Departament de Teoria
del Senyal i Comunicacions

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Ph.D. Thesis

Multi Look-Up Table Digital Predistortion for RF Power Amplifier Linearization

Author: Pere Lluis Gilabert Pinal

Advisors: Dr. Eduard Bertran Albertí
Dr. Gabriel Montoro López

Control Monitoring and Communications Group
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya

Barcelona, December 2007

Chapter 4

Power Amplifier Behavioral Models for Digital Predistortion

4.1 Introduction

The digital baseband predistortion linearization technique is independent on the final frequency band of operation and avoids the complexity of the RF hardware, every time operating at higher frequencies. This makes the DPD linearizer more versatile and it is also a huge field of research for engineers for optimizing DSP architectures and algorithms. However, the cancelation performance of DPD can be seriously degraded due to its sensitivity to memory effects generated in RF power amplifiers. As we have already explained in Chapter 2, the envelope filtering technique is a memory effects compensation technique aimed at reproducing the inverse memory effects that are generated inside the PA.

Therefore, when designing digital baseband adaptive predistorters at least three major issues have to be taken into account:

- the need for accurate PA behavioral models capable to achieve the demanded requirements (in terms of nonlinear memory effects reproduction).
- an efficient model inversion procedure for the identification and adaptation of the baseband digital predistorter.
- an efficient implementation of the DPD function in a digital processing device (DSPs or FPGAs) without an excessive computational cost.

The first topic is discussed in this Chapter, while the second and third one will be covered in Chapter 5 of this thesis.

PA models can be classified according to the type of data needed for their extraction in

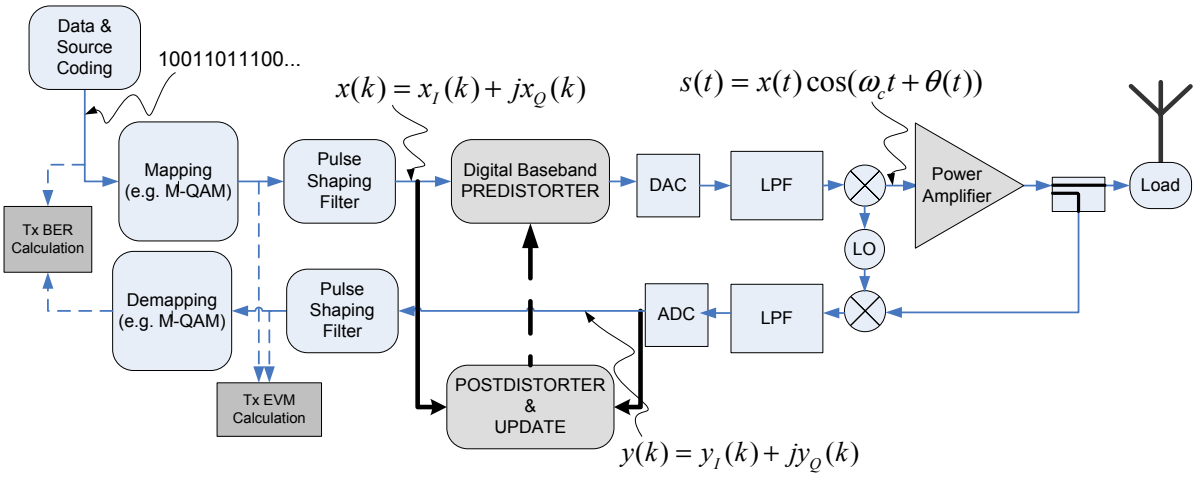

Figure 4.1: Block diagram of a transmitter with digital baseband predistortion.

physical and *empirical* models [Ped05]:

- *Physical models* require the knowledge of the electronic elements that comprise the PA, their constitutive relations and the theoretical rules describing their interactions. Physical models make use of a nonlinear model of the PA active device and other passive components to form a set of nonlinear equations relating terminal voltages and currents. The equivalent circuit description of the PA, whose topology is usually derived from the direct inspection of the real PA, permits accurate results very suitable for *circuit-level* simulation. However, their elevated simulation time and the need for a detailed description of the PA internal structure make them unsuitable for system level linearization purposes.

- *Empirical or behavioral models* do not need an *a priori* knowledge of the PA internal composition (very useful for traveling-wave-tube (TWT) PAs for example), for that reason they are also known as *black-box models*. Their extraction relies on a set of input-output observations. Therefore their accuracy is highly sensitive to the adopted model structure and the parameter extraction procedure.

PA behavioral (or black-box) models at system level are single input single output (SISO) systems. The extraction of PA behavioral model for DPD linearization purposes is carried out by means of input and output *complex envelope* signal observations. Fig. 4.1 shows a simplified transmitter block diagram presenting digital baseband predistortion linearization. We can observe the different type of signals (band-pass, baseband, analogue, digital) present after some of the transmitter subsystem blocks. A band-pass modulation can be defined as the process whereby the amplitude, frequency or phase of an RF carrier, or a combination of them, is varied in accordance with the information to be transmitted. An amplitude and phase modulated

band-pass signal can be described as:

$$s(t) = A(t)\cos\left(\omega_c t + \theta(t)\right) \tag{4.1}$$

where $\omega_c = 2\pi f_c$ is the angular carrier frequency, $A(t)$ and $\theta(t)$ are the time-varying amplitude and phase, respectively. This *band-pass signal* can also be expressed in a polar form as,

$$s(t) = Re\left\{x(t)e^{j\omega_c t}\right\} \tag{4.2}$$

with $x(t)$ being the *complex envelope* defined as,

$$x(t) = A(t)e^{j\theta(t)} = A(t)\cos\left(\theta(t)\right) + jA(t)\sin\left(\theta(t)\right) = x_I(t) + jx_Q(t) \tag{4.3}$$

where $x_I(t)$ and $x_Q(t)$ are the in-phase (I) and quadrature (Q) components of the *complex envelope*, respectively. Using trigonometric identities the *band-pass signal* can be also expressed by means of their Cartesian I and Q components:

$$s(t) = Re\left\{\left(x_I(t) + jx_I(t)\right)\left(\cos\left(\omega_c t\right) + j\sin\left(\omega_c t\right)\right)\right\} = x_I(t)\cos\left(\omega_c t\right) - x_Q(t)\sin\left(\omega_c t\right) \tag{4.4}$$

Therefore, considering the complex envelope $x(t)$ as a baseband signal described by a sequence of ideal pulses $x_I(t)$ and $x_Q(t)$ appearing at discrete times (k=1,2,3...) in a digital baseband environment, from (4.3) it is possible to express the complex envelope in the discrete form as,

$$x(t)|_{t=kT_s} = x_I(kT_s) + jx_Q(kT_s) \cong x_I(k) + jx_Q(k) \tag{4.5}$$

Thus, in order to extract the PA behavioral models that will permit to design the digital predistorter at baseband, it is necessary to have input ($x(k)$) and output ($y(k)$) discrete complex envelope signals, as it is schematically shown in Fig. 4.1.

It is not the scope of this thesis to provide an extended overview of the existing PA behavioral models, which can be found in [Ped05] and [Isa06]. However, since digital predistortion nonlinear cancelation performance depend on the accuracy of these models to reproduce PA nonlinear and memory effects (envelope filtering technique), some of the most significant PA models, in terms of digital predistortion implementation, will be here presented and discussed to finally present our specific contribution of a PA dynamic model capable to cope with digital predistortion linearization.

4.2 Memoryless Power Amplifier Behavioral Models

In memoryless models, the output signal is assumed to be a nonlinear function of the instantaneous input signal only. The envelope frequency of the input signal is much smaller than the envelope bandwidth of the amplifier. Consequently, the amount of amplitude and phase distortion depends only on the input signal level at the corresponding respective time instant.

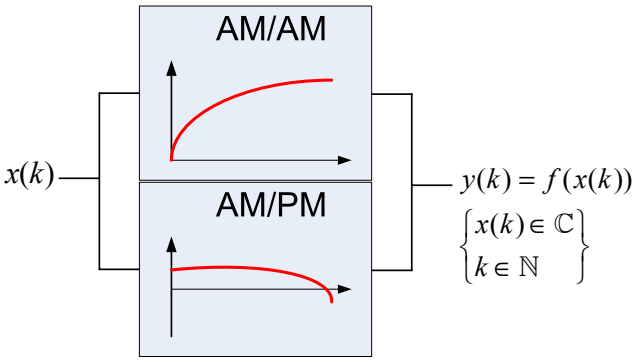

Figure 4.2: Memoryless behavioral model.

4.2.1 Saleh Model

Saleh proposed two general functions to approximate the AM/AM and AM/PM envelope characteristics, initially applied to traveling wave tube (TWT) amplifier models [Sal81], but later also used to characterize solid state power amplifier (SSPA) ones. The polar representation corresponds directly to AM/AM and AM/PM characterization and can be expressed as:

$$f_A(|x(k)|) = \frac{\alpha_A |x(k)|}{1 + \beta_A |x(k)|^2} \tag{4.6}$$

$$f_\phi(|x(k)|) = \frac{\alpha_\phi |x(k)|^2}{1 + \beta_\phi |x(k)|^2} \tag{4.7}$$

where α_A, β_A, α_ϕ, β_ϕ are constant parameters chosen to approximate the real PA characteristics.

4.2.2 Memoryless Polynomial Model

Together with Saleh's models the memoryless polynomial model is one the most known and used models to describe the PA nonlinear static behavior. A general form of p^{th} order power series approximation can be expressed by:

$$y(k) = \sum_{p=0}^{P} \gamma_p \cdot x(k) \cdot |x(k)|^p \tag{4.8}$$

where $x(k)$ and $y(k)$ are the PA input and output complex envelope at sample k, P the order of the polynomial function and γ_p complex coefficients that have to be identified in order to characterize the PA memoryless nonlinear behavior.

4.2.3 Fourier Series Model

In the Fourier series model, the output signal may be expressed as a complex Fourier series expansion of the instantaneous input signal [TAR06]. The general expressions of input-output complex voltage signals of a memoryless nonlinear PA may be expressed by:

$$s_0(t) = G\left(s_i(t)\right) \tag{4.9}$$

where $s_i(t)$ and $s_0(t)$ are the instantaneous input and output signals respectively. The function G can be expanded by any suitable complex series and simplified by introducing the effects of physical constraints.

Assuming the instantaneous voltage transfer characteristic can be represented by a complex Fourier series expansion of its periodic extension where the period is D, defined by the maximum dynamic range of the input signal $s_i(t)$, or the maximum dynamic range for which it is desired to model the PA, then the output can be written as:

$$s_0(t) = \sum_{k=-\infty}^{\infty} c_k e^{j\alpha k s_i(t)} \tag{4.10}$$

where α is defined by the dynamic range, $D = \frac{2\pi}{\alpha}$ and c_k are the coefficients of the Fourier series:

$$c_k = \frac{1}{D} \int_a^b G(x) e^{j\alpha k x} dx \tag{4.11}$$

4.2.4 Bessel Functions Memoryless Polynomial Model

The memoryless complex Fourier series expansion of the instantaneous voltage transfer characteristics is the basis for the derivation of the memoryless complex Bessel Function series model of PA envelope characteristics [TAR06]. Applying complex Bessel series to the instantaneous complex Fourier series model, it is possible to derive a highly computable decomposed Bessel function based model suitable for both large and small signal behavioral modeling [Fue73, O'D89].

Let us consider a multi-carrier signal:

$$s_i(t) = \sum_{m=1}^{N} A_m(t) \cos\left(\omega_m t + \theta_m(t)\right) \tag{4.12}$$

where ω_m is the angular frequency, $A_m(t)$ the amplitude and $\theta_m(t)$ the phase of an individual carrier. Introducing equation (4.12) into equation (4.10), the output of a nonlinear device can be written as:

$$s_0(t) = \sum_{k=-\infty}^{\infty} c_k e^{j\alpha k \sum_{m=1}^{N} A_m(t)\cos(\omega_m t+\theta_m(t))} = \sum_{k=-\infty}^{\infty} c_k \prod_{m=1}^{N} e^{j\alpha k A_m(t)\cos(\omega_m t+\theta_m(t))} \tag{4.13}$$

Applying the Bessel function series expansion:

$$e^{jx\cos\theta} = \sum_{n=-\infty}^{\infty} j^n J_n(x) e^{jn\theta} \tag{4.14}$$

where $J_n(x)$ are the Bessel functions of the first kind, then it is possible to show that:

$$s_0(t) = Re\left\{ \sum_{k=-\infty}^{\infty} jc_k \sum_{n_1,n_2\ldots,n_N=-\infty}^{\infty} \left((-j) \prod_{m=1}^{N} \left[J_n\left(\alpha k A_m(t)\right) (j)^{n_m} \right] \right) e^{j\sum_{m=1}^{N} n_m(\omega_m t+\theta_m(t))} \right\} \tag{4.15}$$

The expression (4.15) is a 'decomposed' one, and it shows that the output resolved into the fundamental and all harmonics and IMD products. Their angular frequencies, ω_j, are $\omega_j = \sum_{m=1}^{N} n_m \cdot \omega_m$ for all integer values of n_m in the range from $-\infty$ to $+\infty$.

Setting the condition $\sum_{m=1}^{N} n_m = 1$, yields the zonal components of output signal, i.e. those in the output band corresponding to the input signal band:

$$s_0(t) = Re\left\{ \sum_{k=-\infty}^{\infty} jb_k \sum_{n_1,n_2\ldots,n_N=-\infty}^{\infty} \left(\prod_{m=1}^{N} \left[J_{nm}\left(\alpha k A_m(t)\right) \right] \right) e^{j\sum_{m=1}^{N} n_m(\omega_m t+\theta_m(t))} \right\} \tag{4.16}$$

where $b_k = j\left(c_k - c_{-k}\right)$.

The single unmodulated tone conditions are satisfied by setting N = 1, $\omega_m = \omega$, $A_m(t) = A$ and $\theta_m(t) = 0$. Equation (4.16) then simplifies to:

$$s_0(t) = \sum_{k=1}^{\infty} b_k J_1\left(\alpha k A(t)\right) e^{j\omega t} \tag{4.17}$$

From (4.17), the coefficients b_k may be derived from measurements, through a least squares extrapolation procedure.

4.3 Power Amplifier Behavioral Models with Memory Effects

As we have discussed in Chapter 2, the precise gain in RF power amplifiers presenting memory effects, is not only a function of the input signal amplitude at the same instant, but also dependent on the recent history of the input-output signals as well. Coping with high speed envelope signals (presenting significant bandwidths) makes engineers reconsider the degradation suffered from memory effects. Time responses are convolved by the impulse response of the system and thus memory effects are no more irrelevant when predistortion type linearization is employed to cancel out the intermodulation sidebands.

To develop amplifier models that include memory effects, the amplifier has to be characterized using dynamic measurement systems. The most common *dynamic nonlinear models* considered

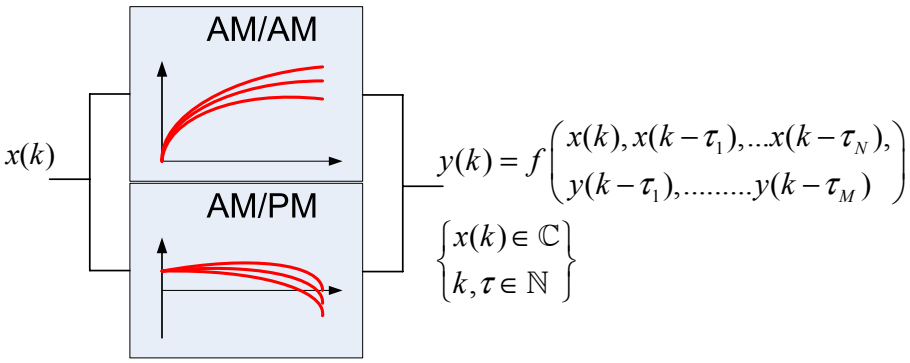

Figure 4.3: Behavioral models with memory.

in literature for characterizing the PA behavior are neural networks (NNs), Volterra series and their simplified versions. In the following, the most common dynamic structures used in PA behavioral modeling are presented and discussed, to finally introduce a *Nolinear Auto-regressive Moving Average* (NARMA) PA behavioral model that will be considered for DPD purposes.

4.3.1 Artificial Neural Networks: Time Delay Neural Networks

An *Artificial Neural Networks* (*ANNs*) is a system composed of a large number of basic elements arranged in layers and that are highly interconnected [Zha00]. In general, the structure is formed by several inputs and outputs, which may be trained to react to the inputs stimulus in a particular desired way. These systems pretend to emulate the human brain by learning how to behave in function of the former knowledge of the environment's problem.

Multi-layer Perceptrons (*MLP*) is one of the most popular structures of *ANN* models. They follow a general class of feedforward *ANN* structure, since the flow of data is from inputs to outputs, without feedback. They have the capability to produce the general approximation of any function [Sca98]. The MLP model consists of a finite number of units called *neurons* or *perceptrons*, where each unit of each layer is connected to each unit of the subsequent/previous layer. These connections are called links or synapses and they only exist between neurons in subsequent layers but connections among neurons within a layer do not exist, nor are there any direct connections across layers. As depicted in Fig. 4.4, the first layer is called input layer, it is followed by intermediate groups of neurons called hidden layers, and the result of the network is obtained in the output layer. The number of neurons in the hidden layer is usually determined by trial-and-error, to find the trade-off that provides the simplest network with acceptable performance. Since the numbers of inputs and outputs are fixed (the independent and dependent variables of the problem, respectively) the number of hidden neurons determines the number of weights that must be optimized during the training process to obtain the best

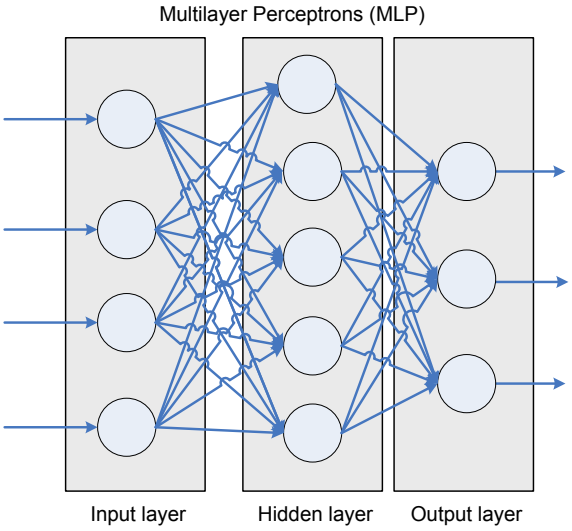

Figure 4.4: Multi layer perceptron (MLP) neural network.

function approximation.

The rules for the MLP model are the following:

- The j^{th} neuron of the k^{th} layer receives as input each x_i of the previous layer. Each value x_i is then multiplied by a weight w_{ji}, and then all the values are summed.

- A shift θ_j (called threshold o bias) is applied to the above sum, and over the results an activation function $\psi(\cdot)$ is applied, resulting in the output of the j^{th} neuron of the k^{th} layer. The MLP neuron model is shown in equation (4.18), where $i = 1 \cdots N$, $j = 1 \cdots M$.

$$y_j = \psi\left(\sum_{i=1}^{N} w_{ij} x_i \pm \theta_j\right) \tag{4.18}$$

Some common activation functions $\psi(\cdot)$ are:

- The logistic sigmoid

$$\psi(x) = \frac{1}{1 + e^{-x}} \tag{4.19}$$

- The hyperbolic tangent

$$\psi(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.20}$$

- The linear function

$$\psi(x) = x \tag{4.21}$$

Neural Networks were applied almost exclusively to model static functions of the input variables alone. Although this approach has been largely exploited for DC simulation, their application

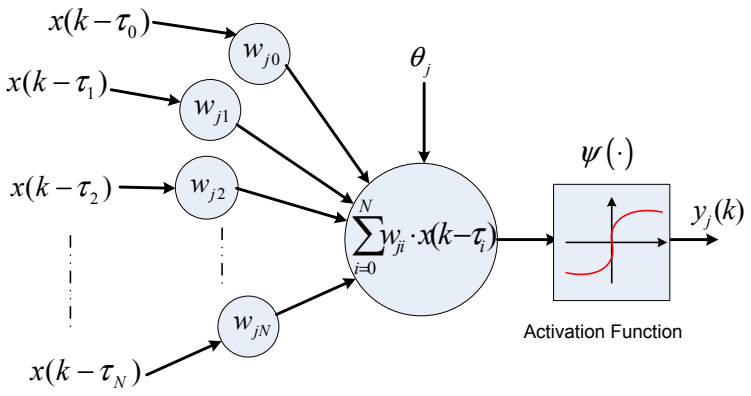

Figure 4.5: TDNN neuron model.

to dynamic system is a rather new research field. Recently have appeared models taking into account the dynamic phenomena, i.e. for power amplifier (PA) modeling [O'T00, Xu02] proposing the use of a special type of MLP model, the *time-delayed neural networks (TDNN)*. They have been successfully applied for solving the temporal processing problems in signal processing, speech recognition, system identification and control [Liu04]. A TDNN is based on the feedforward MLP NN with the addition of tapped delay lines which generate delayed samples of the input variables. They are used to add the history of the input signals to the model, needed for memory effects modeling. The TDNN entries include not only the current value of the input signal, but also its previous values, as shows the TDNN neuron model in Fig. 4.5 and described in (4.22).

$$y_j(k) = \psi\left(\sum_{i=0}^{N} w_{ij} x(k-\tau_i) \pm \theta_j\right) \tag{4.22}$$

The TDNN structure consists of two sections, namely a *Linear Time Invariant* (LTI) system and a non-linear memoryless system (MLP), as shown in Fig. 4.6. LTI provides the TDNN with the capability of performing dynamic mappings which depend on past input values, making them suitable for time series prediction, non-linear system identification, and signal processing applications. In this case, the most popular type are finite impulse response (FIR) neural networks which are obtained by replacing input with FIR filters. The memory depth of the element or system analyzed is reflected on the length of the taps. The strategy followed to set the system memory is the same adopted for the design of numerical filters, where the input tap number is imposed by the bandwidth accuracy required. This model is formed by a three-layer feedforward NN, consisting of an input layer, a hidden layer and an output layer. From the topology of TDNN, it is very clear that the time delay lines are used so as to consider the history of the input signal, which is needed for modeling systems with memory. Thus a TDNN structure can be viewed as a special case of the Wiener model, which is analogous to a FIR filter followed by a memoryless non-linear function. Moreover, a general structure of a recursive single-hidden-layer

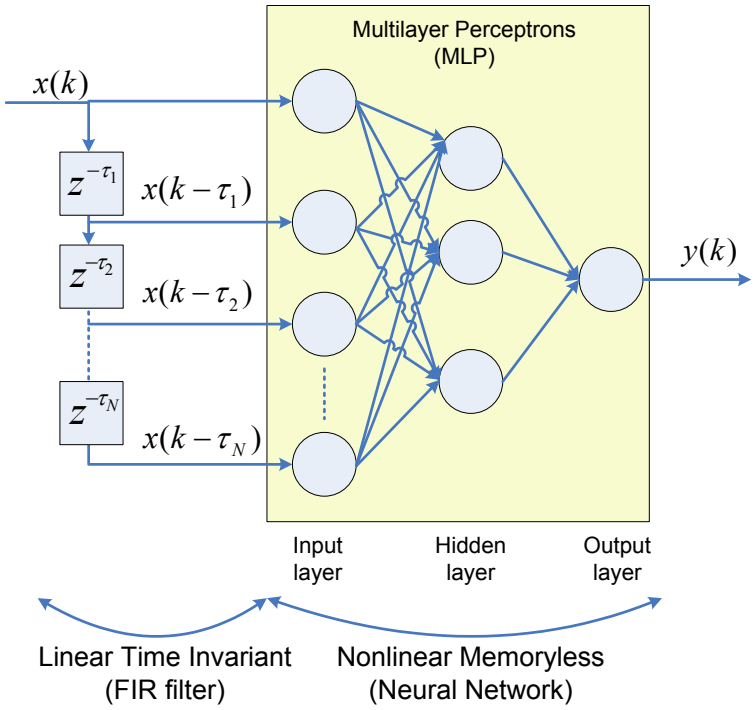

Figure 4.6: Time-delayed neural network (TDNN).

perceptron dynamic ANN for modeling PA nonlinear dynamics is presented in [Ped05], however, no results showing its performance are provided.

Comparing the NN approach with the polynomial one, contrary to the inherent local approximating properties of polynomials, NNs can behave as global approximations: a potential advantage when modeling strongly nonlinear systems. Besides, NNs, work, in principle, better than the classical models in extrapolating beyond the zone exploited for parameter extraction.

4.3.2 Volterra Series Model

The Italian mathematician Vito Volterra was the first to study the functional series [Vol59], considered after him as an extension of the Taylor series expansion [Mat00]. Volterra kernels are reliable descriptors of the system's function, covering a large number of possible system states. Thus, they can be considered to model nonlinear systems with memory such as PAs. Therefore, considering $x(k)$ and $y(k)$ as the complex envelope input-output relation in PA, the discrete-time

complex Volterra series expansion can be expressed as,

$$\begin{aligned} y(k) &= \sum_{i=0}^{M_1-1} h_1(i)\cdot x(k-i) + \\ &+ \sum_{m=0}^{M_3-1}\sum_{n=0}^{M_3-1}\sum_{p=0}^{M_3-1} h_3(m,n,p)\cdot x(k-m)\cdot x(k-n)\cdot x^*(k-p) + \\ &\vdots \\ &+ \sum_{m=0}^{M_Q-1}\sum_{n=0}^{M_Q-1}\cdots\sum_{q=0}^{M_Q-1} h_Q(m,n,\cdots q)\cdot x(k-m)\cdot x(k-n)\cdots x^*(k-q) \end{aligned} \tag{4.23}$$

where h_1, h_3,$\cdots$, h_Q are the so called low-pass equivalent Volterra kernels, k denotes discrete time and M_1, M_3,$\cdots$, M_Q are the number of delays considered in each kernel respectively.

The number of parameters in Volterra series grow exponentially when considering higher order kernel extractions, what can add an unnecessary computational complexity when considering them for identification and linearization purposes. The number of coefficients (n_c) that we have to extract when considering the discrete-time Volterra series is described by the following equation:

$$n_c = \sum_{i=1}^{\frac{N_k+1}{2}} D^{2i-1} \quad (M_1 = M_3 = \cdots = M_Q = D) \tag{4.24}$$

with N_k being the highest order kernel considered (for the sake of simplicity only odd kernels have been considered) and assuming the same memory depth for each kernel ($M_1 = M_3 = \cdots = M_Q = D$).

A common solution consists in using pruning techniques to reduce the complexity of the general Volterra series or using alternative configurations that reproduce Volterra kernels:

- ***Pruned Volterra series***. This technique consists in applying reduction methods to the general Volterra series, selecting only the most important terms and thus reducing the Least Squares Hessian used in the linear estimation process. In [Zhu07a, Zhu07b] pruned Volterra series based on the physical characteristics of the amplifier are presented. The authors report an accurate identification of the PA model by significantly reducing the number of coefficients required in the identification.

- ***Wiener-Bose model***. This technique is based on the Wiener-Bose architecture and can be used to model systems with fading memory, such as the PA [Sil07]. The main advantage of this model is that considers all possible interaction among the signals and thus obtaining good accuracy. However it demands a large number of parameters to estimate and represent the model, therefore the use of pruning techniques to reduce its complexity is advisable.

Alternatively, more simplified deviations of the general Volterra series are the so called *modular approaches*, consisting in structures of linear time-invariant filters and memoryless nonlin-

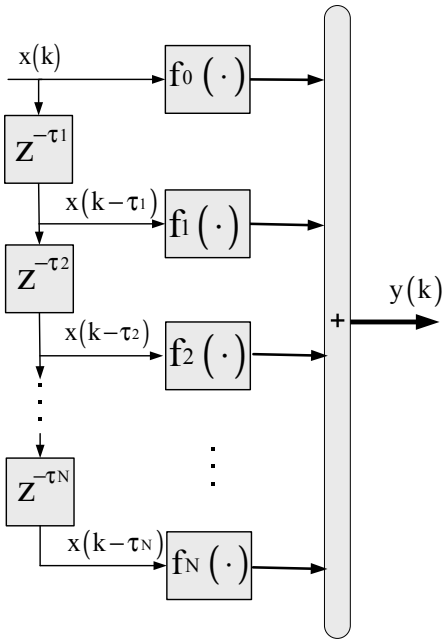

Figure 4.7: Block diagram of a Nonlinear Moving Average (memory polynomial).

earities. Some of the most important published models based in a modular approach will be discussed in the following.

Memory Polynomial: Nonlinear Moving Average Behavioral Models

Memory polynomials or nonlinear moving average (NMA) models are one of the most simple models that take into account the nonlinear dynamic behavior of a PA and have been used for predistortion applications [Kim01,Bos89]. The block scheme of a NMA model is depicted in Fig. 4.7. The input-output relation of a NMA model can be expressed as

$$y_{NMA}(k) = \sum_{n=0}^{N} f_n\left(x(k-\tau_n)\right) \tag{4.25}$$

where the nonlinear function $f_n(\cdot)$ is expressed by a polynomial function and where τ's are the most suitable delays for describing the PA model ($\tau_0 = 0$).

$$f_n\left(x(k-\tau_n)\right) = \sum_{p=0}^{P} \alpha_{pn} x(k-\tau_n)\left|x(k-\tau_n)\right|^p \tag{4.26}$$

Rewriting (4.25) in a more compact matrix notation,

$$y_{NMA}(k) = \lambda^H \mathbf{\Phi} \tag{4.27}$$

where the superindex ^{H} denotes Hermitian and where

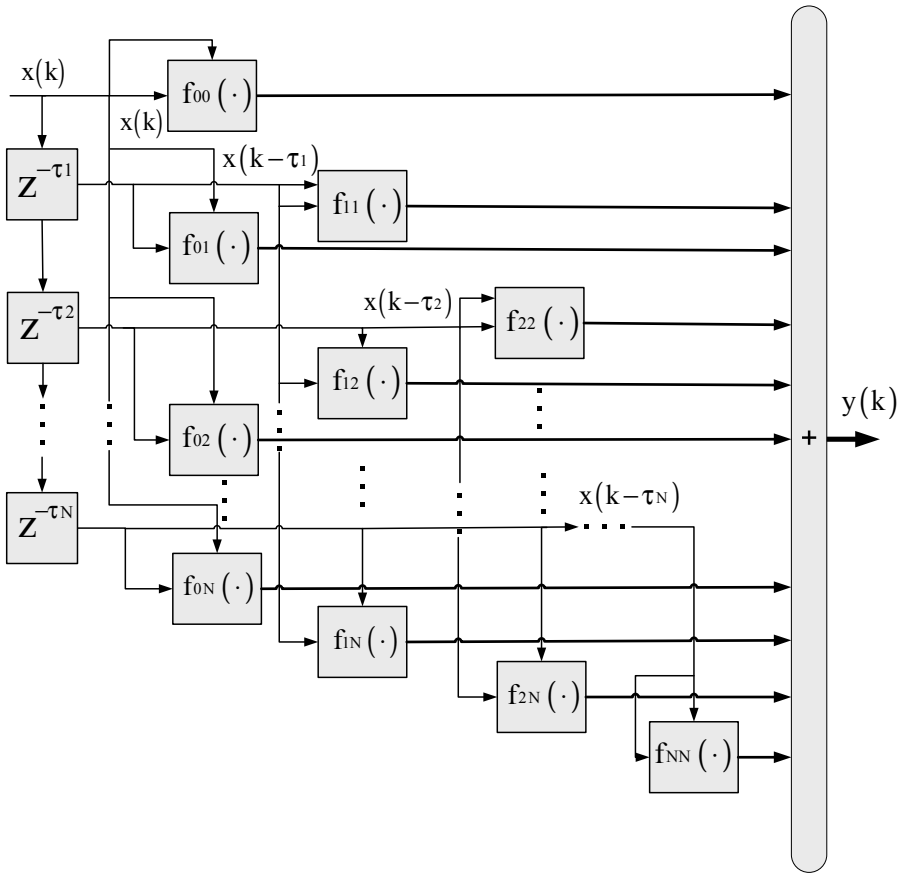

Figure 4.8: Block diagram of an Augmented Nonlinear Moving Average model.

$\lambda = [\alpha_{00},\ \alpha_{10}, ..., \alpha_{P0}, \alpha_{01}, ..., \alpha_{P1}, ..., \alpha_{0N}, ..., \alpha_{PN}]^T$ and

$$\mathbf{\Phi} = \left[\begin{array}{l} x(k),\ x(k)\,|x(k)|\,,\ \cdots, x(k)\,|x(k)|^P\,, x(k-\tau_1), \cdots, x(k-\tau_1)\,|x(k-\tau_1)|^P\,, \\ \cdots, x(k-\tau_N), \cdots, x(k-\tau_N)\,|x(k-\tau_N)|^P \end{array} \right]^T$$

The cost function to be minimized in order to extract the PA model parameters is the *Normalized Square Error* (NSE),

$$J(k) = |\bar{e}(k)|^2 = \left|\left(y(k) - \lambda^H \mathbf{\Phi}\right)/y(k)\right|^2 \tag{4.28}$$

The error is defined as the measured PA output minus the estimated output divided by the measured output.

The ***Augmented Nonlinear Moving Average*** (NMA) PA behavioral model is an extension of the NMA model that introduces p-order nonlinear mainly products of pairs of delayed samples of the input $(x(k-\tau_i), x(k-\tau_j))$ in order to improve nonlinear memory modeling. The augmented NMA model structure is depicted in Fig. 4.8 and its input-output relation can be

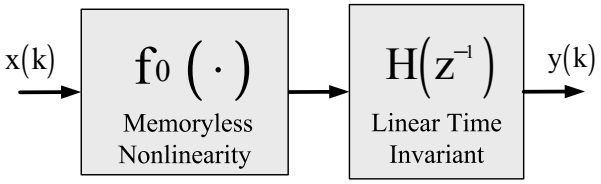

Figure 4.9: Block diagram of a Hammerstein model.

expressed as

$$y_{AugNMA}(k) = \sum_{i=0}^{N} \sum_{j=i}^{N} f_{ij}\left(x(k-\tau_i), x(k-\tau_j)\right) \tag{4.29}$$

where τ's are the most suitable delays for describing the PA model and $f_{ij}(\cdot)$ are static nonlinear functions implemented using polynomials

$$f_{ij}\left(x(k-\tau_i), x(k-\tau_j)\right) = \sum_{p=0}^{P} \alpha_{ij} x(k-\tau_i) \left|x(k-\tau_j)\right|^p \tag{4.30}$$

it is possible to express (4.29) in a more compact matrix notation,

$$y_{AugNMA}(k) = \lambda^H \mathbf{\Phi} \tag{4.31}$$

where $\lambda = [\alpha_{00}, \cdots, \alpha_{0N}, \alpha_{11}, \cdots, \alpha_{1N}, \cdots, \alpha_{(N-1)N}, \alpha_{NN}]^T$

$$\mathbf{\Phi} = \begin{bmatrix} f_{00}\big(x(k), x(k)\big), \cdots, f_{0N}\big(x(k), x(k-\tau_N)\big), \ f_{11}\big(x(k-\tau_1), x(k-\tau_1)\big), \cdots, \\ f_{1N}\big(x(k-\tau_1), x(k-\tau_N)\big), \cdots, f_{(N-1)N}\big(x(k-\tau_{N-1}), x(k-\tau_N)\big), \\ f_{NN}\big(x(k-\tau_N), x(k-\tau_N)\big) \end{bmatrix}^T \quad \text{and}$$

assuming $\tau_0 = 0$.

The cost function to be minimized in order to extract the coefficients defining the augmented NMA model is the same as the one defined in (4.28).

Hammerstein Model

Hammerstein models are composed by a memoryless nonlinearity followed by a LTI system, as it is shown in Fig. 4.9. For the sake of simplicity, the LTI block can be implemented with a FIR filter and thus,

$$y_{Hamm}(k) = \sum_{n=0}^{N} \alpha_n f_0\left(x(k-\tau_n)\right) \tag{4.32}$$

And now, assuming that the memoryless nonlinearity is implemented with polynomials,

$$y_{Hamm}(k) = \sum_{n=0}^{N} \alpha_n \sum_{p=0}^{P} \gamma_p x(k-\tau_n) \left|x(k-\tau_n)\right|^p \tag{4.33}$$

In order to rewrite (4.32) in a matrix notation that permits the extraction of the Hammerstein model coefficients it is necessary to assume that $d_{np} = \alpha_n \gamma_p$, thus the Hammerstein model output can be expressed as:

$$y_{Hamm}(k) = \lambda^H \mathbf{\Phi} \tag{4.34}$$

where $\lambda = [d_{00}, \cdots, d_{0P}, d_{10}, \cdots, d_{1P}, \cdots, d_{N0}, \cdots, d_{NP}]^T$, with $\tau_0 = 0$ and

$$\mathbf{\Phi} = \left[\begin{array}{l} x(k), \cdots, x(k)\,|x(k)|^P, x(k-\tau_1), \cdots, x(k-\tau_1)\,|x(k-\tau_1)|^P, \cdots, x(k-\tau_N), \cdots, \\ x(k-\tau_N)\,|x(k-\tau_N)|^P \end{array} \right]^T.$$

Analogously, its cost function is defined as in (4.28).

Once we have extracted all d_{np} parameters, to identify α_n and γ_p separately it is possible to apply the two stages identification algorithm proposed in [Bai98], consisting in *singular value decomposition* (SVD) of the $\mathbf{H}$ matrix described as:

$$\mathbf{H} = \sum_{i=1}^{\min(N,P)} \mu_i \cdot \sigma_i \cdot \upsilon_i^H = \mathbf{U\Theta V} \tag{4.35}$$

where the $\mathbf{H}$ matrix is defined as:

$$\mathbf{H} = \begin{pmatrix} d_{00} & d_{10} & \cdots & d_{N0} \\ d_{01} & d_{11} & & \vdots \\ \vdots & & \ddots & \vdots \\ d_{0P} & d_{1P} & \cdots & d_{NP} \end{pmatrix} = \begin{pmatrix} \alpha_0\gamma_0 & \alpha_1\gamma_0 & \cdots & \alpha_N\gamma_0 \\ \alpha_0\gamma_1 & \alpha_1\gamma_1 & & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_0\gamma_P & \alpha_1\gamma_P & \cdots & \alpha_N\gamma_P \end{pmatrix} \tag{4.36}$$

where $\mathbf{U} = (\mu_1,\ \mu_2,\ \cdots,\ \mu_N)$, $\mathbf{V} = (\upsilon_1,\ \upsilon_2,\ \cdots,\ \upsilon_P)$ are matrices and where μ_i $(i = 1, 2, \cdots P)$, υ_i $(i = 1, 2, \cdots N)$ are P-M-dimensional orthonormal vectors respectively, and $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_{\min(N,P)} \geq 0$. Finally, α_n and γ_p are estimated as:

$$\begin{aligned} \alpha &= s_\mu \mu_1 \\ \gamma &= s_\mu \sigma_1 \upsilon_1 \end{aligned} \tag{4.37}$$

where $\alpha = [\alpha_0,\ \alpha_1, \cdots, \alpha_N]^T$, $\gamma = [\gamma_0,\ \gamma_1, \cdots, \gamma_P]^T$ and where s_μ denotes the sign of the first nonzero element of μ_1.

The Hammerstein structure has been considered to identify PA behavioral models that later have been used to predistort the PA device. Moreover, IIR filters have also been considered for implementing the LTI block of the Hammerstein scheme, showing better performance than using only FIR filters but having the risk of introducing instabilities, as it has been presented in [Gil06b, Liu06].

More elaborated structures can be constructed by adding single Hammerstein structures in parallel in order to achieve a more refined PA behavioral model identification. This is the case of the *parallel-Hammerstein* model depicted in Fig. 4.10. The mathematical expression for

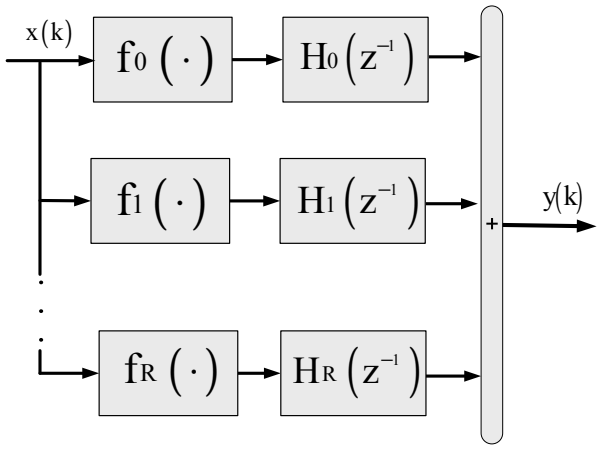

Figure 4.10: Block diagram of a parallel-Hammerstein model.

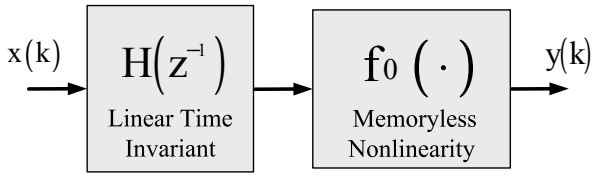

Figure 4.11: Block diagram of a Wiener model.

input-output relation in the parallel-Hammerstein model, assuming again FIR filters for the LTI blocks, can be expressed as

$$y_{P-H}(k) = \sum_{r=1}^{R} \sum_{n=0}^{N_r} \alpha_{nr} f_r \left(x(k - \tau_n) \right) \tag{4.38}$$

where R is the number of branches considered in the parallel-Hammerstein model, and with N_r ($r = 1, \cdots, R$) being the memory depth of each of the FIR blocks considered in each branch respectively.

Results showing the use of the parallel-Hammerstein model to identify PA behavioral models can be found in [Sil06].

Wiener Model

The Wiener model, as the Hammerstein model, can be seen as reduction, simplification of the general Volterra series expansion. The Wiener model is composed by a linear time-invariant system followed by a memoryless nonlinearity, as it is shown in Fig. 4.11. By considering again

a FIR filter for the LTI block, the input-output relation in a Wiener model can be defined as,

$$y_{Wien}(k) = f_0\left(\sum_{n=0}^{N} \alpha_n x(k-\tau_n)\right) \tag{4.39}$$

where $\tau_0 = 0$. And now, assuming that the memoryless nonlinearity is implemented with polynomials,

$$y_{Wien}(k) = \sum_{p=0}^{P} \gamma_p \left(\sum_{n=0}^{N} \alpha_n x(k-\tau_n)\right) \left|\sum_{n=0}^{N} \alpha_n x(k-\tau_n)\right|^p \tag{4.40}$$

As it can be deduced from (4.40), an easy linear regression of its coefficients it is not possible since the filter coefficients are integrated in the power series. In order to solve this problem, it is possible to first estimate an intermediate variable $v(k)$, as it is proposed in [Hag99], for later divide the estimation problem into two steps.

To estimate the intermediate variable, it is necessary to use the following assumptions [Hag99]:

- the linear subsystem (the FIR filter) is stable,
- the nonlinear function (power series) is invertible, and
- there is no noise in the system.

Assuming this, it is possible to calculate the intermediate variable $v(k)$. Therefore to proceed with the two steps identification it is first necessary to rewrite (4.39) by means of an intermediate variable $v(k)$.

$$\begin{aligned} v(k) &= \sum_{n=0}^{N} \alpha_n x(k-\tau_n) \\ y_{Wien}(k) &= f_0\left(v(k)\right) \end{aligned} \tag{4.41}$$

Then, it is possible to proceed with the Wiener model estimation in two steps:

First step: Extraction of the FIR filter parameters

To avoid the trivial solution ($\gamma, \alpha = 0$), it is possible to fix one parameter ($\alpha_0 = 1$) without loss of generality due to the over-parametrization [Hag99]. Then, it is possible to rewrite (4.41) in a manner that permits, by means of the intermediate variable $v(k)$, calculate the rest of filter coefficients:

$$\left.\begin{aligned} x(k) &= v(k) - \sum_{n=1}^{N} \alpha_n x(k-\tau_n) \\ v(k) &= f_0^{-1}\left(y(k)\right) \end{aligned}\right\} \hat{x}(k) = f_0^{-1}\left(y(k)\right) - \sum_{n=1}^{N} \alpha_n x(k-\tau_n) \tag{4.42}$$

So (4.42) describes an equation where the parameters come in linearly and thus can be estimated by linear regression. It is possible to rewrite (4.42) in a matrix notation,

$$\hat{x}(k) = \mu^H \phi \tag{4.43}$$

where $\mu = [\delta_0, \delta_1, \cdots, \delta_P, -\alpha_1, -\alpha_2, \cdots, -\alpha_N]^T$,

$\phi = [y(k),\, y(k)\,|y(k)|, \cdots, y(k)\,|y(k)|^P,\, x(n-\tau_1),\, x(n-\tau_2), \cdots, x(n-\tau_N)]^T$ and the cost function to be minimized is defined as:

$$J(k) = |\bar{e}(k)|^2 = |(x(k) - \hat{x}(k))/x(k)|^2 \tag{4.44}$$

Second step: Extraction of the memoryless polynomial parameters

With the FIR filter parameters already extracted, it is now possible to calculate the intermediate variable $v(k)$:

$$v(k) = x(k) + \sum_{n=1}^{N} \alpha_n x(k-\tau_n) \tag{4.45}$$

Having knowledge of the intermediate variable, it is possible to extract the memoryless polynomial coefficients γ_p.

$$y_{Wien}(k) = \sum_{p=0}^{P} \gamma_p v(k)\,|v(k)|^p \tag{4.46}$$

Rewriting (4.46) in a matrix notation:

$$y_{Wien}(k) = \lambda^H \mathbf{\Phi} \tag{4.47}$$

where $\lambda = [\gamma_0,\, \gamma_1, \cdots, \gamma_P]^T$ and $\mathbf{\Phi} = \left[v(k),\, v(k)\,|v(k)|, \cdots, v(k)\,|v(k)|^P\right]^T$. Its cost function is defined as in (4.28). Some examples using the Wiener model to identify PA behavioral models are presented in [Gil05a, Gil05b].

Similarly to parallel-Hammerstein models, *parallel-Wiener* models can be implemented by adding single Wiener structures in parallel in order to achieve a more refined PA behavioral model identification. A general *parallel-Wiener* structure is depicted in Fig. 4.12. The mathematical expression for input-output relation of a parallel-Wiener model, assuming again FIR filters for the LTI blocks, can be expressed as

$$y_{P-W}(k) = \sum_{r=1}^{R} f_r \left(\sum_{n=0}^{N_r} \alpha_{nr} x(k-\tau_n) \right) \tag{4.48}$$

where R is the number of branches considered in the parallel-Wiener model, and with N_r ($r = 1, \cdots, R$) being the maximum number of considered delays (memory depth) of each of the FIR blocks considered in each branch respectively.

Results showing the use of the parallel-Wiener model to identify PA behavioral models can be found in [Sil05].

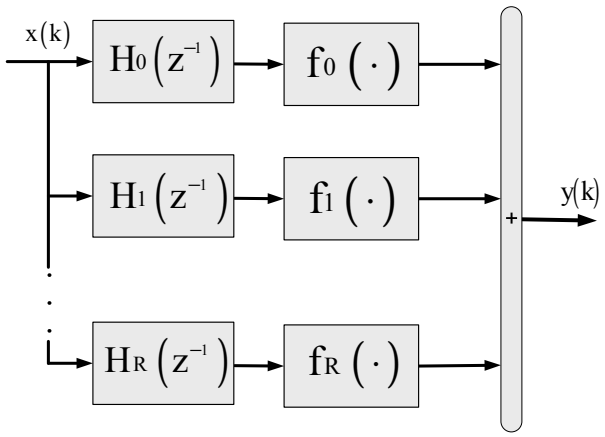

Figure 4.12: Block diagram of a parallel-Wiener model.

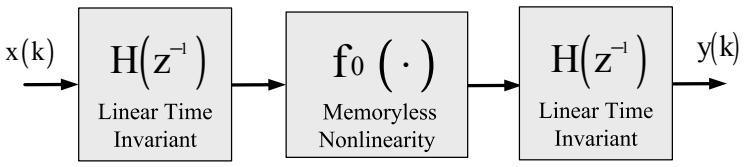

Figure 4.13: Block diagram of a Wiener-Hammerstein cascade model.

Wiener-Hammerstein Cascade: Three box modeling

The two box model methodology (Hammerstein or Wiener) can be extended to the three-box model topology. If we consider a Wiener model, an additional filter is included at the output of the memoryless nonlinearity to yield a filter-nonlinearity-filter cascade configuration. The Wiener-Hammerstein cascade model is a three-box model consisting in the cascade interconnection among a LTI block followed by a memoryless nonlinearity block and finally followed by another LTI block, as it is schematically shown in Fig. 4.13. If we consider both LTI blocks implemented as FIR filters, the input-output relation in a Wiener-Hammerstein cascade model can be defined as,

$$y_{Wien-Hamm}(k) = \sum_{j=0}^{M} \beta_j f_0 \left(\sum_{i=0}^{N} \alpha_i x(k - \tau_i - \tau_j) \right) \tag{4.49}$$

In order to use this model for PA behavioral modeling purposes, it is necessary to use the two-steps identification technique described for the Wiener model, identifying first intermediate variables that permit the final extraction of the coefficients of this three-box model.

Such a configuration is commonly used for satellite communication channels, where the PA at the satellite transponder is driven near saturation to exploit the maximum power efficiency

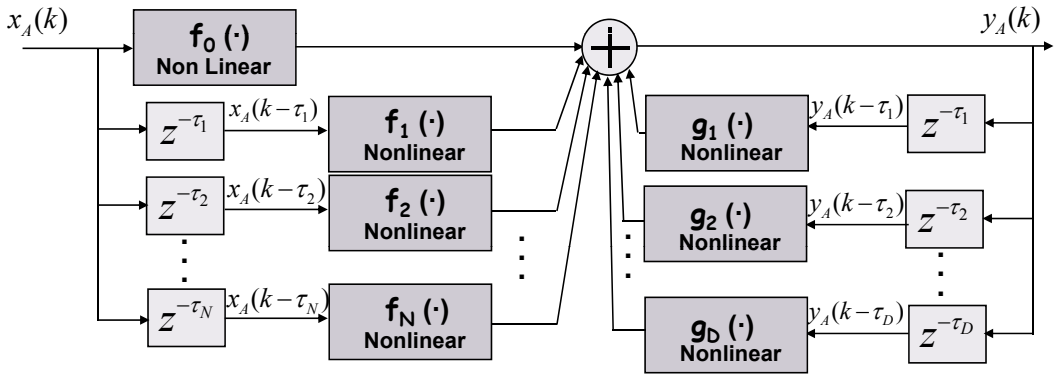

Figure 4.14: Block diagram of the Nonlinear Auto Regressive Moving Average (NARMA) model.

for the downlink. The Wiener-Hammerstien configuration has been used for modeling and predistortion linearization purposes in [Kim01] and [Cha01] respectively.

4.3.3 Nonlinear Auto-Regressive Moving Average Models

Soury et al. in [Sou02] point out that in systems where memory effects are secondary effects to a static nonlinear behavior, it is possible to consider that the individual signal pulses propagate nonlinearly in time but tend to sum up linearly. By considering that memory effects due to biasing network and thermal effects will fall in this category, it is possible to model the PA using a nonlinear auto regressive moving average (NARMA) model.

The advantage of using a NARMA model is the introduction of a nonlinear feedback path (*Infinite Impulse Response* -IIR- terms) that may permit relaxing the number of delayed samples considered to model the PA. Reducing the complexity of the PA model acquires importance when, for example, the PA model is considered for linearization purposes, such in digital predistortion linearization. However, one of the main weaknesses of the NARMA model is its stability, since the use of nonlinear feedback paths can result in the overall system instability. Therefore, in order to guarantee the stability of a NARMA PA model, a stability test based in the small-gain theorem will be presented in following.

Formulation of the NARMA Model

The NARMA model can be seen as extension of a simpler NMA model that incorporates a nonlinear auto regressive block. The block diagram describing the NARMA model is depicted in Fig. 4.14. The general input-output mathematical expression in a NARMA model can be

described as

$$y_{NARMA}(k) = \sum_{i=0}^{N} f_i\left(x(k-\tau_i)\right) - \sum_{j=1}^{D} g_j\left(y(k-\tau_j)\right) \tag{4.50}$$

with $f_i(\cdot)$ and $g_j(\cdot)$ being nonlinear memoryless functions, and where τ_i and τ_j ($\tau \subset$ N; and with $\tau_0 = 0$) are the most significant sparse delays of the input and the output respectively, contributing at the description of the PA memory effects. Next section of this Chapter will be focused in the techniques used for obtaining these sparse delays.

The present output sample ($y(k)$) depends on the sum of different static nonlinearities related to the present sample of the input ($x(k)$), and both input ($x(k\text{-}\tau_i)$) and output ($y(k\text{-}\tau_j)$) past samples ($i=1,2\ldots N$; $j=1,2\ldots D$), as it is shown in Fig. 4.14. By considering polynomials to implement the $f_i(\cdot)$ and $g_j(\cdot)$ nonlinear functions, it is possible to rewrite (4.50) as:

$$\begin{aligned} y_{NARMA}(k) = &\sum_{i=0}^{N} \alpha_i \left(\sum_{p=0}^{P} \gamma_p x(k-\tau_i) \left|x(k-\tau_i)\right|^p \right) - \\ &- \sum_{j=1}^{D} \beta_j \left(\sum_{p=0}^{P} \delta_p y(k-\tau_j) \left|y(k-\tau_j)\right|^p \right) \end{aligned} \tag{4.51}$$

where P is the order of the memoryless polynomial and N and D are the number of the delayed input and output samples (memory depth) considered for modeling the PA dynamics.

It is possible to rewrite (4.51) in a more compact matrix notation assuming that $a_{ip} = \alpha_i \cdot \gamma_p$ and $b_{jp} = \beta_j \cdot \delta_p$. Then (4.51) results in

$$y_{NARMA}(k) = \lambda^H \mathbf{\Phi} \tag{4.52}$$

where $\lambda = \left[\begin{array}{l} a_{00},\ \alpha_{01}, \cdots, a_{0P},\ a_{10}, \cdots, a_{1P}, \cdots, a_{N0}, \cdots, a_{NP}, \\ -b_{10}, \cdots, -b_{1P}, \cdots, -b_{D0}, \cdots, -b_{DP} \end{array} \right]^T$,

$$\mathbf{\Phi} = \left[\begin{array}{l} x(k),\ x(k)\left|x(k)\right|, \cdots, x(k)\left|x(k)\right|^P, x(k-\tau_1), \cdots, x(k-\tau_1)\left|x(k-\tau_1)\right|^P, \\ \cdots, x(k-\tau_N), \cdots, x(k-\tau_N)\left|x(k-\tau_N)\right|^P,\ y(k-\tau_1), \cdots, y(k-\tau_1)\left|y(k-\tau_1)\right|^P, \\ \cdots, y(k-\tau_D), \cdots, y(k-\tau_D)\left|y(k-\tau_D)\right|^P \end{array} \right]^T$$

and where its cost function is defined as in (4.28).

Stability Test: Small-Gain Theorem

The small-gain theorem is an input-output stability method based in bounded norms. It was developed by G. Zames in 1966 and it is well explained and applied in [Des75]. The small-gain method is capable of determining the stability in nonlinear systems when these are bounded by some kind of norm. Considering a block diagram of a feedback system as the one depicted in Fig. 4.15 and applying the methodology described in [Des75], the following inequalities are

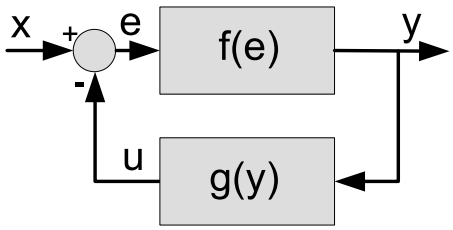

Figure 4.15: Block diagram of a feedback system for applying the small-gain stability method.

obtained:

$$\begin{aligned} \|y\|_p &\leq \gamma_f \|e\|_p \\ \|e\|_p &\leq \|x-u\|_p \leq \|x\|_p + \|u\|_p \\ \|u\|_p &\leq \gamma_g \|y\|_p \end{aligned} \tag{4.53}$$

where $\|\cdot\|_p$ is the p-order norm. Combining the inequalities in (4.53) leads to the following inequality:

$$\|y\|_p \leq \frac{\gamma_f}{1-\gamma_f\gamma_g} \|x\|_p \tag{4.54}$$

In other words, the output norm is bounded and this bound exists if $\gamma_f\gamma_g < 1$. This is a sufficient stability condition.

Now it is possible to apply the small-gain stability method to the proposed NARMA model. Defining $|\cdot|$ as the modulus, and $\|\cdot\|_2$ as the second order norm, it follows:

$$|f_i(v)| \leq \alpha_i |v| \rightarrow \|f_i\|_2 \leq \alpha_i \|v\|_2 \tag{4.55}$$

$$|g_j(v)| \leq \gamma_j |v| \rightarrow \|g_j\|_2 \leq \gamma_j \|v\|_2 \tag{4.56}$$

where α_i and γ_j are bounds of the second order norm gain of each of the polynomial terms of both $f_i(\cdot)$ and $g_j(\cdot)$ nonlinear functions, respectively. Therefore, for the particular NARMA structure the following inequality is obtained:

$$\|y\|_2 \leq \alpha_0 \|x\|_2 + \alpha_1 \|x\|_2 + \cdots + \alpha_M \|x\|_2 + \gamma_1 \|y\|_2 + \cdots + \gamma_N \|y\|_2 \tag{4.57}$$

Finally, derived from (4.57) it is possible to obtain the following inequality:

$$\|y\|_2 \leq \frac{\sum_{i=0}^{M} \alpha_i}{1 - \sum_{j=1}^{N} \gamma_j} \|x\|_2 \tag{4.58}$$

Therefore, the stability of the NARMA structure is assured (a sufficient condition) if the following inequality is accomplished [Des75]:

$$\sum_{j=1}^{N} \gamma_j \leq 1 \tag{4.59}$$

Results of PA Behavioral Modeling Using the NARMA Model

In the following, a practical example of identification of a PA behavioral model using the NARMA structure is presented. In order identify the PA low-pass complex envelope behavior based on a NARMA model, we have used input and output discrete complex data extracted from an LDMOS 3-stage class AB main amplifier.

The test signal used was a 16-QAM RRC filtered (roll-off 0.25) modulated signal. This signal presents power spectral efficiency but lacks of constant envelope and becomes highly sensitive to PA nonlinearities. Measurements were performed for an RF signal presenting a bandwidth of 1.25 MHz, at a center frequency of 1.96 GHz. The LDMOS 3-stage class AB main amplifier was operating at 2 dB of input back-off (IBO), and presented the following nominal characteristics: frequency range of $1.93 - 1.96$ GHz, maximal output power of +48 dBm and 36 dB Gain.

The normalized mean square error (NMSE) is a figure of merit to measure the resemblance of the model estimated output ($\hat{y}(k)$) and the PA measured output ($y(k)$), and it is defined as

$$NMSE(dB) = 10 \log \left(\frac{\sum\limits_{k=0} \left(y_I(k) - \hat{y}_I(k)\right)^2 + \left(y_Q(k) - \hat{y}_Q(k)\right)^2}{\sum\limits_{k=0} \left(y_I(k)\right)^2 + \left(y_Q(k)\right)^2} \right) \tag{4.60}$$

where both estimated and measured signals have been separated in their in-phase and quadrature components respectively.

A NARMA model was extracted considering 5 non consecutive taps (sparse delays), in concrete, 3 input delays and 2 output delays, and obtaining a NMSE below -30 dB. By using more coefficients (considering more delays) it is possible to obtain better figures of NMSE, but -30 dB is already an acceptable value taking into account that not only accuracy is important for DPD purposes, but also computational simplicity is a key issue towards DPD implementation.

On the other hand, in order to ensure its stability the small-gain criterion was checked, as it can be observed in Fig. 4.16. The main memoryless nonlinear function is f_0, while f_1, f_2, f_3 are the nonlinear functions related to the three delayed samples of the input, while g_1, g_2 are nonlinear functions related to the delayed samples of the output. As we can observe in Fig. 4.16, the sum of g_1, g_2 nonlinear functions do not exceed the reference bound (the straight line with unitary slope). Therefore since the small-gain criterion is satisfied, the estimated NARMA

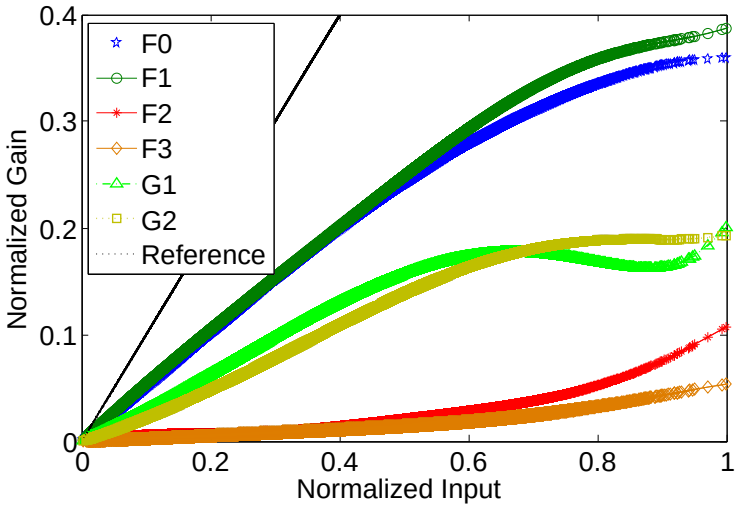

Figure 4.16: Nonlinear functions of the delayed samples in the NARMA model.

model is inherently stable.

In order to highlight the accuracy achieved by the NARMA model following the measured PA data and its significant improvement with respect to a memoryless nonlinear model, some figures pointing out this improvement are presented, such as: the AM/AM and the AM/PM characteristics; the output spectra; both In-phase and Quadrature components and the error vector magnitude (EVM).

Fig. 4.17 and Fig. 4.18 show the AM/AM and the AM/PM characteristics respectively, of the PA measured data, the NARMA model and the memoryless model. While the NARMA model is capable to reproduce the AM/AM and AM/PM data dispersion, the memoryless model cannot. Figure 4.19 shows the output power spectra of the PA measured data, the NARMA model and the memoryless nonlinear model, while Fig. 4.20 and Fig. 4.21 show the In-phase and Quadrature components respectively. It can be observed how the memoryless model cannot perfectly fit the output measured signal since no PA dynamics are considered in this model. Finally, in order to see the capability of the model to accurately reproduce the in-band distortion, we have demodulated the measured PA data to plot its constellation and measure its error vector magnitude (EVM). The PA constellation is shown in Fig. 4.22, presenting an EVM of 7.51%. The NARMA model constellation is similar to the PA one and, in addition, its EVM (7.18 %) is very close to the PA one (see Fig. 4.23-right). On the contrary, the memoryless nonlinear model (Fig. 4.23-left) is incapable of accurately representing the PA measured data constellation presenting an EVM of 4.40 %.

This example has shown how it is possible to obtain NARMA models inherently stables, useful for identifying low pass complex envelope dynamic PA models. Moreover due to its com-

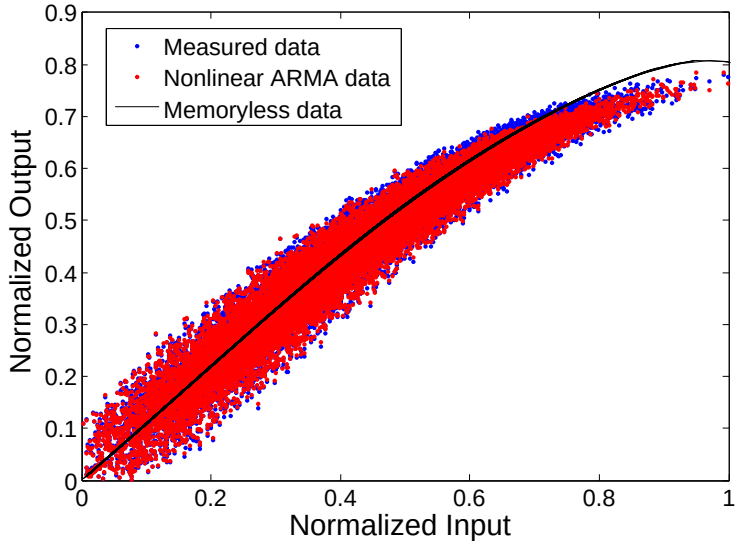

Figure 4.17: AM/AM characteristic of the PA measured data, the NARMA model and a memoryless nonlinear model.

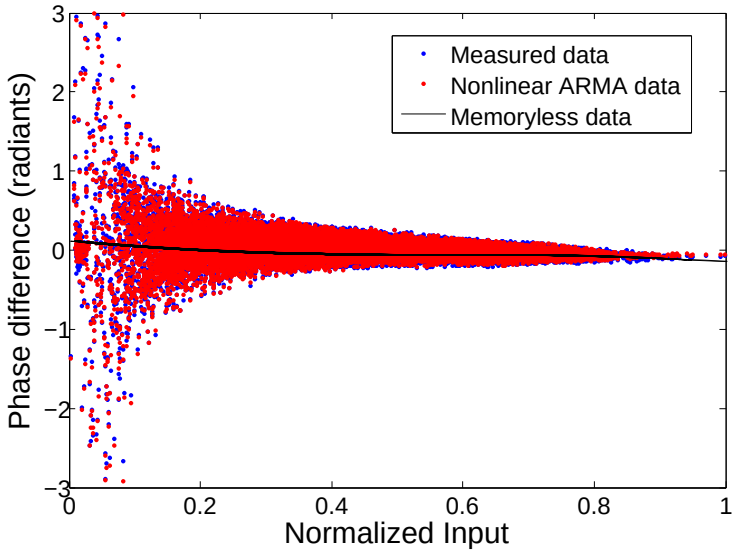

Figure 4.18: AM/PM characteristic of the PA measured data, the NARMA model and a memoryless nonlinear model.

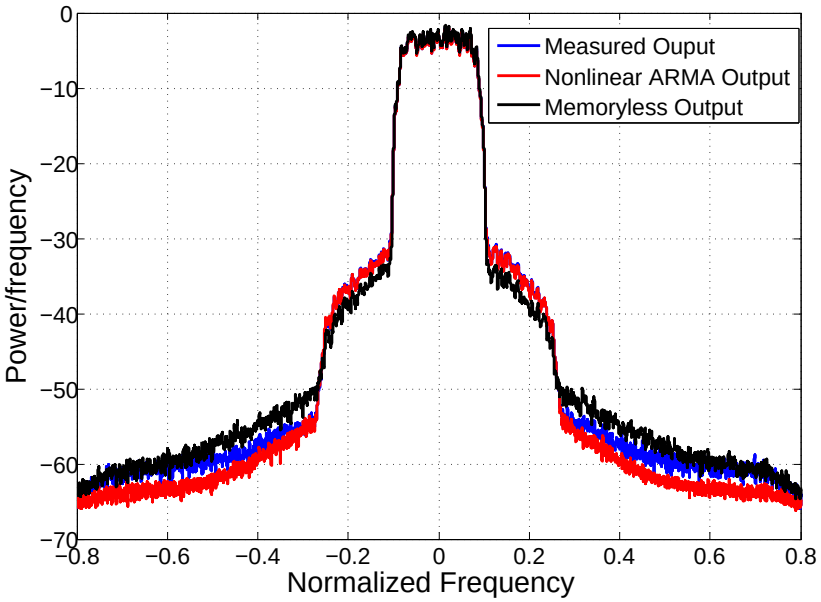

Figure 4.19: Output power spectra of the PA measured data, the NARMA model and a memoryless nonlinear model.

promise between complexity and accuracy, the NARMA model can be seriously considered for digital baseband predistortion linearization [Gil06a].

4.4 Optimum Delays Calculation for Behavioral Models

In order to evaluate the suitability of behavioral models reported in this Chapter to be used for digital predistortion linearization purposes, some issues regarding the complexity introduced by these models have to be taken into account. NNs algorithms perform various combinations of the input data within the hidden layers. As no optimization on the importance of these data combination in hidden layers is performed, NNs are often over dimensioned. On the other hand, Volterra models have the disadvantage of using a huge number of parameters when considering higher order kernel extractions which can lead to an inaccurate identification. As we have seen in previous subchapters, the use of reduced versions of the general Volterra series (pruned Volterra, NMA, Hammerstein, Wiener, three box modeling and its variants) can help to reduce complexity by selecting the more significant delays contributing at modeling the PA nonlinear dynamics. In order to minimize these behavioral models complexity (searching computationally efficient behavioral models) some studies have been aimed at determining the minimum memory length needed to accurately characterize the PA behavioral model, as in [O'B06].

For that reason, in the following, two heuristic search algorithms, the *simulated annealing* (SA) and the *genetic algorithms* (GA), are presented. The use of these heuristic algorithms

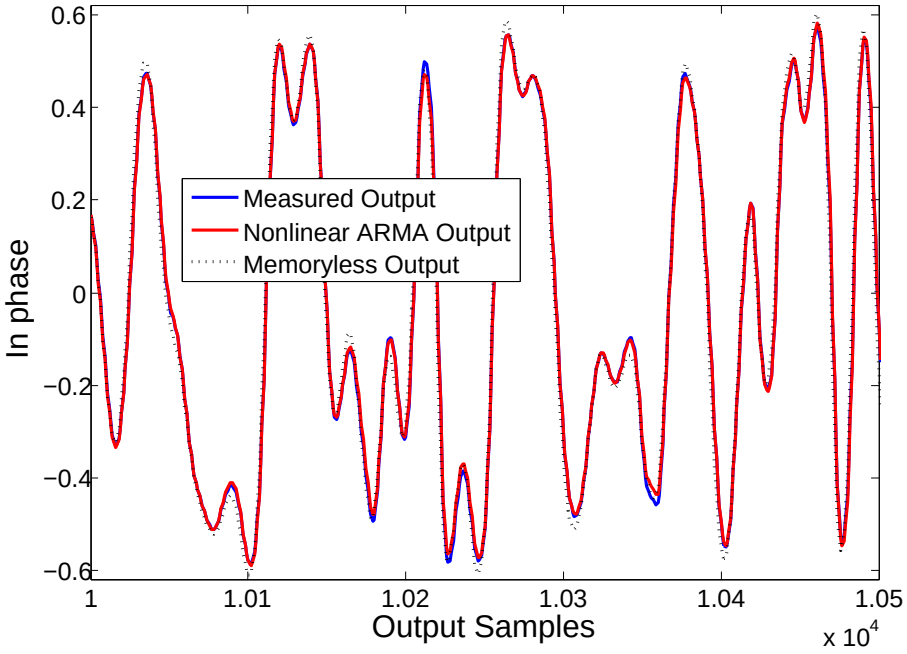

Figure 4.20: In-phase components of the PA measured data, the NARMA model and a memoryless nonlinear model.

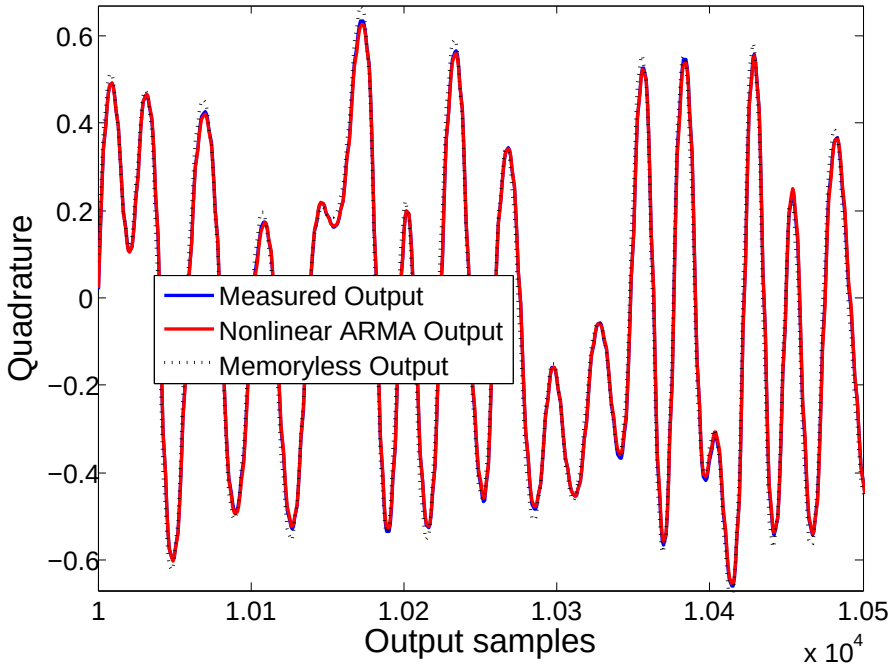

Figure 4.21: Quadrature components of the PA measured data, the NARMA model and a memoryless nonlinear model.

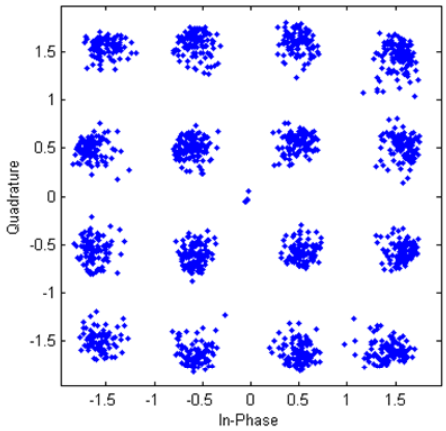

Figure 4.22: 16-QAM RRC filtered PA's constellation (EVM = 7.51%).

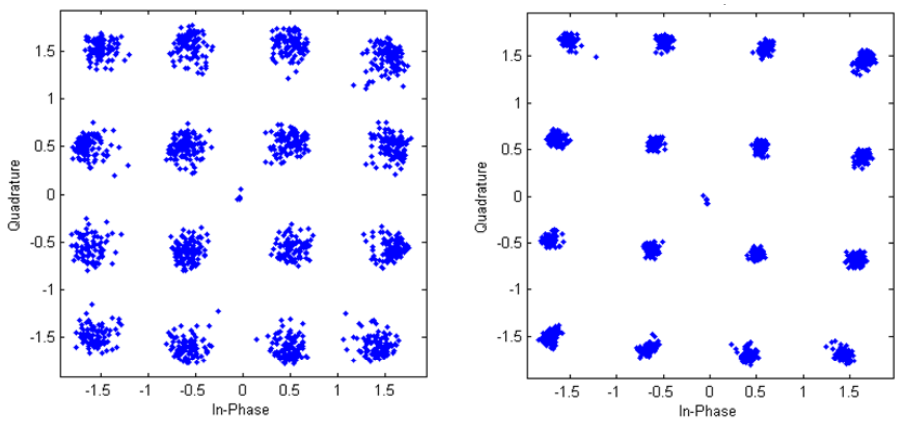

Figure 4.23: 16-QAM RRC filtered constellation of the NARMA model (right)-(EVM = 7.18 %) and the memoryless model (left)-(EVM = 4.40 %).

results useful to deal with the trade-off between behavioral models complexity (memory length and significant sparse delays) and accuracy. These two heuristic search algorithms can be used to determine the sparse delays, expressed as τ's in previous subchapters, that permit a better characterization of the PA nonlinear memory considering the minimum necessary memory length, and so, reducing models complexity.

4.4.1 Simulated Annealing

Simulated Annealing (SA) is a quite straightforward random-search technique based on the Metropolis Monte Carlo (MMC) method and has been proven to be very useful when dealing with nonlinear problems [Aar89]. One of the major advantages of this technique over other methods is its ability to avoid becoming trapped in local minima. The SA algorithm not only accepts changes that decrease the *cost* (or *energy*) *function* ($\sum J(k) = E$), but also some changes

that increase it, accepted with a probability (p).

$$p = e^{-\delta E/T} \tag{4.61}$$

where δE is the increase in the energy function and T is a control parameter called temperature. Focusing in our particular minimization problem (finding the best sparse delays that contribute at the characterization of the PA memory effects) it is necessary to define:

- a representation of possible solutions, e.g. values of the delays considered ($\tau \subset N$) from 1 to "M" delays.
- a random generator that selects some delays (depending on the memory depth) among the representation of possible solutions (defined before). It creates an array of delays (taus) whose suitability will be evaluated.
- a energy function (E) that evaluates the NMSE achieved with a particular solution (a particular array of delays -taus-).
- an annealing schedule, that is, the initial temperature (T) and rules for lowering it as the search progresses.

Once these considerations have been taken into account, the SA algorithm operates as follows (see also the flux diagram in Fig. 4.24):

i) Randomly generate a *solution array* for the problem. In our particular case, an array of N (with N being the memory depth considered) delays that configure the initial solution -taus$_O$-. This becomes the initial (and already old solution) and its energy, E$_O$, is determined. This array of delays and its energy become the *best-to-date values* (taus$_B$, E$_B$).

ii) Set the *effective temperature* (T) to the initial value.

iii) Run a *Metropolis Monte Carlo* (MMC) simulation at this temperature by repeating the following steps:

1. Generate a new solution by replacing the array of delays that is used in the old solution (taus$_O$) with the new one (taus$_N$).
2. Calculate the new energy (E$_N$) of this new array of delays (taus$_N$).
3. Determine if this new solution is kept or rejected.

- If $E_N \leq E_O$, this new array of delays is accepted. It replaces the old solution (taus$_O$) and its energy E$_O$. If this energy is less than the best-to-date value (E$_N$ <E$_B$), it also becomes the best-to-date solution and energy (taus$_B$, E$_B$).

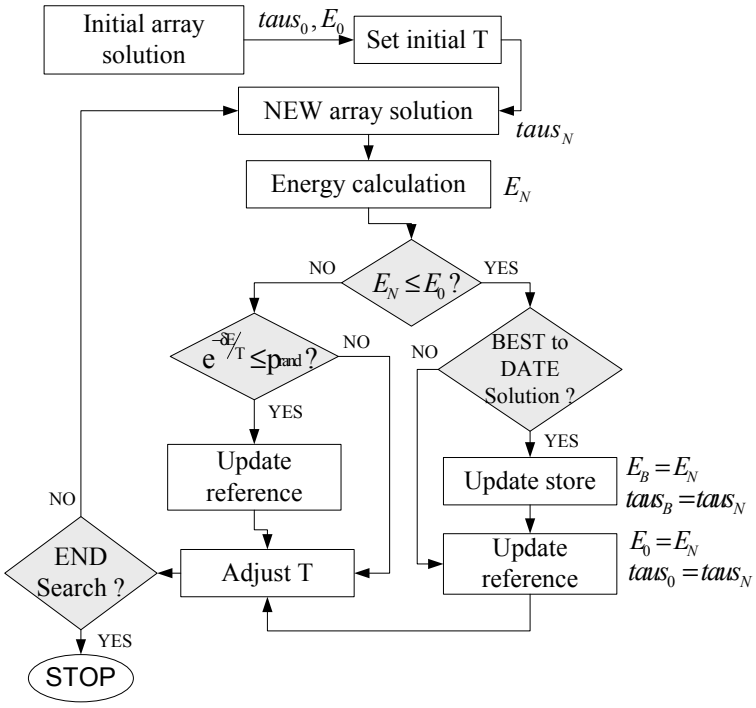

Figure 4.24: Flux diagram of the Simulated Annealing search algorithm.

- If $E_N > E_O$, the acceptance probability is calculated. If this number is greater than a random number between 0 and 1, the new array of delays (taus_N) is accepted and it replaces the old one (taus_O) and its energy E_O. If the acceptance probability is less than the random number, the new solution is rejected and the old solution (taus_O) remains as the reference.

iv) Finally, slightly reduce the effective temperature. If the new effective temperature is greater than the final effective temperature, return to run the MMC simulation. Otherwise the run is finished and the best-to-date solution (taus_B) and energy (E_B) is reported.

The SA has the advantage of performing searches of arrays of delays combinations, where the maximum number of the considered delays per array can be defined by the user. However, in order to obtain reliable results, the SA has to run for several iterations and thus, the study of the best sparse delays defining the PA dynamics in a PA model may consume a considerable amount of time and computational resources.

4.4.2 Genetic Algorithms

Genetic Algorithms (GA) search procedures are based on the mechanisms of natural selection and natural genetics and their basic principles are extensively explained in [Gol88]. A basic

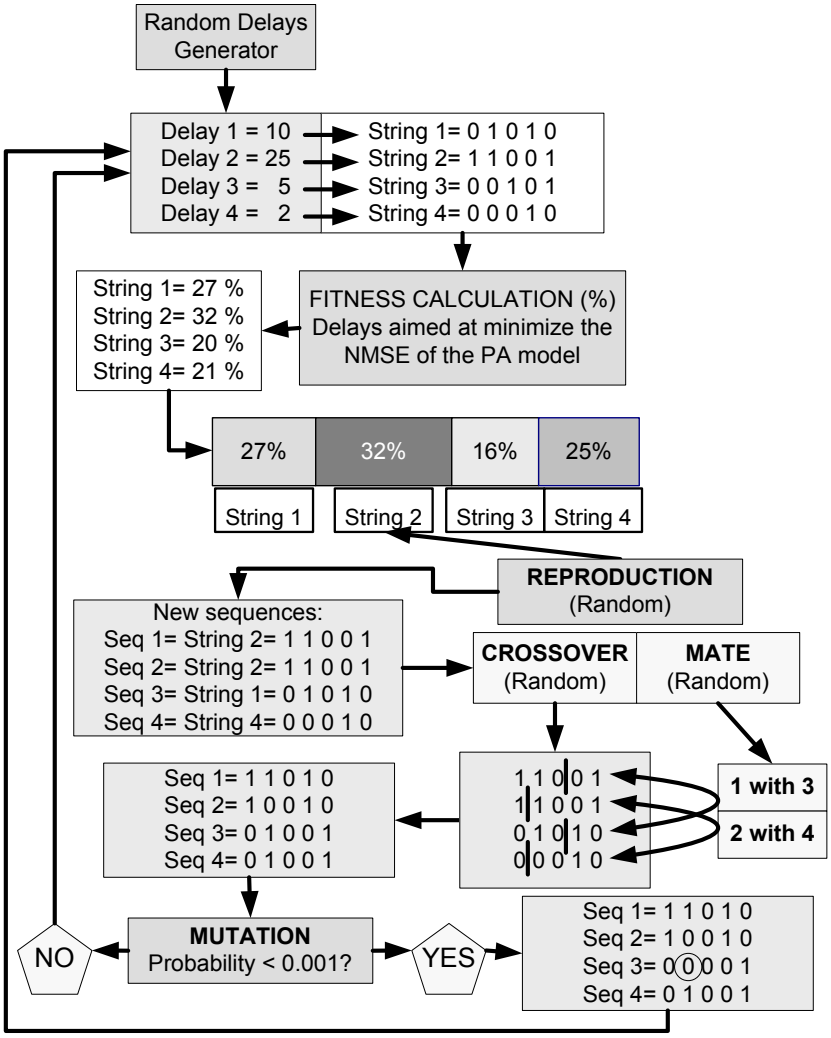

Figure 4.25: Example of the Genetic Algorithm search.

genetic algorithm is composed by three operators: *reproduction*, *crossover* and *mutation*.

Particularizing the GA search process to the obtention of the best sparse delays that characterize PA dynamics in PA models, the GA search operates in the following manner (see also the flux diagram in Fig. 4.25):

Reproduction is a process in which "n" sequences of bit strings (each one representing one delay candidate binary coded) compete for replication. These bit strings (sequences) are selected for being replicated according to their fitness (minimization of the cost function). The string with a higher fitness value has a higher probability of replication, that is, for appearing (even repeated several times) in the new considered group of "n" sequences that appear after reproduction.

Next, *crossover* may proceed into two steps: first, members of the new reproduced strings are

mated at random; second, each pair of string mates interchange a piece of the string (a random number of "b" bits , from last bit position of the bit string (L_b) to position $L_b - b$.

Finally *mutation* randomly changes the bit value of a string with a probability that is fixed by the user. Mutation prevents from losing some potential value not previously appeared with reproduction and crossover operators.

This process is repeated several times and finally there is only one string of all 'n' considered sequences that survives, the one that better fits the minimization function.

The search process to find the best delays that contribute at minimizing the normalized mean square error (NMSE) using GA differ from the one used in SA. While in SA it is possible to search for arrays of N delays at once, with the GA algorithm it is possible to search only one delay at every GA simulation. That is, first, only one best delay is found for a particular PA model. Next, fixing this delay as a good solution, the GA algorithm is run again to find the second best delay that contributes, together with the first delay found, at minimizing the cost function. This operation has to be repeated as many times as the number of delays that one want to consider to model the PA dynamics.

4.4.3 Results of an Optimal Delay Search Using Heuristic Search Algorithms

In order to show the advantages of using these two heuristic search algorithms, the study of the best sparse delays for two behavioral models presented in previous subchapters is exemplified in the following. The selected behavioral models are: an augmented nonlinear moving average (NMA) model and a nonlinear auto-regressive moving average (NARMA) model.

A set of data of a W-CDMA excitation signal with a PAPR of 10 dB was used to extract PA behavioral models. The signal bandwidth and the channels frequency spacing were BW = 3.84 MHz and $f_{spacing} = 5MHz$. The PA used for measurements is a LDMOS class AB main amplifier operating at the mildly nonlinear region (IBO $\sim$ 5 dB) and with the following nominal characteristics: frequency range of $1.93 - 1.96$ GHz, maximal output power of +48 dBm and 36 dB Gain. The PA output signal was measured at a center frequency of 1.96 GHz using an Agilent Performance Spectrum Analyzer (PSA) and an Agilent 89611 Vector Signal Analyzer (VSA). The measurement results were processed using the Agilent 89604 distortion test suit.

Figure 4.26 shows the NMSE achieved in PA behavioral model identification versus the number of delays considered for both NARMA and augmented NMA models when using three types of delay arrays: consecutive delays, delays obtained from both GA and SA searches, respectively. In general, the more delays are considered to model PA dynamics the lower becomes the NMSE. However, at a certain number of delays considered (memory depth) the NMSE remains constant or even increases, which denotes that this particular model cannot further improve in the

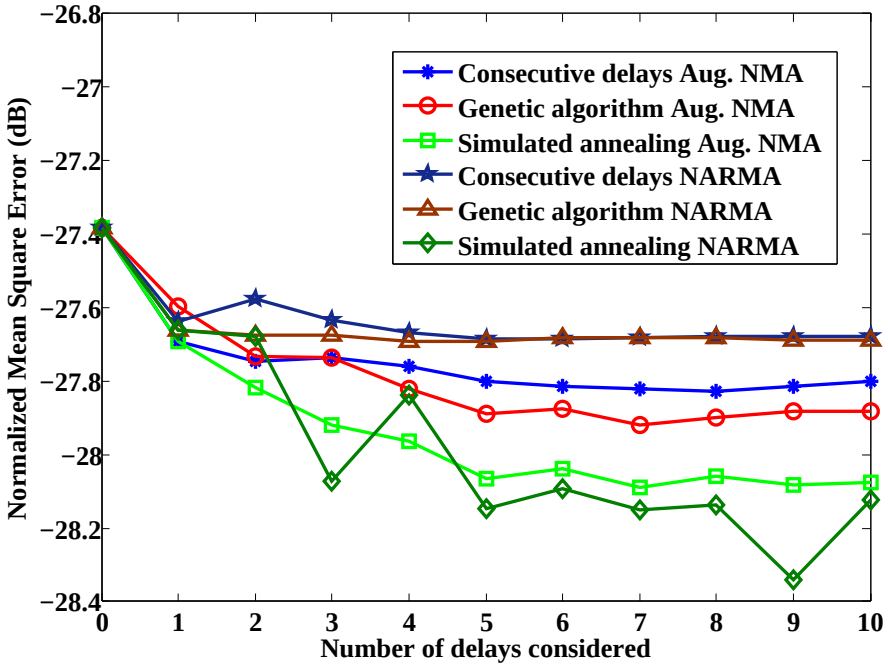

Figure 4.26: NMSE (dB) versus number of considered delays.

identification of PA dynamics.

Attending results shown in Fig. 4.26, we can see that it is possible to obtain better NMSE (for both PA models) using a few proper chosen delays provided by the heuristic search algorithms than using consecutive delays. In addition, SA performance has shown better results than GA. The reason can be related to the limitation of the GA search process regarding the number of delays that can be considered in one search. That is, with the GA instead of searching for the best N sparse delays at once, it is only possible to accumulate optimal delays calculated individually, what clearly limits the possibility of certain beneficial delay combinations.

On the other hand, Table 4.1 shows the NMSE and the adjacent channel error power ratio (ACEPR), for both PA behavioral models. The ACEPR is defined as

$$ACEPR(dB) = 10\log\left(\frac{\int\limits_{Adj\,Chann}\left|Y(f)-\hat{Y}(f)\right|^2 df}{\int\limits_{Chann}|Y(f)|^2 df}\right) \tag{4.62}$$

with $Y(f)$ and $\hat{Y}(f)$ being the discrete Fourier transform of $y(k)$ and $\hat{y}(k)$ respectively. The ACEPR has to be evaluated for both upper and the lower adjacent channels, and the greatest value is used for comparison with other models

For the extraction and validation processes two different sets of measured data (10^5 samples

Table 4.1: Results of Power Amplifier Behavioral Modeling with Study of Delays

Behavioral models	N^o **Delays**	**NMSE (dB)**	**ACEPR (dB)**
Augmented NMA	5 input delays	-34.80	Lower: -48.67 Upper: -46.37
NARMA	3 (1 input and 2 output delays)	-33.62	Lower: -48.64 Upper: -46.27

each set) have been used, respectively. The NMSE and ACEPR figures of merit (FOMs) in Table 4.1 have been obtained considering the first significant minimum number of delays in the SA search shown in Fig. 4.26. Moreover, in Fig. 4.27 it is possible to observe an excellent match between the PA measured output spectrum and the output spectra of the Aug. NMA and NARMA models, respectively. While Fig. 4.28 shows the limiting curves of the AM/AM-conversion of the reference and both modeled output signals. These limiting curves represent the upper and lower boundary covering the AM/AM-conversion plot, and thus it is possible to compare the amount of memory effects introduced by both models here presented.

The SA algorithm has shown to be a powerful heuristic search technique that permits to find the minimum and best sparse delays. This contributes to reduce the PA behavioral models complexity, permitting more computational efficient DPD linearization.

4.5 Comparative among Some Common Behavioral Models

Until now, some of the most significant models or structures for PA behavioral modeling have been presented within this Chapter. Despite the objective of this thesis is not focused in PA behavioral modeling, the selection of an accurate and implementable PA model is a key point towards the DPD linearization. Therefore, a comparison among some of the behavioral models here described can help to understand the trade-off existing between accuracy and complexity when selecting a particular type of model to be used in DPD. For that reason, this comparative takes into account not only the reproduction capabilities (accuracy) of the models, measured in terms of the NMSE figure of merit, but also the number of parameters used, that is, the computational complexity introduced.

The set of models selected for this comparison pretend to be representative of the different behavioral models present in literature (dynamic and memoryless models). In concrete, the considered models are: memoryless Polynomial, memoryless Neural Network (NN), Wiener-Bose, Wiener, Hammerstein, Augmented Nonlinear Moving Average (NMA), Nonlinear Auto-

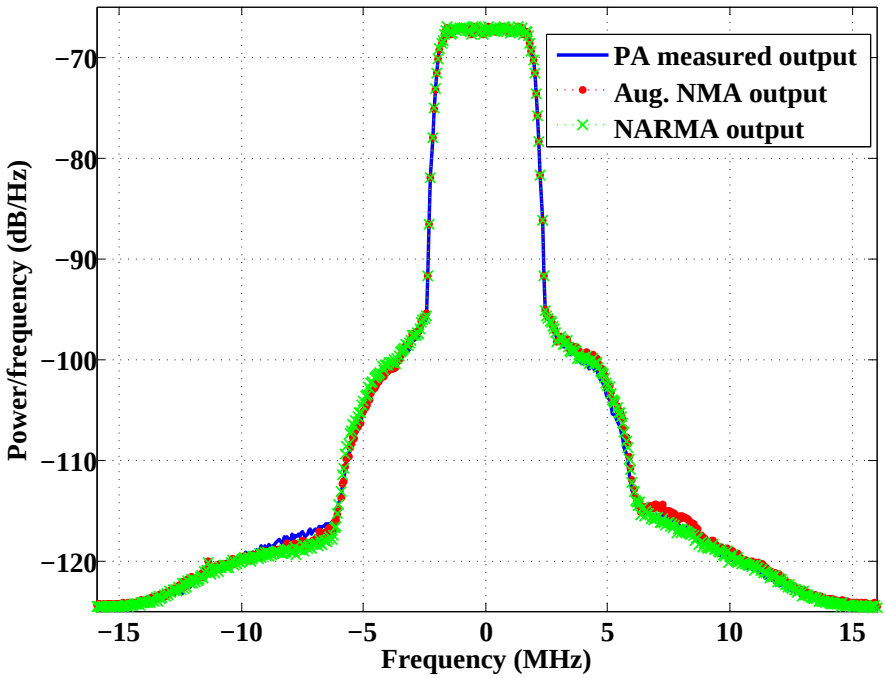

Figure 4.27: Measured, Aug. NMA and NARMA output power spectra.

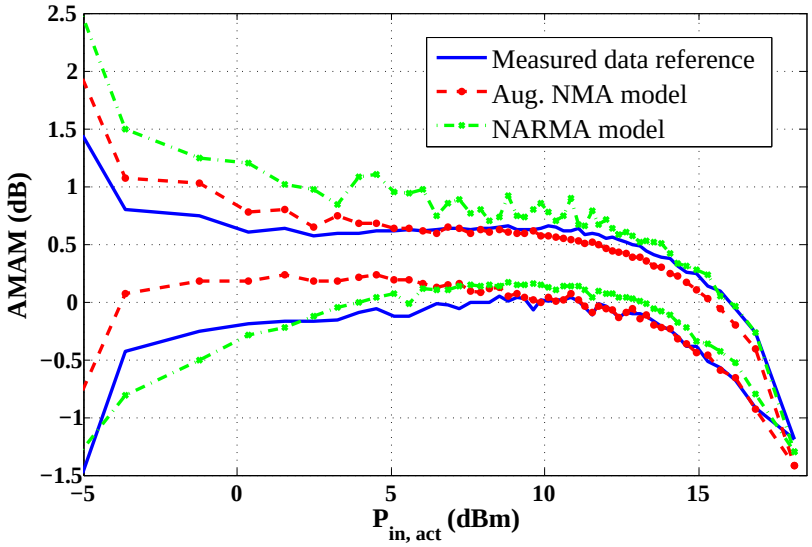

Figure 4.28: AM/AM-conversion limiting curves.

Table 4.2: PA Behavioral Models Comparison

Model	**Nº (Coeffients)**	**noM dB**	**LM (dB)**	**NLM (dB)**	**LM/NLM (dB)**
Neural Network	10 neurons	-27.94	-30.98	-30.84	-31.89
Polynomial	4	-40.45	-36.15	-36.12	-34.58
Wiener	6	-40.00	-40.00	-36.13	-36.77
Hammerstein	12	-40.55	-40.41	-36.95	-37.39
Parallel-Wiener	12	-40.40	-40.35	-36.29	-36.94
3-box	26	-49.62	-47.75	-32.79	-32.80
NARMA**	35 —31 —33 —33	-57.22	-49.86	-37.44	-38.21
Wiener-Bose reduced	102	-41.89	-41.84	-41.70	-41.88
Augmented NMA**	105	-58.17	-50.17	-44.62	-46.39
Wiener-Bose	231	-42.01	-41.97	-43.72	-44.22

** The Simulated Annealing algorithm has been used to find the best delays for modeling

Regressive Moving Average (NARMA), Parallel Wiener and 3-box Wiener-Hammerstein.

Four sets of data have been considered for model extraction and validation. Simulated input-output data sets have been obtained taking into account four different memory situations: (i) memoryless (noM), (ii) linear memory (LM), (iii) non-linear memory (NLM) and (iv) linear and nonlinear memory simultaneously (LM/NLM). Figure 4.29 shows the gain curves for all four different sets of the considered data. The PA has been designed to handle a WCDMA signal with a carrier of 1.9 GHz. In order to show the model capabilities for dealing with nonlinear dynamic systems, the output bias network has been designed to deliberately introduce a reasonable amount of nonlinear or linear memory under demand.

The results of this comparison are listed in Table 4.2. It is possible to observe that memoryless models, incapable of modeling dynamics, provide moderate NMSE figures in comparison to models with memory. On the other hand, normally, the use of more coefficients (more delays combinations) permit achieving higher figures of NMSE. However since computational simplicity is a figure of merit, the use of properly selected coefficients results in a better identification. Therefore, it is not the quantity of coefficients that matters, but their quality, that is, their significance within the model in order to model the PA nonlinear dynamics. Thus, the use of pruning techniques or heuristic algorithms to select the minimum set of significant coefficients contributing at the PA modeling cannot be obviated.

Finally, observing results in Table 4.2 it is possible to conclude that the NARMA model is capable to trade-off accuracy and complexity, obtaining good figures of NMSE with an acceptable set of properly selected coefficients. For that reason, the NARMA model is our selected candidate to perform DPD linearization.

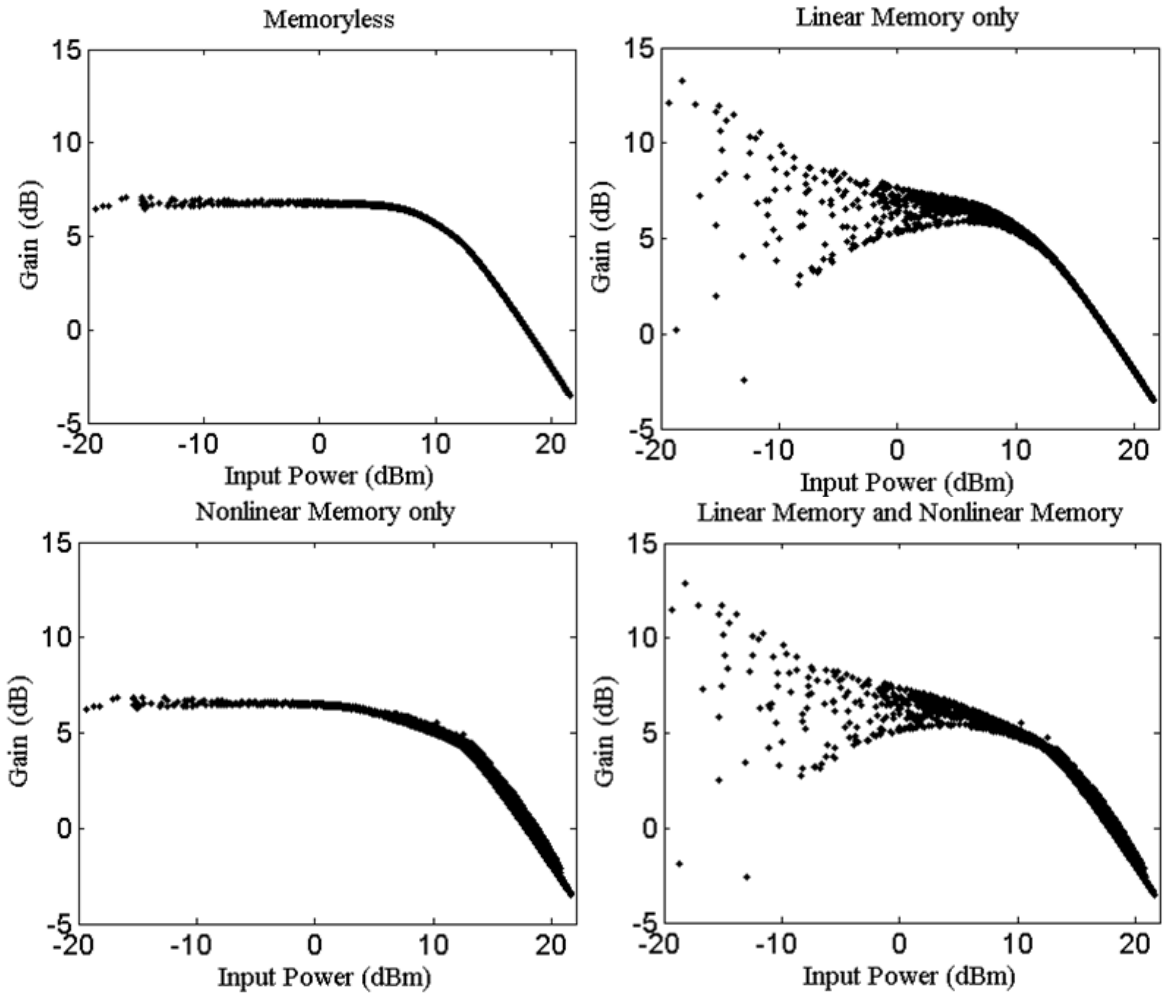

Figure 4.29: Gain characteristics for different memory configurations.

4.6 Summary

In this chapter different behavioral models have been presented. On the one hand, there are behavioral models that can reproduce the PA nonlinear behavior but not take into account possible PA dynamics, those are the so called memoryless behavioral models. On the other, there are behavioral models capable of reproducing PA memory effects as well as its nonlinear behavior and thus suitable for applying the envelope filtering technique in DPD linearization. Among those behavioral models able to reproduce PA memory effects, NNs and Volterra series are very accurate descriptors of nonlinear dynamics since they cover a large sum of possible states and their nonlinear combinations. However, since the optimum can be enemy of the good enough, these models add computational complexity to DPD algorithms which can derive in an inefficient power consume of the DSP device. For that reason several simplified models derived from Volterra series (two and three box modeling) have been proposed as possible candidates

for implementing the DPD function in a DSP. Moreover the NARMA model, which cannot be considered as a simplification of the Volterra series, introduces a conceptual feedback path (IIR filters) that, despite it can take to instabilities, can reduce the number of required coefficients to extract the model (or its inverse) and thus being even more computationally efficient. Related to this last issue, we have presented heuristic search algorithms that can be used to find the optimal sparse delays contributing at describing PA memory effects. By using these search algorithms we have shown how to minimize the number of required coefficients in PA behavioral models and thus finding a compromise between accuracy and computational complexity.