An Efficient CS-CPWL Based Predistorter

M. BRUNO[1], *J. COUSSEAU*[1], *S. WERNER*[2], *J. FIGUEROA*[1], *M. CHEONG*[2], *R. WICHMAN*[2]

[1]CONICET-Dept. of El. and Comp. Eng., Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca, 8000 Argentina
[2]Helsinki University of Technology. Signal Processing Laboratory, P.O. Box 3000, FIN-02015 TKK, Finland

mbruno@criba.edu.ar, [jcousseau, figueroa] @uns.edu.ar, [mycheong, stefan.werner, risto.wichman] @tkk.fi

Abstract. *We study the performance of Hammerstein predistorters (PD) to model and compensate nonlinear effects produced by a high power amplifier with memory. A novel Hammerstein model is introduced that includes, as the basic static nonlinearity, the complex simplicial canonical piecewise linear (CS-CPWL) description. Previous results by the authors have shown that the use of this kind of static nonlinearity leads to an efficient representation of basic nonlinear models. Furthermore, different tradeoffs between modeling capability and performance are considered.*

Keywords

Digital predistortion, adaptive algorithm, Wiener model, Hammerstein model, piecewise linear function.

1. Introduction

Highly efficient power amplifiers (PA) form an important building block for modern wireless communication systems such as Wireless Local Area Network (WLAN), WiMAX and the Long Term Evolution of the 3G system (LTE of 3GPP). Real power amplifiers (PAs) have a nonlinear transfer function causing signal compression and clipping that result in signal waveform distortion and adjacent channel interference. Power backoff and PAPR reduction techniques reduce the nonlinear distortion effects but result in low power efficiency. In other words, there is a clear tradeoff between the allowed level of nonlinear distortion and system power efficiency. In order to maintain satisfactory performance levels while increasing the power efficiency, techniques for compensating the nonlinear distortion are needed. A number of techniques exist to linearize the operating region of the PA, see, e.g., [1] and [2]. Examples of linearization techniques are feedforward and feedback linearizer, envelope elimination and restoration and digital predistortion (PD). In addition to the nonlinear distortion, broadband PAs introduce memory effects, mainly due to impedance mismatch and thermal effects, which do not only increase the computational complexity of the behavioral modeling problem, but also the complexity of the linearization techniques [3], [4]. In this paper we consider digital PDs that efficiently take into account both the nonlinear characteristic and memory effects of broadband PAs.

Traditionally, finite Volterra models [5], [6] were considered for nonlinear digital predistortion mainly because they can describe the ”fading memory behavior” of modern (low cost) PA with acceptable accuracy, and also because specific (not necessarily simple) measurements can be used to obtain their parameters [3].

To overcome the exponential number of parameters required by finite Volterra models, different approaches have been considered. In addition to the known block models, i.e. Hammerstein and Wiener models, there exist different alternatives that combine serially or in parallel the basic building blocks (a linear filter and a static polynomial nonlinearity). Even though Hammerstein and Wiener models have limited modeling capabilities, their simplicity turns them very attractive from the computational point of view.

Recently, a modified finite Volterra model was considered to design a digital predistorter [7]. Particular cases of this model are the more popular ”memory polynomial” predistorter [8] and the ”generalized memory polynomial” [9]. The main concern with these Volterra-based predistorters is perhaps the number of parameters required to obtain a suitable performance in terms of linearization.

In this paper, adaptive Hammerstein model PDs are proposed for linearization of a broadband nonlinear PA using indirect-learning architecture, considering computational complexity as a main issue. The nonlinear part of the proposed Hammerstein PD is parameterized using the Complex Simplicial Canonical Piecewise Linear (CS-CPWL) function [10], [11], while the linear part is modeled with an FIR filter. The CS-CPWL function is able to model general complex static nonlinearities with high accuracy. In addition, and different to polynomial models that tend to cause compression in their output level in response to high input signal level, it does not exhibit compression for high input level. Rather, it saturates after a user-defined maximum input level. That behavior results in better and more efficient modeling capabilities for the kind of strong nonlinearities related to the application at hand. Since the high accuracy obtained with this complete model comes at the cost of high number of parameters, a simplified alternative is also presented that allows to obtain good compromise between modeling capability and complexity.

This paper is organized as follows. The system models are introduced in Section 2. In Section 3, the proposed two-step adaptive predistorter based on indirect learning is presented. An efficient modification of that predistorter is presented in Section 4. Simulation evaluation of the indirect-learning PD is presented in Section 5 using the compact behavioral model of a commercial PA in a high bandwidth application. Finally conclusions are drawn in Section 6.

2. System Description

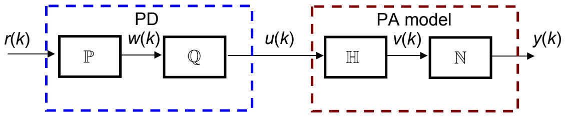


Fig. 1. Adaptive PD system for a Wiener model PA.

In this paper we consider the linearization of a Wiener model PA using a Hammerstein model PD as illustrated in Fig. 1. The Wiener model consists of a linear subsystem $\mathbb{H}(\cdot)$ followed by a static nonlinear subsystem $\mathbb{N}(\cdot)$, while the Hammerstein model is described by cascading a static nonlinear subsystem $\mathbb{P}(\cdot)$ and a linear subsystem $\mathbb{Q}(\cdot)$. As can be inferred from Fig. 1, the PD parameters shall be adapted such that $\mathbb{Q}(\cdot)$ equalizes the effects of $\mathbb{H}(\cdot)$, and $\mathbb{P}(\cdot)$ compensates the nonlinear effect of the static nonlinearity $\mathbb{N}(\cdot)$.

The linear subsystem $\mathbb{Q}(\cdot)$ is parameterized by an L-th order FIR filter, i.e., the PD output is given by

$$u(k) = \mathbf{q}^{\mathrm{H}}(k)\mathbf{w}(k) \tag{1}$$

where

$$\begin{aligned} \mathbf{q}^{\mathrm{H}}(k) &= \left[q_0^*(k)\ q_1^*(k)\ \cdots\ q_{L-1}^*(k)\right] \\ \mathbf{w}(k) &= \left[w(k)\ w(k-1)\ \cdots\ w(k-L+1)\right]^{\mathrm{T}}. \end{aligned} \tag{2}$$

To parameterize the static nonlinear block $\mathbb{P}(\cdot)$, we use the complex-valued simplicial canonical piecewise linear (CS-CPWL) function [10] (see also [12]). Given the baseband (complex-valued) input $r(k) = r_I(k){+}\mathrm{j}r_Q(k) \in \mathbb{C}$, we can form the complex-valued output $w(k)$ from the nonlinearity of the Hammerstein model using a two-dimensional simplicial CPWL function [11]. Based on the efficient representation proposed in [10], we can build the mapping $\mathbb{P}[\cdot] : \mathbb{C} \rightarrow \mathbb{C}$ as

$$w(k) = \mathbb{P}[r(k)] := \mathbf{c}^{\mathrm{H}}\mathbf{\Lambda}[r(k)] \tag{3}$$

where $\mathbf{c} \in \mathbb{C}^M$ is a vector containing the parameters associated with the nonlinear static representation, and $\mathbf{\Lambda} : \mathbb{C} \rightarrow \mathbb{R}^M$ is a vector function depending on the partition of the input $r(k)$, in P equal sectors. Based on this partition of each dimension of the mapping the number of parameters in $\mathbf{c}$ is $M = 1 + 2P + P^2$ [10]. To complete the description of the static nonlinearity, each sector of the simplicial partition [11] is given by $[\beta_{i-1}, \beta_i]$, where $i = 1, 2, \cdots, P$ and $\beta_0 \leq \beta_1 \leq \cdots \leq \beta_P$. These sectors divide real and imaginary components of the input signal into P partitions, and based on this description, $\mathbf{\Lambda}$ is defined by

$$\mathbf{\Lambda}(r(k)) = \begin{bmatrix} \Lambda_0 \\ \mathbf{\Lambda}_1[r(k)] \\ \mathbf{\Lambda}_2[r(k)] \end{bmatrix} \tag{4}$$

where $\Lambda_0 = 1$ is the zero-order basis (nesting level),

$$\mathbf{\Lambda}_1(r(k)) = \begin{bmatrix} \Upsilon^1[Re\{r(k)\}] \\ \Upsilon^1[Im\{r(k)\}] \end{bmatrix} \tag{5}$$

is the first-order basis with $\Upsilon^1 : \mathbb{R} \rightarrow \mathbb{R}^P$ whose i-th entry is given by

$$\Upsilon_i^1(\nu) = \begin{cases} \frac{1}{2}(\nu - \beta_i + |\nu - \beta_i|) & \text{if } \nu \leq \beta_P \\ \frac{1}{2}(\beta_P - \beta_i + |\beta_P - \beta_i|) & \text{if } \nu > \beta_P \end{cases}, \tag{6}$$

and $\mathbf{\Lambda}_2[r] = \Upsilon^2[Re\{r\}, Im\{r\}] : \mathbb{R}^2 \rightarrow \mathbb{R}^{P^2}$ is the second-order basis, whose $[(i-1)P + j]$-th entry is defined by

$$\Upsilon^2_{(i-1)P+j}(\nu_1, \nu_2) = \begin{cases} \Upsilon_i^1(\nu_1) & \text{if } \Upsilon_i^1(\nu_1) \leq \Upsilon_j^1(\nu_2) \\ \Upsilon_j^1(\nu_2) & \text{if } \Upsilon_i^1(\nu_1) > \Upsilon_j^1(\nu_2) \end{cases} \tag{7}$$

for $i, j = 1, ..., P$. Thus, (4) and the terms (6) and (7) define a second-order SCPWL suitable for complex filtering.

3. A CS-CPWL Based Predistorter

The parameters of the Hammerstein-based PD, $\mathbb{P}(\cdot)$ and $\mathbb{Q}(\cdot)$ (see, Fig. 1), can be adaptively identified using an indirect learning strategy, e.g., [14], [13]. It is well-known that the identification of block models, such as the Wiener and Hammerstein models, is often complicated by their non-convex cost function [16], [17], [15]. In order to avoid this problem, we employ a modified Wiener model estimator that provides us with estimates of $\mathbb{P}(\cdot)$ and $\mathbb{H}(\cdot)$.

The estimate of $\mathbb{P}(\cdot)$ is then copied online to the PD while the estimate of $\mathbb{H}(\cdot)$ is used to adapt an estimate of $\mathbb{Q}(\cdot)$. Thus, the indirect learning algorithm proposed in the following consists of a Wiener model estimation loop and a PD linear filter adaptation loop working in tandem.

3.1 Estimation of $\mathbb{P}(\cdot)$ and $\mathbb{H}(\cdot)$

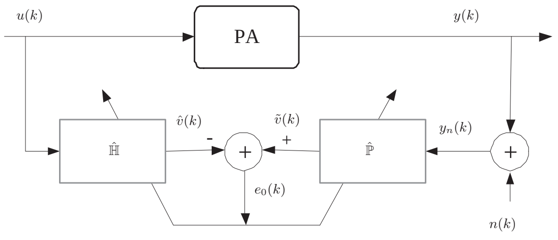


Fig. 2. Identification scheme for the Wiener model PA.

Fig. 2 shows the Wiener model estimator. This configuration allows the model parameters to enter the error equation linearly, resulting in a convex cost function [18]. The basic idea here is to identify the linear subsystem $\mathbb{H}$ and the nonlinear subsystem $\mathbb{P}$, the inverse of $\mathbb{N}$. The proposed algorithm estimates the intermediate signal $v(k)$ in Fig. 1 of the Wiener model from the PA input-output signals $\{u(k), y(k)\}$ by forming the error $e_0(k) = \tilde{v}(k) - \hat{v}(k)$ where

$$\begin{aligned} \tilde{v}(k) &= \hat{\mathbf{c}}^{\mathrm{H}} \mathbf{\Lambda}[y(k)] \\ \hat{v}(k) &= \sum_{i=0}^{N-1} \hat{h}_i^*(k) u(k-i), \end{aligned} \tag{8}$$

and the vectors $\hat{\mathbf{c}}(k) \in \mathbb{C}^{M \times 1}$ and $\mathbf{\Lambda}[y(k)] \in \mathbb{C}^{M \times 1}$ are defined in Section 2. To avoid ambiguity in the filter gain, $\hat{h}_0(k) \equiv \hat{h}_0$ is anchored to a fixed value [16]. The error to be minimized can now be written as

$$e_0(k) = \tilde{v}(k) - \hat{v}(k) = \boldsymbol{\theta}^{\mathrm{H}}(k)\boldsymbol{\phi}(k) - \hat{h}_0^* u(k) \tag{9}$$

where the parameter vector $\boldsymbol{\theta}(k) \in \mathbb{C}^{(M+N-1)\times 1}$ and regression vector $\boldsymbol{\phi}(k) \in \mathbb{C}^{(M+N-1)\times 1}$ are given by

$$\begin{aligned} \boldsymbol{\theta}(k) &= \left[\hat{\mathbf{c}}^{\mathrm{T}}(k)\; \hat{h}_1(k) \cdots \hat{h}_{N-1}(k)\right]^{\mathrm{T}}, \qquad (10) \\ \boldsymbol{\phi}(k) &= \Big[\mathbf{\Lambda}^{\mathrm{T}}[y(k)]\; -u(k-1) \cdots -u(k-N+1)\Big]^{\mathrm{T}}. \end{aligned}$$

Using the instantaneous squared error $|e_0(k)|^2$ as an objective function, a stochastic gradient algorithm that updates $\boldsymbol{\theta}(k)$ is given by

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) - \mu_0 \frac{\partial |e_0(k)|^2}{\partial \boldsymbol{\theta}^*(k)} \\ &= \boldsymbol{\theta}(k) - \mu_0 \boldsymbol{\phi}(k) e_0^*(k) \end{aligned} \tag{11}$$

where μ_0 is the adaptation step size that controls the convergence speed and final error. To ensure convergence, μ_0 is chosen in the range

$$0 < \mu_0 < \frac{1}{\rho_{\max}} \tag{12}$$

where $\rho_{\max}$ is the maximum eigenvalue of $\mathrm{E}[\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{H}}]$. Convergence behavior and stability of algorithm (11) is discussed in detail in [10].

The estimate $\hat{\mathbb{P}}$, defined by the elements of parameter vector $\hat{\mathbf{c}}(k)$ ($\{\hat{c}_i^*(k)\}_{i=0}^{M-1}$), is directly copied to the PD. The estimate of $\hat{\mathbb{H}}$, or $\{h_i^*(k)\}_{i=0}^{N-1}$, is further used for adapting the linear part $\hat{\mathbb{Q}}$ of the PD as detailed next. This copying mechanism is illustrated in Fig. 3.

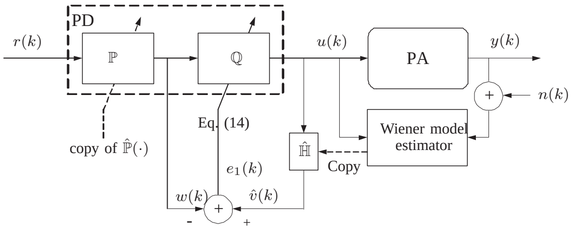


Fig. 3. Indirect-learning configuration for adaptive PD.

3.2 Estimation of $\mathbb{Q}$

In order to estimate the linear part $\mathbb{Q}$ of the PD, we acknowledge the fact that $\mathbb{Q}$ should equalize the memory $\mathbb{H}$ in the Wiener model. In other words, the intermediate signals $w(k)$ and $v(k)$ should ideally be identical. Since the Wiener model estimator, detailed in previous section, provides us with an estimate of $\mathbb{H}$, we can reproduce an estimate of $v(k)$ through (8). By forming the error

$e_1(k) = \hat{v}(k) - w(k)$, where

$$w(k) = \mathbf{c}^{\mathrm{H}} \mathbf{\Lambda}[r(k)], \tag{13}$$

a stochastic gradient algorithm that updates parameter vector $\mathbf{q}(k) = [q_0(k) \cdots q_{M-1}(k)]^{\mathrm{T}}$, describing $\mathbb{Q}$, is given by

$$\begin{aligned} \mathbf{q}(k+1) &= \mathbf{q}(k) - \mu_1 \frac{\partial [|e_1(k)|^2]}{\partial \mathbf{q}^*(k)} \\ &= \mathbf{q}(k) - \mu_1 e_1^*(k) \sum_{i=0}^{N-1} \hat{h}_i^*(k) \frac{\partial u(k-i)}{\partial \mathbf{q}^*(k)} \\ &\approx \mathbf{q}(k) - \mu_1 e_1^* \sum_{i=0}^{N-1} \hat{h}_i^*(k) \mathbf{w}(k-i) \end{aligned} \tag{14}$$

where $\mathbf{w}(k) = [w(k) \cdots w(k-L+1)]^{\mathrm{T}}$. The last approximation in (14) is valid for sufficiently small value of μ_1 so that $\mathbf{q}(k) \approx \mathbf{q}(k-i)$ for $i = 1, \cdots, N-1$.

Equations (11) and (14) constitute the indirect learning Hammerstein PD algorithm. Note that (14) is a filtered-x LMS algorithm. Thus, the stability of the recursion in (14) depends on the quality of the estimates $\{\hat{h}(k)\}_{i=0}^{N-1}(k)$. To ensure stability, the phase response error between the estimate and the actual PA dynamics must be within the range $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ [20], [19]. The CS-CPWL adaptive predistorter is summarized in Tab. 3.2.

4. Simplified PWL Based Predistorter

The main characteristic of the Hammerstein predistorter design is the utilization of the CS-CPWL representation to obtain a complex mapping of the baseband (complex) linear filtered input. This complete complex mapping allows us to approximate a Wiener model with high accuracy when the number of sectors P is increasing. The main drawback of the CS-CPWL representation is that the number of parameters or, equivalently, the PD complexity increases quadratically with the number of employed sectors (partitions). In order to maintain a low complexity at the expense of a reduced modeling capability we may replace the general

```
Definitions:
    $r(k)$ predistorter input
    $\mathbf{\Lambda}[r]$
    $w(k)$ predistorter output
    $u(k)$ estimator input
    $y(k)$ estimator output
Parameters:
    $N$ = number of $\mathbf{q}$ coefficients
    $\beta_i$, $i = 1, \cdots, P$ = sectors of $(Re\{r\}, Im\{r\})$
    $M$ = number of $\mathbf{c}$ coefficients
    $\mu_c$ = step size of $\mathbf{c}$
    $\mu_h$ = step size of $\mathbf{h}$
    $\mu_q$ = step size of $\mathbf{q}$
Initialization:
    $\mathbf{c}(0) = [-1 + j\, 1 - j\, 0 \cdots 0]^{\mathrm{H}}$
    $\mathbf{h}(0) = \mathbf{0}$
    $\mathbf{q}(0) = \mathbf{0}$

For each $k = 1, 2, \cdots$
  For $i = 1$ to $P$,
  $\Upsilon^1_{y_i}$ = eq.(6) with $Re\{y(k)\}$.
  $\overline{\Upsilon}^1_{y_i}$ = eq.(6) with $Im\{y(k)\}$.
  For $i = 1$ to $P$, for $j = 1$ to $P$,
  $\Upsilon^2_{y_{(i-1)P+j}}$ = eq.(7) with $y(k)$.
  $\mathbf{\Lambda}[y] = [1\ \Upsilon^1_y\ \overline{\Upsilon}^1_y\ \Upsilon^2_y]^{\mathrm{T}}$
  For $i = 1$ to $P$,
  $\Upsilon^1_{r_i}$ = eq.(6) with $Re\{y(k)\}$.
  $\overline{\Upsilon}^1_{r_i}$ = eq.(6) with $Im\{y(k)\}$.
  For $i = 1$ to $P$, for $j = 1$ to $P$,
  $\Upsilon^2_{r_{(i-1)P+j}}$ = eq.(7) with $r(k)$.
  $\mathbf{\Lambda}[r] = [1\ \Upsilon^1_r\ \overline{\Upsilon}^1_r\ \Upsilon^2_r]^{\mathrm{T}}$
  $\hat{v}(k) = \mathbf{h}^{\mathrm{H}}(k)u(k)$
  $e_0(k) = \mathbf{c}^{\mathrm{H}}(k)\mathbf{\Lambda}[y] - \hat{v}(k)$
  $w(k) = \mathbf{c}^{\mathrm{H}}(k)\mathbf{\Lambda}[r]$
  $u(k) = \mathbf{q}^{\mathrm{H}}(k)w(k)$
    $e_1(k) = \hat{v}(k) - w(k)$
    $\mathbf{c}(k+1) = \mathbf{c}(k) - \mu_c e^*_0(k)\mathbf{\Lambda}[y]$
    $\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h e^*_0(k)u(k)$
    $\mathbf{q}(k+1) = \mathbf{q}(k) + \mu_q e^*_1(k)\sum_{i=0}^{N-1} h_i(k)w(k-i)$
```

Tab. 1. Indirect learning CS-CPWL predistorter algorithm.

CS-CPWL representation in (3) by [12]

$$\begin{aligned} w(k) &= \mathbb{P}_s[Re\{r(k)\}] + j\mathbb{P}_s[Im\{r(k)\}] && (15)\\ &= Re\{\mathbf{c}_s\}^{\mathrm{T}}\mathbf{\Lambda}[Re\{r(k)\}] + jIm\{\mathbf{c}_s\}^{\mathrm{T}}\mathbf{\Lambda}[Im\{r(k)\}] \end{aligned}$$

where $Re\{\mathbf{c}_s\}, Im\{\mathbf{c}_s\} \in \mathbb{R}^M$ are vectors containing the parameters associated with the simplified nonlinear static representation, and $\mathbf{\Lambda} : \mathbb{R} \to \mathbb{R}^M$ is a vector function depending on the partition of the input $r(k)$ into P equal sectors. Based on this partition of each dimension of the mapping the number of parameters in c_s is $M = 2(1 + 2P)$ (see [12]). Based on this description, $\mathbf{\Lambda}$ is now given by

$$\mathbf{\Lambda}(\nu(k)) = \begin{bmatrix} \Lambda_0 \\ \mathbf{\Lambda}_1[\nu(k)] \end{bmatrix} \qquad (16)$$

where Λ_0 and Λ_1 are as in (4). We see that employing the simplified representation in (16) avoids the cross terms (7) required in the fully complex mapping. As a consequence, considerable computational savings are possible in the predistorter design, because the complexity of each PWL representation is now linear in the number of partitions P. To obtain the simplified CS-CPWL based predistorter we simply employ (16) in the corresponding updating equation. Furthermore, update equation (14) is replaced by

$$\mathbf{q}_s(k+1) = \mathbf{q}_s(k) - \mu_1 e^*_1(k) \sum_{i=0}^{N-1} \hat{h}_i(k) w_s(k-i). \qquad (17)$$

Tab. 4 provides the computational complexity of the PD implementations in terms of number of multiplications and additional operations. For comparison purposes, the memory polynomial predistorter [8] is included. The parameters in each case are

- Memory polynomial PD with a memory of L_{mp} lags and a polynomial order of P.
- CS-CPWL complete PD with a memory of L lags and a Complex PWL with P sectors.
- Simplified CS-CPWL PD with a memory of L lags and a Complex PWL with P Sectors.

Clearly, the simplified algorithm requires lower number of parameters than the complete CS-CPWL. Considering similar memory length, the number of parameters of the simplified algorithm P_{S-PWL} and of the memory polynomial algorithm P_{MP} are equal when $P_{MP} = 1 + (1 + 2P_{S-PWL})/L$. For example, for $L = 5$ and $P_{S-PWL} = 10$ partitions in the simplified CS-CPWL algorithm, this algorithm is more efficient if the order required for the memory polynomial algorithm is higher than 5. Considering the number of multiplications, for the same example, the simplified CS-CPWL algorithm requires 26, the memory polynomial 45 and the complete CS-CPWL scheme 126 multiplications.

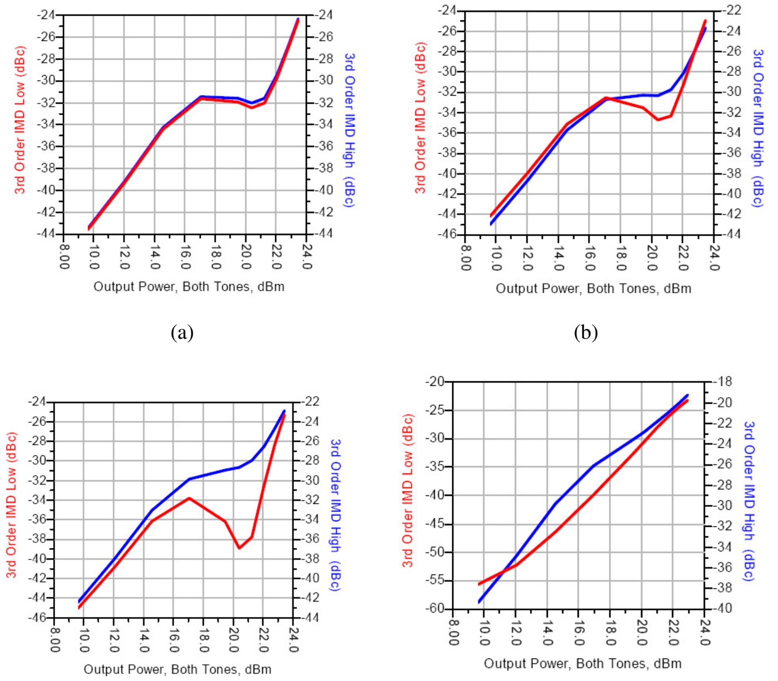
PD	Parameters	Multiplications	Other*
[8]	$L_{\mathrm{mp}}P$	$L_{\mathrm{mp}}P + L_{\mathrm{mp}}(P-1)$	L_{mp}
Complete	$L+1+2P+P^2$	$L+1+2P+P^2$	$2P+P^2$
Simplified	$L+1+2P$	$L+1+2P$	$2P$

* Absolute value

Tab. 2. Algorithm complexity.

5. Simulations

The performance of the algorithms was evaluated using Agilent Advanced Design System (ADS) [21] and Mat-



(a)

(b)

(c)

(d)

Fig. 4. Low and high third-order harmonic intermodulation (IMD3) products. Tone separation: (a) 100 kHz, (b) 1 MHz, (c) 2 MHz, (d) 10 MHz.

lab. This allows us to use realistic nonlinear power amplifier models with memory in ADS and at the same time do system identification in Matlab. The simulations are carried out in both analog and digital domain. Baseband symbols are generated with ADS and imported to Matlab to design the predistorter as described in Section 3. The analog response of the amplifier to the pre-distorted baseband symbols is evaluated with a Time Domain Envelope analysis technique [21] in RF domain.

Baseband signal: The signal is generated according to WLAN 802.11g standard with 54 Mbps data rate, 20 MHz bandwidth, and an FFT of 64 bins.

Radio frequency PA: Our single stage amplifier design is based on Freescale MRF9742 FET model. The model is accurate up to 3 GHz and includes some parasitics effects of circuits design. The specified power gain is $P_G = 9.5$ dB, and the input and output impedances of the amplifier are matched to $50\,\Omega$ by matching networks implemented in the simulator with discrete and micro strip components. These networks are almost flat in the bandwidth of interest centered at 850 MHz. The design also includes via hole inductors connecting to the printed circuit board ground plane, and the DC bias filters [24]. This allows us to include the effect of baseband impedance and determine baseband memory effects from intermodulation products [25], [26], in the case they occur.

Operating conditions: Firstly, to avoid excessive clipping noise, the baseband signal power must be adjusted to a proper PA operating region. To this purpose, the *harmonic balance simulator* (HBS) using one-tone test setup [22], [23], was used to find the 1 dB compression point P_{CP} of the PA that results in 28 dBm output power (HBS is configured up to the ninth order of carrier frequency, 7.65 GHz, in order to include the higher order inter modulation terms). The average baseband input signal power obtained is $P_{AV} = P_{CP} - P_{PAPR} - P_G$, where P_{PAPR} is the output peak-to-average power ratio (PAPR). Verifying that, for the specified signal $P_{PAPR} \cong 8$ dB[1], the result is $P_{AV} \cong 10.5$ dBm.

Next, in order to have a coarse estimation of memory length of the PA circuit, HBS using two-tone test setup [22] is performed to obtain third order intermodulation products (IMD3) [3]. The memory effect is reflected by the asymmetry of low and high IMD3 products. The frequency range in which the asymmetry is visible is defined as *memory bandwidth* L_{BW}. The memory bandwidth of the test PA is approximately 10 MHz as illustrated in Fig. 4. This asymmetry can also be observed by the abrupt change in the phases of the lower and upper IMD3 products, as illustrated in Fig. 5. Since the chosen sampling frequency is $f_s = 80$ MHz (four samples per symbol), a coarse estimation of the PA memory is: $(f_s/2)/L_{\mathrm{BW}} \cong 4$.

Predistorter parameters: The parameters used in the predistorter algorithms are the following: Memory Polynomial: $L_{\mathrm{mp}} = 5$, $P = 8$ (number of parameters= 40); Complete CS-

[1] Despite that its ideal value is equal to the number of subcarriers, PAPR is in general a statistical value [27]. The best output to our modeling purposes is obtained using an ad hoc input power adjustment (close to the calculated value) observing the distortion at the output spectrum.

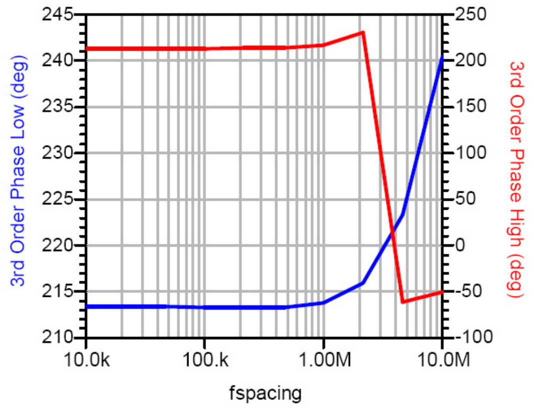


Fig. 5. Low and high IMD3 phases. Phase dispersion vs. different frequencies of tones spacing.

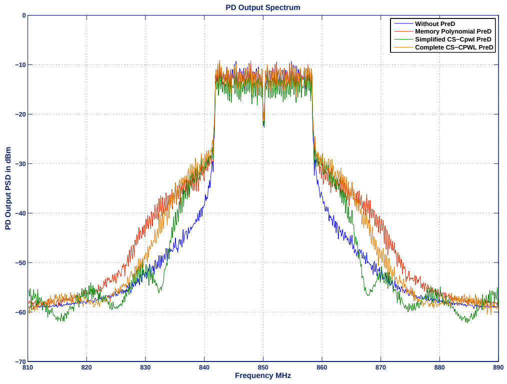


Fig. 6. Output power spectrum obtained with the different predistorters.

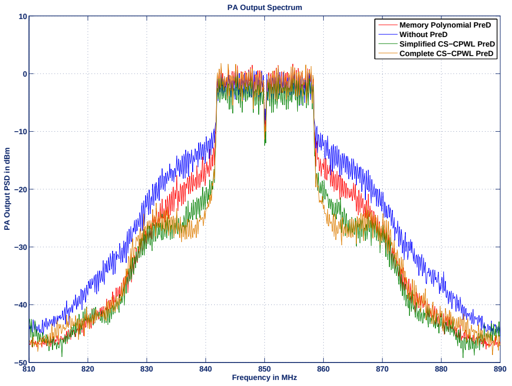


Fig. 7. Power spectrum at the *output* of the PA obtained with the different predistorters.

CPWL: $L = 10$, $P = 10$ (number of parameters= 131) and Simplified CS-CPWL: $L = 10$, $P = 10$ (number of parameters= 31).

Results: The output power spectral density of the different predistorters, is illustrated in Fig. 6. This figure shows the 80 MHz signal bandwidth with a spectral resolution of 50 KHz (1600 frequency bins). It provides an estimate of the actual sampling frequency required by each algorithm.

Fig. 7 compares the PA output power spectral density. We see that both complete and simplified CS-CPWL algorithms outperform memory polynomial algorithm in terms of adjacent channel interference (ACI) reduction. Moreover, simplified CS-CPWL presents the best trade-off between complexity and performance.

Fig. 8 depicts the AM-AM characteristics of the PA without predistortion and using the different predistorters. Scattering of samples in Fig. 8(a) clearly shows the presence of memory effects. We see that the complete CS-CPWL predistorter results in the best performance followed by the simplified CS-CPWL while the memory polynomial predistorter has the worst performance. This is coherent with the better modeling capability of the complete CS-CPWL predistorter. This shows again that the simplified CS-CPWL predistorter offers good trade-off between complexity and reduction of memory effects.

6. Conclusions

We introduced efficient adaptive predistortion based on complex simplicial canonical piecewise linear (CS-CPWL) model emphasizing low computational complexity. The performance of the predistorter was verified with the behavioral model of actual high memory PA. Resulting output power spectral density curves show that the CS-CPWL approach offers good trade-off between performance and complexity.

Acknowledgements

This work was partially supported by Universidad Nacional del Sur, Project PGI # 24-K035, Agencia Nacional de Promoción Cientifica y Tecnologica (ANPCyT), Project 21723, and by the Academy of Finland, Smart and Novel Radios (SMARAD) Centre of Excellence.

References

[1] KENNINGTON, P. B. *High Linearity RF Amplifier Design*. Norwood: Artech House, 2000.

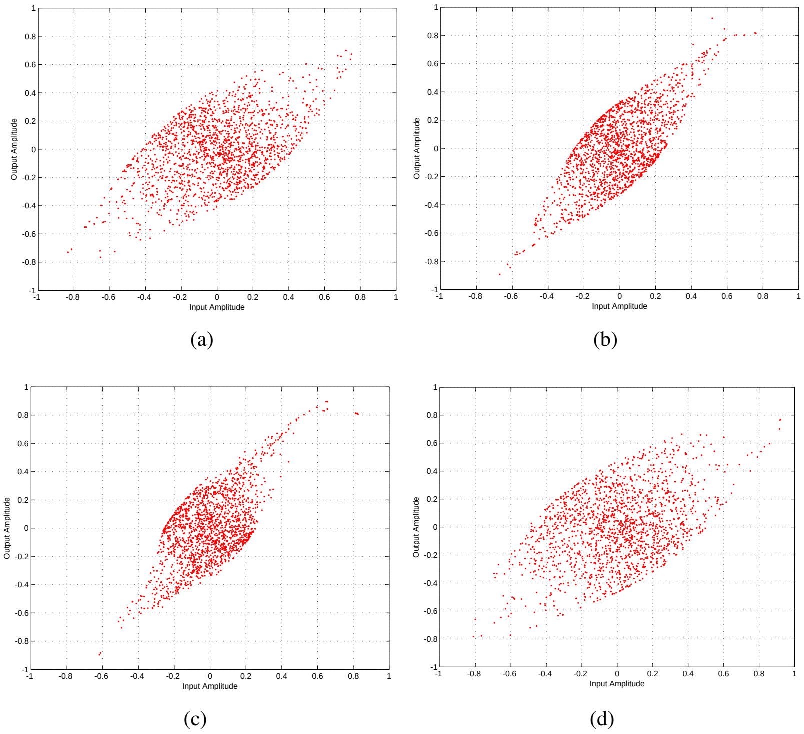
[2] CRIPPS, S. C. *Advanced Techniques in RF Power Amplifier Design*. Norwood: Artech House, 2002.

[3] PEDRO, J. C., CAVALHO, N. B. *Intermodultation Distortion in Microwave and Wireless Circuits*. Artech House, 2003.

[4] ZHU, A., PEDRO, J. C., CUNHA, T. R. Pruning the Volterra series for behavioral modeling of power amplifiers using physical knowledge. *IEEE Transactions on Microwave Theory and Techniques*, 2007, vol. 55, no. 5, p. 813 - 821.

[5] DOYLE, F. J. III, PEARSON, R. K., OGUNNAIKE, B. A. *Identification and Control Using Volterra Models*. Springer, 2001.

[6] SCHETZEN, M. *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.



(a) (b) (c) (d)

Fig. 8. AM-AM characteristic: (a) without predistortion, (b) Complete CS-CPWL, (c) Simplified CS-CPWL, (d) Memory polynomial.

[7] ZHU, A. et al. Open-loop digital predistorter for RF power amplifiers using dynamic deviation reduction-based Volterra series. *IEEE Transactions on Microwave Theory and Techniques*, 2008, vol. 56, no. 7, p. 1524 - 1534.

[8] KIM, J., KONSTANTINOU, K. Digital predistortion of wideband signals based on power amplifier model with memory. *Electronics Letters*, 2001, vol. 37, no. 23, p. 1417 - 1418.

[9] MORGAN, D. R. A generalized memory polynomial model for digital predistortion of RF power amplifiers. *IEEE Transactions on Signal Processing*, 2006, vol. 54, no. 10, p. 3852 - 3860.

[10] COUSSEAU, J., FIGUEROA, J., WERNER, S., LAAKSO, T. I. Efficient nonlinear Wiener model identification using a complex-valued simplicial PWL filter. *IEEE Transactions on Signal Processing*, 2007, vol. 55, no. 5, p. 1780 - 1792.

[11] JULIAN, P., DESAGES, A. C., AGAMENNONI, O. E. High-level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems I.*, 1999, vol. 46, no. 4, p. 463 - 480.

[12] FIGUEROA, J., COUSSEAU, J., DE FIGUEIREDO, R. J. P. A low complexity simplicial canonical piece-wise linear adaptive filter. *Circuits, Systems and Signal Processing Journal*, 2004, vol. 23, no. 5, p. 365 - 386.

[13] DING, L., ZHOU, G. T., MORGAN, D. R., MA, Z., KENNNEY, J. S., KIM, J., GIARDINA, C. R. A robust digital baseband predistorter constructed using memory polynomials. *IEEE Transactions on Communications*, 2004, vol. 52, no. 1, p. 159 - 165.

[14] EUN, C., POWERS, E. J. A new Volterra predistorter based on the indirect learning architecture. *IEEE Transactions on Signal Processing*, 1997, vol. 45, no. 1, p. 223 - 227.

[15] HAGENBLAD, A. *Aspects of the Identification of Wiener Models*. M.Sc. thesis, Elect. Eng. Dept., Linköping: Linköping University, 1999.

[16] WIGREN, T. *Recursive Identification Based on the Nonlinear Wiener Model*, PhD. thesis, Technology Dept., Uppsala: Uppsala University, 1990.

[17] WIGREN, T. Recursive prediction error identification using the nonlinear Wiener model. *Automatica*, 1993, vol. 29, no. 4, p. 1011 - 1025.

[18] KALAFATIS, A. D., WANG, L., CLUETT, W. R. Identification of Wiener-type nonlinear systems in noisy environment. *International Journal of Control*, 1997, vol. 66, p. 923 - 941.

[19] CHEN, G. et al. The stability and convergence characteristics of the delayed-x LMS algorithm in ANC systems. *Journal of Sound and Vibration*, 1998, vol. 216, no. 4, p. 637 - 648.

[20] FEINTUCH, P. L., BERSHAD, N. J., LO, A. K. A frequency domain model for "filtered" LMS algorithms-stability analysis, design, and elimination of the training mode. *IEEE Transactions on Signal Processing*, 1993, vol. 41, no. 4, p. 1518 - 1531.

[21] Agilent Technologies, *Agilent EESoft EDA Advanced Design System*, 2006.

[22] GOLIO, M. *The RF and Microwave Handbook*. CRC Press, 2001.

[23] GAMEZ, G. G. *Measurements for Modelling Wideband Nonlinear Power Amplifiers for Power Communications*. M.Sc. thesis, Dept. of Elec. and Communications Eng., Helsinki: Helsinki University of Technology, 2004.

[24] GILMORE, R., BESSER, L. *Practical RF Circuit Design for Modern Wireless Systems Vol. II*. Artech House, 2003.

[25] BRINKHOFF, J. *Bandwidth Dependent Intermodulation Distortion in FET Amplifiers*. Ph.D. thesis, Sydney: Macquire University, 2004.

[26] BRINKHOFF, J., PARKER, A. E. Effect of baseband impedance on FET intermodulation. *IEEE Transactions on Microwave Theory and Techniques*, 2003, vol. 51, no. 3, p. 1045 - 1051.

[27] ASHRAF, A. et al. Peak-to-average power ratio reduction in OFDM systems using Huffman coding. *Proceddings of World Academy of Science, Engineering and Technology*, 2008, vol.33, p. 266 - 271.

About Authors ...

Marcelo BRUNO was born in Villa Iris, Argentina. He received the B.Sc. degree from Universidad Tecnologica Nacional (UTN), Bahia Blanca, Argentina in 1998, and the M.Sc. degree from the Universidad Nacional del Sur (UNS), Bahia Blanca, Argentina, in 2005. His research interests include the study of nonlinearities in broadband radio frequency power amplifiers and signal processing on high speed circuits. He is currently pursuing his Ph.D. on digital linearization of radio frequency power amplifiers.

Juan COUSSEAU was born in Mar del Plata, Argentina. He received the B.Sc. degree from the Universidad Nacional del Sur (UNS), Bahia Blanca, Argentina, in 1983, the M.Sc. degree from COPPE/ Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1989, and the Ph.D. degree from COPPE/UFRJ, in 1993, all in electrical engineering. Since 1984, he has been with the undergraduate Department of Electrical and Computer Engineering at UNS. He has also been with the graduate program at the same university since 1994. He is a senior researcher of the National Scientific and Technical Research Council (CONICET) of Argentina. He is coordinator of the Signal Processing and Communication Laboratory (LaPSyC) at UNS. He was a visiting professor at the University of California, Irvine, in 1999 and the Signal Processing Laboratory, Helsinki University of Technology, Helsinki, Finland, in 2004 and 2006. Dr. Cousseau was the IEEE Circuits and Systems Chair of the Argentina Chapter from 1997 to 2000 and member of the executive committee of the IEEE Circuits and Systems Society during 2000-2001 (vice-president for Region 9). He is a participant of the IEEE Signal Processing Society Distinguished Lecturer Program 2006.

Stefan WERNER received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998 and the D.Sc. (EE) degree (with honors) from the Signal Processing Laboratory, Smart and Novel Radios (SMARAD) Center of Excellence, Helsinki University of Technology (TKK), Espoo, Finland, in 2002. He is currently an academy research fellow in the Department of Signal Processing and Acoustics, TKK. Dr. Werner is a senior member of the IEEE and a member of the editorial board for the Signal Processing journal. His research interests are in multiuser communications and adaptive signal processing.

José FIGUEROA received the B.Sc. degree in electrical engineering and the Ph.D. degree in systems control from the Universidad Nacional del Sur, Argentina, in 1987 and 1991, respectively. Since 1995, he has been with the Universidad Nacional del Sur and CONICET. His research interests include control systems and signal processing.

Mei Yen CHEONG received both her M.Sc. and Licentiate of Science degrees from the Department of Electrical and Communication Engineering at the Helsinki University of Technology in 2003 and 2006, respectively. She is currently a Ph.D. student at the Department of Signal Processing and Acoustics, Helsinki University of Technology. Her research interests lie in the area of predistortion linearizers and peak-to-average power ratio reduction techniques.

Risto WICHMAN received his M.Sc. and D.Sc. (Tech) degrees in digital signal processing from Tampere University of Technology, Tampere, Finland, in 1990 and 1995, respectively. From 1995 to 2001, he worked at Nokia Research Center as a senior research engineer. In 2002, he joined the Department of Signal Processing and Acoustics, Helsinki University of Technology, where he is a professor since 2003. His major research interests include transceiver algorithms for wireless communication systems.