AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY
Faculty of Information and Natural Sciences

Henri Ruotsalainen

INVESTIGATION OF ORTHOGONAL BASIS EXPANSIONS FOR ADAPTIVE WIENER MODELS

Thesis submitted for examination for the degree of Master of Science in Technology

Espoo 19.04.2010

Thesis supervisor:

Prof. Risto Wichman
(Aalto University School of Science and Technology)

Univ. Prof. Dr.-Ing. Markus Rupp
(Vienna University of Technology)

Thesis instructor:

Dipl.-Ing. Robert Dallinger
(Vienna University of Technology)

Author: Henri Ruotsalainen

Title: Investigation of Orthogonal Basis Expansions for Adaptive Wiener Models

Date: April 19, 2010 Language: English Number of pages: 8+86

Faculty: Faculty of Information and Natural Sciences

Professorship: Signal Processing in Telecommunications Code: S-88

Supervisor: Prof. Risto Wichman

Instructor: Dipl.-Ing. Robert Dallinger

In modern wireless networks, radio frequency (RF) power amplifiers (PA) are essential components which are inherently nonlinear. However, the use of spectrally efficient modulation techniques and densely populated transmission bands require stringently linear behaviour of the PAs. Currently, commercial RF PAs which are operating in their linear region, show poor power efficiency. In order to increase the efficiency, linearization methods are necessary. Predistortion in the digital baseband is considered to be an efficient method to compensate for the nonlinear effects of PAs and can also be combined with other linearization methods.

In this thesis, static nonlinear models based on orthogonal polynomials are presented. It is investigated whether one orthogonal polynomial basis outperforms other bases in the context of PA identification, with respect to digital predistortion. With increasing transmission bandwidth, the memory effects become more prevalent which requires the use of dynamic nonlinear models. This can be achieved in a simple way by cascading a static nonlinearity with a leading linear filter. If such a structure, known as the simplified Wiener model, is used to model the PA, the predistorter (PD) is obtained as a Hammerstein model.

Additionally, the characteristics of the PAs changes over time due to changes in temperature and aging. Therefore, the identification of the PA and the PD needs to be performed adaptively. In this thesis, three architectures for adaptive PDs are considered: the direct learning architecture, the indirect learning architecture and the direct learning architecture based on the nonlinear filtered-x least mean squares algorithm.

The work first analyses the numerical properties of different orthogonal polynomials. Then, the adaptative identification of the models, with and without memory, is investigated based on their convergence behaviour. Finally, the performance of the PD architectures is evaluated by simulations and burst measurements. For a commercial PA, the results of the study demonstrated that it was possible to increase the power efficiency by 55%

Keywords: orthogonal polynomial, digital baseband predistortion, power amplifier, Wiener model, adaptive system identification

AALTO-YLIOPISTON TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Henri Ruotsalainen

Työn nimi: Tutkielma ortogonaalisten sarjakehitelmien hyödyntämisestä adaptiivisissa Wiener malleissa

Päivämäärä: 19.04.2010 Kieli: Englanti Sivumäärä: 8+86

Tiedekunta: Informaatio- ja luonnontieteiden tiedekunta

Professuuri: Tietoliikenteen signaalinkäsittely Koodi: S-88

Valvoja: Prof. Risto Wichman

Ohjaaja: Dipl.-Ing. Robert Dallinger

Radiotaajuuksinen tehovahvistin, jonka vaste on usein epälineaarinen, on olennainen osa langattomia tietoliikennejärjestelmiä. Nykyaikaiset modulaatiotekniikat vaativat kuitenkin siirtotieltä korkeaa lineaarisuutta. Kun tehovahvistinta käytetään sen lineaarisella alueella, on vahvistimen hyötysuhde usein huono. Hyötysuhteen parantamiseksi voidaan hyödyntää kantataajuuksisia esisäröttimiä, joita on usein käytetty tehovahvistimien linearisoinnissa.

Työssä esitetään tehovahvistin- ja esisärötinmalleja, jotka perustuvat ortogonaalisiin sarjakehitelmiin. Yksi tutkimuksen pääaiheista oli erilaisten ortogonaalisten polynomien vertailu tehovahvistin- ja esisärötinmallinnuksessa kantataajuuksisille syötesignaaleille. Vertailun tavoitteena oli tutkia mikä ortogonaalinen sarjakehitelmä soveltuu parhaiten tehovahvistin- ja esisärötinmallinnukseen numeeristen ominaisuuksiensa puolesta. Laajakaistaisilla syötesignaaleilla tehovahvistimen toiminnassa ilmenee muisti-ilmiöitä. Näitä haitallisia ilmiöitä voidaan mallintaa pelkistetyllä Wiener mallilla. Tässä tapauksessa muisti-ilmiöiden kompensointiin voidaan puolestaan käyttää Hammerstein esisärötinmallia.

Tehovahvistimen ominaisuudet muuttuvat pääasiassa johtuen käyttölämpötilojen vaihteluista ja ikääntymisestä. Kun tehovahvistin- ja esisärötinmallinnus toteutetaan adaptiivisesti, voidaan aikariippuvat ominaisuudet ottaa huomioon. Työssä selostetaan useiden adaptiivisten tunnistusmenetelmien rakennetta ja toimintaa kun esisärötin on mallinnettu käyttäen Hammerstein mallia.

Työssä tutkittiin erilaisten ortogonaalisten polynomien numeerisia ominaisuuksia. Adaptiivisen tunnistuksen tehokkuutta mitatiin sekä muistittomille että muistillisille malleille. Lopulta esisärötysmenetelmien toimintaa tutkittiin käyttäen simuloitua tehovahvistinmallia sekä kaupallista tehovahvistinta. Käytettäessä esisärötystä, mitatun tehovahvistimen hyötysuhde parani noin 55%, mikä on osoitus ortogonaalisten sarjakehitelmien soveltuvuudesta tehovahvistin- ja esisärötysmallinnuksessa.

Avainsanat: ortogonaalinen polynomi, digitaalinen kantataajuusesisärötin, tehovahvistin, Wiener malli, adaptiivisen järjestelmän tunnistaminen

Preface

This thesis was carried out at the Institute for Communications and Radio-Frequency Engineering, Vienna University of Technology.

Firstly, I want to thank my supervisors Univ. Prof. Dr.-Ing. Markus Rupp and Prof. Risto Wichman. Secondly, I would like to thank my instructor Dipl.-Ing. Robert Dallinger for his excellent guidance. Without his suggestions and insightful discussions on the topic, the thesis would not be in the shape it is now.

I would also like to express my gratitude to the diploma student colleagues and the people at the institute for the great working atmosphere. My ten months period working on the thesis would have been much less fun without you.

Next, I would like to thank my parents for their love and support .

Finally, I would like to thank my girlfriend Silvia for her encouragement, understanding and love.

Vienna, 19.4.2010

Henri A. Ruotsalainen

Contents

Abbreviations

ACPR	Adjacent channel power ratio
ADC	Analog-to-digital converter
AM/AM	Amplitude modulation/Amplitude modulation
AM/PM	Amplitude modulation/Phase modulation
DAC	Digital-to-analog converter
DLA	Direct learning architecture
IBO	Input back-off
IDLA	Indirect learning architecture
IF	Intermediate frequency
IM	Intermodulation
FPGA	Field-programmable gate array
FxLMS	Filtered-x least mean squares
LMS	Least mean squares
NLMS	Normalized least mean squares
NFxLMS	Nonlinear filtered-x least mean squares
OFDM	Orthogonal frequency division multiplexing
PA	Power amplifier
PAE	Power added efficiency
PAPR	Peak-to-average power ratio
PC	Personal computer
PSD	Power spectral density
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase-shift keying
RAM	Random access memory
RF	Radio frequency
RRC	Root-rised cosine
SSPA	Solid state power amplifier
TCP/IP	Transmission control protocol/Internet protocol
TWTA	Traveling wave tube amplifier
VISA	Virtual instrument software architecture
W-CDMA	Wideband code-division multiple access
WLAN	Wireless local area network
WSS	Wide sense stationary

Chapter 1

Introduction

1.1 Motivation

Wireless data transmission rates have been rising significantly during the past decade due to the growing popularity of wireless networks and mobile Internet access. In modern wireless local area networks (WLAN), the data traffic consists more and more of high resolution images, videos and sound data [1]. Therefore, the available wireless bandwidth needs to be utilized as efficiently as possible to achieve the requested data transmission rates and to provide good service quality to the end users.

To meet the requirements addressed to the shared transmission medium, modulation techniques with higher spectral efficiency must be chosen. The receding GSM networks, utilizing digital transmission with time domain multiple access to allow multiple calls on the same frequency channel, introduced a modest demand for linearity along the transmission path [2]. Ever since, new modulation techniques that are utilizing the spectrum more efficiently have been introduced to handle the growing data transmission rates.

Most modern modulation techniques, such as orthogonal frequency division multiplexing (OFDM) and quadrature amplitude modulation (QAM) utilize wide transmission band and offer high spectral efficiency. These modulation techniques utilize variable amplitude modulation, where the non-constant envelope signal exhibits large deviations in the amplitude. On the other hand, these modulation techniques require very tight linearity conditions in order to avoid any defects in the signal shape. This requirement becomes an important issue, when the signal is amplified.

The radio frequency (RF) power amplifiers (PAs) are inherently nonlinear devices. In practice, the best efficiency is reached when the PA is driven into its saturation area. This nonlinear behavior increases the amount of distortion in both amplitude and phase. Especially, the aforementioned modulation techniques with high peak-to-average power ratio (PAPR) suffer from this nonlinear distortion. A simple way to reduce the nonlinearities is to decrease the input power. However, this action reduces the efficiency of the PA significantly.

A common measure for the PA efficiency is the power added efficiency (PAE) [3]. The

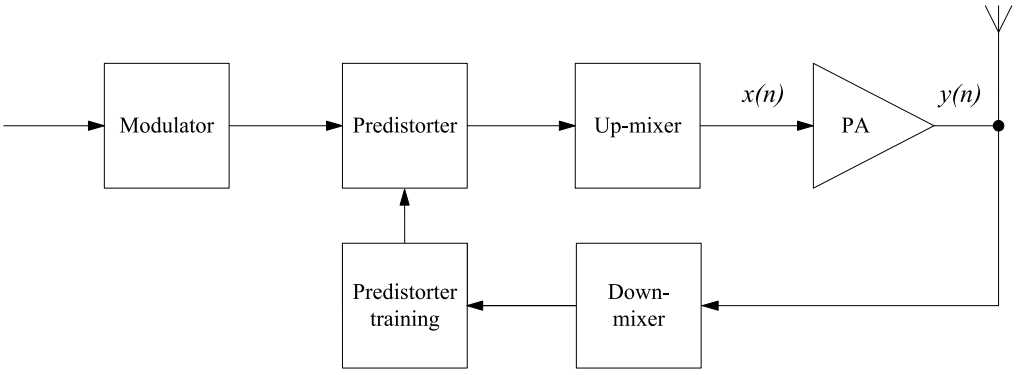

Figure 1.1: Illustration of the RF transmitter including predistorter configuration

PAE is defined as

$$\text{PAE} = \frac{\mathrm{P}_{\text{out}} - \mathrm{P}_{\text{in}}}{\mathrm{P}_{\text{dc}}} 100\%, \tag{1.1}$$

where P_{out} denotes the output power of the PA, P_{in} the respective input power of the PA and P_{dc} the direct current (DC) power of the PA.

Although the recent improvements in PA design have increased the efficiency [4, 5], the typical PAE rates range between 10% - 50% [6]. Consequently, the power amplifiers are usually dominating the power consumption in mobile communication devices. It is reported that the PA inside the sender consumes approximately 70% of the available battery life [7]. In a mobile terminal, it is therefore desireable to minimize the power dissipation to gain longer battery life [8].

In base stations, the importance of efficient PAs is even bigger since the operating power levels are higher. Better efficiency of this kind of PA will lead to longer PA lifetime and savings in the electricity consumption for the mobile service providers [9]. The importance of increased PA efficiency can be demonstrated by the given example in [9], where annual savings of 164 million kilowatt hours were obtained in a network of approximately 10000 base sites using more efficient high power RF amplifiers .

1.2 Objective

The PA as part of an RF transmitter as depicted in the Figure 1.1, is a component which converts a weak RF signal into an amplified version. Ideally, the performance of the PA is entirely linear and the output signal equals the input signal multiplied by a certain gain constant K, i.e.,

$$y\left(t\right) = Kx\left(t\right). \tag{1.2}$$

In reality, the output power of the PA cannot rise up to arbitrarily high levels as shown in Figure 1.2. At some point, when the input amplitude exceeds a certain level the output of

the PA starts to saturate and the gain drops, meaning that the PA does not behave linear in this region. Hence, if nonlinear behavior cannot be accepted, the operating range of the PA has to be limited to its linear area which causes a notable proportion of the output power range to be unusable.

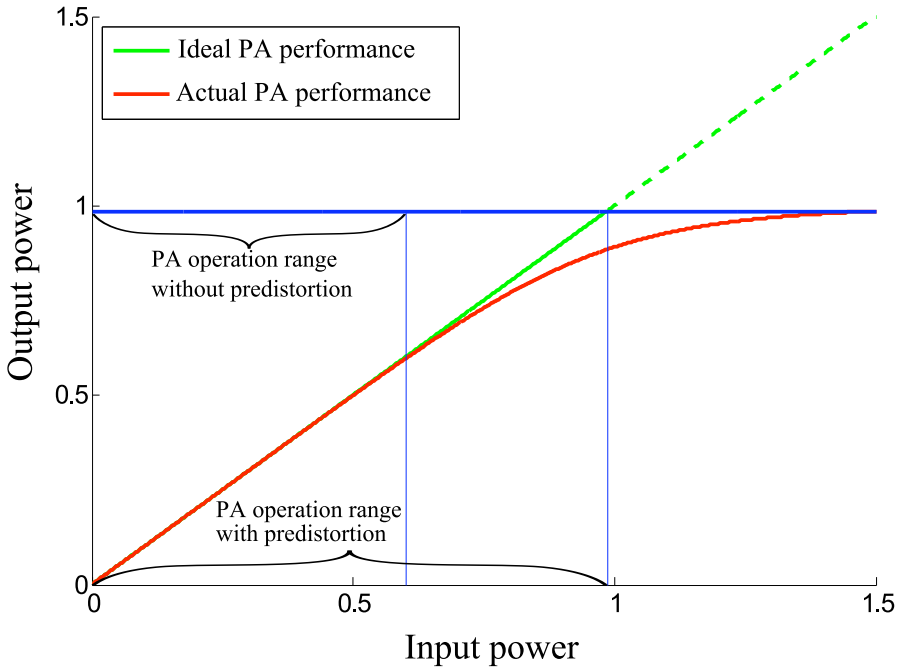

Figure 1.2: The desired and the actual gain performance of a PA

To extend the operational range, for higher input levels the nonlinear behavior must be reduced. In order to do so, it is assumed that the nonlinear behavior is known and representable by some arbitrary time-dependent function $P\left[t, x\left(t\right)\right]$. Then, the output of the PA can be expressed as

$$y\left(t\right) = P\left[t, x\left(t\right)\right]. \tag{1.3}$$

Furthermore, if an inverse P^{-1} of the nonlinear function P is exists such that

$$x\left(t\right) = P^{-1}\left[t, y\left(t\right)\right], \tag{1.4}$$

the cascade of these two functions, P and P^{-1}, would produce an output with the desired constant gain. However, a perfect inverse is rarely available for the entire input range. Moreover, since P depends on the time t, it would also need to be known for all t. Nevertheless, even for an approximation of P^{-1}, the operation range can be extended significantly.

Linearization methods, which modify the input signal before the amplification, are better known as predistortion. Commonly, predistorter is implemented in the digital baseband domain using digital signal processors (DSPs) and/or field-programmable gate arrays (FPGAs), which leads to very flexible designs. For the predistorter, it would be also necessary

to have the ability to track the variations in PA characteristics dynamically whenever they occur. For the digital predistorter this is not an obstacle, since the predistorter algorithms can be implemented adaptively.

In this thesis, static and dynamic nonlinear models based on orthogonal basis expansions are considered for modeling the PAs, respectively the predistorters. For the dynamic nonlinear models, simplified Wiener model and Hammerstain model are considered which are represented by a combination of a linear filter and an orthogonal basis expansion. It is investigated how different basis expansions perform in the context of adaptive system identification. Moreover, performance of several adaptive predistorter learning structures is evaluated by simulations and burst measurements.

1.3 Outline

This thesis is structured as follows. In Chapter 2, several models for PA modeling are presented. The PA models include memoryless models and models with memory based on orthogonal polynomials. In addition, the characteristics and nonlinear behavior of the PAs are discussed. In Chapter 3, adaptive identification algorithms, adaptive system identification method for nonlinear dynamic models and adaptive inverse system identification methods are presented. In Chapter 4, three predistorter learning architectures are introduced: the direct learning architecture, the indirect learning architecture and the direct learning architecture based on the nonlinear filtered-x least mean squares algorithm. In Chapter 5, the numerical properties and the convergence properties of the orthogonal basis expansions are investigated. Furthermore, the performance of the different adaptive predistorter structures is evaluated for the theoretical PA models and for a commercial PA.

Chapter 2

Power amplifier models

PA modeling is a largely studied topic in the context of wireless communications. It is well known that the non-ideal behaviour of the PA affects the quality of the transmitted signal. Thus, the limitations caused by the impairments are a common interest. Depending on the characteristics of the PA and the application, several modeling approaches are available. Recent studies of the PA modeling are concentrating on behavioral modeling, where the model is constructed from the measured input/output relation. Behavioral PA modeling originates from the more general (nonlinear) system identification theory, which has a strong theoretical framework [10]. Thus, the applicability of the behavioral models can be justified.

In this chapter, several behavioral power amplifier models are discussed. At first, in Section 2.1 the characteristics and the nonlinear behaviour of PAs are discussed. In Section, 2.2 memoryless models are reviewed, which have relatively simple structure and serve as a good introduction to the more complex models. Memoryless PA models are often adequate when the input stimulus is narrowband [11]. When the bandwidth of the input signal increases, memory effects introduce additional distortions to the amplified output signal. PA models capable of modeling the memory effects are presented in Section 2.3.

2.1 Characteristics of power amplifiers

High efficiency and linearity are characteristics often desired from the PAs. Unfortunately, these two features which depend on the design of the PA, can hardly be achieved simultaneously. In order to establish precise PA models, it would be necessary to have knowledge of the impairments caused by the nonlinearities.

2.1.1 Classification of power amplifiers

PAs can be classified according to the design of their inner circuitry [3]. The classification is basicly based on the conduction angle, which determines the proportion of the input signal cycle that is conducted during amplification. This measure determines also roughly

the efficiency of the PA. Usually, simpler implementations mean more linear behavior and less distortion, but worse efficiency. The efficiency can be improved by more elaborate design which means also an increase in complexity and often also in distortion. In modern wireless networks, high linearity and high efficiency are required from the PAs. However, these characteristics are more or less mutually excluding each other and require usually a compromise.

Following PA classes with their corresponding efficiencies[1] are usually used to categorize PA circuits [3, 12, 13]:

Class	Theoretical maximum efficiency
A	50%
B	78.5%
AB	50-78.5%
C	90%
D	100%
E	100%

Class A amplifiers are known to be highly linear with low amount of distortion. These operating conditions are achieved by conducting over the whole conduction angle. The PAs operating in class A have very poor efficiency. This kind of PAs suffer also from serious heating problems.

For class B amplifiers, the efficiency is improved by conducting only over half of the conduction cycle. In practical applications so called the push-pull design is used, which combines two halves of the amplified signal thus allowing full conductance. However, this combining procedure results in cross-over distortions due to the synchronization error. In class AB amplifiers, the issue of cross-over distortion is mitigated by allowing conductance of more than 50% for each of the amplifiers. This means, however, a slight decrease in overall efficiency.

Class C type amplifiers conduct less than 50% of the input signal cycle. Therefore, the dynamic range is reduced and the nonlinear effects are very strong. However, very high efficiencies can be achieved.

Class D and E amplifiers utilize switching to achieve nearly optimal efficiency. For example, the switching can be implemented by using pulse width modulated input signal. The modulated signal with constant amplitude can then be amplified so that the amplifier is conducting only during the pulses. By this, a substantial increase in efficiency can be obtained.

Furthermore, there are numerous hybrid circuits based on combinations of the previously mentioned classes. One of them is so called Doherty amplifier which is a combination of class AB and C amplifiers. Due to better compatibility with variable amplitude modulation techniques, the popularity of Doherty amplifiers has increased in the base stations and handset RF amplifiers [14].

[1] Theoretical efficiency rates presented here are based on the ideal PA performance for sinusoidal excitation. However, the practical efficiency rates of the same type of PAs can be much smaller.

It is well known that the efficiency of the class A,B and AB amplifiers can be very poor when they are used in their linear region. Therefore, the use of linearization procedures may result in major improvements in efficiency of the amplifiers based especially on the aforementioned classes. On the other hand, the linearization can be also used to reduce the strong nonlinear effects of the class C,D and E PAs.

2.1.2 Nonlinear effects of power amplifiers

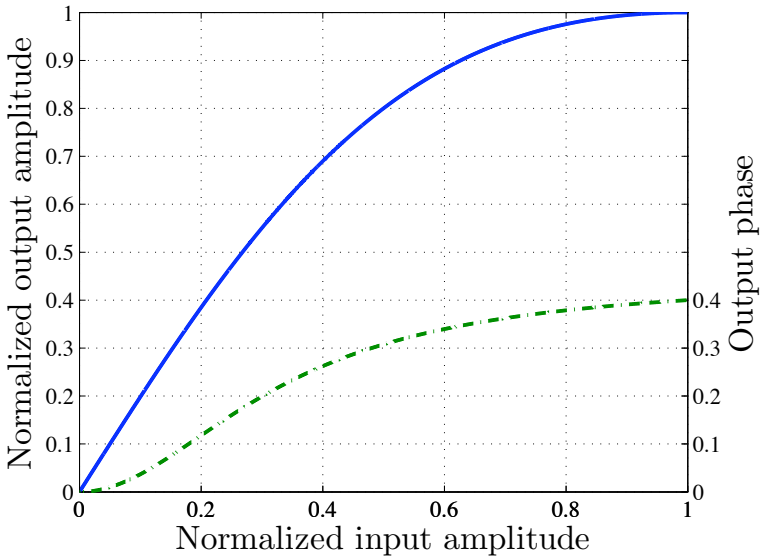

Figure 2.1: Example of AM/AM (solid line) and AM/PM (dashed line) curves

As mentioned in Section 1.1, a PA operating at its best efficiency will produce distortions in the amplitude as well as in the phase. A careful analysis of the amplitude and phase distortions will give the basic information of the characteristics of the PA.

A common way to model the nonlinear effects of a PA is to use Amplitude modulation/Amplitude modulation (AM/AM) and Amplitude modulation/Phase modulation (AM/PM) conversions. In the AM/AM conversion, the nonlinearly distorted output amplitude is obtained as a function of the instantaneous values of the input amplitude [1].

The input baseband signal $V_{\text{in}}(t)$ up-converted to the transmission frequency f_C can be expressed by its complex envelope representation [1]

$$y(t) = |V_{\text{in}}(t)|\, e^{j[2\pi f_C t + \angle V_{\text{in}}(t)]}, \tag{2.1}$$

where $|V_{\text{in}}(t)|$ denotes the input amplitude and $\angle V_{in}(t)$ the input phase. To describe the conversions to both amplitude and phase, corresponding mapping functions $A[.]$ and $\Phi[.]$ are introduced. Assigning these two nonlinear mapping functions to the complex envelope representation (2.1), the output of the PA can be formulated as

$$y(t) = A[|V_{\text{in}}(t)|]\, e^{j[2\pi f_C t + \angle V_{\text{in}}(t) + \Phi[|V_{\text{in}}(t)|]]}. \tag{2.2}$$

The behavior of both functions is illustrated in Figure 2.1. At low amplitude levels the amount of distortion in both amplitude and phase is notably lower, i.e., the amplifier operates in its linear region. The compressive effect becomes dominant when the input amplitude grows beyond the saturation level (about 40% of the input amplitude in Figure 2.1).

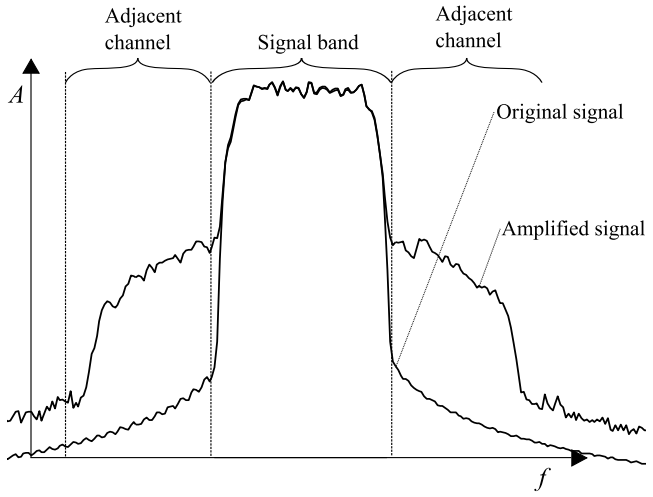

Figure 2.2: Nonlinear distortion in frequency domain, where the signal band is spread to the adjacent channels.

Modulation techniques such as QAM and OFDM used in wireless wideband communication systems are spectrally efficient. To utilize the spectrum as efficient as possible, multiple signals are transmitted simultaneously on the same frequency band. When the input signals are converted up from the baseband to the RF frequency, and amplified by the nonlinear power amplifier, addition distortion components are induced to them. These distortion components are classified depending on their spectral location with respect to the fundamental frequency [3]. Far-off harmonic distortion components are located outside the frequency range of the single transmission channel.They can be suppressed by simple filtering operation. Thus, they have less effect on the transmitted signal. Therefore, the main concern is with in the intermodulation (IM) products located inside or very close to the transmission band itself.

The IM causes the signal to spread to the adjacent channels which is illustrated in the Figure 2.2. This phenomenon is also termed spectral regrowth. When two adjacent channels exhibit spectral spreading, the out-of-band distortion products of one channel interfere with signal components in the transmission band of the other channel. This, in the end, causes a discrepancy between the transmitted and received signal and leads to an increase in the bit error rate (BER). The amount of spectral regrowth is usually measured by the adjacent-channel power ratio (ACPR). It is defined as the ratio of the power within the transmission band and the power spread to the adjacent channels [9, 10]:

$$\mathrm{ACPR} = 10\log_{10}\left\{\frac{\int_{\omega_{\mathrm{adj}}}\mathrm{PSD}_{\mathrm{ADJ}}\left(\omega\right)d\omega}{\int_{\omega_{\mathrm{in}}}\mathrm{PSD}_{\mathrm{IN}}\left(\omega\right)d\omega}\right\}, \tag{2.3}$$

where ω_{adj} and ω_{in} denote the bandwidths of the adjacent channel and the transmission channel, respectively. The frequency range of the adjacent channels is not necessarily the same as the frequency range of the transmission channel.

The nonlinear amplification in the saturation area introduces compression to the input signal amplitude.Especially, the modulation schemes with high PAPR suffer from this kind of inband distortion. The resulting clipping of the signal peaks increases errors in the demodulation at the receiver. For example, in M-QAM modulation, the signal constellation will become corrupted due to the saturation of higher amplitudes leading to an increased BER.

In addition to the amplitude and phase distortions, for transmit signals with higher bandwidth, RF PAs exhibit memory effects. Memory effects can be modelled as variations in the AM/AM and AM/PM characteristics with respect to the modulation frequency [2]. Memory effects can be classified into two groups according to their origin. Thermal memory effects are caused by power dissipation in the electronic components. The dissipated power, which depends on the input power level, affects the transfer characteristics for proximately following portions of the input signal, thus creating long term dependencies to the transmitted signal. Electrical memory effects are another source of memory, caused mainly by the frequency dependent impedances in the bias networks. This kind of non-constant distortion behavior poses challenges for the design of the PAs, for the PA system identification schemes and especially for implementation of the linearization procedures.

2.2 Memoryless nonlinear models

The static nonlinearity of a PA can be modelled using trucated power series. However, the power series modeling suffers from numerical instabilities which can become remarkable when modeling higher order nonlinearities. In order to overcome the numerical stability problems of the polynomial based models, orthogonal polynomials can be used instead.

2.2.1 Conventional polynomial model

In case of a (strictly) memoryless PA, the truncated power series model is sufficient to model the PAs baseband input-output relation [15]. This kind of polynomial modeling technique which can be used to model the bandpass nonlinearities is based on the work of [16, 17, 15].

The bandpass model, depicted in the figure 2.3, consists of a PA and a bandpass filter. The bandpass filter acts as a zonal filter with adequate bandwidth, picking out only the frequency components which fall into the range of the fundamental frequency. In the

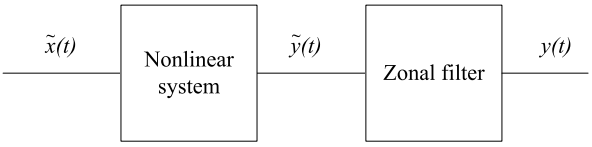

Figure 2.3: Bandpass nonlinearity model

passband, the PA can be described using truncated power series representation

$$\tilde{y}(t) = \sum_{k=0}^{K} \tilde{b}_k \tilde{x}^k(t), \tag{2.4}$$

where $\tilde{x}(t)$ denotes the passband input, $\tilde{y}(t)$ the passband output and $\tilde{b}_k$ the real valued polynomial coefficient. When the passband input signal $\tilde{x}(t)$ passes through the static nonlinearity, distortions occurs in the corresponding output signal $\tilde{y}(t)$ as mentioned previously in Chapter 1. The spectral components due to even order terms in (2.4) are all located outside the fundamental frequency band. By the zonal filtering, they are all suppressed and only $y(t)$ is affected only by odd order terms. The baseband behavior of the respective bandpass model can then be shown to obey the following equation [1, 15]

$$y(t) = \sum_{k=0}^{\left\lfloor \frac{K}{2} \right\rfloor} b_k |x(t)|^{2k} x(t), \tag{2.5}$$

where the baseband distortion coefficient is defined as

$$b_k = 2^{2-k} \begin{pmatrix} k+1 \\ \frac{k}{2} \end{pmatrix} \tilde{b}_k. \tag{2.6}$$

Depending on the presence of short term memory effects, the nonlinear system can be regarded as quasi-memoryless or strictly memoryless. In the latter case, the distortion coefficients b_k are real valued and thus the power series model is only capable of modeling the nonlinearities in the amplitude. In reality, though, PAs always exhibit phase distortions and so the baseband model (2.5) needs to be extended to the complex domain.

According to the Weierstrass approximation theory, any function which is analytical in closed interval can be approximated by a polynomial of arbitrary order [18]. Furthermore, if the order of the polynomial is allowed to grow infinitely, the error between the function and the function estimate is minimized.

However, when a physical PA is modelled using the polynomial model (2.5), only the models with finite polynomial orders can be utilized. Therefore, the baseband behaviour of the PA cannot be modeled exactly, but only approximated with an arbitrary accuracy. It has been suggested that the presence of even order distortion products may improve the overall accuracy of the approximation when the systems input stimulus is broadband type [19,20]. This also allows to use lower order polynomials which have better numerical

properties. Under these assumptions Equation (2.5) becomes

$$\hat{y}(t) = \sum_{k=1}^{K} b_k |x(t)|^{k-1} x(t). \tag{2.7}$$

When referring to the *conventional polynomial baseband model*, this model will be used throughout the thesis unless otherwise noted.

2.2.2 Numerical stability of the conventional polynomial baseband model

The coefficients b_k in (2.5) and (2.7) can be obtained using least squares approximation [21]. For the sake of simplicity, let $\phi_k(t) = |x(t)|^{k-1} x(t)$ represent the monomial of the input baseband data at time t and let $\{t_1, t_2, \ldots, t_N\}$ represent N different time instants. Then, the Van der Monde type matrix $\mathbf{\Phi}$ of size $N \times K$, the $1 \times N$ output data vector $\mathbf{y}$, the $1 \times N$ estimated data vector $\hat{\mathbf{y}}$, and the $1 \times K$ coefficient vector $\mathbf{b}$ are of the form

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1(t_1), & \phi_2(t_1), & \ldots, & \phi_K(t_1) \\ \phi_1(t_2), & \phi_2(t_2), & \ldots, & \phi_K(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(t_N), & \phi_2(t_N), & \cdots & \phi_K(t_N) \end{bmatrix}, \tag{2.8}$$

$$\mathbf{b} = \begin{bmatrix} b_1, b_2, \ldots, b_K \end{bmatrix}^T, \tag{2.9}$$

$$\mathbf{y} = \begin{bmatrix} y(t_1), y(t_2), \ldots, y(t_N) \end{bmatrix}^T, \tag{2.10}$$

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{y}(t_1), \hat{y}(t_2), \ldots, \hat{y}(t_N) \end{bmatrix}^T. \tag{2.11}$$

Using these definitions, the baseband model (2.5) can be expressed as a set of linear equations in the form

$$\hat{\mathbf{y}} = \mathbf{\Phi}\mathbf{b}. \tag{2.12}$$

Using the matrix notation, an estimator minimizing the least square error can be constructed according to [22]. At first, let $\mathbf{e}$ denote the estimation error vector defined by

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{\Phi}\mathbf{b}. \tag{2.13}$$

Then the cost function ξ to be minimized can be defined by

$$\xi = \mathbf{e}^H \mathbf{e}. \tag{2.14}$$

The minimum can be found by setting the derivative of the cost function respect to the coefficients to zero as follows

$$\frac{\partial \xi}{\partial \mathbf{b}} = -2\mathbf{\Phi}^H \mathbf{e} \;=\; 0. \tag{2.15}$$

Then, it follows that

$$\begin{aligned} -2\mathbf{\Phi}^H(\mathbf{y} - \mathbf{\Phi}\mathbf{b}) &= 0, \\ \mathbf{\Phi}^H\mathbf{y} - \mathbf{\Phi}^H\mathbf{\Phi}\mathbf{b} &= 0, \end{aligned} \tag{2.16}$$

which finally leads to

$$\mathbf{b} = \left(\mathbf{\Phi}^H\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^H\mathbf{y}. \tag{2.17}$$

The precision of the approximation depends heavily on the condition of the autocorrelation matrix product $\mathbf{\Phi}^H\mathbf{\Phi}$, which appears in the solution 2.17. For demonstrating the magnitude of the condition number, the input is assumed to be white uniform distributed with amplitude values between $[0..1]$. This allows to calculate the average condition number of $E\left[\mathbf{\Phi}^H\mathbf{\Phi}\right]$. In order to simplify the evaluation, let $\boldsymbol{\phi}_k(\mathbf{x})$ denote a $1 \times N$ vector of monomials defined by

$$\boldsymbol{\phi}_k(\mathbf{x}) = \left[\; \phi_k(t_1), \phi_k(t_2), \ldots, \phi_k(t_N) \;\right]^T. \tag{2.18}$$

Then the evaluation of $E\left[\mathbf{\Phi}^H\mathbf{\Phi}\right]$ becomes

$$E\left[\mathbf{\Phi}^H\mathbf{\Phi}\right] \;=\; E\begin{bmatrix} \boldsymbol{\phi}_1^H(\mathbf{x})\,\boldsymbol{\phi}_1(\mathbf{x}) & \boldsymbol{\phi}_1^H(\mathbf{x})\,\boldsymbol{\phi}_2(\mathbf{x}) & \ldots & \boldsymbol{\phi}_1^H(\mathbf{x})\,\boldsymbol{\phi}_K(\mathbf{x}) \\ \boldsymbol{\phi}_2^H(\mathbf{x})\,\boldsymbol{\phi}_1(\mathbf{x}) & \boldsymbol{\phi}_2^H(\mathbf{x})\,\boldsymbol{\phi}_2(\mathbf{x}) & \ldots & \boldsymbol{\phi}_2^H(\mathbf{x})\,\boldsymbol{\phi}_K(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}_K^H(\mathbf{x})\,\boldsymbol{\phi}_1(\mathbf{x}) & \boldsymbol{\phi}_K^H(\mathbf{x})\,\boldsymbol{\phi}_2(\mathbf{x}) & & \boldsymbol{\phi}_K^H(\mathbf{x})\,\boldsymbol{\phi}_K(\mathbf{x}) \end{bmatrix} \tag{2.19}$$

For stationary processes, the following statement is valid

$$E\left[\boldsymbol{\phi}_n^H(\mathbf{x})\,\boldsymbol{\phi}_k(\mathbf{x})\right] = N\,E\left[\phi_n^*(t)\,\phi_k(t)\right]. \tag{2.20}$$

The product $\phi_n^*(t)\,\phi_k(t)$ can be evaluated as follows

$$
\begin{aligned}
& \phi_n^* (t)\, \phi_k (t) \\
= \;& |x (t)|^{n-1}\, x (t)^*\, |x (t)|^{k-1}\, x (t) \\
= \;& |x (t)|^{n+k-2}\, x (t)^*\, x (t) \\
= \;& |x (t)|^{n+k} \\
= \;& r^{n+k} \qquad , (r = |x (t)|).
\end{aligned} \tag{2.21}
$$

Then the expectation $E [\phi_n^* (t)\, \phi_k (t)]$ equals the statistical moment of order $n + k$ of the random variable r, which is defined by

$$
\mu_{nk} = E [\phi_n^* (t)\, \phi_k (t)] = \int_a^b r^{n+k} f (r)\, dr, \tag{2.22}
$$

where $f (r)$ denotes the probability density function of r. For the uniformly distributed r on the $[0..1]$, the moment (2.22) reduces to

$$
\begin{aligned}
\mu_{nk} &= \int_0^1 r^{n+k} dr, \\
&= \frac{1}{n+k+1} r^{n+k+1}, \\
&= \frac{1}{n+k+1} - 0, \\
&= \frac{1}{n+k+1}.
\end{aligned} \tag{2.23}
$$

By arranging these moments to a matrix, the product $E \left[\mathbf{\Phi}^H \mathbf{\Phi}\right]$ is constructed. The obtained matrix is a form of a generalized Hilbert matrix [21], which is a classical example for an ill-conditioned matrix.

For polynomials of very low order and at low noise levels, the conventional polynomial model may be adequate as stated in [11]. Therefore, the baseband representation could be used to model the systems that are mostly linear and show only very weak nonlinearities [10]. However, for example in power amplifier modeling, the degree of nonlinearity can become remarkable. This is the case, if the PA is driven up to saturation which, as pointed out in Chapter 1, is necessary to achieve the highest possible power efficiency. Consequally, in the context of PA modeling numerical stability requires additional attention.

2.2.3 Orthogonal polynomials

In Section 2.2.2, the numerical stability problem arose when the least squares estimate using conventional polynomials was obtained. Indeed, when the autocorrelation matrix $E\left[\mathbf{\Phi}^H\mathbf{\Phi}\right]$of the conventional polynomial basis was examined, it became evident that the accuracy of the inverse degrades very fast with increasing order of the polynomial base. For better estimation results, it would be beneficial to consider polynomial bases which are different to the conventional polynomial bases, with the aim to achieve better conditional autocorrelation matrices. Here, this leads to the introduction of the orthogonal polynomial bases.

Given two polynomial functions

$$\begin{aligned}\psi_n(x) &= \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \ldots + \alpha_{n+1} x^{n+1},\\ \psi_m(x) &= \beta_1 + \beta_2 x + \beta_3 x^2 + \ldots + \beta_{m+1} x^{m+1},\end{aligned}$$

with two positive integers m and n, thay said to be orthogonal [23], if

$$E\left[\psi_n^*(x)\,\psi_m(x)\right] = \int_b^a \psi_n^*(x)\,\psi_m(x)\,w(x)\,dx = \delta_{nm}N_{nm}; \quad m,n \in \mathbb{N}, \tag{2.24}$$

where $w(x)$ denotes a weight function, δ_{nm} the Kronecker delta, N_{nm} an arbitrary order dependent normalization factor, and the$[a,b]$ the range of orthogonality. If the equality in (2.24) holds for all combinations of n and m with $n \leq m$, the obtained set of polynomials $S_n = \{\psi_1(x), \psi_2(x), \ldots, \psi_n(x)\}$ forms an *orthogonal polynomial basis.* In case that all normalization factors N_{nm} equal one, the set of polynomials is called an *orthonormal polynomial basis.*

A special group of orthogonal polynomials are so called classical orthogonal polynomials [24]. They have important applications in physics, approximation theory and numerous other fields. Three kinds of polynomials are often classified as classical orthogonal polynomials:

- Jacobi polynomials
- Laguerre polynomials
- Hermite polynomials

Additionally, there are many important subclasses of the previously mentioned polynomials. For example, some well known orthogonal polynomials are specializations of the Jacobi polynomials.

- Legendre polynomials
- Chebyshev polynomials
- Gegenbauer polynomials

Several variations of the classical polynomial families exist where the basis is either scaled or shifted in order to maintain the orthogonality property. The related counterparts to the original polynomials are often called shifted or associate polynomials. For example the Legendre polynomials are defined by default on the range $[-1, 1]$. Sometimes though, it is more useful to have the range of orthogonality shifted between positive values. With a simple shifting operation $y = 2x - 1$ the range can be transformed onto $[0, 1]$ and so the new obtained basis is called *shifted* Legendre polynomials. Some of the classical polynomials are also extended to a complex variable, where the range of orthogonality is defined on the unit circle.

2.2.4 Orthogonal basis expansion

Consider an arbitrary function $\varphi(x)$ that is analytical on some closed region in $L_2(a, b)$. Then, $\varphi_n(x)$ can be approximated by a linear combination of orthogonal polynomials $\psi_k(x)$ as follows

$$\varphi(x) = \sum_{k=0}^{N} \alpha_k \psi_k(x), \tag{2.25}$$

where α_k denotes the k^{th} coefficient of the corresponding polynomial basis function. This definition is also known as *orthogonal basis expansion*. In other words, for every function which is continuous function on a finite region can on this region be represented as a linear combination of orthogonal polynomial basis functions.

Another important property concerning the orthogonal polynomials is their ability to produce an estimate that is minimum in the least squares sense. This is a direct consequence from the completeness property, i.e.

$$\lim_{N\to\infty} \left\| \varphi(x) - \sum_{k=0}^{N} \alpha_k \psi_k(x) \right\| = 0. \tag{2.26}$$

Similar to (2.12), the matrix representation of the orthogonal polynomial expansion for the orthogonal baseband model can be expressed as

$$\hat{\mathbf{y}} = \mathbf{\Psi}\mathbf{b} \tag{2.27}$$

and

$$\mathbf{\Psi} = \mathbf{\Phi}\mathbf{U} = \begin{bmatrix} \psi_1(t_1), & \psi_2(t_1), & \ldots, & \psi_K(t_1) \\ \psi_1(t_2), & \psi_2(t_2), & \ldots, & \psi_K(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(t_N), & \psi_2(t_N), & \cdots & \psi_K(t_N) \end{bmatrix} \tag{2.28}$$

where $\mathbf{b}$ denotes the coefficient vector $[b_1, \ldots, b_K]^T$, $\mathbf{\Phi}$ is given by (2.8), $\mathbf{U}$ the orthogonal polynomials coefficient matrix, and $\psi_k(.)$ orthogonal polynomial of order k.

The derivation of the orthogonal polynomial basis expansion for the baseband model (2.7) with an arbitrarily distributed (complex) input sequence is generally a difficult task. As mentioned before, the Van-der-monde type matrix (2.8), which appears in the least squares solution is numerically instable. Thereby, finding a proper set of orthogonal polynomials will increase the numerical stability considerably. The least squares estimator for the coefficients $\mathbf{b}$ can be determined in similar fashion to the conventional baseband model (2.17).

$$\mathbf{b} = \left(\mathbf{\Psi}^H \mathbf{\Psi}\right)^{-1} \mathbf{\Psi}^H \mathbf{y} \tag{2.29}$$

It is desirable to keep the condition number of the resulting matrix multiplication product low, because the LS estimator involvesan inversion of $\mathbf{\Psi}^H\mathbf{\Psi}$. Since $\mathbf{\Psi} = \mathbf{\Phi}\mathbf{U}$, it is possible to rewrite the inversion component as $\mathbf{U}^H\mathbf{\Phi}^H\mathbf{\Phi}\mathbf{U}$. Thus, the search for an optimum polynomial basis "reduces" to the identification of a set of orthogonal polynomials leading to a matrix $\mathbf{U}$ which leads to the best numerical stability in the least squares solution (2.29).

For the orthogonal basis to be found, the existence of statistical moments of the input probability density function is required. Thus, the primary concern is to study the expectation of $\mathbf{\Psi}^H\mathbf{\Psi}$.

$$E\left[\mathbf{\Psi}^H\mathbf{\Psi}\right] \;=\; E\left[\mathbf{U}^H\mathbf{\Phi}^H\mathbf{\Phi}\mathbf{U}\right] = \mathbf{U}^H E\left[\mathbf{\Phi}^H\mathbf{\Phi}\right]\mathbf{U}. \tag{2.30}$$

Once the matrix of the orthogonal polynomial coefficients $\mathbf{U}$ is found, according to the orthogonality property (2.24), the expectation of $E\left[\mathbf{\Psi}^H\mathbf{\Psi}\right]$ will become a diagonal matrix (under the assumption that the respective input probability density function matches with the one that was used to derive set of orthogonal polynomials).

$$E\left[\mathbf{\Psi}^H\mathbf{\Psi}\right] = \begin{bmatrix} E\left[|\psi_1(t_1)|^2\right] & 0 & \dots & 0 \\ 0 & E\left[|\psi_2(t_2)|^2\right] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E\left[|\psi_K(t_K)|^2\right] \end{bmatrix}. \tag{2.31}$$

In the ideal case of exact orthonormality, the obtained matrix (2.31) would reduce to the identity matrix and the matrix inversion in the LS-solution (2.29) would become unnecessary. However, this rarely the case since the exact moments for the input sequences of the baseband model are not available or are hard to derive. Therefore, it makes sense to use known classical orthogonal polynomials which use an adequate weight function, instead of deriving a new unique orthonormal basis. Then, the weight function will act as an approximation to the input probability density function, yielding to a lower condition number of $\mathbf{\Psi}^H\mathbf{\Psi}$ compared to $\mathbf{\Phi}^H\mathbf{\Phi}$.

2.2.5 Orthogonal polynomial models for the conventional baseband model

In modern wireless communication systems, modulation schemes such as OFDM, M-QAM and QPSK with different variations are very popular. Depending on the modulation technique, the amplitude distribution of the modulated signal will vary notably. For example for OFDM, the modulated signal is approximately circularly symmetrical complex Gaussian distributed [16, 25]. This assumption motivates to search for a basis expansion which is orthogonal with respect to the complex Gaussian distribution. However, other modulation methods like M-QAM which have different and more complicated amplitude characteristics, cannot be related directly to a priori known probability density function.

The variations in the characteristics of the amplitude distribution have an effect on the performance of the orthogonal basis expansion. For an adequate modelling accuracy, though, it is not necessary to have an amplitude distribution which perfectly matches the corresponding weight function of the orthogonal polynomials. It is more important to make sure that the condition number of the LS estimator of the orthogonal baseband model (2.29) keeps rather low. From this point of view, it would make sense to study different basis expansions and their sensitivity to different amplitude distributions.

In this thesis, three orthogonal polynomials and respective orthogonal sets for the conventional baseband model are considered:

- Complex Hermite polynomials
- Shifted Legendre polynomials
- Shifted Chebyshev polynomials of the second kind

These particular polynomials were selected by comparing the weight functions of the corresponding classical orthogonal polynomials and the amplitude distributions of the different baseband signals. It was noticed that the weight functions of the three previously mentioned polynomials are close to the amplitude distributions of the digital baseband signals modulated with for example M-QAM and OFDM. There are numerous orthogonal polynomials available. Hence, for this thesis it was necessary to restrict to only a few of them. The above listed three polynomials were seen to be an adequate choice.

Complex Hermite polynomials

The circular complex Gaussian distribution with unit variance $\sigma^2 = 1$ and zero mean $\mu = 0$, has the following probability density function

$$f(z) = \frac{1}{\pi} e^{|z|^2} \tag{2.32}$$

Classical orthogonal polynomials having a weight function $e^{|z|^2}$ which closely relate to the complex Gaussian distribution, are the so called univariate complex Hermite polynomials

$\psi_{G,1}(z) = z$
$\psi_{G,2}(z) = \frac{1}{\sqrt{2}}\left(z\left\|z\right\|^2 - 2z\right)$
$\psi_{G,3}(z) = \frac{1}{\sqrt{12}}\left(z\left\|z\right\|^4 - 6z\left\|z\right\|^2 + 6z\right)$
$\psi_{G,4}(z) = \frac{1}{\sqrt{144}}\left(z\left\|z\right\|^6 - 12z\left\|z\right\|^4 + 36z\left\|z\right\|^2 - 24z\right)$
$\psi_{G,5}(z) = \frac{1}{\sqrt{2880}}\left(z\left\|z\right\|^8 - 20z\left\|z\right\|^6 + 120z\left\|z\right\|^4 - 240z\left\|z\right\|^2 + 120z\right)$

Table 2.1: First five *complex Gaussian polynomial* functions

[26]. In this work they will be briefly referred to by complex Hermite polynomials. They can be derived using Rodriguez formula [27],

$$H^{m,n}(z, z^*) = (-1)^{m+n} e^{|z|^2} \frac{\partial^{m+n}}{\partial z^n \partial z^{*m}} e^{-|z|^2} \tag{2.33}$$

Complex baseband model for the complex Gaussian processes was given in [28]. Relationship between the given baseband model and the complex Hermite polynomials can be indicated straightforwardly. Since the single monomial of the baseband model can be written as $|x|^{2k} x$ or on the other hand $x^k x^{*k+1}$ the complex Hermite polynomials related to the complex Gaussian baseband model are of the form

$$H^{n-1,n}(z, z^*) = (-1)^{2n-1} e^{|z|^2} \frac{\partial^{2n-1}}{\partial z^{n-1} \partial z^{*n}} e^{-|z|^2}, n \geq 1 \tag{2.34}$$

The complex Hermite polynomial functions include only odd order terms. Thus, the basis expansion obtained using complex Hermite polynomials equals to the odd order baseband model (2.5). The complex Hermite polynomials can be shown to be orthogonal with respect to the weight $e^{|z|^2}$ function as follows

$$\int_{-\infty}^{+\infty} H^{n-1,n}(z, z^*) H^{l-1,l}(z, z^*) e^{|z|^2} dz = \pi \delta_{nl} n! (n-1)!. \tag{2.35}$$

From Equation (2.35) it is visible that the kernel of the inner product does not match exactly with the original weight of the complex Gaussian distribution in (2.32). Therefore an additional scaling operation is required to obtain a complete orthonormal basis [28]. Multiplying Equation (2.34) by the the normalization factor $\frac{1}{\sqrt{n!(n-1)!}}$ leads to the orthonormal polynomial basis function

$$\psi_{G,n}(z) = \frac{1}{\sqrt{n!(n-1)!}} H^{n-1,n}(z, z^*). \tag{2.36}$$

First five polynomials related to the complex Hermite polynomials are listed in Table 2.1. In this thesis, these polynomials will be briefly referred to by *complex Gaussian polynomials*.

Shifted Legendre polynomials

The Legendre polynomials arise from the power series solution to the Legendre's differential equation [29], which is a differential equation of second order which is especially popular in physics. The Legendre polynomials are orthogonal with respect to a weight function that equals the probability density function of a random variable which is uniformly distributed on the interval $[-1, 1]$. The orthogonality property can thus be written as

$$\int_{-1}^{1} L_n(x) L_m(x) dx = \frac{2\delta_{mn}}{2n+1}, \tag{2.37}$$

where $L_i(x)$ denotes the Legendre polynomial of order i and δ_{mn} denotes the Kronecker delta.

In order to obtain the orthogonal baseband model related to the Legendre polynomials, a few modifications are required. Firstly, since the amplitude distribution has positive values only, the region of orthogonality must be shifted to the range between zero and the maximum amplitude. Secondly, the desired orthogonal baseband model should not include a constant term as it is not included in the conventional baseband model (2.7) either. The derivation of the set of basis functions can be established either by the Gram-Schmidt orthogonalization procedure [30] or by the inversion of a generalized Hilbert matrix [21]. In the latter case, a closed form representation for the orthogonal set can be found

$$\psi_{L,k}(z) = \sum_{l=1}^{k} (-1)^{l+k} \frac{(k+l)!}{(l-1)!\,(l+1)!\,(k-l)!} |z|^{l-1} z. \tag{2.38}$$

The orthogonality property for the Legendre related orthogonal basis set can be written as

$$\int_{0}^{1} \psi_{L,n}(z) \psi_{L,m}(z) dz = \frac{\delta_{mn}}{2n+1}. \tag{2.39}$$

In other words, the obtained polynomials are shifted Legendre polynomials without the constant term. The first few polynomials are listed in the Table 2.2.

Shifted Chebyshev polynomials of the second kind

The Chebyshev polynomials of the first and the second kind are frequently used in approximation theory and have a broad range of applications [31]. For example, in digital signal processing, Chebyshev polynomials are used in the context of digital filter design. An interesting and important application of Chebyshev polynomials is related to the polynomial interpolation. When a polynomial interpolant of high order is fitted to the data

$\psi_{L,1}(z) = z$
$\psi_{L,2}(z) = 4z\lvert z\rvert - 3z$
$\psi_{L,3}(z) = 15z\lvert z\rvert^2 - 20z\lvert z\rvert + 6z$
$\psi_{L,4}(z) = 56z\lvert z\rvert^3 - 105z\lvert z\rvert^2 + 60z\lvert z\rvert - 10z$
$\psi_{L,5}(z) = 210z\lvert z\rvert^4 - 504z\lvert z\rvert^3 + 420z\lvert z\rvert^2 - 140z\lvert z\rvert + 15z$

Table 2.2: First five shifted Legendre based polynomial functions

set, strong fluctuations can be observed at the borders of the interpolation interval. This is better known as Runge's phenomenon [32]. When the roots of the Chebyshev polynomials are applied as abscissas of the interpolation, the fluctuations are minimized thus also the Runge's phenomenon in mitigated.

This thesis uses the Chebyshev polynomials of the second kind. The weight function of the Chebyshev polynomials of the second kind is related to the probability density function of the Wigner semicircle distribution [29]. With proper scaling of the original probability density function, the weight function can be written as

$$f(x) = \sqrt{1 - x^2}. \tag{2.40}$$

The orthogonality property for the respective polynomials reads

$$\int_{-1}^{1} T_n(x)\, T_m(x)\, \sqrt{1 - x^2} dx = \frac{\pi}{2}\delta_{mn}, \tag{2.41}$$

where δ_{mn} denotes the Kronecker delta.

Similar to the Legendre polynomials, the same modifications are necessary when constructing a Chebyshev based orthogonal basis for the conventional baseband model. The orthogonal baseband model can be achieved by applying the Gram-Schmidt orthogonalization procedure to the conventional baseband model (see Appendix A for a more detailed derivation). As a result of shifting the range of orthogonality to $[0, 1]$, the obtained polynomials are called the shifted Chebyshev polynomials corresponding to the weight function [29]

$$f(z) = \sqrt{z - z^2}. \tag{2.42}$$

The orthogonality property for the orthogonal baseband model based on the shifted Chebyshev polynomials of the second kind can be written as

$$\int_{0}^{1} \psi_{T,n}(z)\, \psi_{T,m}(z)\, \sqrt{z - z^2} dz = \frac{\pi}{2}\delta_{mn}. \tag{2.43}$$

The first five polynomials are listed in the Table 2.3.

$\psi_{T,1}(z) = z$
$\psi_{T,2}(z) = \frac{1}{12}(40z\|z\| - 64z)$
$\psi_{T,3}(z) = \frac{1}{16}\left(112z\|z\|^2 - 448z\|z\| + 348z\right)$
$\psi_{T,4}(z) = \frac{1}{18}\left(240z\|z\|^3 - 1728z\|z\|^2 + 3456z\|z\| - 2048z\right)$
$\psi_{T,5}(z) = \frac{1}{22}\left(440z\|z\|^4 - 4928z\|z\|^3 + 16896z\|z\|^2 - 22528z\|z\| + 10240z\right)$

Table 2.3: First five polynomial functions based on the shifted Chebyshev polynomials of the second kind

2.2.6 System-level memoryless power amplifier models

The conventional polynomial baseband model and the orthogonal baseband model introduced in the Sections 2.2.1 and 2.2.5 are suitable in modelling a broad scale of PA with different nonlinear characteristics. However, when these models are used to approximate a specific PA, a high number of parameters is required.

For simulation purposes, it would be desirable to have PA models which are capable of modelling commercial PAs with good accuracy. With the system-level memoryless behavioral PA models it is possible to describe the AM/AM and AM/PM characteristics of certain type of amplifiers with a few amount of parameters. In literature, several system-level memoryless PA models are proposed [33, 34, 35] which are based on model which describe nonlinearities as a function of the input amplitude. When the digital baseband signal is represented in the complex envelope presentation as (see Section 2.1.2), the nonlinear effects in amplitude and phase can be described by AM/AM function $A[.]$, respectively AM/PM function $\Phi[.]$. In general, these models do not cover memory effects and can thus be regarded as quasi-memoryless.

The Saleh model is widely used as a reference model for standard PAs [33]. Originally the Saleh model was used to represent the behavior of travelling wave tube amplifiers (TWTA) in satellite communications. The TWTA exhibit nonlinearities over the whole range of input power levels. Therefore, the Saleh model servers as a good reference model in the evaluation of robustness with respect to model variations in the behaviors of the PA. The functions for AM/AM and AM/PM of the Saleh model are defined as

$$\begin{aligned} A[|V_{in}(t)|] &= \frac{\alpha_A |V_{in}(t)|}{1 + \beta_A |V_{in}(t)|^2}, \\ \Phi[|V_{in}(t)|] &= \frac{\alpha_\Phi |V_{in}(t)|^2}{1 + \beta_\Phi |V_{in}(t)|^2}, \end{aligned} \tag{2.44}$$

where the $\alpha_{A,}\beta_A, \alpha_\Phi, \beta_\Phi$ are the model parameters which specify the amplitude distortion and phase distortion.

A further PA model used to model solid state PAs is the so called Rapp model [35]. Compared to TWTAs, the solid state PAs show a higher linearity for lower input power levels. Furthermore, the AM/PM function is not considered in the Rapp model. The function for AM/AM of the model is given by

$$A\left[|V_{in}\left(t\right)|\right] = \frac{|V_{in}\left(t\right)|}{\left(1+\left(\frac{|V_{in}(t)|^{2p_A}}{V_{A,max}}\right)\right)^{0.5p_A}}, \tag{2.45}$$

where $V_{A,max}$ denotes the limiting output amplitude and p_A the parameter that controls the sharpness of the transition from the linear region to the saturation region.

A third common PA model, suitable to approximate the behaviour of solid state PAs is the so called Ghorbani model [36]. This model allows to incorporate distortions for low input power levels and it includes the expression of phase distortions. The AM/AM and AM/PM functions for Ghorbani model are given by

$$\begin{aligned} A\left[|V_{in}\left(t\right)|\right] &= \frac{\alpha_1 |V_{in}\left(t\right)|^{\alpha_2}}{1+\alpha_3 |V_{in}\left(t\right)|} + |V_{in}\left(t\right)|\,\alpha_4, \\ \Phi\left[|V_{in}\left(t\right)|\right] &= \frac{\beta_1 |V_{in}\left(t\right)|^{\beta_2}}{1+\beta_3 |V_{in}\left(t\right)|} + |V_{in}\left(t\right)|\,\beta_4, \end{aligned} \tag{2.46}$$

where the $\alpha_1, \alpha_2, \alpha_3, \alpha_{4,}\beta_1, \beta_2, \beta_3, \beta_4$ are the parameters determining the shape of the AM/AM and AM/PM the functions.

2.3 Nonlinear models with memory

When the bandwidth of the input signal grows, the PA exhibits memory effects which cannot be sufficiently expressed by the memoryless models. In the literature, the most frequently encountered models which incorporate the memory effects are the Volterra and Wiener models. These models originate from the work of two mathematicians, Vito Volterra and Norbert Wiener. To make them more suitable for implementation (e.g. of realtime applications), several specializations are extracted from these two general models.

2.3.1 Volterra and Wiener models

The Volterra series expansion has been used widely to model different nonlinear systems with memory. The general Volterra model can be expressed using Volterra functionals [37]

$$V_n\left[x\left(t\right)\right] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} h_n\left(\tau_1, \tau_2, \dots, \tau_n\right) x\left(t-\tau_1\right) x\left(t-\tau_2\right) \dots x\left(t-\tau_n\right) d\tau_1 d\tau_2 \dots d\tau_n, \tag{2.47}$$

where $h_n\left(\tau_1, \tau_2, \dots, \tau_j\right)$ represents the Volterra kernel of n^{th} order. The model itself can be written as a sum of these functionals

$$y\left(t\right) = V_0\left[x\left(t\right)\right] + V_1\left[x\left(t\right)\right] + \dots = \sum_{i=0}^{\infty} V_i\left[x\left(t\right)\right]. \tag{2.48}$$

Analogously, the discrete time version of Volterra functional reads

$$V_n\left[x\left(l\right)\right] = \sum_{m_1=0}^{\infty} \dots \sum_{m_n=0}^{\infty} h_n\left(m_1, m_2, \dots, m_n\right) x\left(l-m_1\right) x\left(l-m_2\right) \dots x\left(l-m_n\right), \tag{2.49}$$

where for simplicity reasons the same symbols V_n, h_n and x as in (2.47) have been used.

The computational complexity of Volterra series grows exponentially with the higher kernel orders. Therefore, simplifications need to be done to obtain feasible implementations. Since it is impossible to implement Volterra functionals (2.49) of infinite memory, the memory has to be truncated.

The amount of summation terms in (2.49) can be decreased by using the property of symmetric kernels, meaning that the kernel is invariant to permutations of indices $m_1, \dots, m_n$. Using this property, the truncated Volterra functional exploiting symmetric kernel property can be written as

$$V_n\left[x\left(l\right)\right] = \sum_{m_1=0}^{M-1} \sum_{m_2=m_1}^{M-1} \dots \sum_{m_n=m_{n-1}}^{M-1} h_n\left(m_1, m_2, \dots, m_n\right) x\left(l-m_1\right) x\left(l-m_2\right) \dots x\left(l-m_n\right), \tag{2.50}$$

where M denotes the memory length.
Another way to view the Volterra series is to consider it as a Taylor series with memory [38]. Consequently, some of the properties of the Taylor series apply also to the Volterra series. The major drawbacks of the Volterra series are its slow convergence in the context of adaptive kernel identification due to the high number of parameters and its incapability to model systems with discontinuities. Analogously as for the Taylor series, it also holds for the Volterra series that it can only model a reduced class of nonlinear systems with memory [38]. However, for PA modeling this is of less importance.

In 1958, the Wiener theory for nonlinear systems was introduced by Norbert Wiener [39]. There, he used the Gram-Schmidt orthogonalisation procedure to orthogonalise the Volterra functionals. The resulting ortonormal functionals are called the G-functionals. These functionals are orthogonal with respect to an input that is real valued and white Gaussian distributed. The advantage of the G-functionals is the completeness of the orthogonal set. This means that a continuous function can be fully expressed using the orthogonal basis expansion of G-functionals, similar to the case of orthogonal polynomials, cmp. (2.26). As a result of the completeness, the estimated system model using the orthogonal basis will always be optimum in the mean square sense. The general expression of G-functionals [40] is given by

$$\begin{aligned} G_n\left[h_n; x\left(t\right)\right] &= \sum_{m=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \frac{\left(-1\right)^m n! \left(\sigma_x^2\right)^m}{\left(n-2m\right)! m! 2^m} \times \\ &\quad \int_{-\infty}^{+\infty} \cdots \int h_n\left(\tau_1, \ldots \tau_{n-2m}, \gamma_1, \gamma_2, \ldots, \gamma_m, \gamma_m\right) \times \\ &\quad x\left(t-\tau_1\right) \ldots x\left(t-\tau_{n-2m}\right) d\tau_1 \ldots d\tau_{n-2m} d\gamma_1 \ldots d\gamma_m, \end{aligned} \tag{2.51}$$

where σ_x^2 denotes the variance of the real, white Gaussian distributed input signal $x\left(t\right)$.

Similar to the orthogonal polynomials, the orthogonality property of the G-functionals can be written as

$$E\left\{G_n\left[h_n; x\left(t\right)\right] G_k\left[h_k; x\left(t\right)\right]\right\} = \delta_{nk}, \tag{2.52}$$

where δ_{nk} denotes the Kronecker delta.

2.3.2 Simplified Wiener model

An alternative interpretation of the general Volterra series expansion respectively the G-functional expansion, is to consider the model of order p as a p-dimensional convolution [37]. The relationship with multidimensional linear systems allows to apply the same properties of multidimensional systems to nonlinear models with memory. The separability property, which applies to both, Volterra and Wiener kernels, can be used to simplify the multidimensional convolution. The n^{th} order kernel is said to be separable, if it can

be represented by a product of lower order kernels. A discrete Wiener kernel that is completely separable can be expressed as a product of first order kernels as follows.

$$h_n\left(m_1,\ldots,m_n\right)=\prod_{i=1}^{n}h_{1,i}\left(m_i\right). \tag{2.53}$$

Using this property discrete-time G-functionals with finite memory can be written as

$$G_n\left[h_1;x\left(l\right)\right] = \sum_{m=0}^{\left\lfloor\frac{n}{2}\right\rfloor}\frac{\left(-1\right)^m n!\left(\sigma_x^2\right)^m}{\left(n-2m\right)!m!2^m}\left[\sum_{m_1=0}^{M-1}h_1\left(m_1\right)x\left(l-m_1\right)\right]^m, \tag{2.54}$$

where σ_x^2 denotes the variance of the input signal and M the memory length. As it can be seen from Equation (2.54), the n^{th} order G-functional can be separated into a linear part with memory and a memoryless nonlinear part. Since the linear part is simply a convolution of the input samples with the first order kernel of length M, it can be regarded as an $M-1$ order finite impulse response (FIR) filter. The orthogonal nonlinear function in the discrete-time G-functionals is the result of the orthogonalization procedure originally applied to the Volterra kernels (see Section 2.3.1) and is equal to the generalized Wiener kernels (2.51). The same basis can be obtained for the completely separable kernels when the orthogonalization procedure is applied to a Taylor series under the assumption that the input is real valued and white Gaussian distributed with unit variance and zero mean. In this case, the resulting orthonormal polynomials are proportional to the classical Hermite polynomials [21]. The corresponding orthonormal polynomials are related to the Hermite polynomials in following way

$$W_n\left(x\right)=\frac{\left(-1\right)^n}{\sqrt{2^n n!}}H_n\left(\frac{x}{\sigma_x^2\sqrt{2}}\right), \tag{2.55}$$

and the orthonormality property of the obtained basis can be written as

$$\int_{-\infty}^{+\infty}W_n\left(x\right)W_m\left(x\right)\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{\sigma_x^2 2}}dw=\delta_{nm}. \tag{2.56}$$

The discrete-time G-functionals (2.54) are valid primarily for the real-valued Gaussian processes as derived originally by Wiener [39]. More generally, the setup including a linear sub-system with memory and a nonlinear memoryless sub-system, is also known as two-box model. If, as above mentioned, the linear sub-system with memory precedes the memoryless nonlinearity, the system is called *simplified Wiener model*[2] as illustrated in the figure 2.4.

[2]The simplified Wiener model shall not be mixed up with the (general) Wiener model in (2.51).

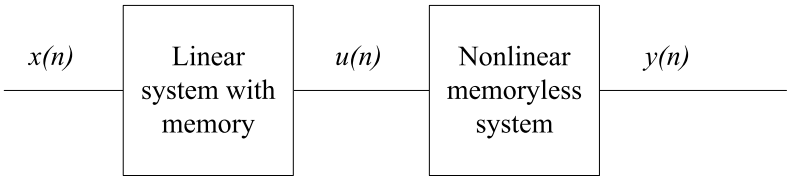

Figure 2.4: The simplified Wiener model

Commonly, the linear sub-system implemented as a FIR filter, as it occurs also in the completely separable Wiener kernels (2.54). Alternatively, the linear sub-system can be realized by an infinite impulse response (IIR) filter, which can be used to model longer impulse responses with less parameters. However, the popularity of the FIR filter approach can be explained by the better stability conditions [41]. Throughout this thesis, the linear sub system of the simplified Wiener model is represented by FIR filter.

The nonlinear sub-system in the simplified Wiener model as defined in (2.54) is based on a power series expansion. However, when modeling RF PAs, it is more desirable to consider modeling the baseband nonlinearities using the orthogonal baseband models (see Section 2.2.5). With an orthogonal basis substituted for W_n, the Wiener kernel can be written as

$$G_n\left[h_1; x\left(l\right)\right] = \psi_n\left[\sum_{m_1=0}^{M-1} h_1\left(m_1\right) x\left(l-m_1\right)\right], \tag{2.57}$$

where ψ_n denotes the orthogonal basis function for the respective baseband model, and M the memory length. The model output is then a linear combination of the basis functions,

$$y\left(l\right) = \sum_{i=0}^{P} c_i G_i\left[x\left(l\right)\right] = \sum_{i=0}^{P} c_i \psi_i\left[\sum_{m_1=0}^{M-1} h_1\left(m_1\right) x\left(l-m_1\right)\right], \tag{2.58}$$

where P denotes the number of the orthogonal basis functions in the orthogonal baseband model and M the memory length.

2.3.3 Hammerstein model

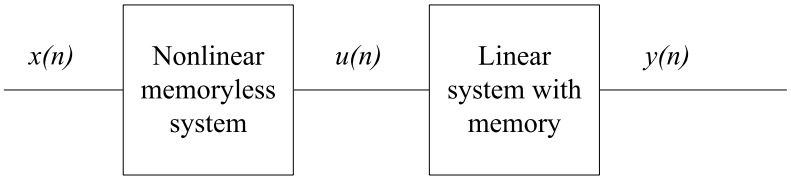

Figure 2.5: The Hammerstein model

An alternative two-box structure was introduced in [42]. Similar to the simplified Wiener

model, in this thesis the nonlinear part of the Hammerstein model is described using a basis of orthogonal polynomials.

The Hammerstein model can be considered as a special case of the general Wiener model. If the general Wiener kernel is assumed to be completely separable as formulated in Section 2.3.2, the relationship to the general Wiener kernels can be again visualized using the multidimensional convolution property. Hammerstein kernels are obtained when the separable kernel of order p satisfies the following condition [37]

$$h_n\left(q_1, q_2, \ldots, q_n\right) = \begin{cases} h_1\left(q_1\right) & q_1 = q_2 = \ldots = q_n, \\ 0 & \text{otherwise} \end{cases} \tag{2.59}$$

In other words, the only nonzero kernels are the ones with equal memory index. The equation of the Hammerstein model can then be formulated by

$$y\left(t\right) = \sum_{m_1=0}^{M-1} \left[h_1\left(m_1\right) \sum_{i=0}^{P} \psi_i\left[x\left(t - m_1\right)\right] \right], \tag{2.60}$$

where M denotes the length of the linear memory and P the number of orthogonal basis functions.

Chapter 3

Adaptive system identification

The behaviour of nonlinear systems such as PAs, can be assumed to be time-varying by nature. With respect to modeling, fluctuations in the characteristics of a nonlinear device complicates the estimation of the model parameters. In general, in such a case, it is necessary to perform the system identification adaptively. This chapter focuses on adaptive gradient search algorithms and structures for adaptive system identification.

Two-box models of Wiener type and Hammerstein type were introduced in Chapter 2. For such models, many adaptive identification techniques have been proposed in the literature [43, 44, 45]. The identification of Wiener models is often regarded as a difficult problem. Due to the inner structure of the Wiener model, a simple closed form estimator for the model parameters is not available. For algorithms to adaptively estimate the parameters of Wiener models are suggested in, e.g., [46, 45]. The difficulty with these estimation methods is that they require output measurements free of noise, which limits their practical applicability. In this thesis, a relatively simple least mean squares (LMS) based estimation method is used to identify the used two-box models.

The rest of this chapter is devoted to the identification of the inverse system. Based on examples of linear inverse estimation problems, a more elaborate inverse learning structure for nonlinear systems is introduced. The methods presented here can be seen as a basis for the adaptive digital predistortion introduced in Chapter 4.

3.1 Gradient search algorithms

3.1.1 Stochastic gradient algorithm

Stochastic gradient algorithm is a method for adaptive system identification. In this context it is employed to estimate a set of parameters to fit the behaviour of a model to the behaviour of the unknown reference system [37]. The principle of the method is to find a global minimum of a multivariable cost function utilizing the method of steepest decent. The global minimum is localized iteratively by moving in sufficiently small steps in the direction of the negative gradient. The cost function is a measure to quantify the quality

of an instantaneous approximation. It can be defined as a function of the form,

$$J(n) = E\{\mathcal{F}[e(n)]\}, \tag{3.1}$$

where $e(n)$ denotes the measured estimation error and $\mathcal{F}[\cdot]$ is an arbitrary, deterministic, even, and convex function.

Then, the update of the parameters which describe the estimated system model, can be formulated using the method of steepest decent

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \mu \nabla_{\mathbf{w}} J(n) \\ &= \mathbf{w}(n) - \mu \frac{\partial E\{\mathcal{F}[e(n)]\}}{\partial \mathbf{w}}, \end{aligned} \tag{3.2}$$

where $\mathbf{w}$ denotes the corresponding vector of parameter and μ denotes the positive update step size of the algorithm.

Adaptive signal processing applications are causal in general. Consequally, the estimated parameters at time instant n can only incorporate the data available from past observations. This means that even for an ergodic[1] error sequence $e(n)$, neither the expectation in (3.2), nor the cost function in (3.1), nor the gradient of it in (3.2) can be calculated exactly in reality. However, assuming ergodicity, the expression operation can be approximated by the mean over past values. Or much simpler but quite common just by the most recent sample $\mathcal{F}[e(n)]$. This keeps the computational requirements for the algorithm low. Thus, the parameter update procedure (3.2) can be approximated by

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{\partial \mathcal{F}[e(n)]}{\partial \mathbf{w}} \tag{3.3}$$

Although this estimate can substantially differ from the original gradient case in (3.2), for sufficiently small step sizes, the results obtained by (3.3) are reasonably accurate for many applications [37].

3.1.2 Least mean squares algorithm

In the early 1960ies, the LMS algorithm for adaptive linear filtering was introduced in [47]. Due to its simplicity and robustness, this algorithm gained popularity in a multitude of fields involving digital signal processing [48]. For the LMS algorithm, the cost function is defined as

$$J(n) = |e(n)|^2 = e(n)\, e^*(n), \tag{3.4}$$

[1] A random process is said to be ergodic if the time average over one realization is equal to the ensemble average.

where the estimation error $e(n)$ is defined as the difference between the desired output sequence and estimated output sequence

$$e(n) = d(n) - \hat{d}(n) = d(n) - \mathbf{w}^T(n)\,\mathbf{x}(n).$$

Substituting (3.4) into the stochastic gradient approximation procedure of one sample mean (3.3), leads to the update equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}} e(n)\, e^*(n),$$

where the gradient $\nabla_{\mathbf{w}} e(n)\, e^*(n)$ is defined as [22]

$$\nabla_{\mathbf{w}} e(n)\, e^*(n) = e(n)\, \nabla_{\mathbf{w}} e^*(n) = e(n)\, \frac{\partial \left(d(n) - \mathbf{w}(n)^T \mathbf{x}(n) \right)^*}{\partial \mathbf{w}^*}$$

and

$$\frac{\partial \left(d(n) - \mathbf{w}(n)^T \mathbf{x}(n) \right)^*}{\partial \mathbf{w}^*} = -\mathbf{x}^*(n). \tag{3.5}$$

This leads to, the LMS update equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\, \mathbf{x}^*(n). \tag{3.6}$$

Apparently, for the complex valued case, each adaptation step with the coefficient update vector $\mathbf{w}(n)$ of length p, requires $8p$ real additions and $8p+2$ real multiplications [49]. Therefore, due to its low complexity, the LMS algorithm is computationally very efficient and can be implemented with very little use of resources.

Whether the LMS converges to the global minimum depends on step size variable μ. In general, the optimum step size can hardly be derived analytically. Nevertheless, some guidelines for convergence conditions are available. If the input sequence $\mathbf{x}(n)$ is assumed to be wide-sense stationary (WSS) and if the corresponding input autocorrelation matrix $\mathbf{R}_{\mathbf{xx}} = E\left[\mathbf{x}(n)\,\mathbf{x}(n)^H\right]$ is known, the LMS algorithm is ensured to converge in the mean-square sense if the step size satisfies [37]

$$0 < \mu < \frac{2}{\eta_{\max}}, \tag{3.7}$$

where the $\eta_{\max}$ denotes the maximum eigenvalue of the autocorrelation matrix $\mathbf{R}_{\mathbf{xx}}$.

However, in most cases, however, $\eta_{\max}$ is not known a priori and its estimation is too complex. Therefore, according to [37], a looser but simpler and more practical convergence condition is found to be

$$0 < \mu < \frac{2}{\mathrm{Tr}\left\{\mathbf{R}_{\mathbf{xx}}\right\}}, \tag{3.8}$$

where $\mathrm{Tr}\left\{\cdot\right\}$ denotes the trace of a matrix.

3.1.3 Normalized least mean squares algorithm

The estimation of an adequate step size is not a trivial task. Especially, when the input sequence does not fulfill the criteria of a WSS process. Furthermore, the continuous estimation of the autocorrelation matrix would be too complex and impractical task.

One way to mitigate the estimation problem, is to use the following identity which is valid for wide sense stationary processes [48]

$$\mathrm{Tr}\left\{\mathbf{R}_{\mathbf{xx}}\right\} = E\left[\left\|\mathbf{x}\left(n\right)\right\|_2^2\right] \tag{3.9}$$

The variance $E\left[\left\|\mathbf{x}\left(n\right)\right\|_2^2\right]$ can be approximated with one sample mean, i.e. $E\left[\left\|\mathbf{x}\left(n\right)\right\|_2^2\right] \approx \mathbf{x}\left(n\right)^H \mathbf{x}\left(n\right)$. Then, by substituting this result into the convergence condition equation for the LMS in (3.8), the following new bounds for mean-square convergence is obtained

$$0 < \mu < \frac{2}{\mathbf{x}\left(n\right)^H \mathbf{x}\left(n\right)} \tag{3.10}$$

Now, the upper bound of the convergence condition is dependent on the instantaneous input values of the input sequence. The step size variable can be reformulated as follows

$$\mu = \frac{\alpha}{\mathbf{x}\left(n\right)^H \mathbf{x}\left(n\right)}, \tag{3.11}$$

Substituting (3.11) in (3.10) results in following convergence bounds

$$0 < \alpha < 2. \tag{3.12}$$

If the time variant step size in (3.11) is used in the original LMS update equation in (3.6), the resulting algorithm is the so called normalized least mean squares (NLMS) algorithm [48]. As the name indicates, the functionality of the NLMS algorithm is based on the input signal, whereas the direction of the gradient remains unchanged. Compared to the LMS, the NLMS shows a slightly higher computational complexity. However, due to the normalization the stability becomes independent from the power of the input signal and moreover the convergence speed is increased [49].

3.2 Adaptive system identification using the LMS algorithm

3.2.1 Adaptive identification of a two-box model

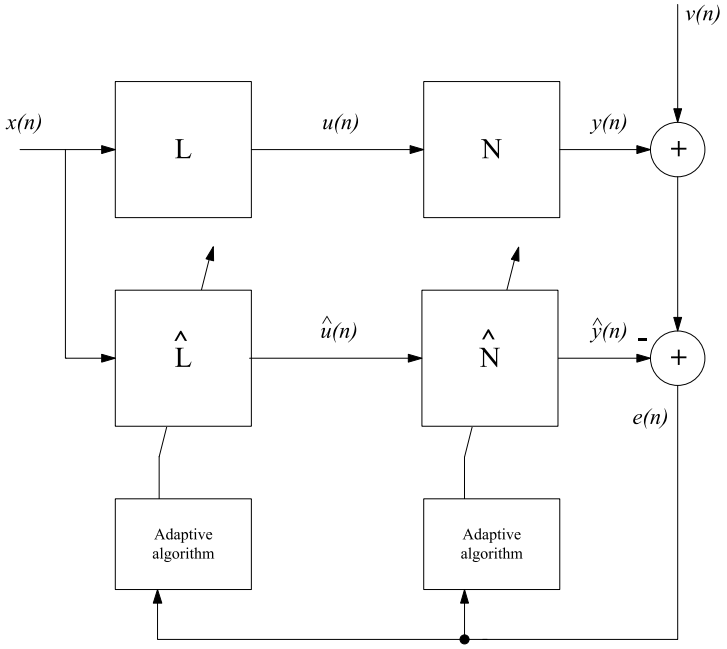

Figure 3.1: Configuration for the adaptive identification of a two-box model

Adaptive system identification based on LMS applied to the general as well as the simplified Wiener model was investigated in [50, 51].

One way to adaptively identify a Wiener model is illustrated in the Figure 3.1. This scheme is an example of a black box modeling approach, where the unknown reference system is assumed to be a Wiener model. The estimation update is derived from the input excitation samples $x(n)$ and the difference between the output samples $y(n)$ of the reference system and the output samples $\hat{y}(n)$ of the estimated system. The error sequence $e(n) = y(n) - \hat{y}(n) + v(n)$ is fed to the adaptive algorithms, which update the parameter vectors $\mathbf{h}(n)$ and $\mathbf{b}(n)$ of the linear filter $\hat{L}$ and the nonlinearity $\hat{N}$, respectively. The same learning architecture can be used for Hammerstein models.

With the following vector notations

$$\begin{aligned} \mathbf{x}(n) &= [x(n), x(n-1), \ldots, x(n-M-1)]^T, \\ \hat{\mathbf{u}}(n) &= [\psi_1(\hat{u}(n)), \psi_2(\hat{u}(n)), \ldots, \psi_K(\hat{u}(n))]^T, \end{aligned} \tag{3.13}$$

and the vectors of parameters

$$\begin{aligned}\mathbf{h}(n) &= [h_1(n), h_2(n), \ldots, h_M(n)]^T, \\ \mathbf{b}(n) &= [b_1(n), b_2(n), \ldots, b_K(n)]^T,\end{aligned}$$

where M denotes the number of taps in the LTI system, ψ_n the orthogonal basis functions (cmp. Section 2.2.5) and K the number of the basis functions, the outputs of the estimated linear filter respectively the estimated nonlinearity can be written as

$$\begin{aligned}\hat{u}(n) &= \mathbf{x}(n)^T \mathbf{h}(n), \\ \hat{y}(n) &= \mathbf{u}(n)^T \mathbf{b}(n).\end{aligned} \tag{3.14}$$

Then, analogously to the LMS algorithm in (3.6) the update equations for the parameter vectors of the linear filter and the static nonlinearity become

$$\begin{aligned}\mathbf{h}(n+1) &= \mathbf{h}(n) + \mu_l e(n) \mathbf{x}^*(n), \\ \mathbf{b}(n+1) &= \mathbf{b}(n) + \mu_n e(n) \hat{\mathbf{u}}^*(n),\end{aligned} \tag{3.15}$$

where μ_l denotes the step size for the adaptation of the LTI system and μ_n the step size for the adaptation of the nonlinear system.

3.3 Inverse system identification

Adaptive inverse control techniques have been used successfully, for example, in the context of channel equalization and echo cancellation [52, 53]. These applications utilize adaptive filtering methods to identify unknown systems which show linear behaviour. For digital predistortion, consequally, a nonlinear inverse control is required, which performs the identification of nonlinear systems. In general, the properties of linear adaptive filtering methods cannot be directly applied to nonlinear inverse control schemes. However, similar structures for the identification of the inverse systems can be utilized. In this section, the inverse control techniques for both linear, as well as nonlinear systems, are considered. At first, the invertibility of the linear reference system is discussed and two linear inverse control structures are presented. Secondly, the invertibility of the simplified Wiener model is discussed and a nonlinear inverse control structure related to linear control structure is presented.

3.3.1 Adaptive linear inverse control

The objective of linear adaptive inverse control is to find the inverse system model of a linear reference system. In general, the functionality of the reference system is assumed

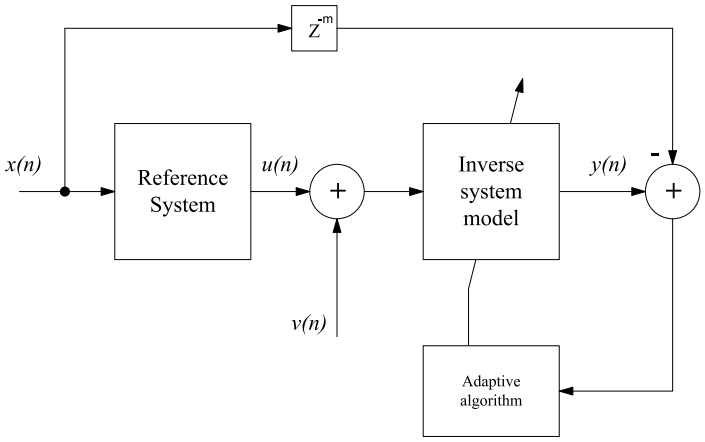

Figure 3.2: Adaptive inverse modeling

to be completely unknown. In practice, however, some basic knowledge of the behaviour of the reference system is required. For example, the information about its response time and about possible variations in the amplitude response of the system can be utilized to optimize the performance of the inverse system model [54]. Since the reference systems behaviour can be assumed linear, the inverse system can be modelled using a LTI system. In the ideal case, the cascade of the inverse system model and the reference system should behave like a simple delay.

A basic configuration for inverse system modeling, which is introduced in [52], is illustrated in Figure 3.2. Here, the input signal $x\left(n\right)$ is fed through an unknown reference system and the corresponding output signal $u\left(n\right)$ is observed. Additionally, $u\left(n\right)$ is assumed to be corrupted by additive noise $v\left(n\right)$. For causal systems with memory, the reference system as well as the estimated inverse will introduce a delay. Hence, after adaptation, the response of the inverse system $y\left(n\right)$ will be a delayed approximation of the input $x\left(n\right)$. The adaptation of the inverse model can be established, for example, using LMS algorithm. Then, the difference between the delayed input and the inverted output is minimized in the mean square sense.

Despite the simplicity of this structure, the existence of an exact system inverse is not always guaranteed. In practice, its accuracy is impaired by several factors, which may include phase errors, and noise, etc. Most of them affect to the quality of the inversion. In the worst, case if the harmful effects are unknown or neglected in the identification procedure, the final estimate will be incorrect.

For example, the presence of the measurement noise $v\left(n\right)$ which is assumed to be uncorrelated with the input signal $x\left(n\right)$, will in general cause the weights of the inverse system to be biased as shown in [55]. Further, stability problems may occur during adaptation, if the reference system does not fulfill the minimum phase property. In the case of an inverse system modelled by an infinite impulse response (IIR) filter, the poles of the filter would be located outside the unit circle, meaning that the filter would become unstable. To reduce the risk of running into an unstable inverse system model, its output can be kept aligned with the input of the reference system $x\left(n\right)$ by a fixed delay, as depicted in the

figure 3.2. Utilizing the delayed inverse modeling, the least mean-squares estimate will be closer to the reciprocal of the reference systems response even if the systems phase response is not minimum-phase [52].

3.3.2 Filtered-x LMS

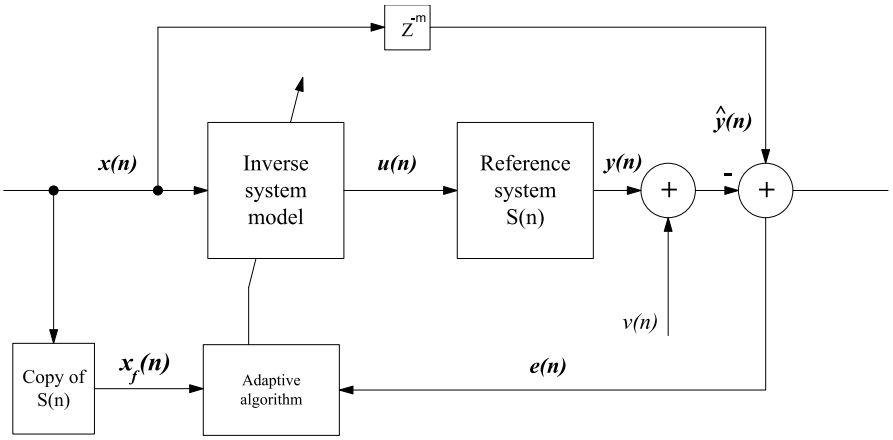

Figure 3.3: Adaptive inverse control utilizing the filtered-x LMS algorithm

In Section 3.3.1, the basic linear inverse control modeling scheme was covered. It was pointed out that the accuracy of the estimated inverse models is deteriorated by a reference system which no minimum-phase system, as well as, by noise. While the former can be alleviated by introducing a delay of the input signal, the latter remains.

An adaptive inverse control scheme which is insensitive to measurement noise and in which the reference system shows a non-minimum phase characteristic was presented in [55]. There, the motivation for improving the identification scheme was to avoid the measurement noise at the input of the inverse system model. The corresponding structure is depicted in Figure 3.3. The essential difference compared to the structure in Section 3.3.1 is the position of the inverse system model. Even though the measurement noise $v(n)$ still disturbs the output of the reference system, it does not directly affect the inverse model. In fact, as shown in [55], the converged solution of the inverse system model is unbiased with respect to the noise $v\,(n)$.

From Figure 3.3 it follows that the error signal $e\,(n) = y\,(n) - \hat{y}\,(n)$ depends on the behaviour and especially the phase response of the reference system, via its output signal $y\,(n)$. If the conventional LMS algorithm would be used for the adaptation, the update procedure for the inverse filter coefficients $\mathbf{w}\,(n)$ would be

$$\mathbf{w}\,(n+1) = \mathbf{w}\,(n) - \mu e\,(n)\,\mathbf{x}^*\,(n)\,. \tag{3.16}$$

Since, the expected phase and amplitude response of the reference system are not present in the incorporated regression vector $\mathbf{x}^*\,(n)$, the LMS algorithms gradient estimate $e\,(n)\,\mathbf{x}^*\,(n)$

is biased and the adaptation is likely to become unstable [56, 52]. In order to obtain stability, the influence of the reference system must be taken into account. If the reference system is a FIR filter, a more precise estimate for the inverse system model can be obtained by the filtered-x LMS algorithm, which moreover circumvents the above mentioned stability problems which would occur for the LMS.

The quadratic cost function which should be minimized is given by the

$$F(n) = |e(n)|^2 \tag{3.17}$$

Substituting (3.17) into the stochastic gradient approximation procedure (3.3), following update equation is obtained

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}^*} |e(n)|^2 . \tag{3.18}$$

Now the gradient $\nabla_{\mathbf{w}^*} |e(n)|^2$ can be evaluated as follows

$$\nabla_{\mathbf{w}^*} |e(n)|^2 = e(n) \nabla_{\mathbf{w}^*} e^*(n) = \frac{\partial e^*(n)}{\partial \mathbf{w}^*(n)}, \tag{3.19}$$

which leads to

$$\frac{\partial e^*(n)}{\partial \mathbf{w}^*(n)} = \frac{\partial \left(y(n) - y\hat{(n)} \right)^*}{\partial \mathbf{w}^*(n)} = \frac{\partial y^*(n)}{\partial \mathbf{w}^*(n)}.$$

Since

$$y(n) = \sum_{j=1}^{M} s_j \left[\sum_{k=1}^{I} w_k x(n-i-k) \right], \tag{3.20}$$

it follows that

$$\frac{\partial y^*(n)}{\partial w_i^*(n)} = \sum_{j=1}^{M} s_j^* x(n-j-i), \tag{3.21}$$

and so

$$\frac{\partial y^*(n)}{\partial \mathbf{w}^*(n)} = \sum_{j=1}^{M} s_j^* \mathbf{x}(n-j) = \mathbf{x}_f^*(n), \tag{3.22}$$

where s_j is the j^{th} filter coefficient of the reference system model, the M the number of filter coefficients in the reference system model and I the number of filter coefficients in the inverse system model. Finally, the coefficient update procedure reads

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\,\mathbf{x}_f^*(n). \tag{3.23}$$

The vector of input samples is now filtered by the reference system model. Hence the algorithm is known as filtered-x least-mean-squares (FxLMS). In comparison to the conventional LMS, the complexity of the FxLMS is increased slightly due to the filtering operation. If the inverse system model is described by a vector $\mathbf{w}(n)$ of length I, and the reference system by a vector $\mathbf{s}$ of length M, $4MI$ real multiplications and $(4M-2)I$ real additions are required.

The functionality of the FxLMS algorithm relies on the fact that the copy of the reference system and the reference system itself are commutable. If the reference system model is linear and time-invariant the commutability is guaranteed. For a nonlinear and time-variant system, the commutability property is questionable. However, when the adaptation is set to be slow enough, i.e., for small step sizes, the characteristics of the inverse system model is approximately time-invariant and the commutation of the reference system model is permitted as stated in [52]. On the other hand, convergence studies have shown that the algorithm works even with larger step sizes [52]. Another issue which concerns the functionality of the algorithm is the quality of the copy of the reference model. Additionally, the identification of the reference model increases slightly the complexity of the algorithm. It turns out that the copy of the reference model does not need to model the behaviour of the actual reference system exactly [57, 52]. Actually, the algorithm will converge even when the magnitude of the phase errors between the identified reference system model and the original reference system stays below 90 degrees [58]. Hence, the algorithm is shown to be applicable to dynamic inverse control problems.

3.3.3 Invertibility of Wiener models

Suppose the reference system is modelled by a Wiener model (see Section 2.3.2) consisting of a FIR filter followed by a static nonlinear system using orthogonal polynomials. To find the inversion of Wiener model, both of its sub-systems must be invertible. Thus, the invertibility conditions for the two sub-systems can be treated separately. In general, the inverse model to the Wiener model is represented by a Hammerstein model.

Input/output relation of a LTI system can be formulated as a convolution or in vector form equivalently:

$$y(n) \;=\; a_1 x(n) + a_2 x(n-1) + \ldots + a_N x(n-N-1) = \mathbf{x}(n)^T \mathbf{a}, \tag{3.24}$$

where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N-1)]^T$ and $\mathbf{a} = [a_1, a_2, \ldots, a_N]^T$.

Consequently, the corresponding impulse response can be written in the Z-domain

$$H(z) = \frac{Y(z)}{X(z)} = a_1 + a_2 z^{-1} + \ldots + a_N z^{-N} \tag{3.25}$$

The exact inverse function $I(z)$, such that $I(z)Y(z) = 1$ would simply be an inverse function of $H(z)$, which is

$$I(z) = \frac{1}{a_1 + a_2 z^{-1} + \ldots + a_N z^{-N}}. \tag{3.26}$$

A minimum-phase system is causal and stable and its inverse is causal and stable as well [48, 52]. However, the existence of a stable inverse for non-minimum-phase system is not guaranteed. When the LTI system is described by a FIR filter, the ideal inverse is all-pole filter and the same stability conditions as discussed in Section 3.3.1 apply. However, the ideal inverse filter can be approximated by a FIR filter.

The static nonlinearity of the Wiener model is a function $\mathcal{F} : \mathbb{C}^m \rightarrow \mathbb{C}^m$. Therefore, its inverse is similarily a function $\mathcal{F}^{-1} : \mathbb{C}^m \rightarrow \mathbb{C}^m$. For the nonlinear transfer function to be exactly invertible, it needs to be bijective. In other words, each input value maps exactly to one unique output value. Again, this condition is not always fulfilled in reality. The AM/AM and AM/PM characteristics of PAs, for example, can be classified as almost strictly monotonic functions. When the PA is driven beyond its saturation level, clipping occurs and the inverse nonlinearity cannot be fully modelled by a single bijective function. In this case, an approximate inversion can be found on the range $[a, b]$ where the AM/AM transfer function is monotonic.

3.3.4 Nonlinear Filtered-x LMS

The adaptive algorithms used for inverse system modeling are usually restricting to systems represented by linear filters. In this context, besides the indirect learning scheme (see Section 3.3.1), the most commonly used algorithm is the filtered-x LMS. In case of linear inverse system identification, the filtered-x LMS algorithm can be used, since it is robust against the noisy output measurements and the phase distortions introduced by the reference system. In contrast, for the identification of nonlinear systems, the additionally introduced nonlinear distortions may cause the filtered-x LMS to become unstable. However, the algorithm can be modified such that it becomes insensitive to those nonlinear distortions. This modified method is called the nonlinear filtered-x least mean squares (NFxLMS) algorithm. Its concept was first introduced in [44, 59]. Later, a more generalized version of the algorithm was presented under the same name in the context of predistortion utilizing Volterra kernels [60]. Recently the NFxLMS algorithm has been successfully used for linearization of PA with several different predistorters [61, 62] based on polynomials. In general the NFxLMS can be applied to the predistorter models, where the system and predistorter models are presented by arbitrary basis functions.

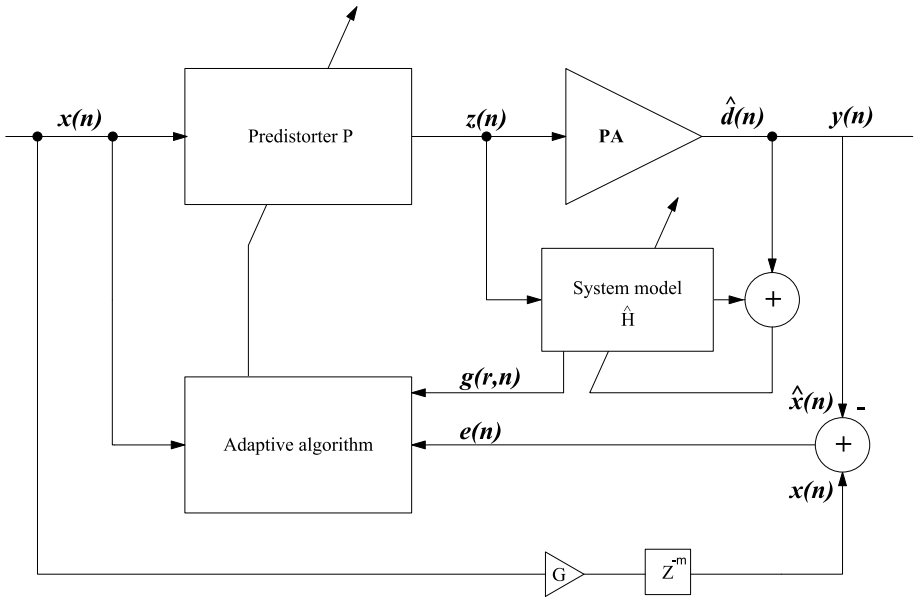

Figure 3.4: Adaptive predistorter using the nonlinear filtered-x LMS algorithm

The structure of the NFxLMS algorithm is illustrated in the Figure 3.4. It is closely related to the structure of the filtered-X LMS (see Figure 3.3). The update equation for NFxLMS based on the stochastic gradient method can be derived in a similar way as it was done for the FxLMS in Section 3.3.2. Let $\mathcal{P}$ denote the nonlinear predistorter with the number of basis functions R and the memory size M and let further denote $\hat{\mathcal{H}}$ the system model with the number of basis functions S and the memory size N. The predistorter model and the system model are represented using arbitrary basis functions functions $p_n\left[x\left(n\right)\right]$and $h_n\left[z\left(n\right)\right]$. For the derivation of the update equations , the following vectors of basis functions are introduced

$$\mathbf{x}\left(n\right)=\left[p_1\left[x\left(n\right)\right],p_2\left[x\left(n\right)\right],\ldots,p_R\left[x\left(n\right)\right]\right]^T,$$

$$\mathbf{y}\left(n\right)=\left[h_1\left[z\left(n\right)\right],h_2\left[z\left(n\right)\right],\ldots,h_S\left[z\left(n\right)\right]\right]^T.$$

Corresponding vectors of expansion coefficients are defined as

$$\mathbf{a}\left(n\right)=\left[a_{1,}a_2,\ldots,a_R\right]^T,$$

$$\mathbf{b}\left(n\right)=\left[b_{1,}b_2,\ldots,b_S\right]^T.$$

Based on these definitions, the input/output relations for the output $z\left(n\right)$ of the predistorter output respectively the output $\hat{d}\left(n\right)$ of the system model become

$$z\left(n\right)=\mathbf{a}^T\left(n\right)\mathbf{x}\left(n\right)=\sum_{i=1}^{R}a_ip_i\left[x\left(n\right)\right],\tag{3.27}$$

$$\hat{d}(n) = \mathbf{b}^T(n)\,\mathbf{y}(n) = \sum_{i=1}^{S} b_i h_i\left[z(n)\right]. \tag{3.28}$$

Under these assumptions that the optimal vector $\mathbf{b}(n)$ describing the system model is given, the objective to find the vector $\mathbf{a}_{opt}$ which minimizes mean square of the error $e(n) = d(n) - \hat{d}(n)$, where $d(n)$ denotes the desired output signal. In this case, $d(n)$ is a delayed and amplified version of the input signal, i.e., $d(n) = Gx(n-k)$, where k denotes the delay between the input of the predistorter and the output of the power amplifier and G is the desired linear gain of the PA including predistortion. Hence, the vector $\mathbf{a}_{opt}$ of coefficients for the optimum predistorter is defined by

$$\mathbf{a}_{opt} = \arg\min_{\mathbf{a}} = E\left[|e(n)|^2\right] \tag{3.29}$$

Since in practice, the expectation cannot be evaluated exactly by a finite set of samples, as in the derivation of the LMS in Section 3.1.2, the rough approximation by one sample is used instead. Consequently, the cost function in (3.29) is approximated by

$$J(n) = |e(n)|^2. \tag{3.30}$$

An adaptive algorithm minimizing the cost function in (3.30) can be derived with respect to $\mathbf{a}$ using the stochastic gradient method [62], i.e.,

$$\begin{aligned}
\mathbf{a}(n+1) &= \mathbf{a}(n) - \frac{1}{2}\mu_p \nabla_{\mathbf{a}}(n) \\
&= \mathbf{a}(n) - \mu_p \frac{\partial MSE(\mathbf{a})}{\partial \mathbf{a}} \\
&= \mathbf{a}(n) + \mu_p e^*(n)\,\frac{\partial \hat{d}(n)}{\partial \mathbf{a}(n)}
\end{aligned} \tag{3.31}$$

where μ_p is the step size for the update of the predistorter coefficients.

The derivation of $\frac{\partial \hat{d}(n)}{\partial \mathbf{a}}$ can be carried out in two parts

$$\frac{\partial \hat{d}(n)}{\partial \mathbf{a}(n)} = \sum_{r=0}^{N-1} \frac{\partial \hat{d}(n)}{\partial z(n-r)} \frac{\partial z(n-r)}{\partial \mathbf{a}(n)}. \tag{3.32}$$

Since the delayed version of the predistorters output is $z(n-r) = \mathbf{a}^T(n-r)\,\mathbf{x}(n-r)$, the derivative $\frac{\partial z(n-r)}{\partial \mathbf{a}(n)}$ becomes

$$\frac{\partial z(n-r)}{\partial \mathbf{a}(n)} = \frac{\partial \mathbf{a}^T(n-r)\,\mathbf{x}(n-r)}{\partial \mathbf{a}(n)}. \tag{3.33}$$

If the step size μ_p is sufficiently small, the update of each expansion coefficient will be small as well. Hence, it is possible to use the approximation $\mathbf{a}(n-r) \approx \mathbf{a}(n)$, as stated

in [62]. This simplifies the derivative in (3.33) and it reduces to

$$\frac{\partial z\left(n-r\right)}{\partial \mathbf{a}\left(n\right)} \approx \mathbf{x}\left(n-r\right). \tag{3.34}$$

The second part of the derivative, i.e., $\frac{\partial \hat{d}(n)}{\partial z(n-r)}$, can be evaluated using the corresponding input-output relationship (3.28),

$$\begin{aligned} \frac{\partial \hat{d}\left(n\right)}{\partial z\left(n-r\right)} &= \frac{\partial \sum_{i=1}^{R} b_i h_i\left[z\left(n\right)\right]}{\partial z\left(n-r\right)}, \\ &= \sum_{i=1}^{R} b_i \frac{\partial h_i\left[z\left(n\right)\right]}{\partial z\left(n-r\right)}. \end{aligned} \tag{3.35}$$

For convenience, this derivative can be abbreviated by a two-parameter function $g\left(r,n\right) = \frac{\partial \hat{d}(n)}{\partial z(n-r)}$, where the parameters n and r refer to the time instant n and the relative delay r, respectively. Substituting the evaluated derivatives in the update equation (3.31) yields the stochastic gradient method,

$$\mathbf{a}\left(n+1\right) = \mathbf{a}\left(n\right) + \mu_p e^*\left(n\right) \sum_{r=0}^{N-1} g\left(r,n\right) \mathbf{x}\left(n-r\right), \tag{3.36}$$

which obviously relies on a regression vector which is obtained by filtering the input signal with a time-varying linear filter. For the algorithm to be feasible, the basis functions for both, the system estimate as well as the predistorter, must be differentiable. The step size μ_p can either be a scalar, meaning that each coefficient update uses the same step-size, or a diagonal matrix with N entries on its diagonal, meaning that each coefficient is updated with a separate step size. An analytical derivation of stability bounds for the step size is not available. However, at least, it is possible to obtain a rough estimate as presented in [62].

Chapter 4

Adaptive digital predistortion

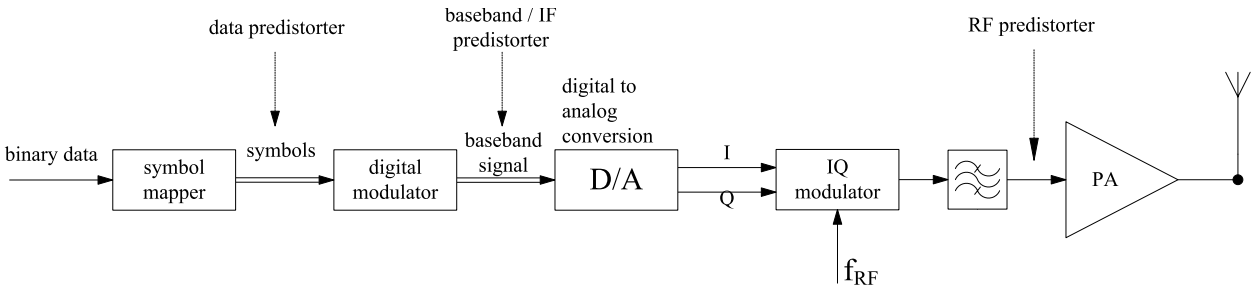

Figure 4.1: Locations of different predistorters in the RF transmitter

Adaptive digital predistortion is one of the most effective ways to linearize PAs with time-variant behaviour. The objective of the linearization procedure is to find a proper model which behaves inverse to the impairments introduced by the PA. Earlier, predistortion was often restricted to compensate for the static nonlinearities of the PA only, memory effects were ignored. However, the high signal bandwidth of modulation schemes require the predistorters also to incorporate the memory effects of the PAs. In modern predistorters, the challenge is to model the reciprocal function to the memory effects in conjunction with the inverse of the static nonlinearity.

Predistorters can be divided into three different subclasses depending on their location in the signal path before the PA [9, 63]

- Baseband predistorter
  - Signal predistorter
  - Data predistorter
- Intermediate frequency (IF) predistorter
- RF predistorter

The locations of these predistorters in the transmitter is depicted are the Figure 4.1. The symbol predistorter[1] is located before the digital modulation and influences the data symbols. One of the disadvantages using the symbol predistortion is the lacking capability to compensate the spectral regrowth outside the signal band. In some cases the baseband signal is digitally modulated to a digital IF before it is converted into an analog signal. In this case, the predistorter can process the IF signal instead of the baseband signal.The baseband predistorter, processes the digital baseband signal, which in general has a higher rate than the symbols. The baseband predistorter as well as the IF predistorter are capable of reducing spectral regrowth. In contrast to the RF predistorter, this advantage is their independence of the actually used RF devices. In this thesis, all the predistorters operate in the digital baseband. The RF predistorter, which directly predistorts the up-converted RF signals, is located immediately before the PA. Due to the high frequencies, the RF amplifiers are implemented in the analog domain. In contrast to the RF predistorters, their advantage is their independence of the actually used RF devices.

In this Chapter, several adaptive digital baseband predistorter architectures with memory are presented. At first, predistorter model and PA model structures are introduced in Section 4.1. The indirect learning architecture, presented in Section 4.2, is a common structure used in predistorter identification. Section 4.3 presents the direct learning architecture, which is an alternative method for the training of predistorter. Finally, in Section 4.4 a direct learning architecture based on the with NFxLMS algorithm is introduced.

4.1 Predistorter model structure

The predistortion models presented in this chapter are based on the Hammerstein structure introduced in Section 2.3.3. Corresponding to the PAs which are modelled by a simplified Wiener model, presented in Section 2.3.2.

For here considered Hammerstein predistorter model and the simplified Wiener PA model, the nonlinear part is represented using the orthogonal basis expansion which in this work is referred to by orthogonal baseband model (see Section 2.2.4). The linear part is represented by a FIR filter. With these models for the PA and the predistorter, the nonlinear behaviour as well as the memory effects of the PA can be modelled respectively compensated.

For the nonlinear subsystems, the coefficients of the orthogonal basis expansion can be written in the vector form as introduced in the Section 2.2.4. Let $\mathbf{c}_s$ and $\mathbf{c}_p$ coefficient vectors related to the basis expansions for the simplified Wiener models respectively the Hammerstein models. These vectors are then defined as follows,

$$\begin{aligned} \mathbf{c}_s(n) &= \left[c_{s,1}(n), c_{s,2}(n), \cdots, c_{s,S_p}(n)\right]^T, \\ \mathbf{c}_p(n) &= \left[c_{p,1}(n), c_{p,2}(n), \cdots, c_{p,P_p}(n)\right]^T, \end{aligned} \tag{4.1}$$

where S_p and P_p denote the number of basis functions used for the simplified Wiener models respectively the Hammerstein models.

[1] Also known as data predistorter

For the linear subsystems, the parameters of the FIR filters can be represented by a column vector. Let $\mathbf{l}_s(n)$ denote filter weights of the simplified Wiener models, and let $\mathbf{l}_p(n)$ denote the filter weights of the Hammerstein models . The corresponding vectors can be written as follows,

$$\begin{aligned} \mathbf{l}_s(n) &= \left[\, l_{s,1}(n),\ \ l_{s,2}(n),\ \ \cdots,\ \ l_{s,S_l}(n) \,\right]^T \\ \mathbf{l}_p(n) &= \left[\, l_{p,1}(n),\ \ l_{p,2}(n),\ \ \cdots,\ \ l_{p,P_l}(n) \,\right]^T \end{aligned} \tag{4.2}$$

where S_l denotes the filter length of the simplified Wiener models and P_l the filter length of the Hammerstein models.

4.2 Indirect learning architecture

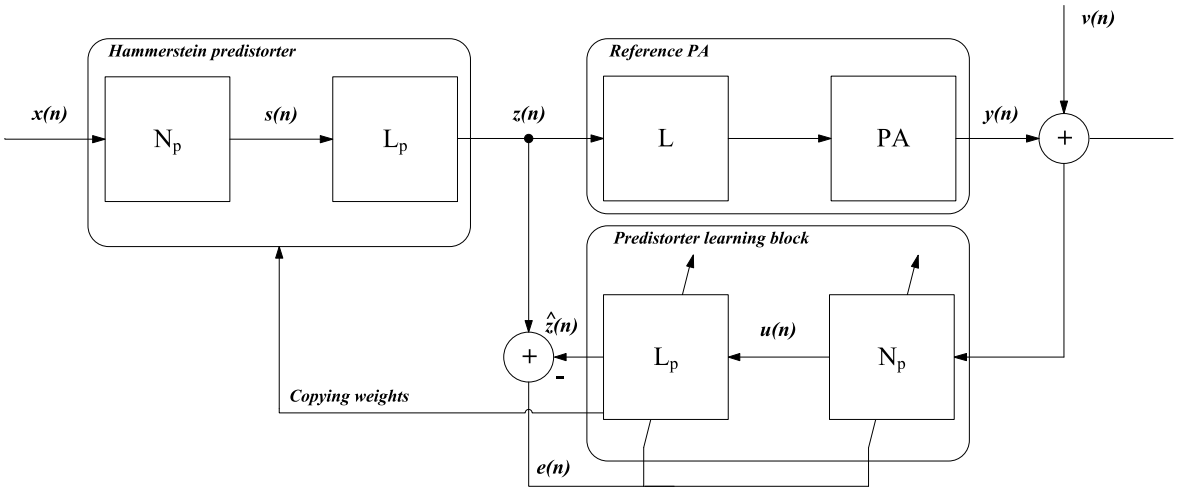

Figure 4.2: A Hammerstein predistorter using the indirect learning architecture to linearize a PA represented by a simplified Wiener model.

The indirect learning architecture (IDLA) is a well-known concept among the predistortion implementations and has been studied intensively in the literature. In [64], the indirect learning architecture is used successfully with Volterra kernels for the linearization of TWTA in the context of satellite communication. Predistorter doing the IDLA based on a Hammerstein model is presented in [43] to linearize PAs which exhibit memory effects. More recently, comparisons have been done between direct and indirect learning architectures. The results indicate that for the predistortion of PAs, the IDLA is a competitive solution. [65, 66, 61].

One of the benefits of IDLA is its simplicity. The predistorter model is derived directly based on the sequences at the input and respectively output of the PA. Thus the overall complexity of the predistorter can be kept relatively low. In contrast, for other algorithms, a system identification needs to be performed before the predistorter can be calculated as an inverse of the system model.

The indirect learning architecture illustrated in Figure 4.2 consists of two separate yet identical Hammerstein models, which represent the predistorter learning block respectively the predistorter block. The functionality of the architecture is based on the basic linear inverse control scheme introduced in Section 3.3.1, i.e., the inverse system model is located in the feedback branch. At first, the Hammerstein predistorter model is identified based on the input samples $z(n)$ of the PA and its response samples $y(n)$. After completion of the predistorter training, the parameters of the identified Hammerstein model can be copied to the Hammerstein predistorter model which is then used to produce out of the desired transmit signal $x(n)$, the predistorted input samples $z(n)$.

Similar to linear control systems, the same sensitivities apply also to the performance of the nonlinear inverse system identification. For proper working conditions, the characteristics of the reference PA are not allowed to change too fast in time, and also the level of additive noise at the output of the PA should be low.

After initialization, the predistorter learning block as well as its copy preceding the PA simply pass the input samples on the output, i.e., $z(n) = x(n)$ and $\hat{z}(n) = y(n)$. The predistorter learning block is adapted such that its output $\hat{z}(n)$ converges to the input of the PA $z(n)$. The nonlinear subsystem N_p of the predistorter learning block and the predistorter itself is modelled using a set of orthogonal polynomials. The linear subsystem L_p is a FIR filter. The input sequences of N_p and L_p of the predistorter learning block can be written in vector form as

$$\begin{aligned} \mathbf{y}(n) &= \left[\psi_{p,1}(y(n)), \psi_{p,2}(y(n)), \ldots, \psi_{p,P_p}(y(n))\right]^T, \\ \mathbf{u}(n) &= \left[u(n), u(n-1), \ldots, u(n-P_l-1)\right]^T. \end{aligned} \tag{4.3}$$

where $\psi_{p,n}(.)$ denotes the orthogonal polynomial basis function of order n, P_p is the number of orthogonal basis functions used in N_p ,and P_l is the length of the FIR filter L_p.

The corresponding input output relations then become

$$\begin{aligned} u(n) &= \mathbf{y}^T(n)\,\mathbf{c}_p(n), \\ \hat{z}(n) &= \mathbf{u}^T(n)\,\mathbf{l}_p(n), \end{aligned} \tag{4.4}$$

where $\mathbf{c}_p(n)$ denotes the coefficient vector for the expansion coefficients of N_p and $\mathbf{l}_p(n)$ contains the coefficients of the FIR filter L_p. Then, the adaptive update equations for the linear filter respectively the nonlinear subsystem of the predistorter learning block are obtained by inserting the error sequence $e(n) = z(n) - \hat{z}(n)$ and the vectors $\mathbf{y}(n)$ and $\mathbf{u}(n)$ into the update equations (3.15) of the LMS,

$$\begin{aligned} \mathbf{l}_p(n+1) &= \mathbf{l}_p(n) + \mu_l e(n)\,\mathbf{u}^*(n), \\ \mathbf{c}_p(n+1) &= \mathbf{c}_p(n) + \mu_n e(n)\,\mathbf{y}^*(n), \end{aligned} \tag{4.5}$$

where μ_l and μ_n denote the step size for the linear part and the step size for the nonlinear part, respectively.

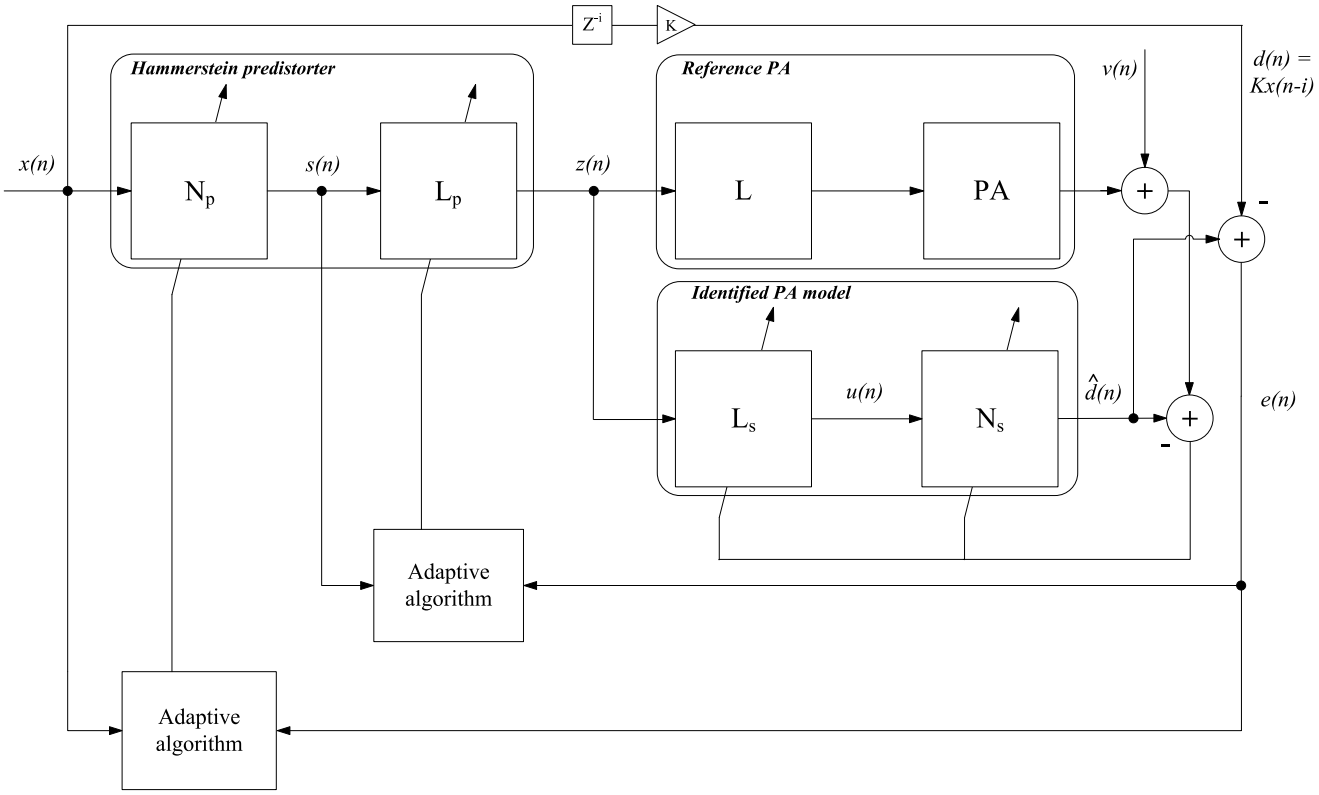

Figure 4.3: Hammerstein predistorter using direct learning architecture used to linearize a PA represented by a simplified Wiener model.

4.3 Direct learning architecture

The direct learning architecture (DLA) is another structure used in predistorter design. Due to its simplicity, most of the predistorters are based on the IDLA. However, the DLA is preferred by many authors because of its robustness and faster convergence rate [62,67]. A Volterra based predistorter for an OFDM transmitter using DLA was introduced and studied in [68]. Polynomial based predistorters using a Hammerstein model and DLA have been considered in many cases since they require less parameters, leading to a allow lower computational complexity compared to the Volterra based predistorters [67,44,60]. For most of the predistorter techniques which are based on the DLA, an increase in the accuracy of estimates and a higher convergence speed has been reported [69, 67].

The direct learning architecture illustrated in the Figure 4.3, consists of a predistorter using a Hammerstein model and a system identification block based on a simplified Wiener model. Since the predistorter block precedes the PA, the structure of the DLA is related to the inverse control for a linear dynamic system, presented in Section (3.3.2). Similar as in the case of a linear dynamic system, the noise component $v\,(n)$ which is assumed to be uncorrelated with the input $x\,(n)$, will not directly affect to the converged solution of the predistorter parameters.

For the DLA, in contrast to the IDLA, a model of the actual PA must be identified first.

This PA model can be then used to train the predistorter. Consequally, this leads to a slight growth of the overall complexity. However, once identified, the PA model can be used as a static model for the predistorter learning, under the assumption that the PA characteristics change slowly. When the mean square error between the PA and the PA model exceeds a certain threshold, the PA can be identified again.

A common choice for PA models which are able to describe nonlinearities together with memory effects is the Volterra model (2.48). It is a very general model and offers a high accuracy to identify or predistort systems with modest nonlinearities. However, for the identification or linearization of systems with "strong" nonlinearities, kernels up to a very high order are required. Due to the exponential growth of the number of parameters with the model order, this results in a not feasible computational complexity. In this case, a two-box models offer a reasonable modelling accuracy of a considerably lower complexity. This is the reason why in this work the focus lies on predistorters using two-box models.

In Figure 4.3, The system identification and predistorter learning procedure are both established with the same LMS based adaptive identification method introduced in Section 3.2.1. Before the predistortion learning starts, the PA model is assumed to have converged to the global minimum in the least mean squares sense. For the adaptation of the predistorter block, the goal is to find a Hammerstein system which minimizes the mean square error function $e(n) = \hat{d}(n) - d(n)$, where $\hat{d}(n)$ denotes the output of the identified PA model, and $d(n) = Kx(n-i)$ is a delayed and appropriately amplified (gain K) version of the sequence at the input of the Hammerstein predistorter. Let $s(n)$ denote the input of the FIR filter in the predistorter block and $z(n)$ its corresponding output. Then, the input sequences for the learning procedure can be formulated using the vector notation,

$$\begin{aligned} \mathbf{x}(n) &= \left[\psi_{p,1}(x(n)), \psi_{p,2}(x(n)), \ldots, \psi_{p,P_p}(x(n))\right]^T, \\ \mathbf{s}(n) &= \left[s(n), s(n-1), \ldots, s(n-P_l-1)\right]^T, \end{aligned}$$

where $\psi_{p,n}(.)$ denotes the orthogonal polynomial basis function of order n, P_p is the number of orthogonal basis functions used in N_p ,and P_l is the length of the FIR filter L_p.

For the predistorter, the outputs of the nonlinearity N_p ,respectively the FIR filter L_p can be expressed as

$$\begin{aligned} s(n) &= \mathbf{x}^T(n)\,\mathbf{c}_p(n), \\ z(n) &= \mathbf{s}^T(n)\,\mathbf{l}_p(n). \end{aligned}$$

where $\mathbf{c}_p(n)$ denotes the coefficient vector for the expansion coefficients of N_p and $\mathbf{l}_p(n)$ contains the coefficients of the FIR filter L_p. Substituting the error sequence $e(n)$ and the vectors $\mathbf{s}(n)$ and $\mathbf{x}(n)$ in the LMS update equation (3.6), the update equation for the coefficient vectors of the Hammerstein predistorter is obtained as,

$$\begin{aligned} \mathbf{l}_p(n+1) &= \mathbf{l_p}(n) + \mu_l e(n)\,\mathbf{s}^*(n), \\ \mathbf{c}_p(n+1) &= \mathbf{c}_p(n) + \mu_n e(n)\,\mathbf{x}^*(n), \end{aligned} \tag{4.6}$$

where μ_l denotes the step size for the linear subsystem, and μ_n the step size for the nonlinear subsystem.

As mentioned before, the problem of noise at the output of the PA can be alleviated using the DLA. However, the PA and the analog components along the transmission path introduce some certain delay which generally is not an integer of the sampling period. Consequently, the choice of the correct delay i is crucial. In the worst case a wrong delay can lead to instability.

4.4 Direct learning architecture using the NFxLMS algorithm

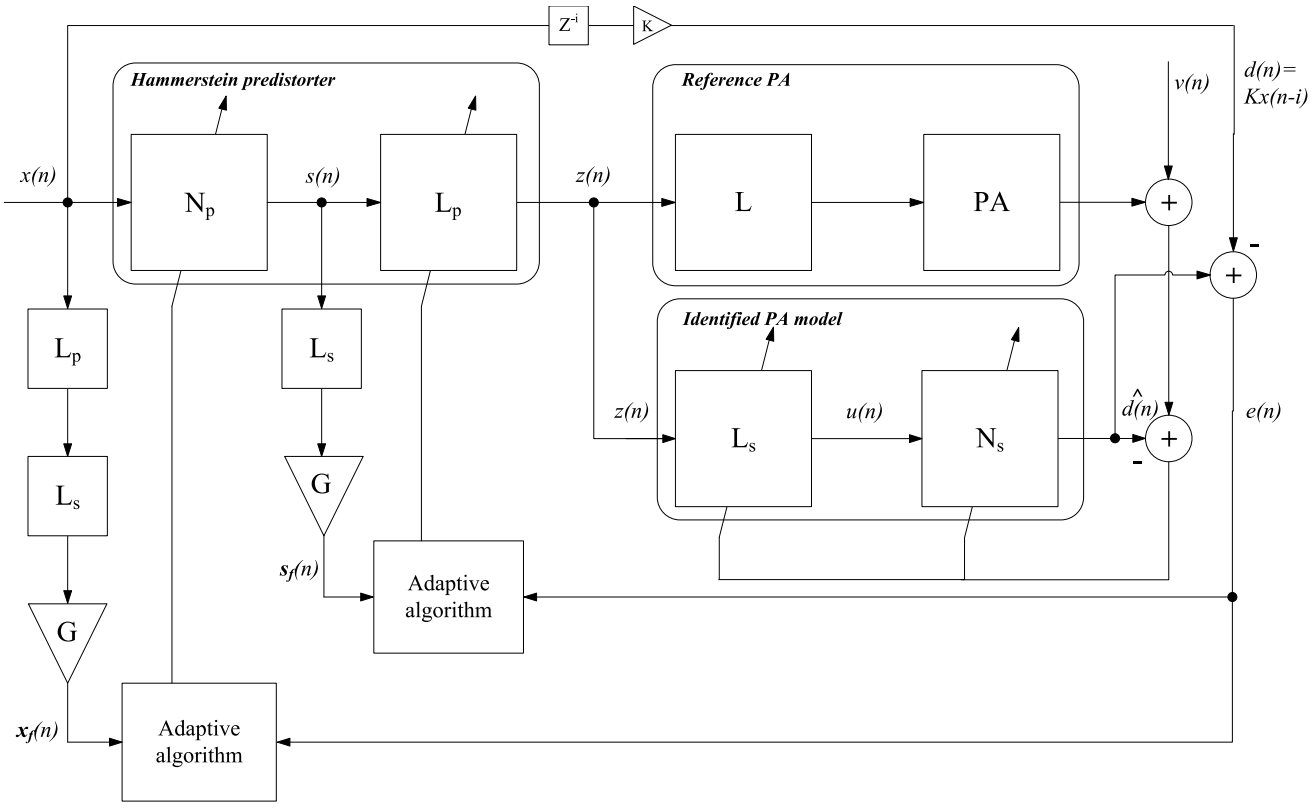

Figure 4.4: Hammerstein predistorter using NFxLMS adaptive algorithm used to linearize a PA represented by a simplified Wiener model.

The model identification of the system inverse for a nonlinear reference system was introduced in the Section 3.3.4. The adaptive update procedure of the coefficients to model the system inverse is given in 3.36, where arbitrary basis functions were used to model the nonlinearity. The resulting algorithm is known as nonlinear filtered-x LMS. Based on these results, in this section, the NFxLMS algorithm can easily be derived for the Hammerstein-DLA predistorter.

The structure of the NFxLMS algorithm using a Hammerstein predistorter is illustrated in the Figure 4.4. Additionally, a simplified Wiener model is used to identify the behaviour of the PA. The simplified Wiener model consists of an FIR filter which is followed by a nonlinearity that is represented by an orthogonal polynomial basis expansion. The FIR filter and the nonlinearity are adapted by the LMS system identification algorithm for two-box models, as introduced in Section 3.2.1.

The identified PA model is used for the training of the Hammerstein predistorter. Let $x(n)$ denote the input to the predistorter nonlinearity N_p, let $s(n)$ be the input of the corresponding FIR filter L_p, and let $z(n)$ represent the output of the predistorter identification. The goal is to minimize the mean square error of the $e(n)$ between the output $\hat{d}(n)$ of the identified PA model and the desired, appropriately delayed and amplified input sequence $Kx(n-d)$. To achieve this goal, in the following the update equations for the stochastic gradient algorithm are derived, which estimate the coefficients for the nonlinearity as well as for the FIR filter. In the update equation (3.36), the function $g(r,n)$ of the time-varying filter must be formulated for the here considered specific case. This requires to evaluate the derivatives $\frac{\partial d(n)}{\partial z(n-r)}$ and $\frac{\partial d(n)}{\partial s(n-r-l)}$. To simplify the notation for the derivation, the input sequences are written using the following vector notations,

$$\begin{aligned}
\mathbf{x}(n) &= \left[\psi_{p,1}(x(n)), \psi_{p,2}(x(n)), \ldots, \psi_{p,P_p}(x(n))\right]^T, \\
\mathbf{s}(n) &= \left[s(n), s(n-1), \ldots, s(n-P_l-1)\right]^T, \\
\mathbf{z}(n) &= \left[z(n), z(n-1), \ldots, z(n-S_l-1)\right]^T, \\
\mathbf{u}(n) &= \left[\psi_{s,1}(u(n)), \psi_{s,2}(u(n)), \ldots, \psi_{s,S_p}(u(n))\right]^T,
\end{aligned} \tag{4.7}$$

where $\psi_{p,n}(.)$ and $\psi_{s,n}(.)$ denote the orthogonal polynomial basis functions for the nonlinearity of the predistorters, respectively the PA model,
P_p is the number of orthogonal basis functions used for the nonlinearity of the predistorter model,
P_l is the length of the FIR filter in the predistorter model,
S_p is the number of orthogonal basis functions used for the nonlinearity of the PA model,
and S_l is the length of the FIR filter in the PA model.

Then the following sequences can be expressed as using the vector notations in 4.7:

$$\begin{aligned}
s(n) &= \mathbf{c}_p^T(n)\,\mathbf{x}(n), \\
\hat{d}(n) &= \mathbf{c}_s^T(n)\,\mathbf{u}(n), \\
z(n) &= \mathbf{l}_p^T(n)\,\mathbf{s}(n), \\
u(n) &= \mathbf{l}_s^T(n)\,\mathbf{z}(n).
\end{aligned} \tag{4.8}$$

The n^{th} order monomial of the baseband polynomial model (2.7) is defined as

$$\begin{aligned}
\phi_n(z) &= |z|^{n-1} z \\
&= (zz^*)^{\frac{n-1}{2}} z \\
&= z^{\frac{n+1}{2}} z^{*\frac{n-1}{2}}.
\end{aligned} \tag{4.9}$$

Thus, the derivative of the n^{th} monomial with respect to its input z can be written as[2]

$$\begin{aligned}\phi_n^{'}(z) = \frac{\partial \phi_n(z)}{\partial z} &= \frac{n+1}{2} z^{\frac{n-1}{2}} z^{*\frac{n-1}{2}} \\ &= \frac{n+1}{2} |z|^{n-1}. \end{aligned} \tag{4.10}$$

When the derivation is applied to the orthogonal polynomial basis functions in vector $\mathbf{u}(n)$, the resulting vector $\mathbf{u}^{'}(n)$ of the derivatives is obtained as

$$\mathbf{u}^{'}(n) = \left[\psi_{s,1}^{'}(u(n)), \psi_{s,2}^{'}(u(n)), \ldots, \psi_{s,N_s}^{'}(u(n))\right]^T. \tag{4.11}$$

The derivatives for the linear part of the Hammerstein predistorters become

$$\begin{aligned}\frac{\partial \hat{d}(n)}{\partial z(n-r)} &= \frac{\partial \hat{d}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial z(n-r)}, \\ &= \left(\mathbf{c}_s^T(n)\,\mathbf{u}^{'}(n)\right) \cdot \begin{cases} l_{s,r}(n) & 0 \le r \le S_l - 1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{4.12}$$

where $l_{s,r}(n)$ denotes the FIR filter coefficient of the beforehand identified PA model. When the same routine is carried out for the nonlinear part of the predistorter, the corresponding derivatives are obtained

$$\begin{aligned}\frac{\partial \hat{d}(n)}{\partial s(n-r-m)} &= \frac{\partial \hat{d}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial s(n-r-m)}, \\ &= \frac{\partial \hat{d}(n)}{\partial u(n)} \frac{\partial u(n)}{\partial z(n-r)} \frac{\partial z(n-r)}{\partial s(n-r-m)}, \\ &= \mathbf{c}_s^T(n)\,\mathbf{u}^{'}(n) \cdot \begin{cases} l_{p,m}(n)\, l_{s,r}(n) & 0 \le r \le S_l - 1, 0 \le m \le S_p - 1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{4.13}$$

where $l_{p,m}(n)$ denotes the m^{th}order FIR filter coefficient of the predistorter estimate, and where $l_{s,r}(n)$ is the r^{th} order FIR filter coefficient of the identified PA model. The amplification function $\mathbf{c}_s^T(n)\,\mathbf{u}^{'}(n)$ in Equations (4.12) and (4.13) is equal to the first derivative of the nonlinearity in the identified PA model for an input $u(n)$.

Now, the derivatives for the linear part (4.12) and the nonlinear part (4.13) can be excited by the respective input vectors $\mathbf{x}(n)$ and $\mathbf{s}(n)$ which leads to the corresponding filtered input vectors $\mathbf{x_f}(n)$ and $\mathbf{s_f}(n)$,

[2]Note that for a complex variable $z \in \mathbb{C}$ the following two indentities hold [26]: $\frac{\partial z^*}{\partial z} = 0, \frac{\partial z}{\partial z^*} = 0$.

$$\begin{aligned}
\mathbf{x_f}(n) &= \mathbf{c}_s^T(n)\,\mathbf{u}'(n)\sum_{m=1}^{P_l}\sum_{r=0}^{S_l} l_{p,m}(n)\,l_{s,r}(n)\,\mathbf{x}(n-r-m), \\
\mathbf{s_f}(n) &= \mathbf{c}_s^T(n)\,\mathbf{u}'(n)\sum_{r=0}^{S_l} l_{s,r}(n)\,\mathbf{s}(n-r).
\end{aligned} \tag{4.14}$$

Finally, the filtered input vectors $\mathbf{x_f}(n)$ and $\mathbf{s_f}(n)$ can be inserted in the NFxLMS update equation (3.36) and the update equations for the linear and the nonlinear subsystems of the predistorter model are obtained as,

$$\begin{aligned}
\mathbf{c}_l(n+1) &= \mathbf{c}_l(n) + \mu_l e^*(n)\,\mathbf{x_f}(n), \\
\mathbf{c}_p(n+1) &= \mathbf{c}_p(n) + \mu_p e^*(n)\,\mathbf{s_f}(n),
\end{aligned} \tag{4.15}$$

where μ_l and μ_p denote the step size for the update of the linear part, respectively the nonlinear part of the predistorter.

Chapter 5

Analysis

This chapter investigates the behaviour of orthogonal polynomials in the context of adaptive system identification. Section 5.1 analyzes the reaction between their numerical properties and the obtained convergence speed of the corresponding adaptive algorithms. Section 5.2 considers the performance of adaptive system identification for Wiener models which use orthogonal polynomials to describe the contained static nonlinearity. Linearization of the theoretical PA model with memory using several predistorter architectures is covered in Section 5.3. Finally, in Section 5.4 the applicability of the orthogonal basis functions is evaluated by simulations which include burst measurements obtained with a commercial PA.

5.1 Orthogonal basis functions

The PA modeling can be established by using system models based on orthogonal polynomials as introduced in the Section 2.2.5. The important question is, how different basis expansions perform if excited by common baseband modulation signals and whether they can be used to model the strong nonlinear effects of PAs adequately. Section 5.1.1 analysis the modeling accuracy and the numerical stability of orthogonal basis functions.

In adaptive system identification, it is important for the adaptive algorithm to be able to track the variations of the reference system with time efficiently. Therefore, it is of special interest to study also the convergence behaviour of adaptive schemes using different orthogonal polynomials. In Section, 5.1.2 the convergence speeds are considered for the identification of a PA using a quasi-memoryless model.

5.1.1 Numerical stability

The least squares estimators for the conventional and the orthogonal baseband model were derived in the Section 2.2.4. For both estimators, the practicability of the estimator is

determined by the condition number[1] of the corresponding autocorrelation matrix. When the conventional baseband model (2.7) is used for the estimation, the condition number of the autocorrelation matrix $\mathbf{\Phi}^H\mathbf{\Phi}$ in 2.19 was is prone to grow exponentially with the number of the polynomial functions (cmp. Section 2.2.1). To improve the numerical stability, three orthogonal baseband models, based on the conventional baseband model, were introduced in Section 2.2.5. In theory, by the use of orthogonal bases, the numerical condition of the autocorrelation matrix $\mathbf{\Psi}^H\mathbf{\Psi}$ can be improved immensely even if the amplitude distribution of the input sequence deviates slightly from the one which leads to exact orthogonality. Prior to this argument, the objective of the comparison is merely to inspect the condition number with different amplitude distribution characteristics.

To investigate the numerical stability of the orthogonal polynomial basis, it would be necessary to evaluate the condition number of the expectation $E\left[\mathbf{\Psi}^H\mathbf{\Psi}\right]$. In general, the calculation of the condition number for arbitrarily distributed input signals is difficult to be carried out analytically. However, under the assumption of ergodic input sequences the sample mean can be used to numerically approximate the actual mean value [21]. Then, it is possible to numerically evaluate the condition number for different polynomial bases and for different amplitude distributions. The precision of the estimates can be achieved by averaging a sufficient number of runs.

In Figure 5.1, the condition number is estimated for five different orthogonal polynomial bases. In order to measure the numerical robustness of a particular basis, the estimated condition numbers are evaluated for four complex valued input sequences with different amplitude statistics. Uniformly respectively Gaussian distributed white noise sequences are representing broadband excitation signals which are typically used for simulations in the context of system identification schemes. 64-QAM and OFDM type sequences are more practical signals, representing common modulation methods used in telecommunication systems. Compared with the uniformly and Gaussian distributed white noise, the bandwidth of the 64-QAM and the OFDM signals is more narrow (approximately 0.1π). Upsampling and pulse shaping of the 64-QAM signal is established according to the UMTS standard TS 25.106 [70] . Accordingly, the anti-aliasing filter is of root-raised cosine type with a roll-off factor of 0.22. The OFDM signal is generated using 16-QAM samples for 128 subcarriers (1024 carriers in total).

The estimated condition number depends on the number of input samples, as presented in [21]. It was also shown in [21] that the estimated condition numbers will be close to the theoretical condition numbers when the number of input samples is large enough (> 10000). The autocorrelation matrices used to calculate the estimated condition numbers are obtained from 15000 samples for each number of orthogonal basis functions. The dimensions of the autocorrelation matrices are thus 15000×15000. Hence, the residual between the estimated and the actual condition number will be low. Additionally, the condition numbers are averaged over 100 independent estimates.

Considering the condition numbers in Figure 5.1, obtained for the complex-valued white Gaussian noise (b) and the OFDM signals (d), the results for the *complex Gaussian poly-*

[1]The condition number is defined as a ratio between the maximum eigenvalue λ_{max} and the minimum eigenvalue λ_{min} of a matrix, i.e., $\frac{\lambda_{max}}{\lambda_{min}}$.

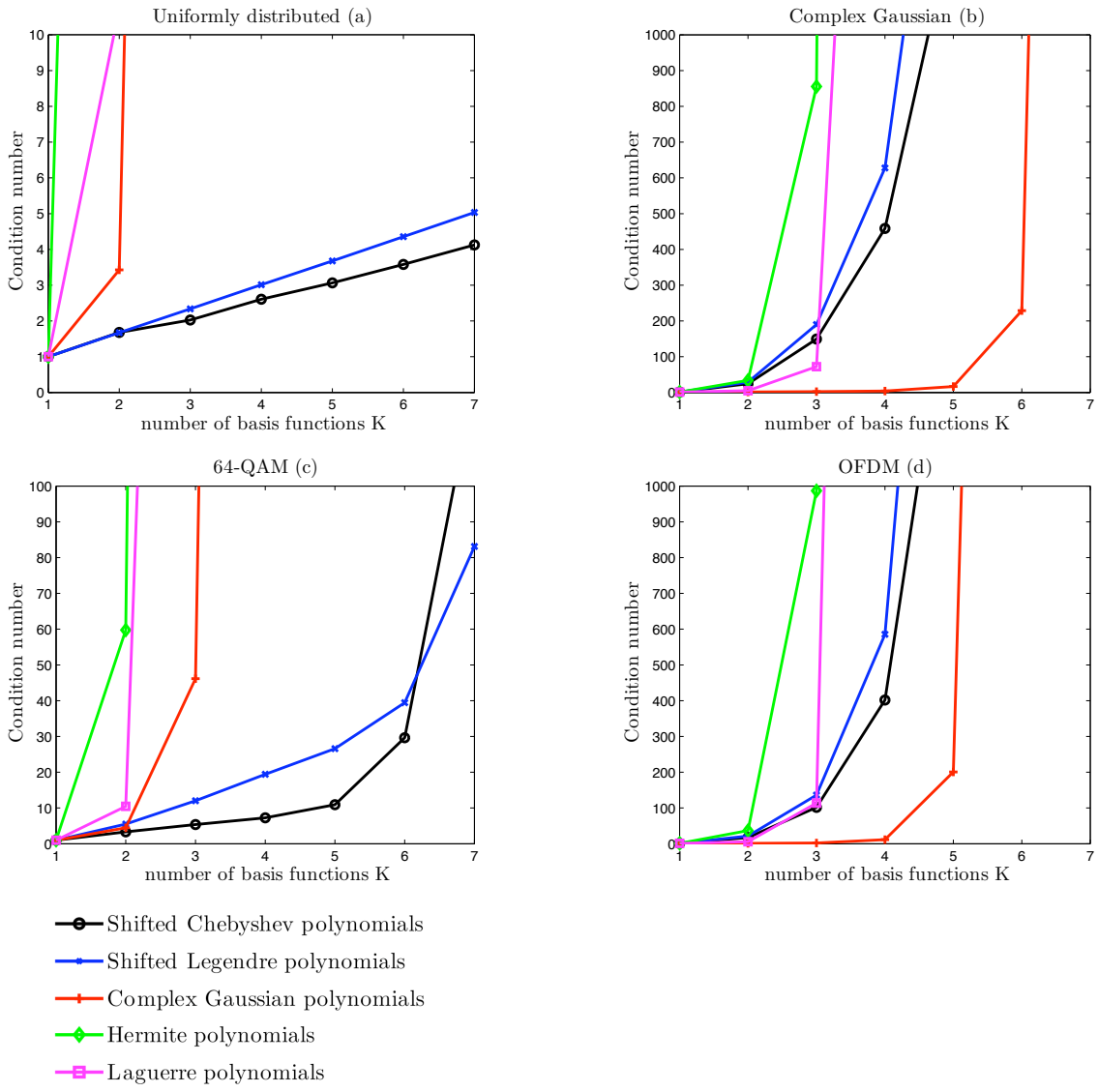

Figure 5.1: Estimated condition number of $E\left[\boldsymbol{\Psi}^H\boldsymbol{\Psi}\right]$ for several orthogonal polynomial bases with the input signal (a) uniformly distributed white noise, (b) complex Gaussian white noise, (c) a root-raised cosine filtered 64-QAM signal, (d) an OFDM signal with 128 subcarriers and 16-QAM as subcarrier modulation scheme.

nomials (2.36) are clearly distinguished from the results for the other polynomials. On the other hand, when the amplitude distribution of the input signal deviates significantly from the complex Gaussian case, the magnitude of the condition number increases rapidly in an exponential way for the *complex Gaussian polynomials*. For the uniformly distributed noise (a) and the 64-QAM (c) signals, the Legendre and the Chebyshev type polynomial bases show superior performance compared to the other bases. Additionally, the condition number estimates for these two bases remain relatively low, even with complex input Gaussian signals. Thereby, it can be concluded that the shifted Legendre basis and the shifted Chebyshev basis will in average lead to good performance which is rather inde-

pendent from the used modulation technique. From Figure 5.1, it can also be observed that the Hermite and Laguerre polynomials, which are defined in the real domain, show weaker performance in comparison to the rest of the polynomials defined in the complex domain.

5.1.2 Convergence analysis

In the context of adaptive system identification, an important property of the used basis functions is how they affect the performance of the adaptation of the nonlinear system.

In this section, the convergence speed is considered based on the a-priori error which is a commonly used measure when comparing adaptive algorithms [71]. In the convergence analysis, a-priori error is calculated for each iteration cycle, which is defined as the distance between the output of the reference system model $y(n)$ and the output of the estimated model $\hat{y}(n)$, i.e.,

$$e_{\mathrm{a}}(n) = y(n) - \hat{y}(n). \tag{5.1}$$

The reference system model is an estimate of the original nonlinear system model, in the sense that based on the used orthogonal polynomials bases, the actually used reference system approximates the original nonlinear model in the least mean square sense using the NLMS algorithm. For the simulations only this approximated system was used, not the original nonlinear model. In this way, a fair comparison can be carried out for different basis functions. The original nonlinear system model is a Saleh PA model (2.44) with the simplified input parameters $\alpha_A = 2, \beta_A = 1, \alpha_\Phi = 2, \beta_\Phi = 1$, according to the recommendation of IEEE 802.16 of the Broadband Wireless Access Working Group [72]. In fact, for the simulations, orthogonal baseband model including the first three basis polynomials is used for the nonlinear function of the actual reference system model. The NLMS algorithm (3.11) was used for the adaptive identification of the reference system model and estimated reference system model. A fixed adaptation step size of 0.2 was used in every identification scheme. The adaptive algorithm converges until the error between the reference system model and the estimated system model is less than the numerical precision of the simulation software[2].

[2] In this case Matlab is used with 64-bit double precision format.

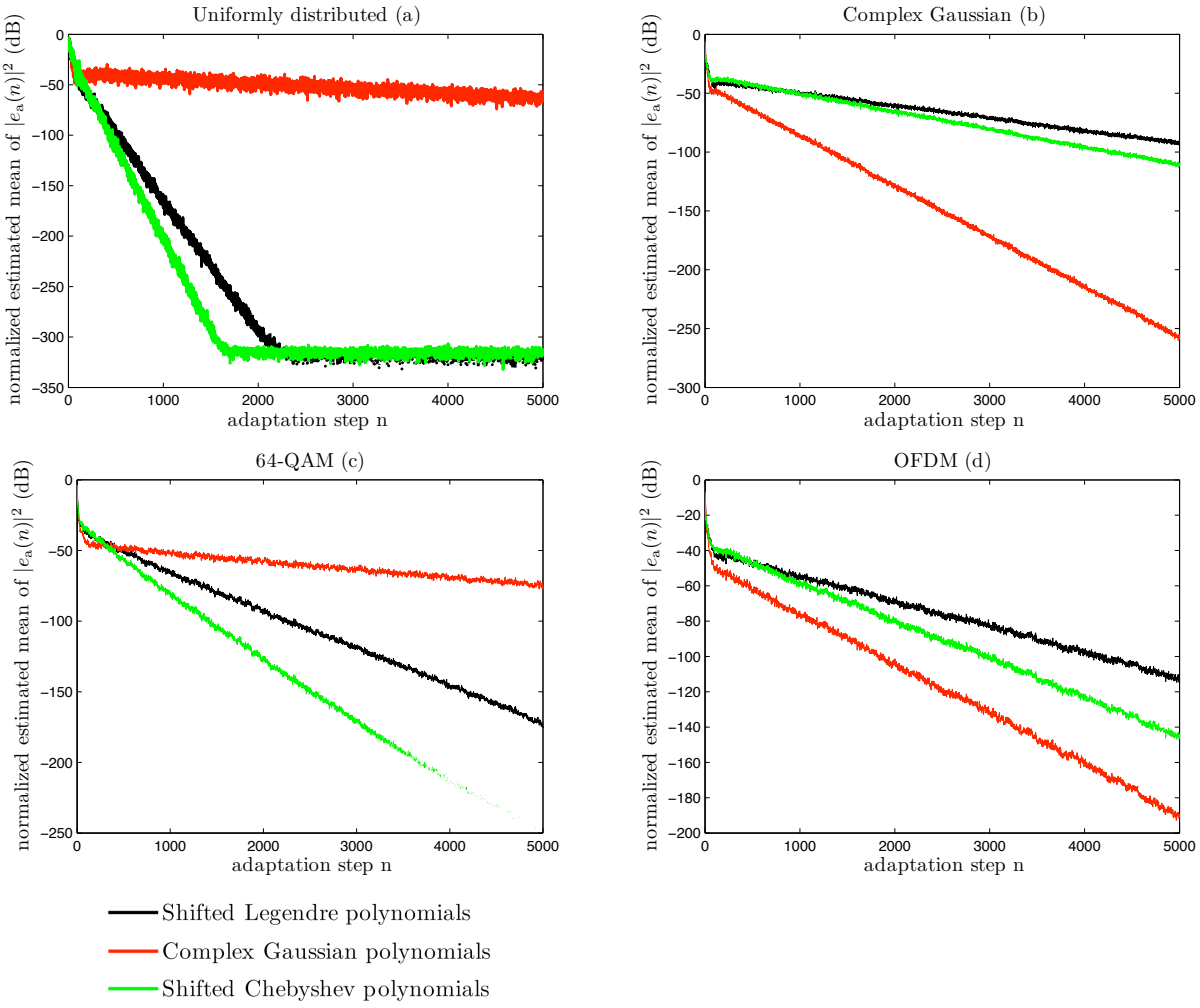

Figure 5.2: A-priori error using the NLMS algorithm for several orthogonal polynomials with the input signal: (a) a uniformly distributed white noise, (b) a complex Gaussian white noise, (c) a root-raised cosine filtered 64-QAM signal, (d) an OFDM signal with 128 subcarriers and 16-QAM as subcarrier modulation scheme

In Figure 5.2, the a-priori error is plotted for the shifted Legendre polynomials, the shifted Chebyshev polynomials and the *complex Gaussian polynomials*. Four different input signals similar to the ones used to estimate the condition numbers in Section 5.1.1 are considered.Comparing Figure 5.1 and 5.1.1, it can be observed that the convergence speed of the a-priori errors reflects roughly the magnitudes of the corresponding estimated condition numbers. Hence, the NLMS adaptation is expected to be faster and more stable when the corresponding condition number of the LS-estimator is lower.

Apparently, for complex Gaussian white noise (b) and the OFDM input signal (d) the *complex Gaussian polynomial* basis achieves faster convergence speeds. However, on average, the fastest convergence speed is achieved with shifted Chebyshev polynomials. The best performance for the shifted Chebyshev polynomials is measured for the uniform and root-raised cosine filtered 64-QAM signals.

5.2 Adaptive Wiener models

Accurate modeling of the PA plays an important role in the design of adaptive digital predistorters. In the end, the performance of the predistorters based on the direct learning structure (Section 4.3) is directly related to the accuracy of the identified system model. Thus, it is important to evaluate the identification process of the system model when the reference PA exhibits memory effects.

In this section, for the estimated PA model a simplified Wiener model is considered, where the nonlinear subsystem is represented using an orthogonal basis expansion. For difference orthogonal polynomial bases, introduced in Section 2.2.5, two analyses are performed. First, a convergence analysis is performed based on the comparison of the a-priori errors. Then, in Section 5.2.2, tests on the modeling accuracy are performed for a varying number of basis functions.

5.2.1 Convergence analysis

The PA reference model used for the identification of the Wiener model is consisting of a linear part represented by a 4^{th} order FIR filter and a nonlinear part represented by a Saleh model. The complex valued and uniformly distributed coefficients of the FIR filter, listed in Table 5.1, were randomly generated using Matlab function **`rand`** for both real and imaginary parts. Additionally, the invertibility of the filter was ensured by reflecting all the zeros with magnitude more than one inside the unit circle with the Matlab function **`polystab`**. Without loss of generality, the first coefficient is forced to be real valued.

filter coefficient	value
h_0	0.4981
h_1	$0.0783 + 0.0055i$
h_2	$-0.0226 - 0.1524i$
h_3	$0.2154 + 0.0608i$
h_4	$0.0446 - 0.0090i$

Table 5.1: Coefficients of the FIR filter used in the Wiener model

The parameters of the Saleh model were chosen according to the recommendation IEEE 802.16 of the Broadband Wireless Access Working Group [72]. A PA with strong nonlinear behaviour can be modelled by a Saleh model with the following parameters $\alpha_A = 2, \beta_A = 1, \alpha_\Phi = 4, \beta_\Phi = 9$.

The identified Wiener model in the adaptive system is also consisting of a linear part represented by a 4^{th} order FIR filter and a nonlinear part represented by the orthogonal baseband model (see Section 2.3.2) using three basis functions. The identification structure is equivalent to the two-box adaptive identification scheme introduced in Section 3.2.1. For both subsystems, the NLMS algorithm was used with a constant step-size of $\alpha = 0.2$. For a reliable convergence analysis, the a-priori error was observed.

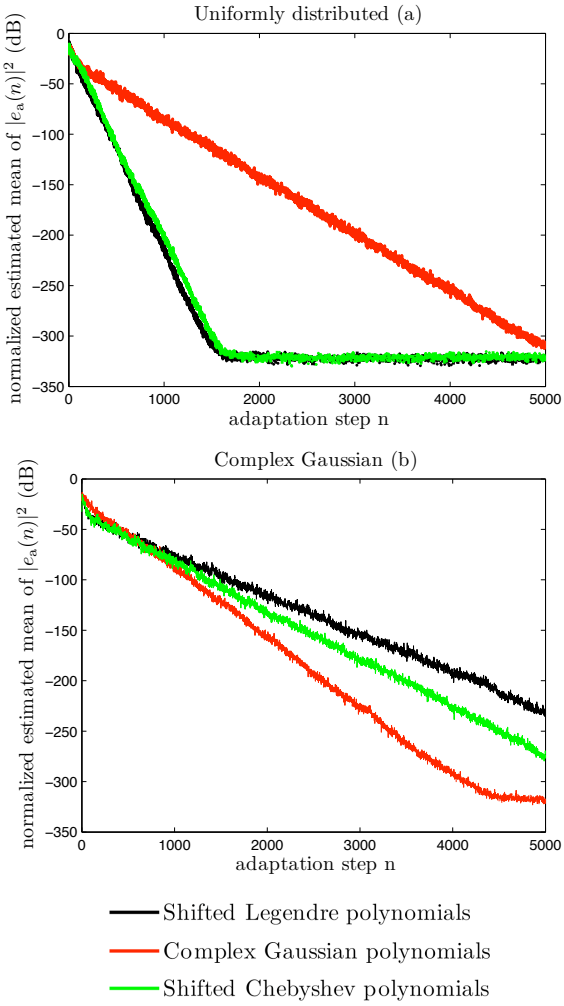

Figure 5.3: A-priori error for the Wiener model using the NLMS algorithm for the adaptation, where the nonlinear part of the Wiener model is represented by three different orthogonal polynomials. Convergence rates can be observed for (a) uniformly distributed white noise and (b) complex Gaussian white noise

In Figure 5.3, the a-priori error is plotted for adaptive Wiener models employing three different orthogonal basis expansions. When the complex valued input amplitude is uniformly distributed (a) the basis of shifted Legendre polynomials and the basis of shifted Chebyshev lead to faster convergence than the basis of *complex Gaussian polynomials*. In contrast, as desired, the basis of *complex Gaussian polynomials* performs better compared to the other bases when the input is complex Gaussian distributed. When comparing the a-priori errors of the nonlinearity in Figure 5.2 (b) and the results for the simulations using a Wiener model in Figure 5.3 (b), the similarity between the two cases becomes obvious. This behaviour suggests that the Wiener model estimators behaviour is strongly depending on the identification of the nonlinear part.

5.2.2 Mean square error analysis

The modeling accuracy of the PA estimator is an important aspect. When the PA is modelled using the simplified Wiener model, a crucial point is to find the lowest order of the used polynomial basis, respectively the smallest number of FIR filter taps, which is necessary to adequately describe the behaviour of the reference system.

System identification using an adaptive Wiener model can be implemented according to the identification structure introduced in Section 3.2.1. The NLMS is used as adaptation algorithm for both subsystems of the Wiener model to guarantee fast and stable convergence even for high degrees of nonlinearity. The step sizes are chosen to be $\alpha_{NL} = 0.5$ for the nonlinear subsystem and $\alpha_L = 0.05$ for the linear subsystem. Again, as in Section 5.2.1, the reference PA Wiener model is consisting of a linear part represented by a 4^{th} order FIR and a nonlinear part represented by the orthogonal baseband model using first three basis functions. A zero-mean complex valued white Gaussian noise with unit variance was used as input signal. To evaluate the modeling error between the reference model and the estimated system, the normalized mean square error (NMSE) is used, which in this case is given by

$$NMSE(dB) = 10\log_{10}\left[\frac{\sum_{n=1}^{N}\left|y\left(n\right)-\hat{y}\left(n\right)\right|^2}{\sum_{n=1}^{N}\left|y\left(n\right)\right|^2}\right], \tag{5.2}$$

where y and $\hat{y}$ denote the output sequences of the reference PA model, respectively the identified PA model and N denotes the number of steps used in the adaptation.

Table 5.2 lists the NMSE, averaged over 100 adaptation runs, for three polynomial bases. As expected, the modeling precision is nearly independent from the used orthogonal polynomial basis. Interestingly, the magnitude of the error is similar although the *complex Gaussian* basis, consisting of polynomials with the odd orders $[1, 3, 5, \ldots]$, has a higher maximum order than the shifted Legendre polynomials and shifted Chebyshev polynomials. This phenomenon could be explained by the better modeling precision of the bases using also even orders. When more than five basis functions were used for the *complex Gaussian* basis, in the Wiener model, the adaptive identification procedure did not converge to the global minimum. However, the most important result indicated by this anal-

ysis is that the identified PA model is able to track the nonlinear effects of the reference PA model adequately, more or less independently of the used polynomial basis

Number of basis functions	Shifted Legendre	Shifted Chebyshev	*Complex Gaussian*
1	-17.00	-17.16	-17.09
2	-33.93	-33.92	-33.05
3	-44.31	-44.36	-46.89
4	-62.70	-62.18	-60.82
5	-69.54	-67.12	-74.97
6	-79.70	-80.31	-44.42
7	-88.18	-89.73	-45.32

Table 5.2: Normalized mean-square errors in dB for the simplified Wiener models after 15000 adaptation steps and averaged over 100 runs.

5.3 Predistorters based on orthogonal basis expansions

In this section the performance of three different Hammerstein type predistorters as introduced in the Sections 4.2-4.4 are evaluated. The essential difference between the predistorters lies in the structure of the learning architecture which also determines the computational complexity of the whole system. The question is, whether the use of a learning architecture with higher computational complexity leads to a better performance. The performance tests are done for the different learning structures, using the same Wiener type reference PA model as in Section 5.2.1.

The efficiency of the predistorters is usually evaluated by the spectral regrowth that remains at the output of the PA if predistortion is used. The power leakage to the adjacent channels can be expressed using the ACPR. However, for the simulations, it is sufficient to visually observe the predistortion performance by comparing the power spectral densities (PSD) of the corresponding predistorter learning architectures.

The indirect learning architecture, introduced in the Section 4.2, shows the most smallest computational complexity since no explicit identification of a PA model is required. In contrast, the predistorter models based on the direct learning architectures (introduced in Section 4.3 and Section 4.4), require the PA model to be identified explicitly. Thus, they lead to a higher computational complexity. Therefore, it is interesting to compare the performance of the identified predistorters for all three learning structures.

For the simulations, the PA was modelled using a Wiener model consisting of a 4^{th} order linear FIR filter and a Saleh model. The Hammerstein predistorter consists of a nonlinear sub-system using a basis of the first five shifted Chebyshev polynomials of the second kind and a linear sub-system represented by a 4^{th} order FIR filter. The parameters of the sub-systems used in the reference PA model are the same as in Section 5.2.1. The identification of the Wiener model is performed according to Section 3.2.1. The NLMS algorithm was used for the adaptation of each learning structure. The constant step sizes

guaranteeing a stable behaviour for the adaptive algorithms were obtained by trial and error separately for each learning structure.

In the Figure 5.4, the estimated PSDs of the obtained PA output signals are plotted for the considered learning architectures after convergence of the predistorter. A root-raised cosine filtered 64-QAM signal (a) was used as excitation signal leading to the corresponding output signal (b) without predistortion. At first, a memoryless predistorter[3] was identified using the direct learning architecture from Section 4.3. With this memoryless predistorter the spectral regrowth is obviously reduced in (c). However, the incapability of compensating the memory effects leads only to minor improvements. Afterwards, a Hammerstein predistorter with memory was identified using both, direct and indirect learning architectures. The overall performance of the predistorter with memory is clearly better than for the memoryless predistorter. However, the performance is rather independent of the used learning scheme (indirect or direct learning). Finally, a Hammerstein predistorter of the same structure as before was identified by using the direct learning architecture employing the NFxLMS algorithm (Section 4.4). Here, a significant improvement in the spectral regrowth can be observed in (f). In some spectral regions the improvement with respect to the spectral regrowth is up to 5 dB compared to results obtained by the direct and the indirect learning architectures.

The same simulations were repeated for the 128 subcarrier OFDM sequence and the results are depicted in Figure 5.5. Again, the memoryless predistorter was not able to compensate the memory effects sufficiently and a similar performance was obtained as with the 64-QAM signal. As expected, using the Hammerstein model, leads to a decreased spectral regrowth compared to the memoryless case. While the direct and the indirect learning architectures achieved good increase in the performance, the best predistorter training architecture turned out to be the direct learning architecture employing the NFxLMS algorithm.

[3] Hammerstein predistorter without memory.

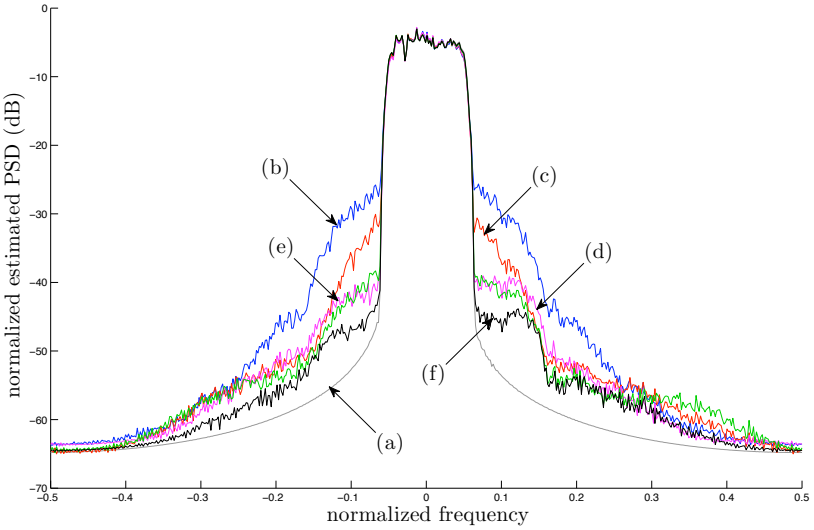

Figure 5.4: Estimated PSDs of the output of a dynamic nonlinear PA model, modelled by a simplified Wiener model, and for reference, the input signal (a) 64-QAM digital baseband signal adequately amplified to allow the comparison with the output signal. The other plots show the PA response (b) without predistorter, (c) with memoryless predistorter, (d) with Hammerstein predistorter adapted by the indirect learning architecture, (e) with Hammerstein predistorter adapted by the direct learning architecture (f) with Hammerstein predistorter adapted by the NFxLMS algorithm.

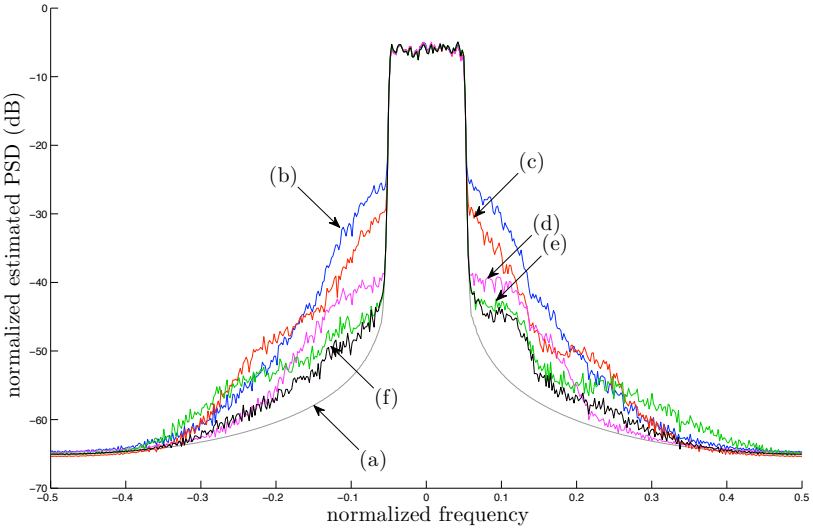

Figure 5.5: Estimated PSDs of the output of a dynamic nonlinear PA model, modelled by a simplified Wiener model, and for reference, the input signal (a) OFDM digital baseband signal adequately amplified to allow the comparison with the output signal. The other plots show the PA response (b) without predistorter, (c) with memoryless predistorter, (d) with Hammerstein predistorter adapted by the indirect learning architecture, (e) with Hammerstein predistorter adapted by the direct learning architecture (f) with Hammerstein predistorter adapted by the NFxLMS algorithm.

5.4 Simulations based on burst measurements

The performance of different learning architectures for digital predistortion was analyzed by pure simulations in the Section 5.3. However, to evaluate the real-world performance of the considered algorithms in a more reliable way, this section presents results obtained by simulations which include burst measurements using a commercial PA.

5.4.1 Simulation setup

The testbed, illustrated in Figures 5.6 and 5.7, is consisting of following devices:

- Sender unit, Rhode & Schwarz Vector Signal Generator SMU200
- Receiver unit, Rhode & Schwarz Signal Analyzer FSQ 26
- Power meter, Rhode & Schwarz Power Sensor NPR-Z11
- Digital multimeter, Agilent 34410A
- PC with Matlab software

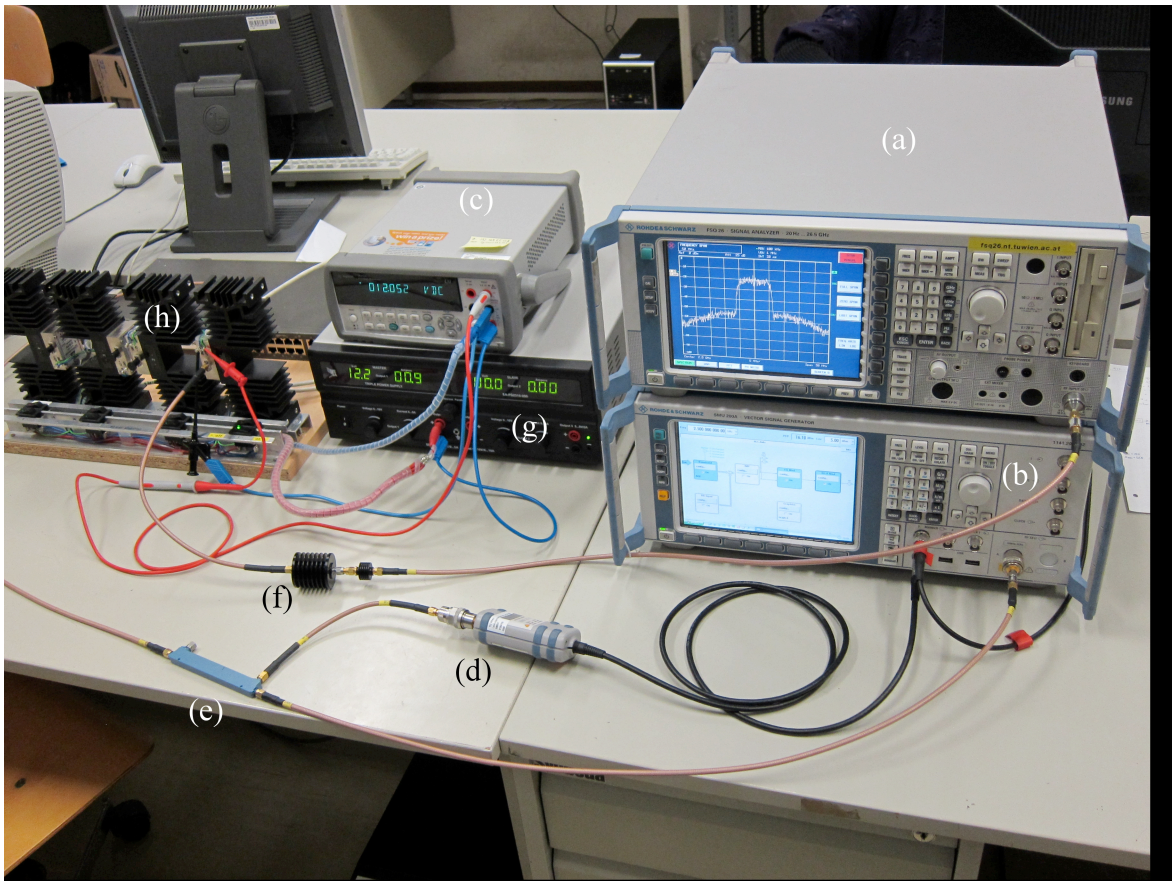

Figure 5.6: The predistorter testbed used for the burst measurements. The following devices are included in the picture (cmp. Figure 5.7): (a) FSQ 26 Signal Analyzer, (b) SMU 200A Vector Signal Generator (c) Digital multimeter, (d) NPR-Z11 power sensor, (e) directional coupler, (f) attenuators (20 dB and 10 dB), (g) DC power source (h) ZVE-8G+ power amplifier (used one out of four).

Communication between the measurement equipment and the personal computer (PC) is established via local area network (LAN) using TCP/IP. The virtual instrument software architecture (VISA) standard is supported by the Instrument Control Toolbox of Matlab. It enables the communication between Matlab and the measurement devices. With this testbed, the functionality of the different predistortion algorithms with commercial PAs can be analyzed. Especially, it allows to measure the figures of merit ACPR and PAE. Additionally, the measurements can be done almost automatically using only the control via Matlab scripts.

5.4.2 Testbed configuration for the predistortion analysis

The structure of the used testbed is illustrated in Figure 5.7. As an initial step, a sequence of 10000 random symbols is generated and modulated using either OFDM modulation or 16-QAM modulation. In the case of 16-QAM modulation, upsampling and pulseshaping are applied to the complex valued symbol sequence (cmp. Section 5.1.1). For the OFDM modulation, 1024 carriers and 128 subcarriers utilizing a 16-QAM constellation are used. Then, the double precision baseband signal $u(n)$ is normalized and quantized to 16 bit values $u_q\,(n)$ and transferred via LAN from the PC to the sender unit (at this point without any predistortion). In the sender unit, the circular signal[4] is first stored on the internal hard drive from where it is copied to high-speed random access memory (RAM). From the high-speed RAM, the digital baseband signal is read out continuously. A digital-to-analog converter (DAC), clocked with 100 Msps, converts the digital baseband signal to its corresponding analog representation. The following anti-aliasing filtering ensures the avoidance of spectral images at the output. The analog signal $v\,(t)$ with the bandwidth of 20 MHz is converted up to the RF frequency of $f_c = 2.5$ GHz and provided at the output of the sender $v_f\,(t)$. Before amplification by the PA, the power level of the signal is monitored and adjusted in order to keep the power level at the input of the PA constant. Following the PA, the signal is attenuated by 30 dB with two fixed attenuators to achieve an adequate power level at the input $w_f\,(t)$ of the receiver unit. At the receiver side, the signal is converted down to baseband and digitized by an analog-to-digital converter (ADC) running at a sampling rate of 100 Msps and providing a resolution of 14 bits.

Once the received baseband signal $y\,(n)$ is stored and transferred back to the PC, the training of the predistorter parameters is performed. However, before that, the output sequence must be synchronized with the corresponding input sequence. The synchronization is established in two steps. Firstly, an integer delay is estimated and compensated using the estimated cross correlation function $R_{xy}\,(n)$ of the sent signal $x\,(n)$ and the received signal $y\,(n)$. The index of the maximum cross correlation value indicates the delay. The

[4]The digital baseband signal is generated so that no discontinuities occur when the signal is transmitted circularily.

integer delay estimator can be defined as follows

$$\begin{aligned} D_{int} &= \arg\max_{n} \left| R_{xy}(n) \right|, \\ R_{xy}(n) &= \sum_{i=0}^{M-n-1} x(i+n)\, y^{*}(i), \end{aligned} \tag{5.3}$$

where M denotes the length of the input and the output sequence.

Secondly, the phase offset between the local oscillator at the receiver and the transmitter needs to be compensated. For low power levels of the transmit signal $v_f(t)$, the output of the PA corresponds considerably well to a linear amplified version of its input signal, .i.e,

$$y_i(n) = a x(n), \tag{5.4}$$

where $x(n)$ denotes the normalized input signal value and $y_i(n)$ denotes the corresponding normalized output signal value with integer delay compensated and a the phase offset. Then, the phase offset can be estimated using a following LS-estimator

$$\hat{a} = \frac{x^H y}{\|y\|_2^2}, \tag{5.5}$$

Once the output signal is available and synchronized, the predistorter training can take place. Depending on the learning structure, an additional system identification is performed before the predistorter learning. After 10000 iterations, the predistorter coefficients can be assumed to have converged to the global minimum in the least mean squares sense. However, the inverse gain of the predistorter after the predistorter training is not reciprocal to the gain of the PA. In order to obtain the intended linearized gain, the coefficients of the polynomial basis must be scaled correctly. This scaling operation can be done by a matrix transformation as originally defined[5] in [28]. The scaling matrix $\mathbf{M}$ and the scaling operation are defined as follows

$$\begin{aligned} \mathbf{M} &= \mathbf{U}^{-1}\mathbf{A}\mathbf{U}, \\ \boldsymbol{\beta} &= \mathbf{M}\boldsymbol{\gamma} \end{aligned} \tag{5.6}$$

and

$$\mathbf{A} = \begin{bmatrix} \alpha & 0 & 0 & \cdots & 0 \\ 0 & \alpha^2 & 0 & \cdots & 0 \\ 0 & 0 & \alpha^3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha^K \end{bmatrix}. \tag{5.7}$$

[5]For more detailed derivation, see the corresponding reference.

where $\mathbf{U}$ denotes the matrix combining the coefficients of the orthogonal baseband model (cmp. (2.28)), $\boldsymbol{\gamma}$ denotes the vector containing the original polynomial coefficients, $\boldsymbol{\beta}$ denotes the vector containing the correctly scaled polynomial coefficients, and α is the intended linearized gain.

Finally, the coefficients with the gain correction can be transferred to the predistorter and the predistorted input can be generated using based on the original input sequence $u(n)$. The parameters required for the calculation of the PAE can be obtained using the power meter [6], the multimeter and the receiver. The output power of the PA is measured directly in the receiver, the input power of the PA is obtained by the power meter, and the overall DC power consumption is measured indirectly via DC voltage and DC current by the multimeter. The ACPR can be calculated using an internal function of the receiver unit.

[6] To obtain the actual input power level of the PA, the power levels measured by the power meter need to be modified (cmp. Fig 5.7). During calibration an offset of 7.82 dB was obtained.

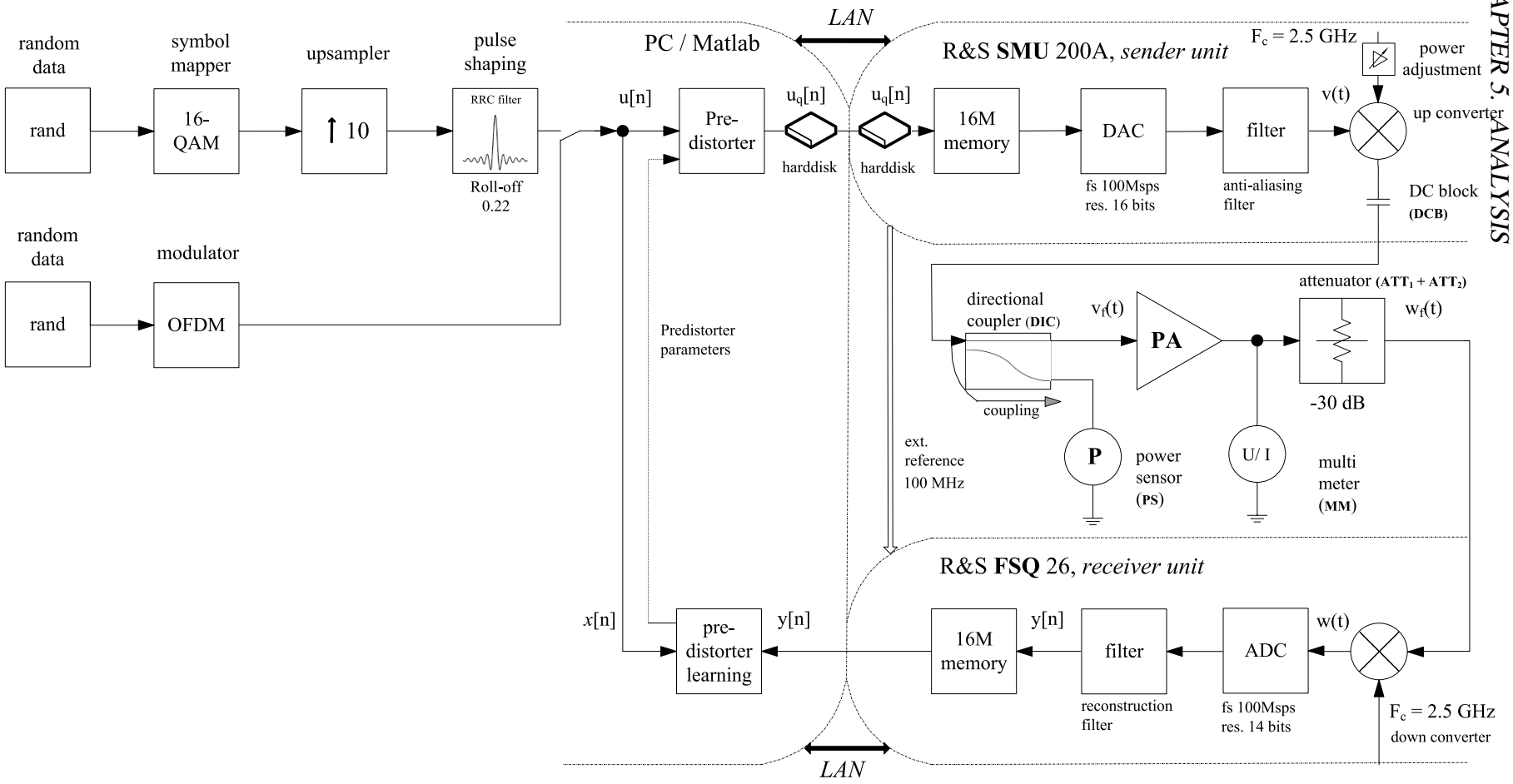

Figure 5.7: Setup of the testbed for the simulation of the predistortion algorithms based on the measurements (cmp. Figure 5.6).

5.4.3 Measured commercial power amplifier

Manufacturer:	Mini-Circuits
Type:	ZVE-8G+
Frequency range	2000-8000 MHz
Gain	30 dB
Maximum power	+30 dBm (1 dB Compression point)
DC power	12V

The PA used for the measurements was a Mini-Circuits ZVE-8G+ commerical PA for example suitable in satellite communication systems and line-of-sight transmitters [73]. When the PA was excited well beyond the saturation point, significant nonlinear compression was measured on the input levels around the saturation area. Therefore, when the ZVE-8G+ amplifier is operated on its linear region, the PAE ratings will be very low ($< 3\%$[7]).

5.4.4 Simulation results

Two separate measurement sets were performed, one with a 16-QAM excitation signal, the other one with an OFDM excitation signal. As for the simulations in the previous sections of this chapter several types of predistorters and learning algorithms were here considered, on one hand, a memoryless predistorter only consisting of a polynomial nonlinearity, and on the other hand, a Hammerstein structure with polynomial nonlinearity combined with a FIR filter. Again, a simplified Wiener model was used to model the PA for the direct learning architectures.

For the static nonlinear part of the Wiener model as well as for the Hammerstein model, orthogonal baseband model containing five basis functions based on shifted Chebyshev polynomials of second kind was used. The linear dynamic part was modelled by a 4th order FIR filter.

The learning architectures used for the predistorter training were, the direct learning architecture from Section 4.3, the indirect learning architecture from Section 4.2 and the direct learning architecture based on the NFxLMS algorithm presented in Section 4.4. For the NLMS, the step size α_{NL} for the adaptation of the nonlinearity, as well as the step size α_L for the adaptation of the FIR filter were chosen manually, such that stability was ensured. The step sizes are listed in Table 5.3.

[7] Based on the measurements

Identification scheme	α_{NL}	α_L
Direct learning	0.1	0.01
Indirect learning	0.3	0.3
NFxLMS	0.1	0.01
Memoryless	0.1	-
Wiener PA model identification	0.1	0.03

Table 5.3: NLMS step sizes used in the simulations.

Another important factor which determines the performance of the linearization procedure is the gain correction of the orthogonal basis as described in Section 5.4.2. The selection of the intended linearized gain is not a trivial task, especially when the PAs amplitude response is strongly nonlinear. Apparently, the gain depends on the input signal, but also on the used predistorter training method. Therefore, the gain values were adjusted by hand individually for each case. The manually optimized gain constants are listed in the Table 5.4.

Predistorter architecture	Gain / 16-QAM	Gain / OFDM
Memoryless predistortion	1.4	1.6
Hammerstein predistortion / Direct learning	1.4	1.6
Hammerstein predistortion / Indirect learning	1.15	1.4
Hammerstein predistortion / NFxLMS	1.4	1.6

Table 5.4: Selected intended linearized gain constant used to adjust the orthogonal basis.

The performance of the predistorters was measured using several metrics. To illustrate the spectral regrowth, the estimated PSDs were evaluated for the original excitation signals, for the respective PA responses, and for the predistorted PA responses. The ACPR was measured to allow a fair and accurate comparison between the predistorters. Finally, the PAE values were calculated to show how the efficiency of the PA can be increased by using the aforementioned predistortion structures.

	PAE (%)	ACPR (Low, dB)	ACPR (Up, dB)
Without predistortion	8.70	-24.8	-24.4
Without predistortion, IBO	4.17	-32.9	-32.1
Memoryless predistortion	7.34	-27.5	-27.4
Hammerstein predistortion / Direct learning	6.89	-31.9	-31.1
Hammerstein predistortion / Indirect learning	6.44	-32.5	-32.4
Hammerstein predistortion / NFxLMS	6.82	-31.9	-31.3

Table 5.5: The ACPR and the PAE for the PA with and without predistortion. The excitation signal was in this case a 16-QAM.

In the Figure 5.8, corresponding to Table 5.5, the estimated PSDs are depicted to visually evaluate the performance of the predistorters in the frequency domain. Here, the PA was

excited by a root-raised-cosine filtered 16-QAM signal and the responses were measured for the different predistorters and for the case without predistortion. The worst predistorter performance was measured for the memoryless predistorter (c). Obviously, in this case, the predistorter was not able to find a proper inverse of the corresponding system model. Remarkable improvements were obtained by additionally incorporating the linear memory in the predistorter. The Hammerstein predistorters based on the direct learning (d) respectively the NFxLMS (e) share the same kind of adaptation structure, and thus, then performance is similar. However, the best results were observed for the indirect learning architecture (f). Obtaining the post-distorter parameters directly without the necessity of an explicit identification of the PA model, gave a slight improvement compared to the predistorters based on the direct learning architecture. The corresponding PAE and the ACPR values for the upper and the lower adjacent channels are listed in Table 5.5. For reference, the simple but inefficient linearization by input back-off (IBO) was also included. The IBO level was chosen such that the evaluated ACPR values matched approximately with the best Hammerstein predistorter. Then PAE was measured to be 4.17%. On the other hand, the PAE of the best Hammerstein predistorter was 6.44%, which means approximately 54% increase in the PA efficiency.

	PAE (%)	ACPR (Low, dB)	ACPR (Up, dB)
Without predistortion	6.04	-24.6	-24.90
Without predistortion, IBO	3.32	-30.4	-31.2
Memoryless predistortion	4.42	-30.3	-30.8
Hammerstein predistortion / Direct learning	4.79	-30.1	-30.2
Hammerstein predistortion / Indirect learning	4.29	-27.5	-27.4
Hammerstein predistortion / NFxLMS	4.44	-30.7	-31.0

Table 5.6: The ACPR and the PAE for the PA with and without predistortion. The excitation signal was in this case an OFDM signal with 128 subcarriers.

In the second set, the measurements were repeated for an input signal according to the OFDM signal presented in Section 5.4.2. The estimated PSDs for the input signal, the PA response without predistortion, and the PA responses using the different predistortion schemes are depicted in the Figure 5.9. Here, the improvements in the ACPR are hardly distinguishable. Thus, the performance difference between these different predistorters is more or less negligible. However, marginally the best performance was obtained for the Hammerstein predistorter utilizing the direct learning architecture based on the NFxLMS algorithm. Interestingly, with the OFDM signal, the indirect learning architecture was unable to find the correct predistorter parameters. Thus the results are worse compared to the 16-QAM case. The corresponding PAE and the ACPR values for the upper and the lower adjacent channels are listed in Table 5.6. When the ACPR values are compared between the 16-QAM case and OFDM case, a performance difference of about 2 dB can be observed. This seems to be caused by the higher peak-to-average power ratio of the OFDM signal. The rare occurance of output samples with high amplitudes causes that the PA is not very frequently excited around the saturation area. This, affects the performance of the predistorter and the system identification. Therefore, longer training

sequences would have been necessary for the measurements with an OFDM excitation signal. However, for the performed measurements the length of the training sequences was kept fixed for all cases. For the OFDM signal, the increase in the PAE compared to the IBO was approximately 33%.

Number of orthogonal basis functions	ACPR (Low, dB)	ACPR (Up, dB)
3	-28.3	-28.6
4	-31.4	-31.6
5	-32.5	-32.4
6	-32.1	-31.9

Table 5.7: Performance of the Hammerstein predistorters versus the order of the orthogonal basis, using the indirect learning architecture for the training. The excitation signal was a RRC filtered 16-QAM signal.

Finally, the performance of the Hammerstein predistorter utilizing the indirect learning architecture was evaluated for different dimensions of the orthogonal polynomial basis. The corresponding estimated PSDs are depicted in Figure 5.10. The corresponding ACPR values are listed in the Table 5.7. From Figure 5.10 and Table 5.7, it can observed that with increasing number of basis functions, the performance of the linearization improves. Already, a relatively large improvement (~ 2 dB) is achieved using a basis with four basis functions. For a higher number of basis functions, e.g., 5 or 6, the improvements concern almost exclusively spectral regions which are located further away from the band occupied by the input signal. For bases including more than five basis functions, the improvements in the ACPR were negligible. This is because the corresponding intermodulation products are located outside the measurable frequency range.

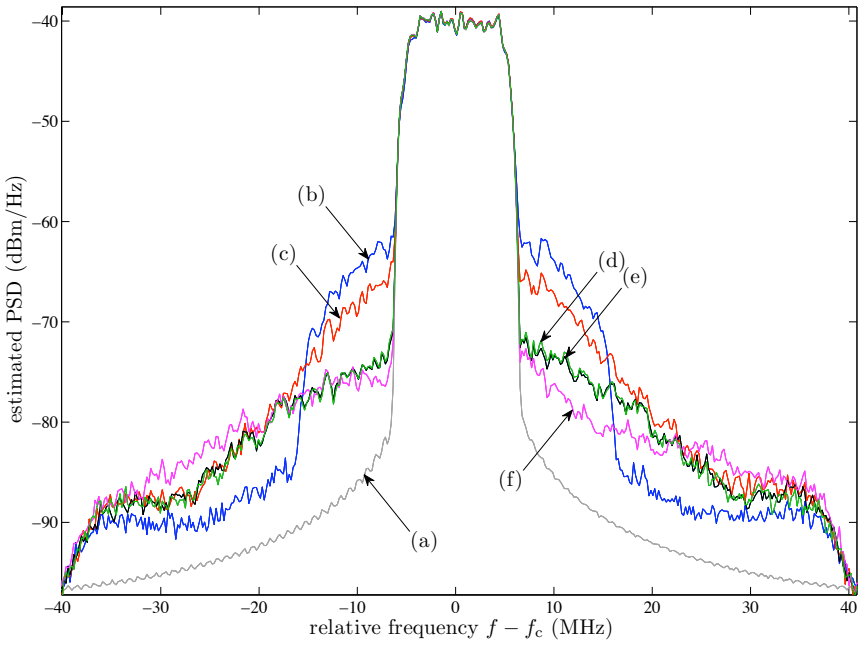

Figure 5.8: Estimated PSDs for the output of the ZVE-8G+ PA with the predistorted 16-QAM signals, generated using different predistorters. For the simplified Wiener model and the Hammerstein model a 4^{th} order FIR filter and an orthogonal basis using the first five shifted Chebyshev polynomials of the second kind were used. The PSDs were estimated for (a) the RRC filtered 16-QAM at the input (using an adequate amplification to allow the comparison to the PA response), (b) for the corresponding PA response. Moreover the PSDs were estimated for he PA response using (c) a memoryless predistorter, (d) a Hammerstein predistorter based on direct learning architecture according to Section 4.3, (e) a Hammerstein predistorter using the NFxLMS and (f) a Hammerstein predistorter based on the indirect learning architecture.

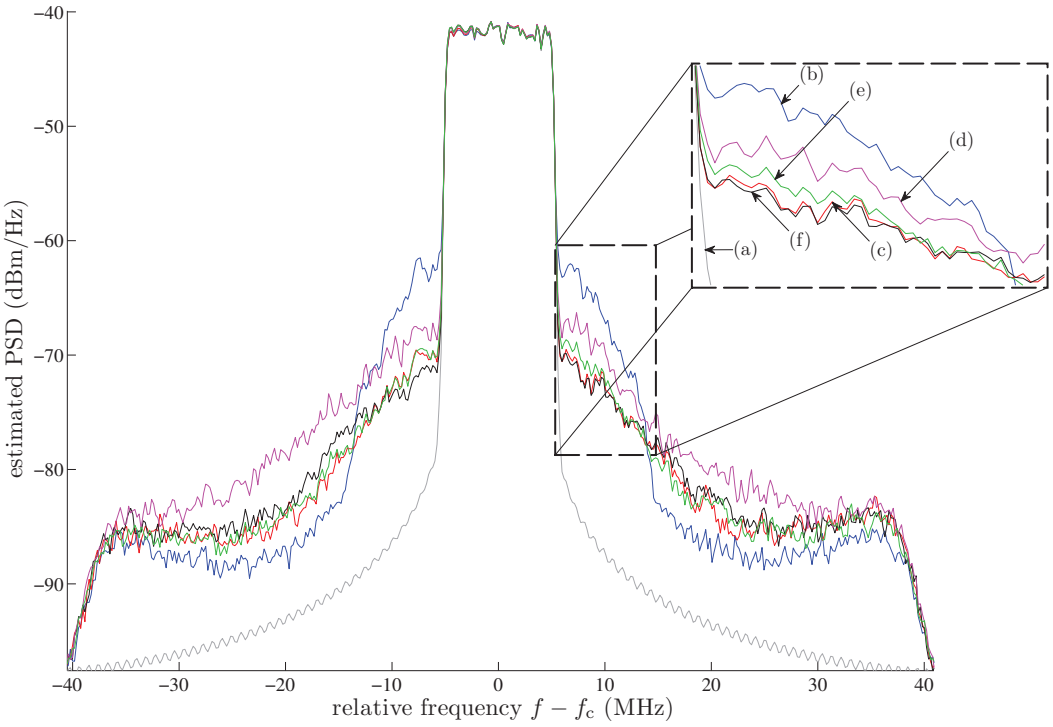

Figure 5.9: Estimated PSDs for the output of the ZVE-8G+ PA with the predistorted OFDM signals, generated using different predistorters. For the simplified Wiener model and the Hammerstein model a 4^{th} order FIR filter and an orthogonal basis using the first five shifted Chebyshev polynomials of the second kind were used. The PSDs were estimated for (a) the OFDM at the input (using an adequate amplification to allow the comparison to the PA response), (b) for the corresponding PA response. Moreover the PSDs were estimated for he PA response using (c) a memoryless predistorter, (d) a Hammerstein predistorter based on direct learning architecture according to Section 4.3, (e) a Hammerstein predistorter using the NFxLMS and (f) a Hammerstein predistorter based on the indirect learning architecture.

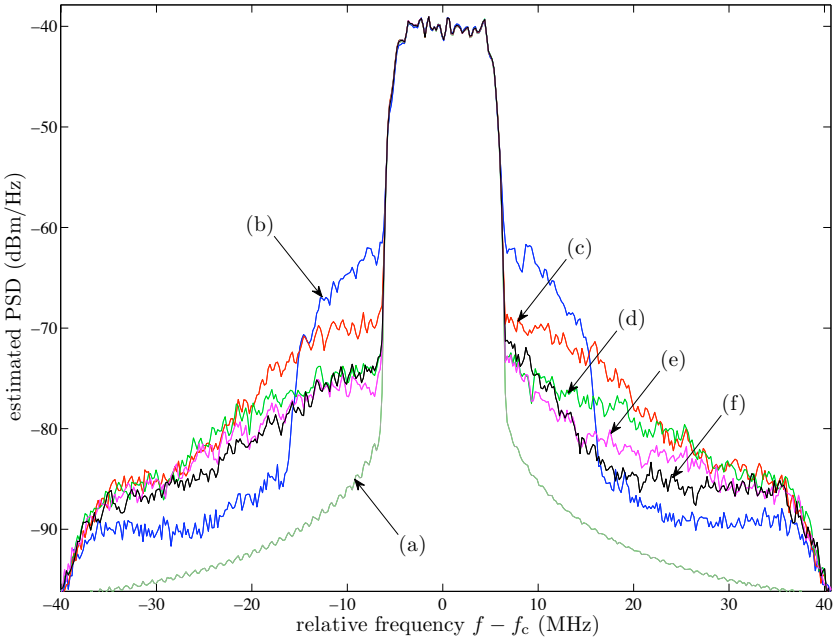

Figure 5.10: Estimated PSDs for the output of the ZVE-8G+ PA with the predistorted 16-QAM signals. The performance of the Hammerstein predistorter was observed for the different dimensions of the orthogonal basis. The indirect learning architecture was used to train the predistorter. The Hammerstein structure consists of a 4^{th} order FIR filter and a orthogonal basis using the shifted Chebyshev polynomials of the second kind with (c) 3, (d) 4, (e) 5, (f) 6 basis functions. Additionally, the PSDs were estimated for (a) the RRC filtered 16-QAM input signal, and (b) the corresponding PA response.

Chapter 6

Conclusions

In this thesis an analysis was performed for adaptive simplified Wiener models which employ an orthogonal basis expansion to represent the nonlinear sub-system. Three different orthogonal polynomial bases lead to superior performance if used with complex valued baseband signals. These three bases were selected for further investigation. The performance of the basis expansions was observed based on their numerical stability. Furthermore, it was determined, which modelling accuracies and convergence speeds were achieveable with these three orthogonal bases if applied to an adaptive simplified Wiener model respectively an adaptive Hammerstein model. The obtained results showed that the use of the presented orthogonal polynomials leads to better numerical stability and faster convergence speeds compared to a model using general non-orthogonal polynomials. This is valid, even if the amplitude distribution of the input sequence does not perfectly fir the kernel for which the polynomials are designed to be orthogonal. On average, the shifted Chebyshev polynomials of the second kind and the shifted Legendre polynomials lead to the best results.

The applicability of afore designed and evaluated models was then analyzed by including them into different adaptive predistorter structures. The predistorters based on the direct learning architecture require the identification of a precise model of the PA. Here, the simplified Wiener model based on the orthogonal polynomials was used. For the direct learning architectures, as well as, for the indirect learning architecture, the predistorter was implemented as a Hammerstein model with orthogonal polynomials. The obtained results indicate that the used models and learning architectures can be adequately used for adaptive predistorter architectures.

Finally, a testbed with a commercial PA was used to evaluate the practical performance of the predistorter structures. For QAM signals and OFDM signals, the predistorters equipped with orthogonal polynomials effectively compensated most of the detrimental effects of the PA and lead to a higher overall efficiency. The simulations also indicated how sensitive the adaptive predistorter algorithms are to the choice of the parameter settings.

Open questions for future work are:

- What are the optimum step sizes, filter lengths and polynomial orders leading to the best performance and can they be determined analytically?
- How to automatically choose the optimal equivalent linear gain used for the predistorter training?

Further investigations concerning the predistortion algorithms with the orthogonal polynomials could focus on the stability analysis of the presented algorithms and more complex predistorter architectures. It would be interesting to study the behaviour of the orthogonal polynomials for example with the Wiener-Hammerstein model or with the memory polynomial structure.

Bibliography

[1] S. Benedetto and E. Biglieri, *Principles of digital transmission: with wireless applications*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[2] J. Vuolevi, T. Rahkonen, and J. Manninen, "Measurement technique for characterizing memory effects in RF power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, pp. 1383–1389, Aug. 2001.

[3] S. C. Cripps, *RF power amplifiers for wireless communications*. Norwood, MA, USA: Artech House, 2006.

[4] D. Kimball, P. Draxler, J. Jeong, C. Hsia, S. Lanfranco, W. Nagy, K. Linthicum, L. Larson, and P. Asbeck, "50% PAE WCDMA basestation amplifier implemented with GaN HFETs," in *Proceedings of the IEEE Compound Semiconductor Integrated Circuit Symposium*, pp. 1–4, Oct. 2005.

[5] "Characteristics of E-pHEMT vs HBTs for PA applications," tech. rep., AVAGO Technologies, 2007.

[6] F. Wang, A. Ojo, D. Kimball, P. Asbeck, and L. Larson, "Envelope tracking power amplifier with pre-distortion linearization for WLAN 802.11g," in *Proceedings of the IEEE Microwave Symposium Digest*, vol. 3, pp. 1543–1546, Jun. 2004.

[7] O'Droma, J. Portilla, E. Bertran, S. Donati, T. Brazil, M. Rupp, and R. Quay, "Linearisation issues in microwave amplifiers," in *Proceedings of the European Microwave Week, 12th GAAS International Symposium*, (Amsterdam), pp. 199–202, Oct. 2004.

[8] J. Vuolevi and T. Rahkonen, *Distortion in RF power amplifiers*. Norwood, MA, USA: Artech House, 2003.

[9] P. B. Kenington, *High linearity RF amplifier design*. Norwood, MA, USA: Artech House, Inc., 2000.

[10] D. Schreurs, M. O'Droma, A. A. Goacher, and M. Gadringer, *RF Power Amplifier Behavioral Modeling*. Cambridge University Press, 2009.

[11] T. Ogunfunmi, *Adaptive nonlinear system identification: The Volterra and Wiener model approaches*. New York, NY, USA: Springer, 2007.

[12] "RF transmitting transistor and power amplifier fundamentals," tech. rep., Philips Semiconductions, 1998.

[13] B. Berglund, J. Johansson, and T. Lejon, "High efficiency power amplifiers," tech. rep., Ericsson, 2006.

[14] T. Yamamoto, T. Kitahara, and S. Hiura, "50% drain efficiency doherty amplifier with optimized power range for W-CDMA signal," in *Proceedings of Microwave Symposium, IEEE/MTT-S International*, pp. 1263 –1266, Jun. 2007.

[15] R. Raich and G. Zhou, "On the modeling of memory nonlinear effects of power amplifiers for communication applications," in *Proceedings of 10th IEEE Digital Signal Processing Workshop and the 2nd Signal Processing Education Workshop.*, pp. 7–10, Oct. 2002.

[16] K. Gard, H. Gutierrez, and M. Steer, "Characterization of spectral regrowth in microwave amplifiers based on the nonlinear transformation of a complex Gaussian process," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, pp. 1059–1069, Jul. 1999.

[17] S. Stapleton and F. Costescu, "An adaptive predistorter for a power amplifier based on adjacent channel emissions," *IEEE Transactions on Vehicular Technology*, vol. 41, pp. 49–56, Feb. 1992.

[18] E. W. Cheney, *Introduction to approximation theory, 2nd ed.* Providence, RI, USA: AMS Bookstore, 1998.

[19] L. Ding and G. T. Zhou, "Effects of even-order nonlinear terms on power amplifier modeling and predistortion linearization," *IEEE Transactions on vehicular technology*, vol. 53, pp. 156–162, Jan. 2004.

[20] O. Schmitz, S. K. Hampel, H. Eul, and D. Schwingshackl, "Considering even-order terms in stochastic nonlinear system modeling with respect to broadband data communication," *Advances in Radio Science*, vol. 5, pp. 265–271, Jun. 2007.

[21] R. Raich, H. Qian, and G. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 1468–1479, Sep. 2004.

[22] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986.

[23] G. Szegö, *Orthogonal Polynomials*. Providence, RI, USA: American Mathematical Society, 1975.

[24] W. Gautschi, *Orthogonal polynomials, computation and approximation*. New York, NY, USA: Oxford University Press, 2004.

[25] G. T. Zhou and R. Raich, "Spectral analysis of polynomial nonlinearity with applications to RF power amplifiers," *EURASIP Journal of Applied Signal Processing*, vol. 2004, pp. 1831–1840, 2004.

[26] W. McGee, "Complex Gaussian noise moments," *IEEE Transactions on Information Theory*, vol. 17, pp. 149–157, Mar. 1971.

[27] A. Ghanmi, "A class of generalized complex Hermite polynomials," *Journal of Mathematical Analysis and Applications*, vol. 340, no. 2, pp. 1395 – 1406, 2008.

[28] R. Raich and G. Zhou, "Orthogonal polynomials for complex Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2788–2797, Oct. 2004.

[29] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. New York, NY, USA: Dover, 1964.

[30] C. D. Meyer, *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: SIAM, 2000.

[31] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. Boca Raton, FL, USA: CRC Press, 2003.

[32] J. F. Monahan, *Numerical methods of statistics*. New York, NY, USA: Cambridge University Press, 2001.

[33] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Transactions on Communications*, vol. 29, pp. 1715–1720, Nov. 1981.

[34] A. Cann, "Nonlinearity model with variable knee sharpness," *IEEE Transactions on Aerospace Electronic Systems*, vol. 16, pp. 874–877, Nov. 1980.

[35] C. Rapp, "Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting system," in *Proceedings of the Second European Conference on Satellite Communications, Liege, Belgium*, pp. 179–184, Oct. 1991.

[36] M. Ghorbani, A. Sheikhan, "The effect of solid state power amplifiers (SSPAs) nonlinearities on MPSK and M-QAM signal transmission," in *Proceedings of 6th International Conference on Digital Processing of Signals in Communications*, pp. 193–197, 1991.

[37] J. Mathews and G. Sicuranza, *Polynomial signal processing*. New york, NY, USA: Wiley, 2000.

[38] M. Schetzen, "Nonlinear system modeling based on the Wiener theory," *Proceedings of the IEEE*, vol. 69, pp. 1557–1573, Dec. 1981.

[39] N. Wiener, *Nonlinear problems in random theory*. Cambridge, MA, USA: Technology Press of Massachusetts Institute of Technology, 1958.

[40] W. J. Rugh, *Nonlinear system theory: The Volterra/Wiener approach*. Baltimore, MD, USA: Johns Hopkins University Press, 1981.

[41] S. K. K. Mitra, *Digital signal processing: a computer-based approach*. Columbus, OH, USA: McGraw-Hill Higher Education, 2000.

[42] K. Narendra and P. Gallman, "An iterative method for the identification of nonlinear systems using a Hammerstein model," *IEEE Transactions on Automatic Control*, vol. 11, pp. 546–550, Jul. 1966.

[43] L. Ding, R. Raich, and G. Zhou, "A Hammerstein predistortion linearization design based on the indirect learning architecture," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2689–2692, 2002.

[44] H. W. Kang, Y. S. Cho, and D. H. Youn, "Adaptive precompensation of Wiener systems," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2825–2829, Oct. 1998.

[45] A. Hagenblad, *Aspects of the Identification of Wiener Models*. PhD thesis, University of Lingköping, Sweden, 1999.

[46] G. Pajunen, "Adaptive control of Wiener type nonlinear systems," *Automatica*, vol. 28, no. 4, pp. 781–785, 1992.

[47] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *IRE WESCON Convention Record*, vol. 4, pp. 96–104, Aug. 1960.

[48] M. H. Hayes, *Statistical digital signal processing and modeling*. Hoboken, NJ, USA: John Wiley & Sons, inc., 1996.

[49] A. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, inc., 2003.

[50] N. Bershad, P. Celka, and J.-M. Vesin, "Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output measurements," *IEEE Transactions on Signal Processing*, vol. 47, pp. 675–689, Mar. 1999.

[51] J. Tsimbinos, *Identification and compensation of nonlinear distortion*. PhD thesis, University of South Australia, Feb. 1995.

[52] B. Widrow and S. D. Stearns, *Adaptive signal processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

[53] B. Widrow, "Adaptive inverse control," in *Proceedings of the 2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*, 1986.

[54] B. Widrow and E. Walach, *Adaptive inverse control: a signal processing approach IEEE Press series on power engineering, 2nd ed.* Wiley-IEEE, 1995.

[55] B. Widrow, D. Shur, and S. Shaffer, "On adaptive inverse control," *Record of the Fifteenth Asilomar Conference on Circuits, Systems and Computers*, pp. 185–189, Nov. 1981.

[56] K. Håkansson, "The filtered-x LMS algorithm," *Lecture notes of the course Adaptive Signal Processing at the University of Karlskrona, Sweden*, 2004.

[57] S. Elliott and P. Nelson, "Active noise control," *Signal Processing Magazine, IEEE*, vol. 10, pp. 12–35, Oct. 1993.

[58] D. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 457–461, Apr. 1980.

[59] H. W. Kang, Y. S. Cho, and D. H. Youn, "On compensating nonlinear distortions of an OFDM system using an efficient adaptive predistorter," *IEEE Transactions on Communications*, vol. 47, pp. 522–526, Apr. 1999.

[60] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1726–1730, Jun. 1998.

[61] E. Abd-Elrady, L. Gan, and G. K. Ove, "Direct and indirect learning methods for adaptive predistortion of IIR Hammerstein systems," *Elektrotechnik & Informationstechnik*, vol. 125, pp. 126–131, Mar. 2008.

[62] D. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Transactions on Signal Processing*, vol. 55, pp. 120–133, Jan. 2007.

[63] M. O'Droma, E. Bertran, M. Gadringer, S. Donati, A. Zhu, P. Gilabert, and J. Portilla, "Developments in predistortion and feedforward adaptive power amplifier linearisers," in *Gallium Arsenide and Other Semiconductor Application Symposium, 2005. EGAAS 2005. European*, pp. 337 – 340, oct. 2005.

[64] C. Eun and E. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, pp. 223–227, Jan. 1997.

[65] P. L. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *Proceedings of the Asia-Pacific Microwave Conference*, vol. 2, pp. 1191–1194, 2005.

[66] R. Marsalek, P. Jardin, and G. Baudoin, "From post-distortion to pre-distortion for power amplifiers linearization," *IEEE Communications Letters*, vol. 7, pp. 308–310, Jul. 2003.

[67] S. Choi, E.-R. Jeong, and Y. Lee, "A direct learning structure for adaptive polynomial-based predistortion for power amplifier linearization," in *Proceedings of the 65th IEEE Vehicular Technology Conference*, pp. 1791–1795, Apr. 2007.

[68] S. Chang and E. Powers, "A simplified predistorter for compensation of nonlinear distortion in OFDM systems," in *Proceedings of the Global Telecommunications Conference*, vol. 5, pp. 3080–3084, 2001.

[69] Y. Qian and T. Yao, "Structure for adaptive predistortion suitable for efficient adaptive algorithm application," *Electronics Letters*, vol. 38, pp. 1282 – 1283, Oct. 2002.

[70] "ETSI-3GPP TS 25.106 version 9.0.0: 3rd generation partnership project; technical specification group radio access network; UTRA repeater radio transmission and reception (release 9)," tech. rep., 2009.

[71] E. Aschbacher and M. Rupp, "Modelling and identification of a nonlinear power-amplifier with memory for nonlinear digital adaptive pre-distortion," in *Proceedings of the 4th IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications, Rome*, pp. 658–662, Jun. 2003.

[72] D. Falconer, T. Kolze, Y. Leiba, and J. Liebetreu, "Proposed system impairment models," *IEEE 802.16 Broadband Wireless Access Working Group*, 2000.

[73] Mini-Circuits, "ZVE-8G+ PA," tech. rep., datasheet, 2010.

[74] C. M. Bender and E. Ben-Naim, "Nonlinear-integral-equation construction of orthogonal polynomials," *Journal of Nonlinear Mathematical Physics*, vol. 15, no. 3, pp. 73–80, 2008.

[75] Mathworks, "Symbolic math toolbox 5," tech. rep., datasheet, 2009.

Appendix A

Orthogonalization of the conventional baseband model

The Gram-Schmidt orthogonalization procedure can be used to develop an orthogonal basis based on arbitrary non-orthogonal basis which spans the whole space [30]. Here, the procedure is applied for the non-orthogonal polynomials of the conventional baseband model in (2.7) in order to obtain the corresponding basis of orthogonal polynomials. Let $\varphi_n(x)$ denote the n^{th} order polynomial of the conventional baseband model (2.7). Then the non-orthogonal basis can be written as

$$\begin{aligned} \varphi_1(x) &= x, \\ \varphi_2(x) &= |x|\,x + x, \\ \varphi_3(x) &= |x|^2\,x + |x|\,x + x, \\ &\vdots \\ \varphi_N(x) &= |x|^{N-1}\,x + \ldots + x. \end{aligned} \tag{A.1}$$

According to the Gram-Schmidt procedure, the polynomials $\psi_{k+1}(x)$ of the orthogonal basis can be constructed iteratively as follows

$$\psi_{k+1}(x) = \varphi_{k+1}(x) - \frac{\sum_{n=1}^{k} E\left[\psi_n^*(x)\,\varphi_{k+1}(x)\right]}{\sum_{n=1}^{k} E\left[\psi_n^*(x)\,\varphi_n(x)\right]}. \tag{A.2}$$

The expected value of the polynomial product is defined as

$$E\left[\psi_n^*(x)\,\varphi_{k+1}(x)\right] = \int_b^a \psi_n^*(x)\,\varphi_{k+1}(x)\,w(x)\,dx, \tag{A.3}$$

where $w(x)$ denotes the probability density function and $[a, b]$ the range of orthogonality.

Another way to perform the Gram-Schmidt procedure is to use a moment matrix representation as introduced in [74]. In the moment representation, two square matrices $\mathbf{A}_k$

and $\mathbf{B}_k$ are defined as follows

$$\begin{aligned}
\mathbf{A}_k &= \begin{bmatrix} x & |x|\,x & \ldots, & |x|^{k-1}\,x \\ m_3 & m_4 & \ldots, & m_{k+3} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k+2} & m_{k+3} & \cdots & m_{2k+2} \end{bmatrix}, \\
\mathbf{B}_k &= \begin{bmatrix} m_2 & m_3 & \ldots, & m_{k+2} \\ m_3 & m_4 & \ldots, & m_{k+3} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k+2} & m_{k+3} & \cdots & m_{2k+2} \end{bmatrix},
\end{aligned} \tag{A.4}$$

where

$$m_k = \int_b^a |x|^k\, w\,(x)\, dx. \tag{A.5}$$

The orthogonal basis function $\psi_k\,(x)$ of the orthogonal baseband model can then be obtained as follows

$$\psi_k\,(x) = \frac{|\mathbf{A}_k|}{|\mathbf{B}_k|}. \tag{A.6}$$

Hence, the iterative orthogonalization procedure is reduced to the evaluation of two Hankel type determinants [74]. If an explicit solution is available for both determinants, a closed form solution for the coefficients of the orthogonal baseband model can be found. In most cases however, the explicit solution is not available or it is difficult to derive manually. In this case, the evaluation of the determinants can be implemented by using, for example, Matlab and Symbolic Math Toolbox [75] in order to avoid rounding errors.

Appendix B - List of devices

Abbreviation	Description	Model	Manufacturer	Operational limits	
PA	Power amplifier	ZVE-8G+	Mini-Circuits	Frequency range: Gain: Maximum input power: Maximum output power:	20 MHz . . . 6 GHz 13 dB 15 dBm 10 dBm
SMU	Vector signal generator	SMU 200A	Rhode & Schwarz	Frequency range: I/Q modulation bandwidth:	100 kHz . . . 6 GHz 200 MHz
FSQ	Vector signal analyzer	FSQ 26	Rhode & Schwarz	Frequency range: Demodulation bandwidth:	20 Hz . . . 3.6 GHz 120 MHz
PS	Power sensor	NRP-Z11	Rhode & Schwarz	Dynamic range:	-67 dBm . . . 23 dBm
DIC	Directional coupler	EMC 4224-10	EPX Microwave	Frequency range:	0.6 . . . 18.0 GHz
ATT_1	20 dB, attenuator	Inmet 6B20W-20	Aeroflex	Frequency range:	0.0 . . . 18.0 GHz
ATT_2	10 dB, attenuator	VAT-10W2+	Mini-Circuits	Frequency range:	0.0 - 6.0 GHz
DC	Triple power supply	EA-PS2316-050	EA Elektro-Automatik	DC voltage range: DC current range:	0 V . . . 16 V 0 A. . . 5 A
MM	Digital multimeter	34410A	Agilent	DC voltage range: DC current range: AC (RMS) voltange range:	100 mV . . . 1000 V 100 μA. . . 3 V 100 mV . . . 750 V
DCB	DC Block	BLK-18-S+	Mini-Circuits	Frequency range: Maximum DC voltage:	10 MHz . . . 18 GHz 50 V