In *IEEE International Conference on Granular Computing, 2009, GRC '09*, pages 450–453, 2009.

[Ogu07] Tokunbo Ogunfunmi. *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*. Springer, 2007.

[OVS94] Vojin G. Oklobdzija, David Villeger, and Thierry Soulas. An integrated multiplier for complex numbers. *Journal of VLSI signal processing systems for signal, image and video technology*, 7(3):213–222, October 1994.

[PCMC08] JC Patra, WC Chin, PK Meher, and G. Chakraborty. Legendre-FLANN-based nonlinear channel equalization in wireless communication system. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, page 1826–1831, 2008.

[PJ99] D. Park and T. Jeong. Complex bilinear recurrent neural network for equalization of a digital satellite channel. In *International Joint Conference on Neural Networks, 1999. IJCNN '99*, volume 3, pages 1485–1490 vol.3. IEEE, 1999.

## BIBLIOGRAPHY


[PPBP99] J.C. Patra, R.N. Pal, R. Baliarsingh, and G. Panda. Nonlinear channel equalization for QAM signal constellation using artificial neural networks. 29(2):262–271, 1999.

[PS08] Proakis and Salehi. *Digital Communications*. McGraw Hill, fifth edition, 2008.

[Rap91] C. Rapp. Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting signal. In *In ESA, Second European Conference on Satellite Communications*, volume 332, pages 179–184, October 1991.

[RKJ+12] R. Rezvani, M. Katiraee, A. H. Jamalian, S.H. Mehrabi, and A. Vezvaei. A new method for hardware design of multi-layer perceptron neural networks with online training. In *Cognitive Informatics Cognitive Computing (ICCI*CC), 2012 IEEE 11th International Conference on*, pages 527–534, 2012.

[RQZ04] R. Raich, H. Qian, and G.T. Zhou. Orthogonal polynomials for power amplifier modeling and predistorter design. *IEEE Transactions on Vehicular Technology*, 53(5):1468–1479, September 2004.

[RVR+10] S. Ramasamy, B. Venkataramani, C.K. Rajkumar, B. Prashanth, and K. Krishna Bharath. The design of an area efficient segmented dac. In *Signal and Image Processing (ICSIP), 2010 International Conference on*, pages 382–387, 2010.

[Sal81] A. Saleh. Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers. *IEEE Transactions on Communications*, 29(11):1715–1720, November 1981.

[SFU05] SFU. *Broadcast Test System R and S SFU Specifications*. Rhode&Schawarz, 2005.

[SG08] M. Shabany and P.G. Gulak. Efficient compensation of the nonlinearity of solid-state power amplifiers using adaptive sequential monte carlo methods. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(10):3270–3283, November 2008.

[SL08] Tim Schenk and Jean-Paul Linnartz. *RF imperfections in high-rate wireless systems: impact and digital compensation*. Springer, February 2008.

[Smi98] G. Smithson. Introduction to digital modulation schemes. In *The Design of Digital Cellular Handsets (Ref. No. 1998/240), IEE Colloquium on*, pages 2/1–2/9, 1998.

[SR12] K. Shi and A. Redfern. Blind volterra system linearization with applications to post compensation of adc nonlinearities. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3581–3584, 2012.

[Str94] R. Strauss. Reliability of sspa's and twta's. *Electron Devices, IEEE Transactions on*, 41(4):625–626, 1994.

[Tay13] Taylor. Taylor's theorem, June 2013. Wikipedia, the free encyclopedia.

[Ter10] TerAsic. *TerAsic DE4 User Manual*. TerAsic Company, 2010.

[TI06] TI. *Intermediate Frequency (IF) Sampling Receiver Concepts*. Texas Instruments, 2006.

[Tsi95] John Tsimbinos. *Identification and Compensation of Nonlinear Distortion*. PhD thesis, University of South Australia, 1995.

[TTD00] J.K. Tugnait, Lang Tong, and Zhi Ding. Single-user channel estimation and equalization. *IEEE Signal Processing Magazine*, 17(3):16–28, 2000.

[YWL04] C. Yen, W. Weng, and Y. Lin. FPGA realization of a neural-network-based nonlinear channel equalizer. 51(2):472– 479, April 2004.

[YZWY10] Yonggang Yan, Junwei Zhao, Zhankui Wang, and Yongpeng Yan. An novel variable step size lms adaptive filtering algorithm based on hyperbolic tangent function. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 14, pages V14–233–V14–236, 2010.

[ZZ09] H. Zhao and J. Zhang. Adaptively combined FIR and functional link artificial neural network equalizer for nonlinear communication channel. 20(4):665–674, April 2009.

# RÉSUMÉ DE THÈSE

Présentée à

## TELECOM Bretagne

pour obtenir le grade de

## DOCTEUR de TELECOM BRETAGNE

Mention: **Sciences et Technologies de l'Information et de la Communication**

par

***Esteban Cabanillas***

---

## *Efficient non-linear ISI compensation techniques for commercial receivers*

---

Soutenue le 14 février 2014 devant la Commission d'examen :

Composition du Jury

- *Rapporteurs :* Olivier Romain, Professeur, Université de Cergy Pontoise
  Jean-François Hélard, Professeur, INSA Rennes
- *Examinateurs :* Didier Lohy, Development Director, NXP Semiconsductors France
  Cyril Lahuec, Maitre de conférences, TELECOM Bretagne
  Michel Jézéquel, Professeur, TELECOM Bretagne

# Résumé

L'objectif de cette Thèse est d'étudier différentes méthodes de compensation des dégradations induites par un amplificateur de puissance (AP) non-linéaire sur des systèmes de communications numériques. Il existe deux effets adverses induits par un AP non-linéaire: l'étalement spectral et l'interférence inter-symbole (IIS) non-linéaire. Ces dégradations peuvent être corrigées au niveau du transmetteur avec une méthode de pré-distorsion. Malgré son efficacité, la pré-distorsion n'est pas implantée dans tous les APs. Il est par conséquent important de compenser l'IIS non-linéaire au niveau du récepteur. Cette technique est nommée post-distorsion. A ce jour, la méthode de post-distorsion la plus efficace est le Volterra Canceller. Une telle technique estime et élimine l'IIS non-linéaire associée aux symboles reçus. Une compensation optimale peut être faite à partir d'un Volterra Canceller travaillant avec des symboles idéaux, ce qui n'est en pratique jamais possible.

Cette Thèse propose une méthode innovante de post-distorsion plus efficace que le Volterra Canceller. Elle est basée sur la linéarisation du canal non-linéaire. La principale difficulté liée à l'implantation de cette technique est l'estimation aveugle de la fonction de transfert inverse du canal. De plus, cette estimation est faite à partir de données bruitées, ce qui complique la tâche. La linéarisation est basée sur une régression polynômiale d'Hermite. La compensation optimale caractérisant le Volterra Canceller idéal est atteinte par la combinaison de cette linéarisation polynomiale et d'un Volterra Canceller. Ce comportement a été vérifié par simulations faites à partir du modèle mathématique d'un système de TV numérique satellite Digital Video Broadcasting Satellite 2 (DVB-S2), cas d'application de cette Thèse. Les performances du récepteur qui compensent l'IIS non-linéaire sont établies en termes du gain de $E_S/N_0$ par rapport à un récepteur sans compensation.

L'architecture est optimisée pour un récepteur commercial DVB-S2. La méthode proposée a été implantée sur FPGA dans un récepteur DVB-S2 et testée. Le récepteur a ensuite été inclus dans une chaîne de transmission émulant un système DVB-S2. Les résultats obtenus sont corrélés avec les résultats des simulations. Par rapport à un Volterra Canceller seul, la méthode proposée améliore le gain de 60% pour une complexité ajoutée de 40%. Ceci confirme l'efficacité de cette compensation en termes de performance et de complexité.

# Sommaire

# 1 Non-linéarité dans un système de communication numérique

Un système de communication numérique est composée d'un transmetteur, d'un canal et d'un récepteur, figure 1.1. Les bits $c_j$ sont obtenus après l'encodage canal des bits d'information $a_i$. Les symboles $s_n$ sont générés par le bloc Mapping. Après sur-échantillonnage, le filtre numérique adapté cosinus surélevé (bloc CSF$(\cdot)$) limite la bande de transmission du signal $x_t$. Cette bande est définie par le coefficient de roll-off associé au filtre. Un amplificateur de puissance (bloc AP) donne au signal transmit $x'_t$ la puissance nécessaire pour atteindre le récepteur. Après être bruité par un canal Gaussien, le signal reçu $y_t$ est à nouveau filtré par un bloc CSF$(\cdot)$. Obtenus à partir d'un sous-échantillonnage, les symboles reçus $s'_n$ sont démodulés (bloc Demapping) et décodés pour obtenir une estimation des bits d'information $\hat{a}_i$.

Les standards le plus modernes caractérisant les systèmes de communication numériques cherchent d'une part la réduction de la puissance consommée et d'autre part l'augmentation du débit de données. Cette dernière caractéristique est obtenue à partir des modulations numériques de grand ordre (8-PSK, 16/32-APSK, 16/32/64-QAM, etc). L'utilisation des modulations de grand ordre implique un grand rapport signal à bruit ($E_S/N_0$) au niveau du récepteur. Pour avoir le niveau de $E_S/N_0$ nécessaire avec une puissance consommée minimale, l'efficacité de l'AP au niveau du transmetteur doit être la plus grande possible. Il est bien connu que pour avoir une efficacité élevée, l'AP doit

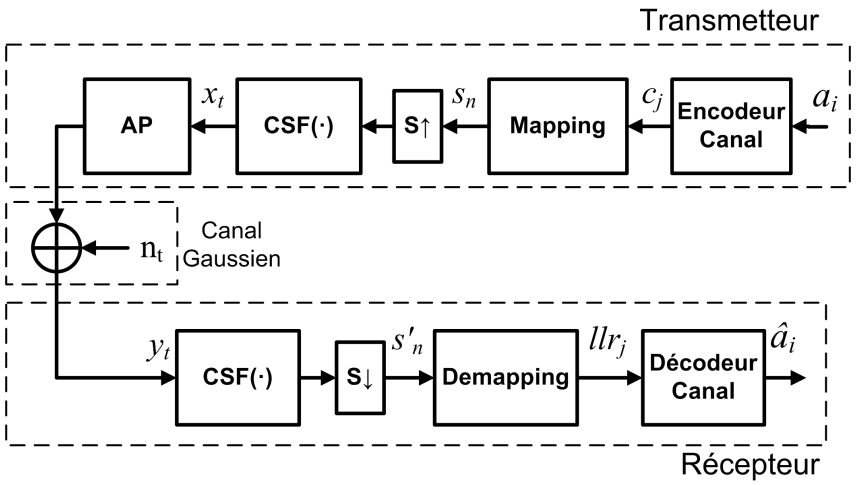


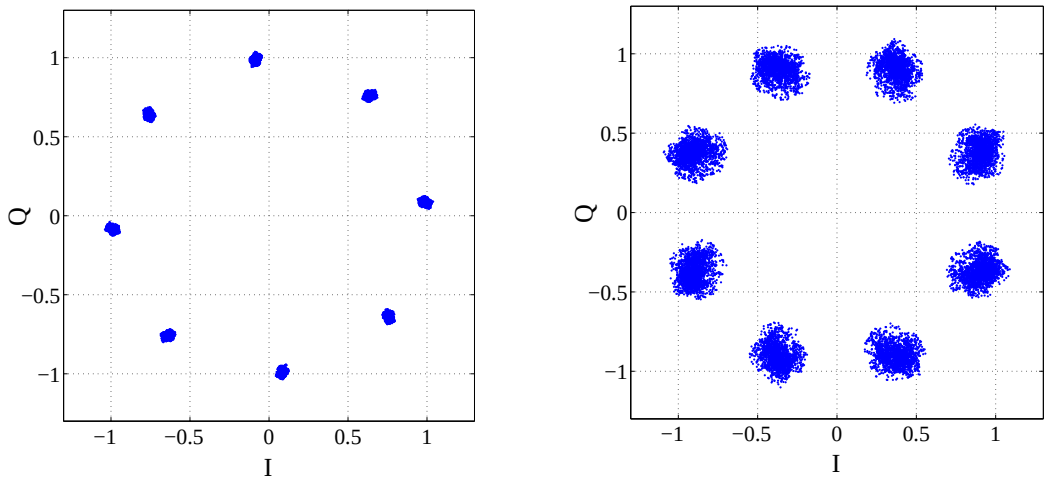

Figure 1.1: Modèle d'un système de communication numérique

travailler proche de sa saturation, ajoutant de la non-linéaire au système. L'input back-off (IBO) est une métrique permettant de déterminer le niveau de saturation d'un AP. Il est défini comme la différence entre la puissance de saturation du signal d'entrée de l'AP et sa puissance moyenne. Donc, un faible IBO signifie que la puissance moyenne est proche de celle de saturation, donc l'AP est fortement non-linéaire.

L'interférence d'ordre deux, trois et supérieur qui caractérisent un système non-linéaire est responsable de deux effets qui dégradent les systèmes de communication numériques. Le premier effet est l'étalement spectral, qui est manifesté comme un élargissement de la bande de transmission qui peut interférer des canaux adjacents. La seconde adversité vient du fait que le canal Gaussien de la figure 1.1 devient un canal non-linéaire sélectif en fréquence, i.e. le critère fréquentiel de Nyquist n'est plus respecté. Un tel canal induit une interférence connue comme Interférence Inter-Symbole (IIS) non-linéaire. Cette interférence est modélisée par la régression des séries de Volterra:

$$s'_n = \sum_{i=-M}^{M} q_i s_{n+i} + \sum_{i=-M}^{M}\sum_{j=-M}^{M} q_{i,j} s_{n+i} s_{n+j} + \sum_{i=-M}^{M}\sum_{j=-M}^{M}\sum_{k=-M}^{M} q_{i,j,k} s_{n+i} s_{n+j} s_{n+k} + O^3. \tag{1.1}$$

Où $M$ est la mémoire (délai maximal) de l'interférence, $O^3$ fait référence aux termes d'ordre supérieur à trois et $c_i$ est le $i^{ème}$ coefficient de la régression, qui est une fonction de l'IBO et du roll-off. La dégradation induite par l'IIS non-linéaire est exemplifié à partir des résultats de simulation considérant un système de TV satellite Digital Video Broadcasting-Satellite 2 (DVB-S2), qui est le cas d'application de ce travail. Ces résultats sont présentés dans la figure 1.2, montrant la constellation reçue du type 8-PSK pour IBO 10 dB et 0 dB et $E_S/N_0 = 100$ dB. Il est montré qu'une grande valeur de IBO est associée à une faible IIS non-linéaire. Dans le cas contraire, l'IBO = 0 dB induit une grande dispersion dans les symboles reçus.




(a) IBO = 10 dB (b) IBO =0 dB

Figure 1.2: Dispersion des symboles reçus en fonction de l'IBO

# 2 État de l'art de méthodes de post-distorsion

Implantée au niveau du transmetteur, la pré-distorsion est une méthode efficace qui corrige l'étalement spectrale et l'IIS non-linéaire. Cette efficacité est d'une part liée à l'absence du bruit Gaussien dans les signaux du transmetteur, et d'autre part au fait que l'information transmise est connue par le transmetteur. Malgré ses points forts, la pré-distorsion n'est pas utilisée dans tous les transmetteurs. Donc, il est recommandé de compenser l'IIS non-linéaire au niveau du récepteur. Une telle méthode, appelée post-distorsion, est implantée dans un système de communication numérique comme représente sur la figure 2.1, bloc Post. Dist. Ce bloc va donc supprimer l'IIS non-linéaire des symboles reçus $s'_n$ pour générer les symboles $s''_n$.

Il existe plusieurs techniques de post-distorsion qui définissent l'état de l'art. La technique la plus ancien et la plus simple est l'égalisateur de Volterra, qui à partir d'une régression des séries de Volterra applique l'inverse de la fonction de transfère du canal non-linéaire. Le calcul de l'inverse du canal ne peux être fait avec précision pour les cas où le plancher de bruit est supérieur à l'atténuation du canal. Donc, cette technique n'est pas efficace pour des systèmes travaillant dans des environnements fortement bruités. Une autre technique de simple implantation est le Volterra Canceller, figure 2.2. Le Volterra Canceller estime et supprime l'IIS non-linéaire $I_n$ associée à un symbole reçu $s'_n$. L'IIS non-linéaire est déterminée à partir d'une estimation a-priori qui assigne un point de constellation $\hat{s}_n$ à chaque symbole $s'_n$. Le bloc NL(·) estime $I_n$ avec une régression de Volterra. L'estimation de $I_n$ faite à partir d'une régression d'ordre trois est la suivante:

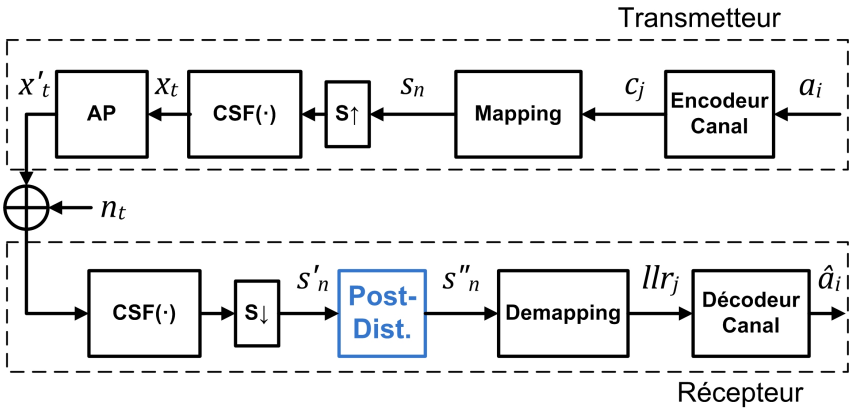



Figure 2.1: Méthode de post-distorsion dans un système de communications numérique

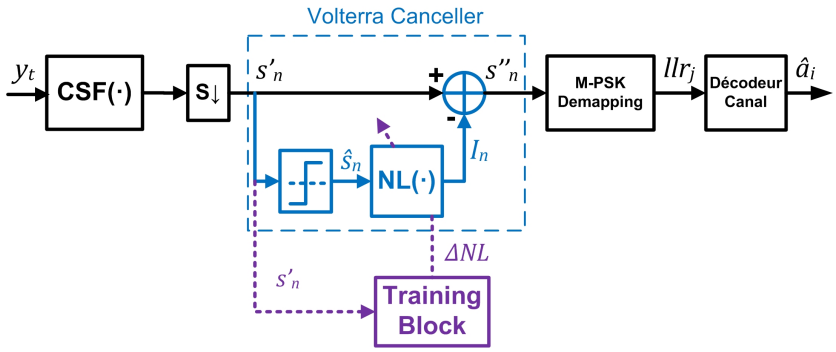



Figure 2.2: Volterra Canceller dans un récepteur

$$I_n = \sum_{i=-M}^{M} vt_i \hat{s}_{n+i} + \sum_{i=-M}^{M} \sum_{j=-M}^{M} \sum_{k=-M}^{M} vt_{i,j,k} \hat{s}_{n+i} \hat{s}_{n+j} \hat{s}_{n+k}. \tag{2.1}$$

Où $vt_i$ est le $i^{ème}$ coefficient de régression. La technique de post-distorsion la plus performante est le turbo canceller. Cette technique travaille de manière itérative avec un turbo décodeur pour estimer l'IIS non-linéaire.

Il est intéressant de comparer l'efficacité de chacune des techniques de post-distorsion. L'efficacité est définie comme le compromis entre le gain de $E_S/N_0$ et la complexité associée à une technique. Ce gain est la différence de $E_S/N_0$ en dB nécessaire pour que deux systèmes, un avec post-distorsion et l'autre sans, aient un taux d'erreur similaire. Le gain de plusieurs techniques de l'état de l'art est présenté dans la table 2.1. Tous ces travaux considèrent une modulation numérique du type 16-QAM.

Pour faire une analyse d'efficacité, il est nécessaire d'étudier ensemble le gain et la complexité de chaque technique. Le turbo canceller a le gain le plus élevé, mais il a besoin d'une grande quantité de mémoire associée au délai dû aux itérations. Le gain du Volterra Canceller est proche de celui du turbo canceller, et la complexité est similaire à celle d'un l'égalisateur typique. Il est donc correct d'affirmer que le Volterra Canceller est la solution la plus efficace. Le reste de ce travail est consacré à trouver une nouvelle méthode plus efficace que le Volterra Canceller.

Table 2.1: Performance des méthodes de post-distorsion de l'état de l'art

| Technique | Références | Roll-off | Gain [dB] |
|---|---|---|---|
| Égalisateur de Volterra | [Giugno et al., 2004] | 0.3 | 0.60 |
| | [Beidas and Seshadri, 2010] | 0.25 | 1.35 |
| Volterra Canceller | [Beidas and Seshadri, 2010] | 0.25 | 1.7 |
| | [Burnet, 2006] | 0.2 | 0.68 |
| Turbo canceller | [Burnet and Cowley, 2005] | 0.2 | 0.79 |

# 3 Nouvelle méthode de post-distorsion à partir de linéarisation polynomiale

L'IIS non-linéaire est générée par l'interaction entre l'AP non-linéaire et les filtres numériques CSF. Il est donc possible de réduire l'intensité de l'IIS si la non-linéarité est compensée au niveau du récepteur avant d'appliquer le CSF. Cette compensation est faite en multipliant la fonction de transférer de l'AP par sa fonction inverse, figure 3.1.

Une façon de faire cette linéarisation est à partir d'une régression de polynômes orthogonaux. La plus efficace des linéarisations est faite avec la famille des polynômes d'Hermite présentes dans la table 3.1. Considérant $y_t$ le signal reçu bruité et affecté par la non-linéarité d'un AP, le signal linéarisé $y'_t$ est une fonction de $y_t$ défini comme:

$$y'_t = \sum_{i=0}^{N} b_i H_i(y_t). \tag{3.1}$$

Où $N$ est l'ordre de la régression, $H_i(\cdot)$ est le $i^{ème}$ polynôme d'Hermite et $b_i$ est le

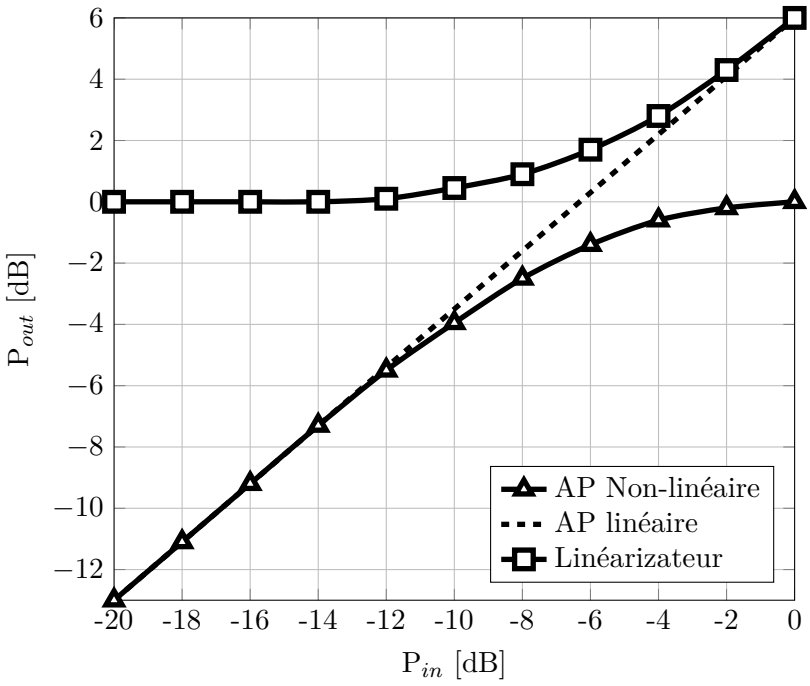



Figure 3.1: Linéarisation par l'application de la fonction inverse d'un AP

Table 3.1: Polynômes d'Hermite

$$
\begin{aligned}
H_0(y) &= 1.\\
H_1(y) &= y.\\
H_2(y) &= 2yy^* - 1.\\
H_3(y) &= 2y^2y^* - 3y.\\
H_4(y) &= 4y^2(y^*)^2 - 12yy^* + 3.\\
H_5(y) &= 4y^3(y^*)^2 - 20y^2y^* + 15y.\\
H_6(y) &= 8y^3(y^*)^3 - 60y^2(y^*)^2 + 90yy^* - 120.
\end{aligned}
$$



coefficient de régression. Les $b_i$s sont adaptés à la non-linéarité et au canal Gaussien à partir de l'algorithme du gradient stochastique implanté dans le Training Block. Cette technique innovante est le principal apport de cette Thèse.

À cause de l'existence du bruit Gaussien, l'inverse du canal n'est pas parfaitement calculé. Donc un IIS non-linéaire résiduel est associé aux symboles reçus. Ce résidu est éliminé en ajoutant un Volterra Canceller après la linéarisation, figure 3.2. Comme l'intensité et la mémoire de cette IIS non-linéaire est faible, l'estimation a-priori des symboles reçus est précise, ce qui permet au Volterra Canceller de travailler proche de sa performance idéale. Le Volterra Canceller est aussi adapté avec un algorithme du gradient stochastique.

Des simulations Matlab incluant un modèle du système DVB-S2 sont faites pour déterminer la performance de la technique proposée en figure 3.2. Les résultats confirment que la compensation de l'IIS non-linéaire faite à partir d'une linéarisation d'Hermite d'ordre trois combinée à un Volterra Canceller d'ordre trois est la méthode la plus efficace comparée aux autres méthodes proposées par l'état de l'art.

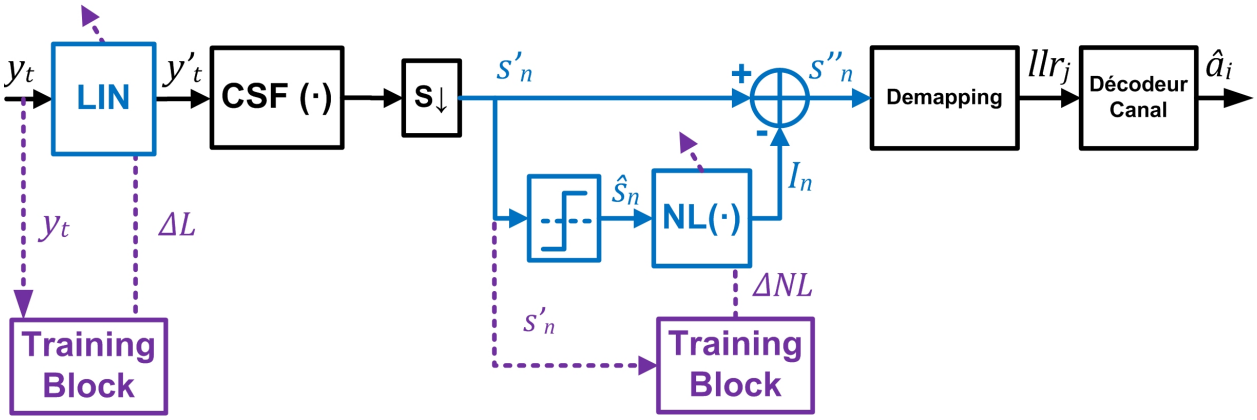



Figure 3.2: Intégration d'une Linéarisation polynômial d'Hermite combinée à un Volterra Canceller dans un récepteur

Une implantation numérique de la technique combinée est conçue afin de valider expérimentalement son efficacité. Cette implantation sera intégrée dans un récepteur commercial de TV satellite. La règle de design d'un tel récepteur est d'avoir la meilleure performance possible à partir d'une complexité réduite. En conséquence, l'implantation numérique de la technique combinée doit être focalisée sur la réduction de la complexité.

L'architecture du Volterra Canceller (incluant son Training Block) implantée est illustrée dans la figure 3.3. Cette architecture est constituée de trois blocs: Un bloc de mapping et de registre à décalage "Mapping and Shiff-Register", bloc NL(·) et un bloc d'adaptation "LMS Training Block". Le bloc "Mapping and Shiff-Register" applique une estimation a-priori aux symboles reçus $s'_n$ pour générer les points de constellation $\hat{s}_n$. L'estimation est faite avec une méthode qui compare l'erreur entre le symbole reçu et les points de constellation adjacents à partir des blocs d'addition. Finalement, ce bloc retarde les $\hat{s}_n$ avec un registre de décalage afin d'obtenir la mémoire nécessaire pour estimer l'IIS non-linéaire $I_n$. Le bloc NL(·) estime la valeur d'IIS non-linéaire $I_n$ et de l'estimation du symbole reçu sans IIS non-linéaire $\check{s}_n$ à partir des estimations $\hat{s}_n$ et le vecteur de coefficients de Volterra $\mathbf{vt_m}$. Le bloc "Volterra Combiner" calcule les combinaisons d'ordre un et trois caractérisant la régression de Volterra montrée en (2.1). Ces combinaisons sont les éléments du vecteur $\mathbf{vc_n}$. Pour éviter la complexité amené par l'implantation de multiplicateurs complexes, les combinaisons sont calculées à partir de tableaux du type "Look-Up Table". Le vecteur $\mathbf{vt_m}$ est estimé par le bloc d'adaptation "LMS Training Block". Ce bloc implémente une version simplifiée innovante de l'algorithme du gradient stochastique. La moitié de l'algorithme du gradient stochastique est basée sur le calcul de l'erreur de l'estimation. Comme cette erreur est déjà calculée dans la soustraction entre $I_n$ et $s'_n$, elle peut être réutilisée pour le bloc d'adaptation. La complexité de l'autre

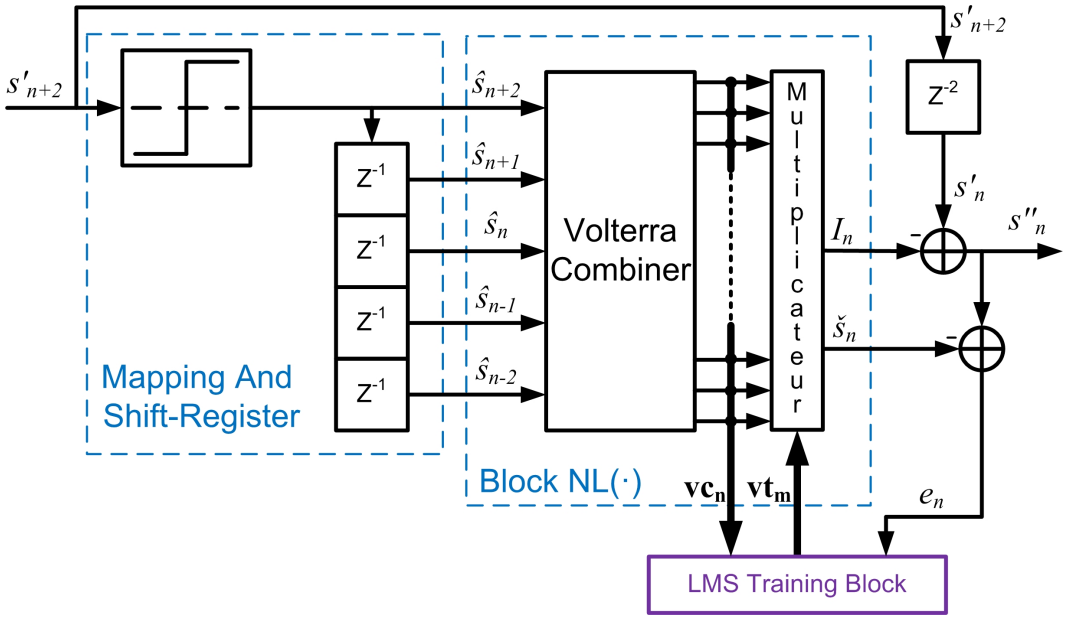



Figure 3.3: Blocs constituant l'architecture du Volterra Canceller implanté

moitié de l'algorithme est réduite grâce au remplacement de multiplicateurs par des blocs de décalage binaire.

L'architecture du linéarisateur d'Hermite est présentée dans la figure 3.4. Cette architecture est aussi constituée de trois blocs: un bloc qui calcule les valeurs des polynômes ("Polynomial Expansion Block"), un Multiplicateur et un bloc d'adaptation "LMS Training Block". Le "Polynomial Expansion Block" calcule la valeur des polynômes d'ordre un et trois ($hc_{1,t}$ et $hc_{3,t}$). Afin d'obtenir ces valeurs avec un minimum de complexité de calcul, ce bloc remplace les multiplicateurs complexes grâce à plusieurs simplifications et blocs de décalage binaire. La sortie linéarisée $y'_t$ est le résultat du produit entre le vecteur ($hc_{1,t}$ ,$hc_{3,t}$) et les coefficients de régression composant le vecteur $\mathbf{vt}_\iota$. Ce dernier vecteur est déterminé par le bloc d'adaptation.

Plusieurs simulations SystemC/VHDL ont été faites afin de vérifier que toutes les simplifications implantées dans le Volterra Canceller et le linéarisateur d'Hermite ne dégradaient pas leur capacité de compenser l'IIS non-linéaire.

.

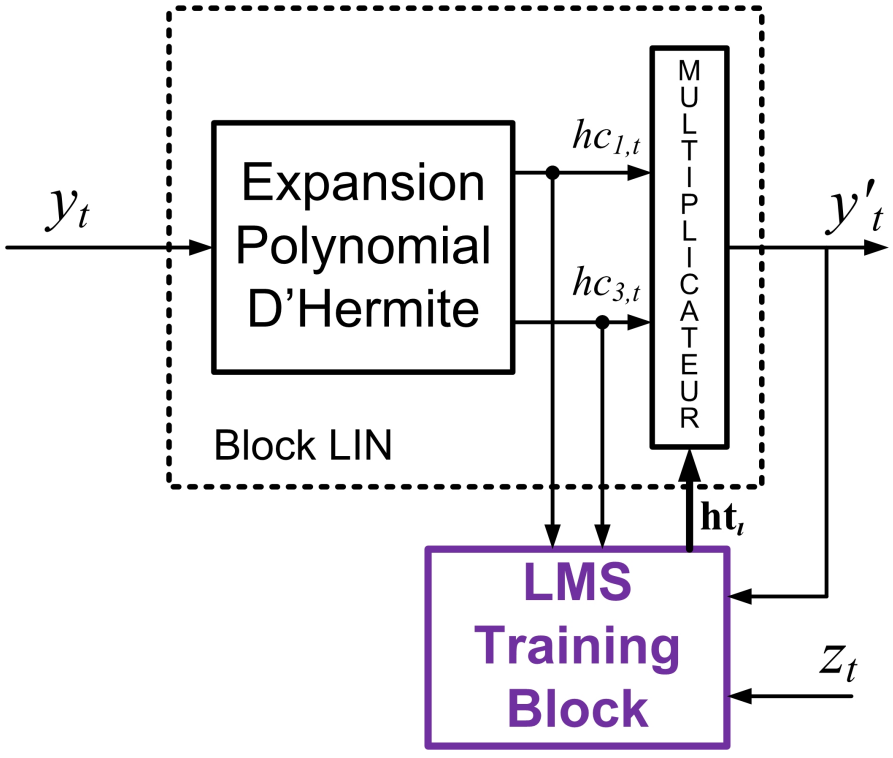



Figure 3.4: Blocs constituant l'architecture du linéarisateur d'Hermite

# 4

# Résultats de mesure

Cette section est consacrée à l'analyse des résultats de mesure d'un récepteur compensé avec la nouvelle technique présentée dans la section 3. Ces résultats ont été obtenus à partir d'une chaine de transmission DVB-S2, figure 4.1. Le transmetteur DVB-S2 et le canal Gaussien sont émulés par deux appareils spécifiques: un générateur de signal TV numérique et un générateur de bruit aléatoire Gaussien. Un modèle de l'AP non-linéaire proposé pour le standard DVB-S2 est implanté dans une FPGA. Cette FPGA partage ses ressources avec le récepteur DVB-S2, qui est un prototype qui a été conçu par NXP semiconductors.

Le récepteur est configuré de trois façons différentes. Dans la première configuration, aucun bloc supplémentaire n'est ajouté. Donc cette configuration ne corrige pas l'IIS non-linéaire. La deuxième configuration intègre un Volterra Canceller tout seul dans le récepteur. La dernière configuration combine le linéarisateur et le Volterra Canceller dans le récepteur DVB-S2.

La banque de test utilisée pour obtenir les résultats de mesure est montré dans la figure 4.2. Le SFU est un appareil construit par Rodhe & Swcharz qui génére le signal de TV satellite. L'AP implanté dans la FPGA ajoute de la non-linéarité à ce signal. Une carte de conversion analogue-numérique et numérique-analogue est connectée à la FPGA. Un coupleur additionne le bruit Gaussien généré par l'appareil AWGN Hewlett Packard avec le signal TV non-linéeaire. Le rapport signal à bruit $E_S/N_0$ est déterminé avec un analyseur de spectre. Le signal bruité est amplifié par un AMC-180 avant d'être traité par le récepteur DVB-S2 embarqué dans la FPGA.

Ce travail propose une nouvelle métrique pour déterminer la performance d'un récepteur DVB-S2. Cette métrique est le nombre moyen d'itérations LDPC (définies par le standard DVB-S2) nécessairs pour décoder sans erreur le message transmit, pour un $E_S/N_0$ donné. Les résultats de mesure sont à la fois comparés aux résultats de simulation afin d'analyser le comportement de l'implantation numérique. L'AP non-linéaire travaille avec un IBO=0dB, le pire des cas.


Figure 4.1: Chaine de transmission DVB-S2


Figure 4.2: Banque de test

Le cas d'un récepteur sans compensation d'IIS non-linéaire est considéré dans la figure 4.3. Le code rate LDPC est 9/10. Les courbes rouges décrivent le nombre moyen d'itérations pour le cas simulé, et en noire les performances mesurées. Pour les deux cas, simulé et mesure, les courbes pointillés montrent le cas d'un transmetteur dont l'AP est linéaire, c'est-à-dire le cas idéal. Le pire des cas est montré par les courbes continues, qui représentent le cas d'un AP non-linéaire. Un offset de 0.09 dB est remarqué entre les courbes simulées et les mesures. La dégradation induite par l'IIS non-linéaire est de 0.76 dB pour le cas simulé et de 0.85 dB pour le cas mesuré. Donc, l'implantation est plus sensible à l'IIS non-linéaire que la simulation.

Les performances mesurées et simulées d'un récepteur qui compense avec un Volterra Canceller (cas "VC") sont illustrées dans la figure 4.4. Le gain de $E_S/N_0$ donné par l'utilisation d'un Volterra Canceller est de 0.29 dB pour le cas simulé. Pour le cas mesuré le gain est légèrement plus bas (0.24 dB).

La figure 4.5 illustre les performances d'un récepteur qui intègre la méthode combinée ("HPL+VC") présentée dans la section 3. Il est montré que le gain est supérieur au cas VC considéré dans la figure 4.4. Le gain par rapport au pire cas est de 0.45 dB pour les simulations et 0.37 dB pour les mesures.

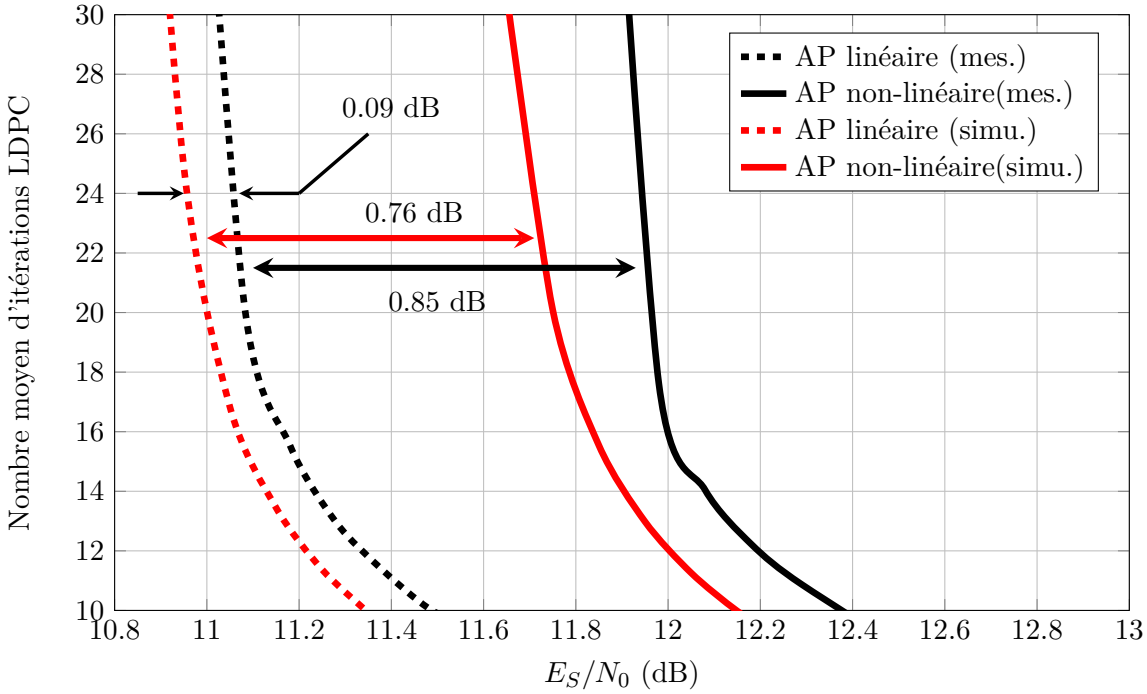



Figure 4.3: Comparaison des performances entre résultats de mesure et simulation. Définitions des limites des performances. Code Rate=9/10.

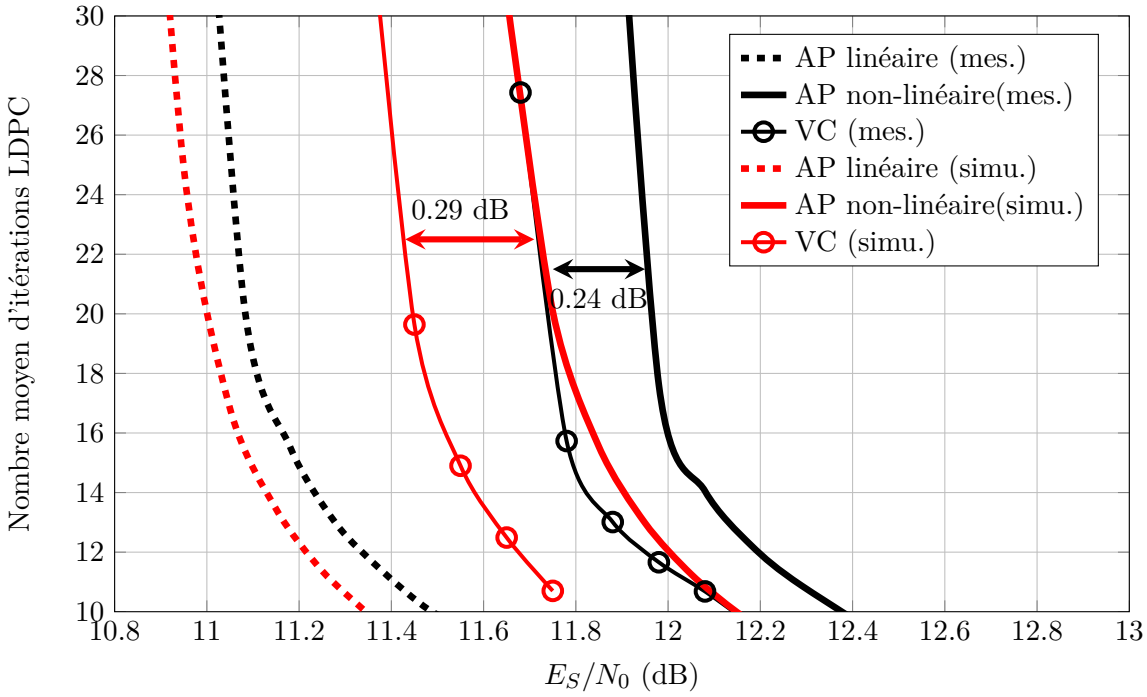



Figure 4.4: Comparaison des performances entre résultats de mesure et simulation avec un Volterra Canceller (VC). Code Rate=9/10.

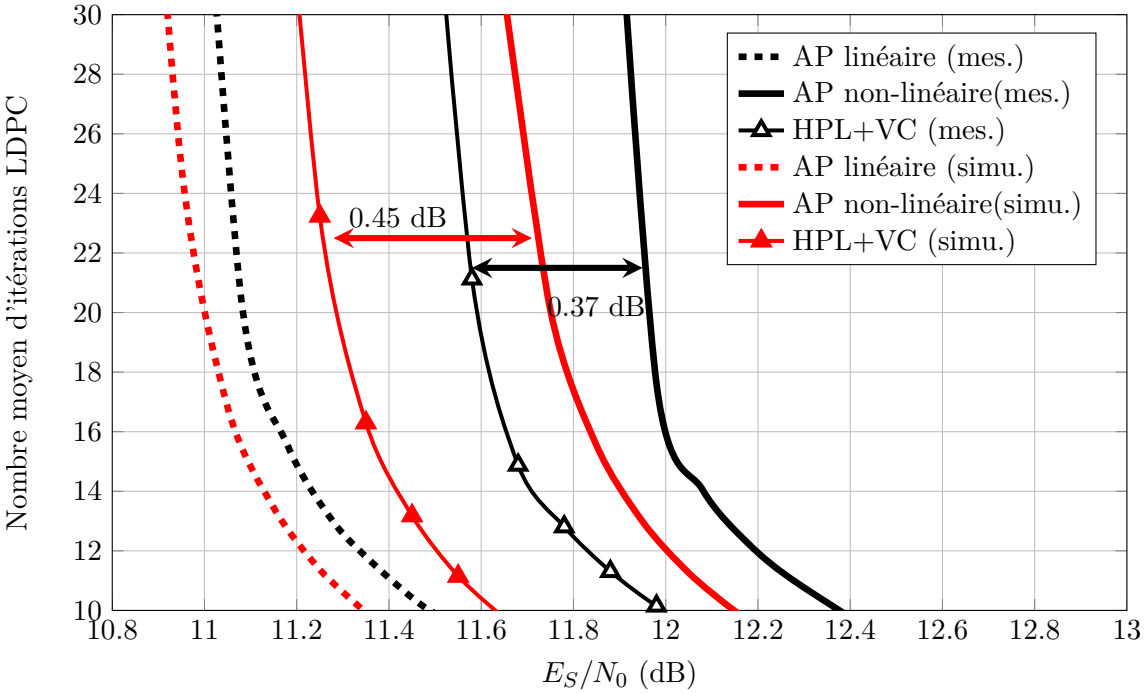



Figure 4.5: Comparaison des performances entre résultats de mesure et simulation avec une combinaison de linéarisation d'Hermite et Volterra Canceller (HPL+VC). Code Rate=9/10.

Table 4.1: Gain de $E_S/N_0$ [dB] par rapport au cas non-compensé et non-linéaire, 22 itérations LDPC de moyen

| *Case* / *CodeRate* | AP linéaire (mes.) | AP linéaire (simu.) | VC (mes.) | VC (simu.) | HPL+VC (mes.) | HPL+VC (simu.) |
|---|---|---|---|---|---|---|
| 2/3 | 0.48 | 0.43 | -0.09 | 0 | -0.13 | 0.01 |
| 3/4 | 0.57 | 0.51 | -0.02 | 0.06 | 0.02 | 0.11 |
| 5/6 | 0.67 | 0.6 | 0.08 | 0.16 | 0.19 | 0.27 |
| 8/9 | 0.81 | 0.74 | 0.22 | 0.28 | 0.34 | 0.43 |
| 9/10 | 0.85 | 0.76 | 0.24 | 0.29 | 0.37 | 0.45 |

Le gain par rapport au pire cas pour un nombre moyen de 22 itérations LDPC et plusieurs code rates sont montrés dans le tableau 4.1.

Il est intéressant de déterminer l'efficacité de cette implantation. La synthèse FPGA a montré que le linéarisateur Hermite ajoute seulement 42% de complexité au Volterra Canceller. Puisque le gain additionnel peut être de 60% (cas du code rate = 9/10), il est valide d'affirmer que cette nouvelle technique est efficace.

# 5

# Conclusions

Cette Thèse a été consacrée à la correction d'une des imperfections induites par un AP travaillant proche de la saturation: l'IIS non-linéaire. Il a été montré que cette interférence dégrade la qualité des symboles reçus, et donc, la performance d'un récepteur.
Par conséquence, plusieurs méthodes de l'état de l'art en compensation de l'IIS non-linéaire au niveau du récepteur ont été présentées. La méthode la plus efficace est le Volterra Canceller. Cette Thèse propose une nouvelle méthode qui permet d'améliorer le Volterra Canceller en ajoutant un bloc linéarisateur. La linéarisation est faite à partir d'une régression de polynômes d'Hermite. Cette méthode combinée a été implantée sous la forme d'un circuit numérique. Une telle implantation est optimisée en termes de fonctionnement et de complexité pour travailler dans un récepteur commercial DVB-S2.

Les résultats de mesure de cette implantation sont corrélés à ceux de simulation. Le gain du Volterra Canceller est amélioré de 60% en ajoutant un bloc linéariseur d'Hermite. Ce gain est assez élevé pour justifier les 42% de complexité additionnelle.

5

# Références

[Beidas and Seshadri, 2010] Beidas, B. and Seshadri, R. (2010). Analysis and compensation for nonlinear interference of two high-order modulation carriers over satellite link. *IEEE Transactions on Communications*, 58(6):1824–1833.

[Burnet and Cowley, 2005] Burnet, C. and Cowley, W. (2005). Performance analysis of turbo equalization for nonlinear channels. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 2026 –2030.

[Burnet, 2006] Burnet, C. E. (2006). *Mitigation of Adverse Effects in the Nonlinear Satellite Channel for 16QAM Transmission*. PhD thesis, University of South Australia.

[Giugno et al., 2004] Giugno, L., Luise, M., and Lottici, V. (2004). Adaptive pre- and post-compensation of nonlinear distortions for high-level data modulations. *IEEE Transactions on Wireless Communications*, 3(5):1490–1495.