Adaptive Piecewise Linear Predistorters for Nonlinear Power Amplifiers With Memory

Mei Yen Cheong, Stefan Werner, *Senior Member, IEEE*, Marcelo J. Bruno, Jose L. Figueroa, *Senior Member, IEEE*, Juan E. Cousseau, *Senior Member, IEEE*, and Risto Wichman

***Abstract*—We propose novel direct and indirect learning predistorters (PDs) that employ a new baseband simplicial canonical piecewise linear (SCPWL) function. The performance of the proposed PDs is easily controlled by varying the number of segments of the SCPWL function. When comparing to polynomial-based PDs, our SCPWL-based PDs are more robust for modeling strong nonlinearities and are less sensitive to input noise. In particular, we show that noise appearing in the feedback path of an indirect learning SCPWL-PD has negligible effect on the performance while the polynomial counterpart suffers from a noise-induced coefficient bias. We consider adaptive implementations of both Hammerstein-based and memory-based SCPWL PDs; the former featuring less parameters to be identified while the latter renders more straightforward parameter identification. When deriving the PD algorithms, we avoid a separate PA identification step which allows for a true real-time, or sample-by-sample, implementation without an alternating PA and PD identification procedure. However, to arrive at efficient sample-by-sample algorithms for Hammerstein PDs we need to bypass the problem of the associated nonconvex cost function. This is done by employing a modified, linear-in-the-parameter, Wiener model whose parameters can be explicitly or implicitly used for both indirect and direct learning. Extensive simulations confirm that the proposed SCPWL PDs outperform their polynomial counterparts, especially when noise is present in the feedback path of the indirect learning structure. The same is also verified by circuit level simulations on the Freescale MRF6S23100H class-AB PA in an 802.16d WiMAX system.**

***Index Terms*—Adaptive predistorter, digital predistorter, direct learning, indirect learning, piecewise linear function, power amplifier with memory.**

I. Introduction

MODERN communications systems such as the third generation (3G) system, the long term evolution of 3G system (3G-LTE), and WiMAX all emphasize spectral efficiency and power efficiency. These systems employ linear modulation techniques to fill more bits into the available spectrum and multicarrier transmission schemes such as orthogonal frequency-division multiplexing (OFDM) to gain spectral efficiency. The drawback of the OFDM scheme is its high peak-to-average power ratio (PAPR). High PAPR signals require the power amplifier (PA) to be operated in its linear region to avoid nonlinear distortion due to excessive signal clipping and compression. In wideband systems, memory effects result in a nonflat PA transfer function. Effects of nonlinear distortion include inband signal distortion and spectral spreading into the adjacent channels. Spectral spreading has to be mitigated at the transmitter side to avoid adjacent channel interference (ACI). On the other hand, linear PAs are power inefficient. For battery operated devices, power efficiency helps prolong battery life. In mobile network base stations, the PA systems account for more than 50% of the total power consumption [1]. Thus, power efficiency of the PA is crucial for effective operational cost savings as well as for alleviation of heat dissipation problems.

Linearization techniques can be employed to achieve a balance between power efficiency and linearity. Linearizers alter the transmitted signal so that the resulting output of the PA is linear. They can be realized using extra circuitry or by signal processing methods. Recent examples of the former include the reverse MM-LINC technique [2] and an adaptive predistortion employing a $\Delta\Sigma$ modulator [3]. Digital predistortion is a method most commonly implemented by a signal processing approach. For narrowband systems, memoryless predistorters (PDs) can be employed [4], [5]. For broadband systems, PDs that compensate nonlinearity with memory are required, e.g., [6]–[11], and are also considered in this paper. Recent developments of digital PDs have mainly focused on the following two areas:

Firstly, many recent papers have introduced efficient memory structures into the PD models to enable linearization of broadband PAs that exhibit memory effects. Although the Volterra model is known as the most complete nonlinear model with memory [12], its complexity prohibits its practical application. Block models such as the Wiener and Hammerstein systems and their variants are often used for PA and PD modeling [6], [13]–[15] as they can be described by much less parameters. However, parameter estimation of these models is often complicated by their nonconvex cost function. Simplified Volterra models, e.g., memory polynomial models [7], [8], [16], generalized memory polynomial [9], or the dynamic deviation reduction-based Volterra model [10] are linear-in-the-parameter models that have been used for efficient modeling of PA and PD with memory. Recently, a more general expression formulated using sum of separable functions has emerged [11], in which the Volterra model and its variants mentioned above are shown to be special cases of the expression, where polynomials are used as the bases to form the separable functions. However, a drawback of polynomial model is the numerical problem associated with higher order polynomials. Thus, polynomials are limited for modeling mild nonlinearities. Various orthogonal polynomials have been proposed [17], [18] for alleviating the numerical problem, but with the trade-off of having nonlinear basis functions which are more complex to generate.

Manuscript received July 14, 2011; revised September 19, 2011; accepted October 21, 2011. Date of publication January 18, 2012; date of current version June 22, 2012. This work was supported in part by Universidad Nacional del Sur, Project PGI # 24-K035 and by the Academy of Finland, Smart Radios and Wireless Research (SMARAD) Center of Excellence. This paper was recommended by Associate Editor C.-C. Tseng.

M. Y. Cheong, S. Werner, and R. Wichman are with the School of Electrical Engineering, Department of Signal Processing & Acoustics, Aalto University, 00076 Aalto, Finland (e-mail: meiyen.cheong@aalto.fi; stefan.werner@aalto.fi; risto.wichman@aalto.fi).

M. J. Bruno, J. L. Figueroa, and J. E. Cousseau are with the CONICET-Department of Electrical and Computer Engineering, Universidad Nacional del Sur, Bahía Blanca, 8000 Argentina (e-mail: mbruno@criba.edu.ar; figueroa@uns.edu.ar; jcousseau@uns.edu.ar).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2011.2177007

Secondly, adaptive PD algorithms that allow for tracking of time-varying PA characteristics are becoming more important in modern communications systems. In the literature, PD adaptation algorithms are mainly based on the indirect learning [7], [19] and direct learning [13], [20], [21] approaches. The former is inherently adaptive to fluctuation in the PA characteristics. It identifies the postinverse model of the PA which is directly used for predistortion. Its computational complexity is lower compared to the latter. A major concern for the method is the effect of measurement noise due to downconversion and analog-to-digital (A/D) conversion, at the feedback path. Due to the nonlinear basis functions of polynomial models, noise causes coefficient bias effects [9] in polynomial PDs, leading to spectral regrowth at the PA output [9], [21], [22]. The direct learning method identifies the preinverse model of the PA [13], [21], [23] and measurement noise is not a concern. However, it requires explicit knowledge of the PA model. The need for PA model estimation and input signal filtering with the PA model significantly increase the computational complexity. In addition, an online or periodically active PA model estimator is required to enable tracking of changes in the PA characteristics. Due to the complexity of implementation, some authors assumed the PA model to be known [20], [21] or is identified offline [13]. In [23], in order to reduce the computational complexity, the PA static nonlinearities are modeled using piecewise linear (PWL) functions. However, the AM/AM and AM/PM functions are modeled as two separate PWL functions.

In this paper, we propose novel complex-valued simplicial canonical piecewise linear (SCPWL) [24] functions, namely a static SCPWL model and a memory-SCPWL model, that are suitable for modeling baseband nonlinearities. Unlike previous real-valued PWL functions used for modeling PA and PD (e.g., see [23]), the proposed baseband SCPWL functions are able to capture both amplitude and phase information in a single function. The proposed SCPWL models are more robust for modeling strong nonlinearities as they do not exhibit numerical problem as higher order polynomials. Their modeling capability can be improved by increasing the number of breakpoints. In terms of intermodulation distortion (IMD) products, it has been shown that the SCPWL model is capable of modeling infinite IMD products [25]. Unlike polynomial models, its modeling capability does not degrade but saturates when the amount of breakpoints exceeds an optimum number. It also allows flexible distribution of the breakpoints to better fit a given shape of a nonlinearity [26].

The proposed SCPWL models are used for implementing a Hammerstein-SCPWL PD and a memory-SCPWL PD, respectively. We derive both the indirect and direct learning algorithms for adaptive identification of the PD parameters. To overcome the nonconvex cost function problem of the Hammerstein model PD identification, we employ a modified Wiener model estimator. For the direct learning PDs, we incorporate online PA model estimators so that the PA characteristics variation can be tracked. We present a solution to simplify the overall complexity of the direct learning Hammerstein PD algorithm. Concerning measurement noise at the feedback path of the indirect learning PD, we show that due to its linear basis function, the SCPWL model does not suffer from noise induced coefficient bias effects which affect polynomials. Thus, noise has negligible effects on the SCPWL PD spectral regrowth compensation performance. We evaluate and compare the proposed SCPWL PDs with existing polynomial PDs [7], [9], [21], first by baseband system-level simulations in MATLAB® followed by circuit level simulations in the Agilent Advanced Design System (ADS) simulator. The PA used in the circuit level simulations was designed in ADS system for WiMAX system based on the Freescale MRF6S23100H LDMOS PA component model. The PA design has been prototyped and validated in a laboratory by IMD test [27].

In Section II, we present the proposed baseband memoryless and memory-SCPWL models. The indirect learning algorithms for the SCPWL PDs are derived in Section III. In the same section, we describe the modified Wiener model estimator that is employed with the Hammerstein PD algorithms in order to circumvent the nonconvex cost function of block models. The section ends with an analysis and comparison of measurement noise effects on the SCPWL and polynomial models. The direct learning algorithms for the proposed SCPWL PDs are derived in Section IV. We present solutions for simplifying the direct learning algorithms, e.g., by exploiting the known structure of the SCPWL function and by employing the modified Wiener model estimator. Section V presents the simulation results. The SCPWL PDs are first evaluated and compared with polynomial PDs in baseband level simulations in MATLAB®. Effects of measurement noise on the indirect learning PDs are also evaluated. Then, the indirect learning memory-SCPWL PD is further evaluated on the broadband PA in 802.16d downlink by circuit level simulations in the ADS simulator. The computational requirements for the indirect and direct learning algorithms are also provided in this section. Finally, conclusions are drawn in Section VI.

II. Baseband Memory SCPWL Function

The SCPWL PDs proposed in the following sections are developed for the case of complex-valued (baseband) input-output signals. In this section, we develop a new baseband SCPWL model by modifying the single-input real-valued SCPWL mapping $\mathbb{R} \rightarrow \mathbb{R}$, which is a particular case of the multidimensional SCPWL mapping [24] given by

$$f(x) = c_0 + \sum_{i=1}^{\sigma-1} c_i \lambda_i(x), \tag{1}$$

where $x \in \mathbb{R}$, σ is the number of breakpoints used for defining the piecewise linear (PWL) segments and $\{c_i\}_{i=0}^{\sigma-1}$ are the real-valued coefficients. The SCPWL basis function is given by

$$\lambda_i(x) = \begin{cases} \frac{1}{2}(x - \beta_i + |x - \beta_i|), & x < \beta_\sigma \\ \frac{1}{2}(\beta_\sigma - \beta_i + |\beta_\sigma - \beta_i|), & x \geq \beta_\sigma \end{cases} \tag{2}$$

where $\beta_1 < \beta_2 < \cdots < \beta_\sigma$, are the user-defined breakpoints.

In order to model baseband signals, we need to modify (1) to include phase information and allowing coefficients c_i to be complex-valued. Furthermore, memory effects originating from past input samples $\{r(n-\ell)\}_{\ell=1}^{L-1}$ are included. The resulting complex-valued SCPWL mapping $\mathbb{C} \rightarrow \mathbb{C}$ can be viewed as a simplified variant of a parallel Hammerstein structure [9]. The output of the memory-SCPWL model is formed by summing the output signals described by a baseband mapping $\mathbb{C} \rightarrow \mathbb{C}$ at the current and $L-1$ past time instants, i.e.,

$$\begin{aligned} f[\mathbf{r}(n)] &= \sum_{\ell=0}^{L-1} \left\{ c_{\ell 0}^* + \sum_{i=1}^{\sigma_\ell - 1} c_{\ell i}^* \lambda_i(|r(n-\ell)|) \right. \\ &\qquad \left. \times \exp(j\varpi_r(n-\ell)) \right\} \\ &= \sum_{\ell=0}^{L-1} \mathbf{c}_\ell^{\mathrm{H}} \boldsymbol{\lambda}[r(n-\ell)], \end{aligned} \tag{3}$$

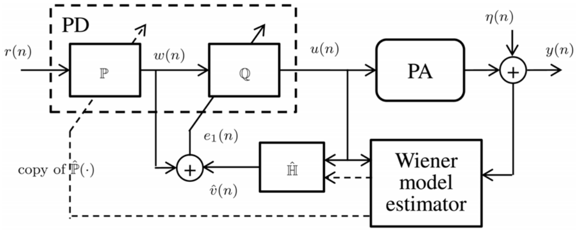

Fig. 1. Indirect learning Hammerstein model PD.

where $\mathbf{r}(n) = [r(n) \quad r(n-1) \quad \cdots \quad r(n-L+1)]^{\mathrm{T}}$ is the vector of input samples, $|r(n)|$ and $\varpi_r(n)$ are the respective amplitude and phase of the baseband signal $r(n) \in \mathbb{C}$, and vectors $\mathbf{c}_\ell \in \mathbb{C}^{\sigma \times 1}$ and $\boldsymbol{\lambda}[r(n)] \in \mathbb{C}^{\sigma \times 1}$ are given by

$$\begin{aligned} \mathbf{c}_\ell &= [c_{\ell 0} \quad c_{\ell 1} \quad \cdots \quad c_{\ell(\sigma_\ell - 1)}]^{\mathrm{T}}, \\ &\qquad\qquad \ell = 0, \ldots, L-1, \\ \boldsymbol{\lambda}[r(n)] &= [1 \quad \lambda_1[|r(n)|]\exp(j\varpi_r(n)) \quad \cdots \\ &\qquad \lambda_{\sigma-1}[|r(n)|]\exp(j\varpi_r(n))]^{\mathrm{T}}. \end{aligned} \tag{4}$$

The term in the curly bracket in (3) has a similar expression as the baseband equivalent of passband quasi-memoryless model (AM/AM and AM/PM conversion) discussed in [14, Sec. IV]. Equation (3) reduces to a memoryless (static) mapping when $L = 1$.

In what follows, we propose a Hammerstein-SCPWL PD and a memory-SCPWL PD. The memoryless mapping is used as a building block in the proposed Hammerstein predistorter. The more general memory mapping, i.e., $L > 1$, is used to model the broadband PD as a single nonlinear function with memory. The memory-SCPWL function is particularly attractive for PD design as it renders simple adaptive predistorter algorithms. This is due to its linear-in-the-parameter expression, which in turn, makes parameter adaptation more straightforward than the Wiener or Hammerstein systems, which is often complicated by the nonconvex cost function problem [26], [28], [29].

III. Indirect Learning SCPWL Predistorters

The indirect learning method is a conceptually simple approach where a learning loop is closed around the PA system. Input and output signals of the PA are fed to the indirect learning filter such that the postinverse of the PA model is identified. Then the parameters are directly copied to the PD block, see, e.g., [7], [19]. In this section, we derive indirect learning algorithms for Hammerstein-SCPWL PD and memory-SCPWL PD. Then, we analyze the effects of noise at the feedback path on the identified polynomial and SCPWL coefficients.

A. Indirect Learning Hammerstein-SCPWL PD

A Hammerstein model PD consists of a static nonlinearity $\mathbb{P}$ followed by a linear filter $\mathbb{Q}$. Fig. 1 illustrates the configuration of the proposed indirect learning Hammerstein model PD. To identify the Hammerstein model PD, the PA is assumed to be a Wiener system, consisting of a linear subsystem $\mathbb{H}(\cdot)$ followed by a static nonlinear subsystem $\mathbb{N}(\cdot)$. In order to avoid the nonconvex cost function problem of Wiener/Hammerstein system identification, we employ a modified Wiener model estimator shown in Fig. 2. The estimator provides the estimates of the postinverse of the PA static nonlinearity $\hat{\mathbb{P}}(\cdot)$ and the PA memory $\hat{\mathbb{H}}(\cdot)$. The estimate $\hat{\mathbb{P}}(\cdot)$ is then directly copied to the nonlinear block of the PD. The estimate $\hat{\mathbb{H}}(\cdot)$ is used for adapting the PD linear filter $\mathbb{Q}(\cdot)$. Thus, the indirect learning Hammerstein PD consists of *two identification algorithms running simultaneously*. The algorithms for the two loops are derived in the following.

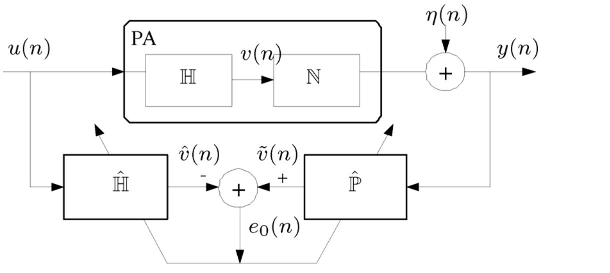

Fig. 2. Modified Wiener model estimator.

1) Loop 1 : Estimation of $\hat{\mathbb{H}}$ *and* $\hat{\mathbb{P}}$*:* Fig. 2 illustrates the modified Wiener model estimator employed in the indirect learning branch. The configuration of the estimator renders a linear-in-the-parameter error equation, leading to a convex cost function [30]. The algorithm estimates the intermediate signal $v(n)$ in Fig. 2 from the input-output signals $\{u(n), y(n)\}$ by minimizing the error $e_0(n) = \tilde{v}(n) - \hat{v}(n)$, where

$$\begin{aligned} \hat{v}(n) &= \sum_{i=0}^{N-1} \hat{h}_i^*(n)u(n-i) \quad \text{and} \\ \tilde{v}(n) &= \hat{\mathbf{c}}^{\mathrm{H}}(n)\boldsymbol{\lambda}[y(n)], \end{aligned} \tag{5}$$

and $\hat{\mathbf{c}}(n)$ and $\boldsymbol{\lambda}[y(n)]$ are defined in Section II. To avoid ambiguity in the filter gain, $\hat{h}_0(n) \equiv \hat{h}_0$ is anchored to a fixed value [26], [28]. The error signal to be minimized can now be written as

$$e_0(n) = \tilde{v}(n) - \hat{v}(n) = \boldsymbol{\theta}^{\mathrm{H}}(n)\boldsymbol{\phi}(n) - \hat{h}_0^* u(n) \tag{6}$$

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

$$\begin{aligned} \boldsymbol{\theta}(n) &= \left[\hat{c}_0(n) \cdots \hat{c}_{\sigma-1}(n), \hat{h}_1(n) \cdots \hat{h}_{N-1}(n)\right]^{\mathrm{T}} \\ \boldsymbol{\phi}(n) &= [\boldsymbol{\lambda}^{\mathrm{T}}[y(n)], \quad -u(n-1) \quad \cdots \quad -u(n-N+1)]^{\mathrm{T}}, \end{aligned} \tag{7}$$

and vector $\boldsymbol{\lambda}[y(n)] \in \mathbb{C}^{\sigma \times 1}$ is defined by (4). Using the instantaneous squared error $|e_0(n)|^2$ as an objective function, the stochastic gradient algorithm that updates $\boldsymbol{\theta}(n)$ is given by

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \mu_0 \boldsymbol{\phi}(n) e_0^*(n), \tag{8}$$

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 [31]

$$0 < \mu_0 < \frac{1}{\varrho_{\max}}, \tag{9}$$

where $\varrho_{\max}$ is the maximum eigenvalue of $\mathrm{E}[\boldsymbol{\phi}(n)\boldsymbol{\phi}^{\mathrm{H}}(n)]$. A more practical step size range is obtained by replacing $\varrho_{\max}$ with its upper bound

$$\begin{aligned} \varrho_{\max} &\leq \mathrm{tr}(\mathrm{E}[\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{H}}]) = \mathrm{tr}(\boldsymbol{\lambda}[y(n)]\boldsymbol{\lambda}^{\mathrm{H}}[y(n)]) \\ &\qquad + \mathrm{E}\left[\sum_{i=1}^{N-1} |u(n-i)|^2\right] \\ &\leq 1 + \sum_{i=1}^{\sigma-1} (\beta_\sigma - \beta_i)^2 + (N-1)\sigma_u^2, \end{aligned} \tag{10}$$

where σ_u^2 is the variance of the signal $u(n)$. The inequality in (10) resulted from the fact that the first entry of $\boldsymbol{\lambda}[y(n)]$ is bounded in magnitude by 1, and the rest by $(\beta_\sigma - \beta_i)$. Equation (10) indicates that the PWL partition sizes of $\hat{\mathbb{P}}(\,\cdot\,)$, the filter length N of $\hat{\mathbb{H}}$ and the input signal power affect the adaptation step size, and therefore the convergence speed of the estimator. For example, for a given N and a fixed set of $\boldsymbol{\beta}$, increasing the input signal power renders a smaller upper bound for μ_0, leading to slower convergence.

Remark 1: Considering an SCPWL PD, the PD output $u(n)$ is bounded in amplitude by the saturation imposed by the function following the last breakpoint, see (2). Thus, there is a practical limit for the upper bound of μ_0 which is determined by the saturation level and the gain of the linear filter $\mathbb{Q}(\,\cdot\,)$. In contrast, the polynomial PD has an expanding characteristic without a saturation level. In case of high level input, μ_0 has to be very small in order to avoid algorithm instability, which in turn, leads to very slow convergence.

2) Loop 2 : Adaptation of the PD Linear Filter $\mathbb{Q}$: The linear filter $\mathbb{Q}(\,\cdot\,)$ of the PD shall equalize the memory $\mathbb{H}(\,\cdot\,)$ of the Wiener model PA. In other words, the intermediate signals $w(n)$ and $\hat{v}(n)$ in Figs. 1 should ideally be identical. Since the Wiener model estimator provides us with an estimate $\hat{\mathbb{H}}(\,\cdot\,)$, we can reproduce an estimate $\hat{v}(n)$ through (5). By forming the error

$$\begin{aligned} e_1(n) = \hat{v}(n) - w(n) &= \sum_{i=0}^{N-1} \hat{h}_i^*(n) u(n-i) - w(n) \\ &= \sum_{i=0}^{N-1} \hat{h}_i^*(n)\{\mathbf{q}^{\mathrm{H}}(n)\mathbf{w}(n-i)\} - w(n), \end{aligned} \tag{11}$$

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

$$\begin{aligned} \mathbf{q}(n+1) &= \mathbf{q}(n) - \mu_1 e_1^*(n) \sum_{i=1}^{N-1} \hat{h}_i^*(n) \frac{\partial u(n-i)}{\partial \mathbf{q}^*(n)} \\ &\approx \mathbf{q}(n) - \mu_1 e_1^*(n) \sum_{i=1}^{N-1} \hat{h}_i^*(n) \mathbf{w}(n-i) \end{aligned} \tag{12}$$

where $\mathbf{w}(n) = [w(n) \cdots w(n-M+1)]^{\mathrm{T}}$. The last approximation in (12) is valid for sufficiently small value of μ_1 so that $\mathbf{q}(n) \approx \mathbf{q}(n-i)$ for $i = 1, \cdots, N-1$.

Equations (8) and (12) constitute the indirect learning Hammerstein PD algorithm. Note that (12) is a filtered-x LMS algorithm. Thus, the stability of the recursion in (12) depends on the quality of the estimates $\{\hat{h}(n)\}_{i=0}^{N-1}$. To ensure stability, the phase response error between the estimate and the actual PA dynamics must be within the range $-(\pi)/(2)$ and $(\pi)/(2)$ [32], [33].

B. Indirect Learning Memory-SCPWL PD

Fig. 3 illustrates the indirect learning configuration for the memory-SCPWL PD. The postinverse of the broadband PA is modeled using the memory-SCPWL model in (3). Following from Fig. 3, the output of the memory-SCPWL filter $\hat{\mathbb{P}}(\,\cdot\,)$ is given by

$$\hat{u}(n) = \sum_{\ell=0}^{L-1} \hat{\mathbf{c}}_\ell^{\mathrm{H}}(n) \boldsymbol{\lambda}[y(n-\ell)]. \tag{13}$$

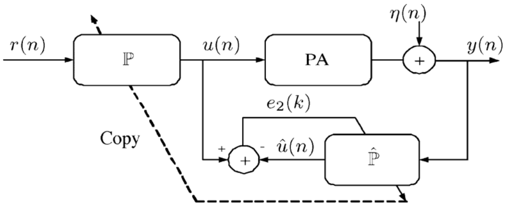

Fig. 3. Indirect learning memory SCPWL PD.

Then, the error signal is given by

$$e_2(n) = u(n) - \sum_{\ell=0}^{L-1} \hat{\mathbf{c}}_\ell^{\mathrm{H}} \boldsymbol{\lambda}[y(n-\ell)]. \tag{14}$$

The adaptation of the memory-SCPWL PD is greatly simplified compared to that of the Hammerstein PD, as the filtering problem is linear in the parameters. By minimizing the the instantaneous squared error $|e_2(n)|^2$ with respect to $\hat{\mathbf{c}}_\ell(n)$, we obtain the update equation

$$\hat{\mathbf{c}}_\ell(n+1) = \hat{\mathbf{c}}_\ell(n) + \mu_2 e_2^*(n) \boldsymbol{\lambda}[y(n-\ell)], \tag{15}$$

where $\ell = 0, \ldots, L-1$. Finally, the memory-SCPWL PD $\mathbb{P}(\,\cdot\,)$ is updated by directly copying the postinverse filters as $\mathbf{c}_\ell(n) = \hat{\mathbf{c}}_\ell(n-1)$ for $\ell = 0, \ldots, L-1$.

The computational complexity of implementing the indirect learning algorithms for the two SCPWL PDs are presented in Table II.

C. Measurement Noise Effects

Measurement noise at the PA output, which constitutes perturbations due to down-conversion and A/D conversion is shown to cause errors in the identified polynomial PD coefficients, known as coefficient bias effect [22] caused by the nonlinear model. Here, we examine the disturbance in the mean amplitude of each noise corrupted basis of the polynomial and SCPWL models.

In the context of predistortion, we assume in the following that convergence has taken place. Furthermore, the step size is assumed sufficiently small so that effects due to parameter copying (associated with indirect learning) can be neglected. We also assume that the input signal is backed off sufficiently so that no signal is clipped by the PD nor PA. Then, $y(n)$ is Gaussian distributed with zero mean and variance σ_y^2. Its amplitude $|y(n)|$, denoted by $x(n)$ follows the Rayleigh distribution with parameter $\sigma_x = \sqrt{(\sigma_y^2)/(2)}$.

1) Polynomial Model: The mean amplitude of the kth basis $\psi_k[x(n)] = x^k(n)$ can be expressed as [18]

$$E[\psi_k[x(n)]] = \begin{cases} \frac{k}{2}!(\sigma_x)^k, & k \text{ is even} \\ \left(\sqrt{\frac{\pi}{2^{k+1}}}\right) k!!(\sigma_x)^k, & k \text{ is odd,} \end{cases} \tag{16}$$

where $k!! = (1 \cdot 3 \cdot \cdots \cdot k)$. Let $\tilde{x}(n) = |y(n) + \eta(n)|$ be the noisy input signal where $\eta(n)$ is a zero-mean Gaussian noise process with variance σ_η^2. Then,

$$\begin{aligned} &E[\psi_k[\tilde{x}(n)]] - E[\psi_k[x(n)]] \\ &= \begin{cases} \left(\frac{k}{2}\right)! \left[\sqrt{\sigma_y^2 + \sigma_\eta^2}^{\,k} - \sqrt{\sigma_y^2}^{\,k}\right], & k \text{ is even} \\ \left(\sqrt{\frac{\pi}{2^{k+1}}}\right) k!! \left[\sqrt{\sigma_y^2 + \sigma_\eta^2}^{\,k} - \sqrt{\sigma_y^2}^{\,k}\right], & k \text{ is odd} \end{cases} \end{aligned}$$

$$\approx \begin{cases} \left(\frac{k-1}{2\gamma}\right) \frac{k}{2}!(\sigma_y)^k, & k \text{ is even} \\ \left(\frac{k-1}{2\gamma}\right) \left(\sqrt{\frac{\pi}{2^{k+1}}}\right) k!!(\sigma_y)^k, & k \text{ is odd} \end{cases} .$$
$$= \frac{(k-1)}{2\gamma} E[\psi_k[x(n)]], \tag{17}$$

where $\gamma = (\sigma_y^2)/(\sigma_\eta^2)$ denotes signal-to-noise-ratio (SNR). The approximation in the third line is valid for sufficiently high SNR so that terms with $((\sigma_\eta^2)/(\sigma_y^2))^2$ and higher order can be neglected. The last line of (17) shows that the effect of noise on the polynomial basis is a bias that is proportional to the mean of the unperturbed basis and increases with the order of the basis k. Consequently, the identified polynomial coefficients are biased by different factors, with larger bias for higher order coefficients. In effect, this alters the spectral shape of the PD, leading to an increase in out-of-band power at the PA output.

2) SCPWL Model: The mean amplitude of the ith SCPWL basis, derived in the Appendix, is given by

$$E[\lambda_i[a(n)]] = \sigma_y \sqrt{\pi} \left[\Phi_{0,\frac{\sigma_y^2}{2}}(\beta_\sigma) - \Phi_{0,\frac{\sigma_y^2}{2}}(\beta_i) \right]$$
$$= \sigma_y \sqrt{\pi} \times \Pr(\beta_i < x(n) < \beta_\sigma). \tag{18}$$

The expression in the square bracket is the probability that the signal amplitude $a(n)$ falls between β_i and β_σ. For sufficiently high SNR,[1]$\Pr(\beta_i < \tilde{x}(n) < \beta_\sigma) \approx \Pr(\beta_i < x(n) < \beta_\sigma)$. Then from (18), the error in the SCPWL basis caused by input noise can be expressed as

$$E[\lambda_i[\tilde{x}(n)]] - E[\lambda_i[x(n)]]$$
$$= \sqrt{\pi}\left(\sqrt{\sigma_y^2 + \sigma_\eta^2} - \sqrt{\sigma_y^2}\right) \times \Pr(\beta_i < x(n) < \beta_\sigma)$$
$$\approx \frac{1}{2\gamma}\sigma_y\sqrt{\pi} \times \Pr(\beta_i < x(n) < \beta_\sigma)$$
$$= \frac{1}{2\gamma}E[\lambda_i[x(n)]]. \tag{19}$$

The approximation in the third line is valid for sufficiently high SNR, i.e., $\gamma \gg 1$. We see from (19) that the noise-induced bias is only proportional to the unperturbed basis but not the basis number i. In effect, each SCPWL basis is scaled by the factor $(1 + (1)/(2\gamma))$, which also leads to uniform scaling of the coefficients when a stochastic gradient algorithm as in (15) is employed. Thus, the spectral profile of the identified SCPWL PD is not affected by the measurement noise.

IV. Direct Learning SCPWL Predistorters

This section derives the direct-learning versions of the Hammerstein-based and memory-based SCPWL predistorters. Comparing to the indirect learning versions of the previous section, the direct learning algorithms attempt to minimize the more relevant error signal formed by the PA output and the desired output (i.e., a scaled version of the PD input). The error function used for parameter updates depends explicitly on an assumed PA model. As a consequence, direct learning algorithms require knowledge of the PA model. The estimate of the PA model is required to be accurate to within 90° of the actual PA phase response [33], [34] to ensure the algorithm is stable. Our solutions will simultaneously find the necessary PA and PD parameters. In the case of the Hammerstein PD, we show how to avoid a complicated Wiener model identification by exploiting the estimates provided by the modified Wiener model described in Section III, instead.

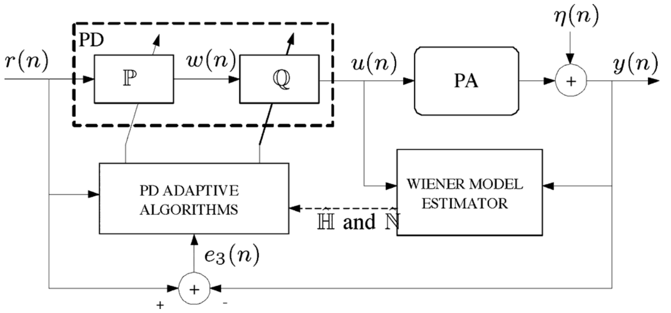

Fig. 4. Direct learning Hammerstein model PD.

A. Direct Learning Hammerstein SCPWL Predistorter

Fig. 4 illustrates the direct learning configuration for the Hammerstein model PD. In the following, the PA is modeled by a Wiener system characterized by the cascade system $\mathbb{H}(\cdot) - \mathbb{N}(\cdot)$ as depicted in Fig. 3. At each algorithm iteration n, estimates of $\hat{\mathbb{H}}$ and $\hat{\mathbb{N}}$ obtained from the Wiener model estimator are used in the PD adaptive algorithms as shown in Fig. 4. Later on we show that the PD algorithm can be implemented using knowledge of $\hat{\mathbb{H}}$ and $\hat{\mathbb{P}}$. This will significantly simplify online adaptation, since we only need to identify the modified Wiener model which is linear in the parameters.

The error signal that drives the direct learning algorithm can be written as

$$e_3(n) = r(n) - y(n)$$
$$\approx r(n) - \hat{\mathbb{N}}(\hat{v}(n)) \tag{20}$$

where $\hat{v}(n)$ is as defined in (5). Note that the approximation of the error signal in (20), which involves $\hat{\mathbb{N}}(\hat{v}(n))$, is used when deriving the adaptive algorithms. By minimizing the instantaneous squared error $|e_3(n)|^2$ with respect to $\mathbf{q}(n)$ and $\mathbf{c}(n)$, and using the chain rule, we obtain the stochastic gradient algorithms

$$\mathbf{q}(n+1) = \mathbf{q}(n) + \mu_3 e_3^*(n) \frac{\partial \hat{\mathbb{N}}(\hat{v}(n))}{\partial \hat{v}(n)} \frac{\partial \hat{v}(n)}{\partial \mathbf{q}^*(n)}$$
$$= \mathbf{q}(n) + \mu_3 e_3^*(n) G(n) \sum_{i=0}^{N-1} \hat{h}_i^*(n) \frac{\partial u(n-i)}{\partial \mathbf{q}^*(n)}$$
$$\approx \mathbf{q}(n) + \mu_3 e_3^*(n) G(n) \sum_{i=0}^{N-1} \hat{h}_i^*(n) \mathbf{w}(n-i), \tag{21}$$

and

$$\mathbf{c}(n+1) = \mathbf{c}(n) - \mu_4 e_3^*(n) \frac{\partial \hat{\mathbb{N}}(\hat{v}(n))}{\partial \hat{v}(n)} \frac{\partial \hat{v}(n)}{\partial \mathbf{c}^*(n)}$$
$$= \mathbf{c}(n) + \mu_4 e_3^*(n) \frac{\partial \hat{\mathbb{N}}(\hat{v}(n))}{\partial \hat{v}(n)}$$
$$\times \sum_{i=0}^{M+N-1} s_i^*(n) \frac{\partial w(n-i)}{\partial \mathbf{c}^*(n)}$$
$$\approx \mathbf{c}(n) + \mu_4 e_3^*(n) G(n)$$
$$\times \sum_{i=0}^{M+N-1} s_i^*(n) \boldsymbol{\lambda}[r(n-i)], \tag{22}$$

[1] Simulations verify that $\gamma \geq 20$ dB is sufficient.

where $G(n) = (\partial \mathbb{N}(\hat{v}(n)))/(\partial \hat{v}(n))$, $\mathbf{s}(n) = \{s_i(n)\}_{i=0}^{N+M-1} = \hat{\mathbf{h}}(n) \otimes \mathbf{q}(n)$, and $\otimes$ stands for convolution. The approximations on the last lines of (21) and (22) are valid for sufficiently small step sizes such that $\mathbf{q}(n) \approx \mathbf{q}(n-i)$ for $i = 1, \cdots, N-1$ and $\mathbf{c}(n) \approx \mathbf{c}(n-i)$ for $i = 1, \cdots, M+N-1$, respectively.

Equations (21) and (22) resemble the nonlinear filtered-x algorithms as in [35], and constitute the adaptive algorithms for the direct-learning Hammerstein model PD. We see that the computation of $G(n)$ requires the knowledge of $\mathbb{N}$. Estimation of $\mathbb{N}$ is not attractive from the perspective of computational complexity. Our solutions are summarized in the following remarks.

Remark 2: To reduce the overall complexity of the direct learning Hammerstein PD in (21) and (22), we may employ the modified Wiener model estimator described in Section III, which provides us with $\hat{\mathbb{H}}$ and $\hat{\mathbb{P}}$. Although an estimate of $\mathbb{N}$ is not available, we recognize that $\hat{\mathbb{P}}$ approximates the inverse of $\mathbb{N}$. By using the fact that the derivative of inverse functions are reciprocals of each other we may use the parameters of $\hat{\mathbb{P}}$, see (8), to obtain the following approximation of $G(n)$ (see Appendix for details)

$$\hat{G}(n) = \begin{cases} \left[\frac{\hat{v}(n)}{y(n)} + \sum_{i=1}^{N_s} c_i^*(n)\right]^{-1}, & |y(n)| \leq \beta_\sigma \\ \frac{y(n)}{\hat{v}(n)}, & |y(n)| > \beta_\sigma \end{cases}, \quad (23)$$

where N_s specifies the PWL segment number that $|\hat{v}(n)|$ falls into. In (23) c_i are the parameters of the PD static nonlinearity, where the first two parameters c_0 and c_1 determine the small signal gain of the PD. In our simulations, the initial value of c_1 is chosen to be 1 and 0 for other c_i's. This in effect gives a linear response of gain 1, which means that initially (begining of the first iteration), the PD does not change the signal that passes through it.

Remark 3: The complex-valued gain $G(n)$ can be dropped at the expense of a decrease in convergence speed [35].

B. Direct Learning Memory SCPWL Predistorter

Fig. 5 illustrates the direct learning configuration for the memory-SCPWL PD. In the following we assume that the PA system is accurately modeled as a memory-SCPWL system with memory length L_1. That is, the PA model output may be written as

$$\hat{\mathbb{N}}[\mathbf{u}(n)] = \sum_{\ell_1=0}^{L_1-1} \mathbf{a}_{\ell_1}^{\mathrm{H}} \boldsymbol{\lambda}[u(n-\ell_1)]. \quad (24)$$

Since the PA model leads to a linear-in-the-parameter error equation, we can directly formulate the stochastic gradient algorithm for updating PA model parameter vectors $\{\mathbf{a}_{\ell_1}\}_{\ell_1=0}^{L_1-1}$ via input-output measurements $[\{u(n-\ell_1)\}_{\ell_1=0}^{L_1-1}, y(n)]$

$$\begin{aligned} \mathbf{a}_{\ell_1}(n+1) &= \mathbf{a}_{\ell_1}(n) + \mu_5 \epsilon_0^*(n) \boldsymbol{\lambda}[u(n-\ell_1)] \\ \epsilon_0(n) &= y(n) - \sum_{\ell_1=0}^{L_1-1} \mathbf{a}_{\ell_1}^{\mathrm{H}}(n) \boldsymbol{\lambda}[u(n-\ell_1)] \end{aligned} \quad (25)$$

The PA model estimates obtained through (25) are fed to the direct learning algorithm of the PD that will be derived next.

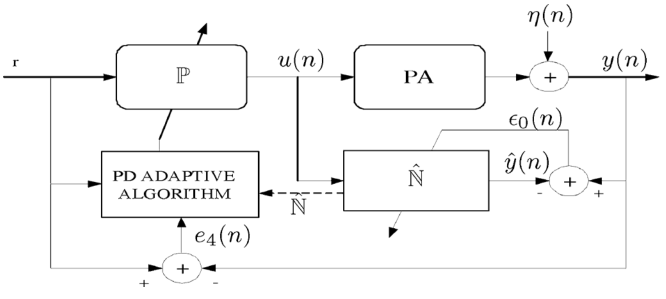

Fig. 5. Direct learning memory SCPWL PD.

Let the memory length of the memory-SCPWL PD be L_2, and PD parameter vectors $\{\mathbf{c}_{\ell_2}\}_{\ell_2=0}^{L_2-1}$. Then, the PD output signal is expressed as

$$u(n) = \sum_{\ell_2=0}^{L_2-1} \mathbf{c}_{\ell_2}^{\mathrm{H}} \boldsymbol{\lambda}[r(n-\ell_2)]. \quad (26)$$

The error signal for the PD adaptive algorithm can be expressed in terms of the PA model as

$$e_4(n) = r(n) - y(n) \approx r(n) - \sum_{\ell_1=0}^{L_1-1} \mathbf{a}_{\ell_1}^{\mathrm{H}} \boldsymbol{\lambda}[u(n-\ell_1)]. \quad (27)$$

By minimizing the instantaneous squared error $|e_4(n)|^2$ with respect to $\mathbf{c}_{\ell_2}(n)$, the stochastic gradient algorithm for adapting the PD is given by

$$\begin{aligned} \hat{\mathbf{c}}_{\ell_2}(n+1) &= \hat{\mathbf{c}}_{\ell_2}(n) + \mu_6 e_4^*(n) \sum_{\ell_1=0}^{L_1-1} \mathbf{a}_{\ell_1}^{\mathrm{H}} \frac{\partial \boldsymbol{\lambda}[u(n-\ell_1)]}{\partial \mathbf{c}_{\ell_2}^*(n)} \\ &= \hat{\mathbf{c}}_{\ell_2}(n) + \mu_6 e_4^*(n) \sum_{\ell_1=0}^{L_1-1} G_{\ell_1}(n) \boldsymbol{\lambda}[r(n-\ell_1-\ell_2)], \end{aligned} \quad (28)$$

where the last line is valid for sufficiently small μ_6 such that $\mathbf{c}_{\ell_2}(n) \approx \mathbf{c}_{\ell_2}(n-\ell_1-\ell_2)$ for $(\ell_1+\ell_2) = 0, \cdots, (L_1+L_2-1)$. Here, the PA parameters are available and using (36) the derivative of the PA model is given by

$$\hat{G}_{\ell_1}(n) = \begin{cases} \frac{\mathbf{a}_{\ell_1}^{\mathrm{H}}(n)\boldsymbol{\lambda}[u(n-\ell_1)]}{u(n-\ell_1)} & \\ \quad + \sum_{p=1}^{N_{\ell_1}} a_{\ell_1,p}^*(n), & |u(n-\ell_1)| \leq \beta_\sigma \\ \frac{\mathbf{a}_{\ell_1}^{\mathrm{H}}(n)\boldsymbol{\lambda}[u(n-\ell_1)]}{u(n-\ell_1)}, & |u(n-\ell_1)| > \beta_\sigma \end{cases}, \quad (29)$$

where N_{ℓ_1} is the segment number that $|u(n-\ell_1)|$ falls into.

Equation (28) constitutes the adaptive algorithm of the direct learning memory-SCPWL PD and (25) is the online PA model estimator.

Remark 4: Unlike the case of a static nonlinearity, $\{G_{\ell_1}(n)\}_{\ell_1=0}^{L_1-1}$ cannot be omitted from (28). It contains the coefficients filtering $\boldsymbol{\lambda}[r(n-\ell_1-\ell_2)]$ as well as the phase response information of the PA which is crucial for the algorithm stability.

V. Simulations

In this section we evaluate the performance of the proposed SCPWL PDs by simulations, first in MATLAB environment, followed by circuit level simulations in the Agilent Advanced Design System (ADS) simulator. In Section V-A, we confirm our analysis results on the effect of measurement noise at the feedback path of the indirect learning PDs by MATLAB simulations. We demonstrate how noise affect the adjacent channel power ratio (ACPR) performance of the memory-SCPWL PD and memory polynomial PD as the number of PD parameters is increased. Then, we compare the performance of the proposed SCPWL PDs with polynomial PDs with similar model structures, i.e., memory-SCPWL PD with memory polynomial [7] and generalized memory polynomial (GMP) PDs [9], and, Hammerstein-SCPWL PD with Hammerstein polynomial PD [37], [38]. The performance comparison results for the indirect and direct learning PDs are presented in Sections V-B and V-C, respectively.

It will be evident from the MATLAB simulation results that the memory-SCPWL PD provides robust performance, whether adapted by the indirect or direct learning methods. Although the direct learning Hammerstein-SCPWL PD shows as good performance as the memory-SCPWL PD, it is shown in Section V-E that it requires 70% more multiplications than the indirect learning memory-SCPWL PD. Thus, in Section V-D, the indirect learning memory-SCPWL PD is further evaluated on a circuit level LDMOS PA in WiMAX 802.16d system in the ADS simulator.

The PD performances are evaluated based on the ACPR and the amount of inband distortion it compensates. The ACPR performance is observed through the spectrum of the linearized PA output. The compensation of inband distortion is evaluated by the error vector magnitude (EVM) given in (30).

$$\mathrm{EVM} = 10\log_{10}\left\{\sqrt{\frac{\sum_{n=1}^{\aleph}|\hat{Z}_n - D_n|^2}{\sum_{n=1}^{\aleph}|D_n|^2}}\right\}\mathrm{dB}, \tag{30}$$

where $\hat{Z}_n$ and D_n are the received symbols and reference symbols, respectively, and $\aleph$ is the total number of QAM symbols used in the calculation.

A. Measurement Noise and Model Order

In all the MATLAB simulations, the transmitted signal is an OFDM signal generated by 128 subcarriers with carrier spacing of 50 kHz, each modulated by 16-QAM data. The OFDM signal was oversampled by 8 times and a root-raised cosine (RRC) filter with rolloff factor $\alpha = 0.2$ was used for pulse shaping. The PAPR of the OFDM signal is approximately 12 dB. In the simulations, the input signal is backed off by approximately 12 dB from the PA saturation point so that the peak amplitude of the signal falls around the PA saturation point. The linearization performances are measured by averaging the ACPR and EVM of 500 linearized OFDM symbols over 10 realizations.

In this part, the PA is implemented as a Wiener system characterized by a minimum phase filter [36] followed by a static nonlinearity given by

$$H_{\mathrm{W}}(z) = 0.7692 + 0.1538z^{-1} + 0.07691z^{-2},$$
$$\mathbb{N}_{\mathrm{W}}[v(n)] = \frac{2|v(n)|}{1+|v(n)|^2}e^{j\left[\angle v(n)+\frac{3.5|v(n)|^2}{1+25|v(n)|^2}\right]}. \tag{31}$$

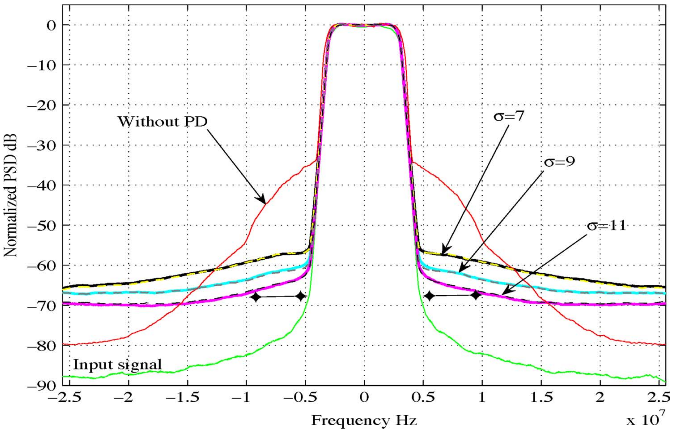

Fig. 6. ACPR performance of the indirect learning memory-SCPWL PDs with $\sigma = 7, 9, 11$. The solid plots indicate SNR = 40 dB and dotted plots indicate $\mathrm{SNR} = 50$ dB. As the number of breakpoints increases, the performance of the PD improves. The effect of measurement noise is insignificant on the memory-SCPWL PD.

As noise effect is more pronounced in models with memory, we use the memory-SCPWL PD and memory polynomial PD for this experiment. The noise at the feedback path $\eta(n)$ is set so that the SNR at the PA output is 30 dB, 40 dB, or 50 dB. The memory-SCPWL PD is implemented with $\sigma = 7, 9, 11$ and memory length $L = 3$ and identified by the indirect learning algorithm in (15). In all the simulations, the breakpoints of the SCPWL PDs are evenly distributed between zero and the saturation point of the PA.[2] The PD parameter were initialized to $c_{01} = 1$ and zero for all other coefficients. Adaptation step size $\mu_2 = 0.1$ was employed when the SNR at the PA output was 30 dB and $\mu_2 = 0.5$ was used otherwise. The memory polynomial PD of order 3, 5, and 7, and memory length 4, are simulated. The memory length is chosen so that the ACPR performance (at the immediate adjacent channels) at SNR 40 dB is almost the same as that of the memory-SCPWL PD. The parameters are identified using the damped-Newton algorithm as in [9]. The PDs are identified using data block of 5 OFDM symbols in each iteration for 35 iterations. The PD parameters were initialized such that the first order, zero memory coefficient was one and all other coefficients were zero. Adaptation step size of 0.4 was employed for adapting the 7th order PD and step size 0.3 were used for adapting PDs of orders 3 and 5.

Fig. 6 shows the ACPR performance of the memory-SCPWL PD as the number of breakpoints is increased from $\sigma = 7$ to $\sigma = 11$. The solid and dotted plots indicate the performance of the PDs when SNR at the feedback path is 40 dB and 50 dB, respectively. It can be seen that the effect noise on SCPWL PD performance is negligible. In contrast, at $\mathrm{SNR} = 40$ dB, increasing the polynomial order from 5 to 7 results in performance degradation as shown in Fig. 7. It can also be seen that, the performance degradation of the 7th order PD is much more significant than the 5th order PD when the noise level is increased, which indicates that noise effect is more pronounced in higher order polynomial.

Fig. 8 shows the average normalized adjacent channel power (ACP) versus SNR at the PA output for the memory-SCPWL PD and memory-polynomial PD. The ACPs are calculated from the

[2] If *a priori* knowledge of the nonlinearity shape is available, more breakpoints can be used in regions where slope of the nonlinearity changes rapidly than regions which are almost linear ([26, Ch. 3]). Otherwise, common practice is to use uniform partition.

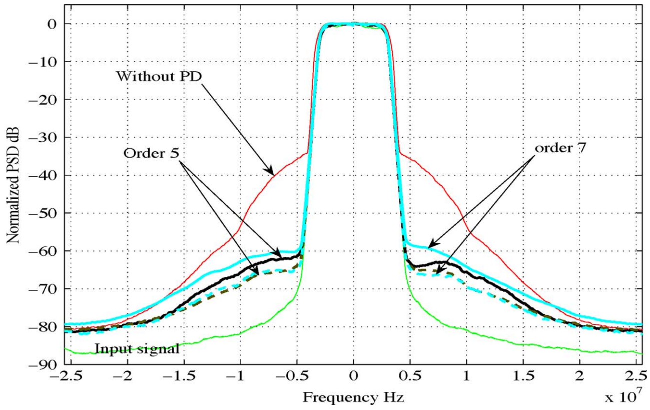

Fig. 7. ACPR performance of the indirect learning memory polynomial PDs of order 5 and 7. The solid plots indicate $\mathrm{SNR} = 40$ dB and dotted plots indicate $\mathrm{SNR} = 50$ dB. The noise effects are more pronounced on the higher order memory-polynomial PD, as can be seen in the performance degradation of the 7th order PD at SNR 40 dB.

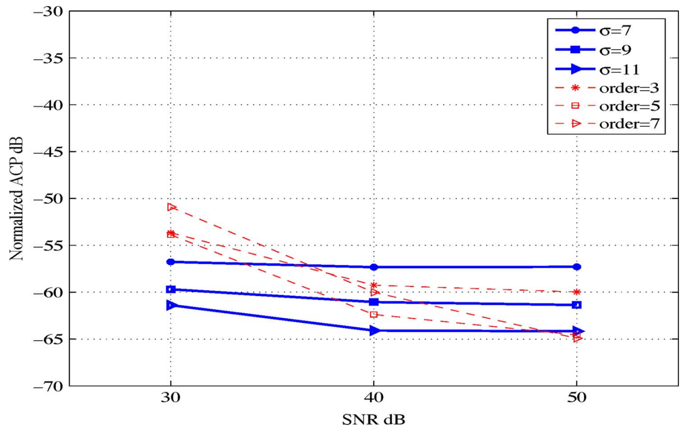

Fig. 8. Normalized ACP of linearized PA output vs. SNR at the PA output. Solid lines and dashed lines represent the performance of the memory-SCPWL PD and memory polynomial PD, respectively.

average power of the immediate adjacent channels indicated by the double arrows in Fig. 6. For different SNR levels at the PA output, it is shown that as the number of breakpoints σ of the memory-SCPWL PD increases, the ACP decreases, see Fig. 6. In contrast, when the SNR at the PA output is low, the ACP at the linearized PA output is higher as the order of the polynomial increases, see Fig. 7. This shows that noise has insignificant effect on the memory-SCPWL PD performance, whereas, it has increasing effect on the memory-polynomial PD performance as the PD model order is increased.

B. Performance Comparison of Indirect Learning PDs

In this section, we compare the performance of the indirect learning memory-SCPWL PDs with memory polynomial [7] and generalized memory polynomial (GMP) [9] PDs, identified using the "damped" Newton algorithm [9]. For comparison with the Hammerstein-SCPWL PD, we have implemented the Hammerstein-polynomial PDs, identified using the Narrendra-Gallman (NG) method as in [37], [38]. In the following examples, the SNR at the feedback path is 40 dB, unless otherwise specified.

The PDs are evaluated on a Wiener-Hammerstein (W-H) PA consisting of a static nonlinear system $\mathbb{N}_{\mathrm{WH}}$ in between two linear filters $H_{\mathrm{WH1}}(z)$ and $H_{\mathrm{WH2}}(z)$. The linear filters are taken from [7, eq. (8)]. Instead of employing the complete W-H PA in [7], we consider a stronger nonlinearity described by a Saleh model. The ACPR of the unlinearized W-H PA is approximately 35 dB at $\mathrm{OBO} = 12$ dB. The static nonlinearity and linear filters of the W-H PA are as follows:

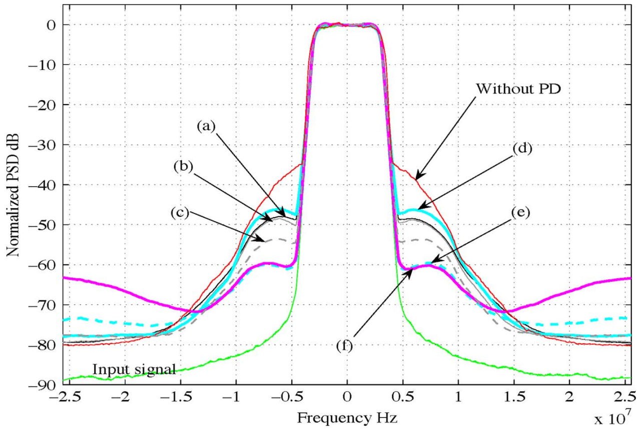

Fig. 9. Comparison of the indirect learning memory-SCPWL PD with various polynomial-based PDs for a W-H model PA. (a) Memory polynomial PD, $K = 5$, odd orders only, $L = 4$. (b) Memory polynomial PD $K = 5$, even and odd orders, $L = 4$. (c) Memory polynomial PD $K = 5$, even and odd orders, $\mathrm{SNR} = 60$ dB. (d) GMP PD, $\mathcal{K}_a = [0:4]$, $\mathcal{L}_a = [0:3]$, $\mathcal{K}_b = [2\ 4]$, $\mathcal{L}_b = [0:3]$, $\mathcal{M}_b = [1:3]$. (e) GMP PD at $\mathrm{SNR} = 60$ dB (dashed plot). (f) The proposed indirect learning memory-SCPWL PD, $\sigma = 11$, $L = 3$.

$$\mathbb{N}_{\mathrm{WH}}[v(n)] = \frac{1.33|v(n)|}{1+|v(n)|^2} e^{j\left[\angle v(n) + \frac{3.5|v(n)|^2}{1+25|v(n)|^2}\right]},$$
$$H_{\mathrm{WH1}}(z) = \frac{1+0.5z^{-2}}{1-0.2z^{-1}}, \quad H_{\mathrm{WH2}}(z) = \frac{1-0.1z^{-2}}{1-0.4z^{-1}}. \tag{32}$$

1) Memory-SCPWL PD vs. Memory Polynomial and GMP PDs: The memory-SCPWL PD employs $\sigma = 11$ breakpoints and memory lengths are $L = 3$. Initially, the PD parameter vector are all zero except $c_{01} = 1$, and the adaptation step size $\mu_2 = 0.7$ was used. The memory polynomial PD is of order $K = 5$ and memory length $L = 4$. The GMP PD employs 5th order aligned terms, i.e., $\mathcal{K}_a = [0\ 1\ 2\ 3\ 4]$ and memory length 4, i.e., $\mathcal{L}_a = [0\ 1\ 2\ 3]$. The cross-terms parameters $\mathcal{K}_b = [2\ 4]$, $\mathcal{L}_b = [0\ 1\ 2\ 3]$ and $\mathcal{M}_b = [1\ 2\ 3]$ are employed.[3] The polynomial PD parameters were initialized to 1 for the first parameter (first order and memory $\ell = 0$) and zero for all other parameters. The "damped" Newton algorithm employed data blocks of 5 OFDM symbols, adaptation step size of 0.4 and 30 iterations (convergence achieved between 10–15 iterations).

Fig. 9 shows the ACPR performance of the PDs in linearizing the W-H model PA. When the SNR at the feedback path is 40 dB, the polynomial PDs are only able to suppress the ACP by approximately 15 dB, achieving an ACPR of approximately 50 dB. The memory-SCPWL PD outperforms the polynomial PDs by at least 10 dB, suppressing the ACP to 60 dB below the in-band signal level. By reducing the noise level to $\mathrm{SNR} = 60$ dB, the GMP PD is able to attain a similar performance as the memory-SCPWL PD, while the memory polynomial PD with odd and even order terms attains an ACPR of 55 dB.

Table I shows the EVMs of the unlinearized and linearized W-H PA. The average EVM without predistortion for the W-H PA output is -5.93 dB. The memory-SCPWL PD attains an

[3] Please refer to [9] for detailed description of $\mathcal{K}_a$, $\mathcal{L}_a$, $\mathcal{K}_b$, $\mathcal{L}_b$ and $\mathcal{M}_b$.

TABLE I
LINEARIZED OUTPUT EVM

Indirect learning PDs for W-H PA	
Without PD	-5.93 dB
M-SCPWL PD, $\sigma = 11$, $L = 3$	-18.31 dB
GMP PD, 5th order	-19.21 dB
M-Poly. PD, $K = 5$ odd, $L = 4$	-19.94 dB
M-Poly. PD, $K = 5$, even & odd orders, $L = 4$	-20.06 dB
Hammerstein-SCPWL PD, $\sigma = 11$, $N = 5$, $M = 7$	-22.76 dB
Hammerstein-poly. PD, $K = 5$ odd, $M = 7$	-18.55 dB
Hammerstein-poly. PD, $K = 5$ odd & even, $M = 7$	-18.50 dB
Direct learning PDs for Wiener PA	
Without PD	-10.64 dB
M-SCPWL PD $\sigma = 11$, $L_1 = L_2 = 3$	-24.01 dB
M-Poly. PD, $K = 3$, $L_2 = 5$	-23.70 dB
M-Poly. PD, $K = 5$, $L_2 = 4$	-21.81 dB
M-Poly. PD, $K = 5$, $L_2 = 5$	-23.81 dB
Hammerstein-SCPWL PD	-24.02 dB
Hammerstein-poly. PD, 5th odd order	-23.66 dB
Hammerstein-poly. PD, 5th odd & even orders	-24.18 dB

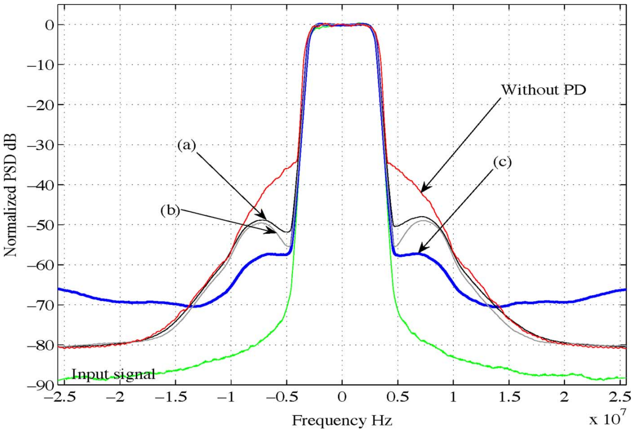

Fig. 10. Comparison of the indirect learning Hammerstein-SCPWL PD with the Hammerstein-polynomial PDs for a W-H model PA. (a) Hammerstein-polynomial PD, $K = 5$, odd orders only, $M = 7$. (b) Hammerstein-polynomial PD, $K = 5$, even and odd orders, $M = 7$. (c) The proposed Hammerstein-SCPWL PD, $\sigma = 11$, $N = 5$, $M = 7$.

EVM of -18.31 dB, compared to approximately -20 dB attained by the memory polynomial PDs and the GMP PDs. An improvement of approximately 2 dB can be obtained by increasing the memory length of the memory-SCPWL PD to $L = 4$, in order to level the EVM performance of the polynomial PDs. However, for a lower computational complexity, the 2 dB can be considered a good tradeoff. Furthermore, it is more crucial to achieve a better ACPR performance than improving EVM performance, which has a less stringent target.

2) Hammerstein-SCPWL PD vs. Hammerstein-Polynomial PD: The Hammerstein-SCPWL PDs are implemented with $\sigma = 11$ breakpoints for the static nonlinear blocks of the PD and PA model estimators. The linear filters of the PD and PA model estimator employ $M = 7$ and $N = 5$, respectively. The initial parameters are all zero except $\hat{h}_0 = 1$, $\hat{c}_1 = 1$, and $q_0 = 1$. For the initial 200 OFDM symbols, the step sizes $\mu_0 = 0.05$ and $\mu_1 = 1e^{-5}$ were used. After the initial adaptation, $\mu_0 = 1e^{-3}$ and $\mu_1 = 0.5$ were employed.

The Hammerstein-polynomial PDs are identified using the Narrendra-Gallman (NG) method [37], [38], employing blocks of data with 5 OFDM symbols in each iteration. The PDs are modeled with fifth order polynomials and linear filter of length $M = 7$. Adaptation step size of 0.2 and 15 iterations were used.

Fig. 10 compares the ACPR performance of the Hammerstein model PDs. The Hammerstein-SCPWL PD reduces the adjacent channel power to 58 dB below the inband signal level compared to ACPR of approximately 50 dB attained by the polynomial PDs. The Hammerstein-polynomial PD that includes odd and even order terms gives no significant improvement in the ACPR performance compared to the odd-order-only PD. In terms of EVM, the Hammerstein-SCPWL PD outperforms the Hammerstein-polynomial PDs by 4 dB, as can be seen from Table I.

C. Performance Comparison of Direct Learning PDs

We compare the direct learning SCPWL PDs with polynomial PDs adapted using the nonlinear filtered-x LMS (NFXLMS) algorithm [21]. The Instantaneous Equivalent Linear (IEL) filter employed in the NFXLMS algorithm in [21] is developed by *assuming a Wiener model PA* and that *the PA model is known*. Thus, for fair comparison, we only consider comparing the direct learning PDs in linearizing the Wiener model PA in (31). As in [21], the polynomial PDs are identified using the NFXLMS algorithm with perfect knowledge of the Wiener PA. In contrast the proposed direct learning algorithm for the SCPWL PDs includes online PA model estimators as described in Section IV, where the algorithm does not necessarily require the PA to be a Wiener system.

1) Memory-SCPWL PD vs. Memory-Polynomial PD: The memory-SCPWL PD is implemented with $\sigma = 11$ and memory length $L_2 = 3$. Its online PA model estimator identifies the PA as a memory-SCPWL model $\sigma = 11$ and $L_1 = 3$ (see Section IV-B). With these parameters, the memory-SCPWL PD converges to an MSE close to the noise floor after a few hundread OFDM symbols.

Although the direct learning method circumvents the problem of measurement noise at the PA output, a major concern for the NFXLMS algorithm is its slow convergence as discussed in [21]. We observed that the convergence speed of the NFXLMS algorithm for the memory polynomial PD reduces drastically as the memory length is increased. On the other hand, with memory length $L_2 < 5$, both the third and fifth order polynomial PDs could not converge to the SNR level of the PA output. Fig. 11 shows the MSE achieved after convergence for the third and fifth order PDs with different memory lengths, adapted using different step sizes. The curves indicate the best step sizes for adapting PDs of different orders and memory length. With $L_2 = 4$, the NFXLMS memory polynomial PDs can only attain an MSE of approximately 38.5 dB, after 2000 OFDM symbols. For $L_2 = 5$, a minimum MSE close to noise floor, i.e., $\mathrm{SNR} = 40$ dB can be attained. However, the algorithm converged approximately after 80000 OFDM symbols. Fig. 12 shows the convergence curves of the direct learning memory-SCPWL PD and the NFXLMS memory polynomial PDs. The memory-SCPWL PD algorithm employs step sizes $\mu_5 = 0.3$ and $\mu_6 = 0.01$ for adapting the PA model and the PD parameters, respectively. The polynomial PDs of order $K = 5$, and $L_2 = 4$ is adapted using step size 0.1. The third and fifth order polynomial PDs with $L_2 = 5$ are adapted using step sizes 0.1 and 0.05, respectively.

Fig. 13 shows the ACPR performances of the PDs. The memory-SCPWL PD attains an ACPR of approximately 65 dB, outperforming both the third and fifth order memory polynomial PDs with memory length $L_2 = 4$. The fifth order memory

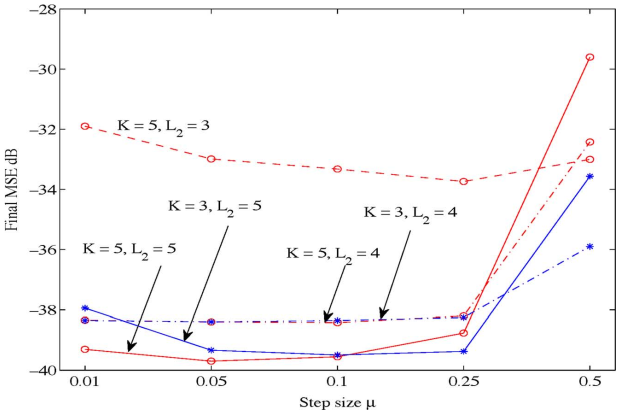

Fig. 11. MSE after convergence with different step sizes for the direct learning memory polynomial PDs. Lines with "*" markers indicate third order polynomial and "o" markers indicate fifth order polynomial. Dashed line indicates memory length $L_2 = 3$, dash-dotted line indicates $L_2 = 4$ and solid line indicates $L_2 = 5$.

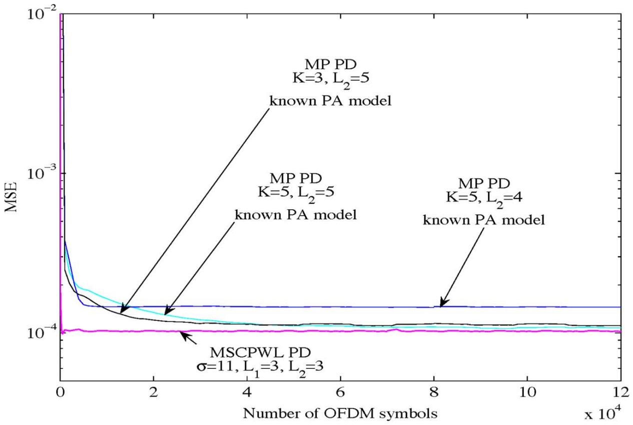

Fig. 12. Learning curves of the direct learning memory-SCPWL PD ($\sigma = 11$, $L_1 = L_2 = 3$) and the memory-polynomial PDs (3rd and 5th order, $L_2 = 3, 4, 5$, known Wiener PA model).

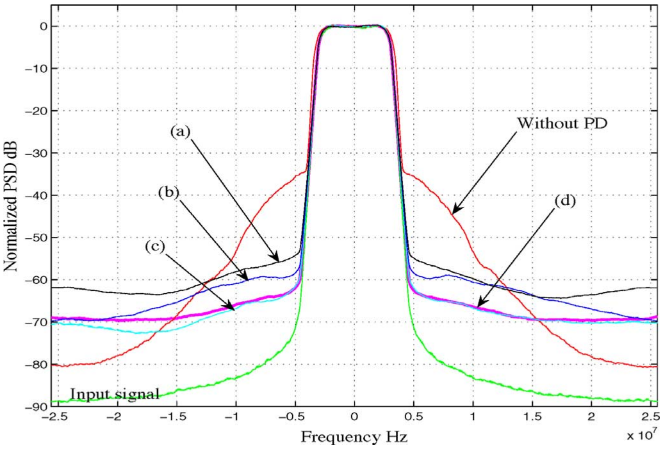

Fig. 13. Comparison of the ACPR performance of the direct learning memory-SCPWL PD with memory polynomial PDs for a Wiener model PA. (a) Memory polynomial PD, $K = 3$, $L_2 = 5$. (b) Memory polynomial PD $K = 5$, $L_2 = 4$. (c) Memory polynomial PD $K = 5$, $L_2 = 5$. (d) The proposed direct learning memory-SCPWL PD, $\sigma = 11$, $L_1 = L_2 = 3$.

polynomial PD with $L_2 = 5$ is able to attained an ACPR of 65 dB after 100 000 OFDM symbols.

2) Hammerstein-SCPWL PD vs. Hammerstein-Polynomial PD: The Hammerstein-SCPWL PD employs a linear filter of length $M = 5$ and $\sigma = 11$ breakpoints for the nonlinear block. The PA model estimator assumes the PA to be a Wiener model. The linear and nonlinear blocks of the PA are identified using the modified Wiener model estimator with memory length $N = 3$ and $\sigma = 11$, respectively. The PD and PA model estimator parameters are initialized to $c_1 = 1$, $\hat{c}_1 = 1$, $q_0 = 1$ and zero for all other parameters. For the initial 200 symbols, $\mu_0 = 0.01$, $\mu_3 = 0.0001$ and $\mu_4 = 0.0001$ were used. After the initial adaptation, the step sizes $\mu_0 = 0.001$, $\mu_3 = 0.1$, $\mu_4 = 0.1$ were employed. The Hammerstein-polynomial PDs employ fifth order models and memory length $M = 5$. The IEL filter for the NFXLMS algorithm is implemented by assuming the Wiener PA parameters are known as in [21]. Thus, there is no initial adaptation. The first parameters of the linear and nonlinear filters of the PD were initialized to 1, and the rest of the parameters were set to zero. The adaptation step sizes for both the linear and nonlinear filters were 0.03.

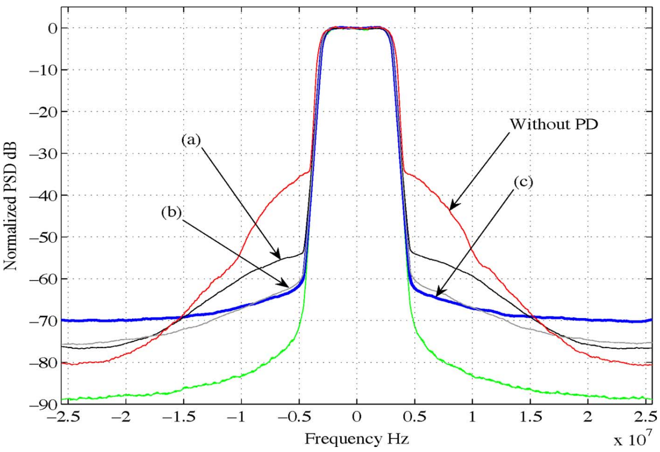

Fig. 14. Comparison of the direct learning Hammerstein-SCPWL PD with the Hammerstein-polynomial PDs for a Wiener model PA. (a) Hammerstein-polynomial PD, $K = 5$, odd orders only, $M = 5$. (b) Hammerstein-polynomial PD, $K = 5$, even and odd orders, $M = 5$. (c) The proposed Hammerstein-SCPWL PD, $\sigma = 11$, $N = 3$, $M = 5$.

Fig. 14 shows the ACPR performances of the Hammerstein model PDs in linearizing the Wiener model PA. The Hammerstein-SCPWL PD attains an ACPR of approximately 63 dB. The odd-order-only Hammerstein-polynomial PD attained an ACPR of approximately 55 dB, 8 dB worse than that of the Hammerstein-SCPWL PD. By including the even order terms, the ACPR performance of the Hammerstein-polynomial PD is leveled close to that of the Hammerstein-SCPWL PD.

From Table I, the Hammerstein-SCPWL PD and the Hammerstein-polynomial PDs are shown to achieve a similar EVM performance of approximately 24 dB.

D. Linearization of Freescale MRF6S23100H LDMOS PA in 802.16d System

In this section, MATLAB and ADS-Ptolemy cosimulation is used to evaluate the performance of the indirect learning memory-SCPWL PD on a PA in a WiMAX 802.16d downlink transmitter.

1) The Freescale MRF6S23100H LDMOS PA: The PA under test in this experiment is designed using the Freescale MRF6S23100H LDMOS device model provided in the ADS component library [27] for a WiMAX base station. It employs a push-pull architecture and is designed to operate between 2.3–2.4 GHz, with 13 dB gain ±0.25 dB gain flatness and 51.2 dBm (131W) at 1 dB compression point. The broadband matching networks are designed using Smith chart [40] and the

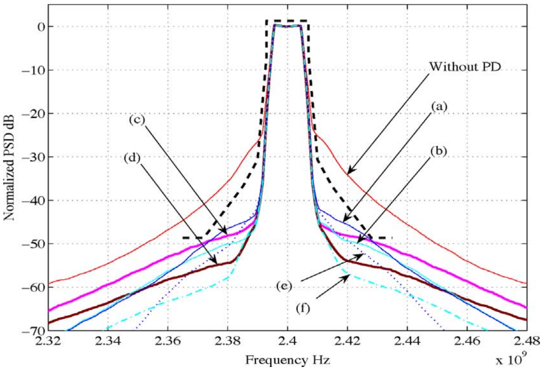

Fig. 15. Linearization of the Freescale MRF6S23100H PA by the indirect learning PDs. (a) Memory polynomial PD, $K = 5$, odd and even orders, $L = 3$. (b) Memory polynomial PD, $K = 7$, odd and even orders, $L = 3$. (c) Memory-SCPWL PD, $\sigma = 11$, $L = 2$.(d) Memory-SCPWL PD, $\sigma = 15$, $L = 2$. (e) Memory polynomial PD, $K = 5$, $L = 3$, noiseless. (f) Memory polynomial $K = 7$, $L = 3$, noiseless.

resulting microstrip components are further optimized using the EM simulator (Momentum). The PA design is based on the Freescale Root LDMOS model. Thus the memory effects evaluated are mainly electrical (short term) memory effects. The harmonic balance method [41] is used to include the effect of up to the seventh order harmonic of the fundamental frequency. The PA is biased at class-AB. The final circuit level PA design has been prototyped and validated in a laboratory by IMD test [27].

2) Performance Evaluation: The indirect learning memory-SCPWL PD and memory polynomial PD are evaluated in a WiMAX 802.16d downlink system. The system operates at carrier frequency 2.4 GHz and channel bandwidth of 14 MHz. It employs an OFDM physical layer with 256 subcarriers. Each symbol is composed of 192 data subcarriers, 1 zero DC subcarrier, 8 pilot subcarriers, and 55 guard carriers and a cyclic prefix ratio of 1/4 is used. The OFDM subcarriers are modulated with 64-QAM symbols encoded with code rate 3/4. The baseband OFDM signal is oversampled by 8 times and filtered using a raised cosine filter with roll-off factor 0.2.

A block of the downconverted PA input and output baseband signal (384 639 samples) is then exported to MATLAB for PD identification and signal predistortion. The SNR at the feedback path is 40 dB, unless otherwise indicated. The predistorted signal is then parsed back to the ADS-Ptolemy simulator to replace the original WiMAX baseband signal. The linearization performance of the PD is evaluated based on the PA output spectrum.

Firstly, the PD parameters are chosen such that the PA output spectrum is at least contained under the spectrum mask, indicated by the black dashed plot. The SCPWL PD with memory length 2 and 10 segments is able to keep the PA output specturm under the mask as shown in Fig. 15. Then, the number of PWL segments is increased to a number where further increment does not give significant performance improvement. Increasing the number of segments to 14 further improves the ACPR performance of the memory-SCPWL PD by approximately 6 dB, reducing the ACP by more than 25 dB. The 5th order polynomial PD with memory length 3 is required to keep the PA output spectrum under the mask. The memory-SCPWL PD with 10 segments outperforms the 5th order memory polynomial PD by approximately 3 dB. The 7th order memory polynomial PD gives slightly better performance than the 5th order PD. However, the 14-segment memory-SCPWL PD outperforms the 7th order memory polynomial PD by 5 dB. The performance of the memory polynomial PDs are affected by noise at the feedback path. This can be seen from plots (e) and (f), showing the performances of the 5th and 7th order memory polynomial PDs, respectively, when there is no noise at the feedback path. On the other hand, the performances of the memory-SCPWL PDs are not affected by the noise.

TABLE II
NUMBER OF COMPLEX MULTIPLICATIONS PER ITERATION

		Multiplications (×)
Memory-SCPWL	PD filtering	$L_2(2\sigma - 1)$
	Indirect	$(2L+1)\sigma + (L-1)$
	Direct	$(2L_2+1)\sigma + (L_2-1)$ $+\{(3L_1 + L_1L_2 - L_2)\sigma + L_1\}$
Hamm-SCPWL	PD filtering	$2\sigma + M - 1$
	Indirect	$3\sigma + (2+M)N$
	Direct	$3\sigma + (2+M)N$ $+\{(M+N+1)\sigma + M + 3\}$

E. Computational Complexity

Table II summarizes the complexity in terms of number of complex multiplications required in each iteration, for implementing PD filtering and the indirect and direct learning algorithms of the SCPWL PDs. Due to PA model estimation and filtering of input signal, the direct learning algorithms are computationally more demanding than the indirect learning algorithms. The additional computations required by the direct learning algorithms are indicated in the curly brackets.

In order to illustrate the difference in computational complexity, we calculate the number of complex multiplications required by the SCPWL PDs with $\sigma = 11$, $L = L_1 = L_2 = 3$, $N = 3$, and $M = 5$. PD filtering by memory-SCPWL PD and Hammerstein-SCPWL require 30 and 26 multiplications, respectively. The indirect learning adaptation of the memory-SCPWL PD requires 79 multiplications compared to 247 multiplications required by the direct learning method. For the Hammerstein-SCPWL PD, 54 and 161 multiplications are required for implementing the indirect learning and direct learning algorithms, respectively. Thus, for both PD models, the direct learning method is approximately 200% more computationally demanding than the indirect learning method.

VI. Conclusion

We have proposed a novel baseband simplicial canonical piecewise linear (SCPWL) function for modeling predistorters (PDs) with memory. The new complex-valued SCPWL function is used for implementing indirect and direct learning Hammerstein-SCPWL predistorter (PD) and memory-SCPWL PD. Our direct learning PDs incorporate an online PA model estimator, which allows changes in PA characteristics to be tracked. To overcome the nonconvex cost function problem of Hammerstein PD identification, we employed a modified Wiener model estimator, which improves the efficiency of the Hammerstein PD algorithms. Compared to polynomial based PDs, the SCPWL PDs are shown to give better modeling capability and is more robust for modeling strong nonlinearities. In addition, the SCPWL PDs are less sensitive to input noise than polynomial-based PDs. Particularly, we showed that noise at the feedback path of the indirect learning filter has negligible effects on the SCPWL PD, due to its linear basis function. On the other hand, polynomial coefficients are affected by noise induced bias, which in turns, cause performance degradation in the polynomial-based PDs, with greater degradation for higher order polynomial. The performance of the proposed SCPWL PDs were evaluated and compared with polynomial-based PDs via extensive baseband level and circuit level simulations. Baseband level simulations confirmed that the SCPWL PDs outperform their polynomial counterparts, especially when noise is present in the feedback path of the indirect learning branch. When adapted using the direct learning algorithm, the memory-SCPWL PD is shown to provide better modeling capability and converge faster than the memory-polynomial PD. In the circuit level simulations, the indirect learning memory-SCPWL PD was evaluated on the Freescale MRF6S23100H PA designed for WiMAX 802.16d system. The PA was designed in the Agilent Advanced Design System (ADS). The design has been prototyped and validated by intermodulation distortion (IMD) test in a laboratory. The simulation results once again confirmed that, with indirect learning implementation, the memory-SCPWL PD outperforms the memory polynomial PD in supressing spectral regrowth, especially when noise is present at the feedback path.

Appendix

A. Derivative of the Static SCPWL Model

Consider a static nonlinear system with the description in (3) $(L = 1)$, we have the system output

$$y(n) = \mathbb{N}[v(n)] = a_0^* + \sum_{i=1}^{\sigma-1} a_i^* \lambda_i[|v(n)|] \exp[j\varpi_v(n)]. \tag{33}$$

The amplitude and phase of the signal $v(n)$ are

$$\begin{aligned} |v(n)| &= [v(n)v^*(n)]^{1/2}, \\ \varpi_v(n) &= \tan^{-1}\left(\frac{v(n) - v^*(n)}{j(v(n) + v^*(n))}\right). \end{aligned} \tag{34}$$

Differentiating the amplitude and phase with respect to $v(n)$ ([31], App. B), we get

$$\begin{aligned} \frac{d|v(n)|}{dv(n)} &= \exp(-j\varpi_v(n)), \\ \frac{d\varpi_v(n)}{dv(n)} &= \frac{-j\exp(-j\varpi_v(n))}{|v(n)|}. \end{aligned} \tag{35}$$

Using (35) we may calculate the derivative of (33) w.r.t. $v(n)$

$$\begin{aligned} \frac{d\mathbb{N}[v(n)]}{dv(n)} &= \sum_{i=1}^{\sigma-1} a_i^* \frac{d\lambda_i[|v(n)|]\exp[j\varpi_v(n)]}{dv(n)} \\ &= \sum_{i=1}^{\sigma-1} a_i^* \left[\lambda_i[|v(n)|]\frac{d\exp[j\varpi_v(n)]}{dv(n)}\right. \\ &\quad \left. + \exp[j\varpi_v(n)]\frac{d\lambda_i[|v(n)|]}{d|v(n)|}\frac{d|v(n)|}{dv(n)}\right] \\ &= \sum_{i=1}^{\sigma-1} a_i^* \left[\frac{\lambda_i(v(n))}{v(n)} + \lambda_i'(|v(n)|)\right]. \end{aligned} \tag{36}$$

The derivative of the SCPWL basis function vector takes the form of (37) at the bottom of the page [39], for $(1 \leq N_s \leq \sigma - 1)$, where N_s is the number of the segment that $|v(n)|$ falls into. Typically, $a_0 = 0$ for PA type nonlinearity. Thus,

$$\frac{d\mathbb{N}[v(n)]}{dv(n)} = \begin{cases} \frac{y(n)}{v(n)} + \sum_{i=1}^{N_s} a_i^*(n), & |v(n)| \leq \beta_\sigma \\ \frac{y(n)}{v(n)}, & |v(n)| > \beta_\sigma \end{cases}. \tag{38}$$

B. Approximation of $G(n)$

Assume that the inverse function of $y(n) = \mathbb{N}\{v(n)\}$ exists such that $\mathbb{P}\{y(n)\} = \mathbb{P}[\mathbb{N}\{v(n)\}] = v(n)$. Then by the inverse function theorem we have

$$\frac{\partial y(n)}{\partial v(n)} = \hat{G}(n) = \left[\frac{\partial \mathbb{P}\{y(n)\}}{\partial y(n)}\right]^{-1}. \tag{39}$$

From (5), $\mathbb{P}[y(n)] = \mathbf{c}^{\mathrm{H}}\boldsymbol{\lambda}[y(n)]$ approximates $v(n)$. The derivative of $\mathbb{P}[y(n)]$ can be obtained by following similar steps as in (36). Then, using (39), $G(n)$ can be approximated from the static nonlinear model of the PD as

$$\hat{G}(n) = \begin{cases} \left[\frac{v(n)}{y(n)} + \sum_{i=1}^{N_s} c_i^*(n)\right]^{-1}, & |y(n)| \leq \beta_\sigma \\ \frac{y(n)}{v(n)}, & |y(n)| > \beta_\sigma \end{cases}. \tag{40}$$

C. Expectation of SCPWL Basis Function Amplitude

The SCPWL basis function in (2) can be rewritten as

$$\lambda_i[x] = \begin{cases} 0, & x \leq \beta_i \\ x - \beta_i, & \beta_i < x < \beta_\sigma \\ \beta_\sigma - \beta_i, & x \geq \beta_\sigma \end{cases}. \tag{41}$$

$$\boldsymbol{\lambda}'(|v(n)|) = \begin{cases} \begin{bmatrix} 0 & \underbrace{1 \cdot\cdot 1}_{N_s \text{ elements}} & \overbrace{0 \cdot\cdot 0}^{\sigma-1-N_s \text{ elements}} \end{bmatrix}^{\mathrm{T}}, & \beta_{N_s} < |v(n)| \leq \beta_{N_s+1}, \\ [0 \quad 0 \cdots\cdots 0]^{\mathrm{T}}, & |v(n)| > \beta_\sigma \end{cases} \tag{37}$$

For zero-mean complex Gaussian signal with variance σ_z^2, its amplitude is Rayleigh distributed with parameter $\sigma_x = \sqrt{(\sigma_z^2)/(2)}$. The probability density function of x is given by $p(x) = (x)/(\sigma_x^2)e^{-(x^2)/(2\sigma_x^2)}$. Then, the mean amplitude of the SCPWL basis function is

$$\begin{aligned} E[\lambda_i(x)] &= \int_{\beta_i}^{\beta_\sigma}(x-\beta_i)p(x)\,dx + \int_{\beta_\sigma}^{\infty}(\beta_\sigma-\beta_i)p(x)\,dx \\ &= \sqrt{\frac{\pi\sigma_x^2}{2}}\left[\operatorname{erf}\left(\sqrt{\frac{\beta_\sigma^2}{2\sigma_x^2}}\right) - \operatorname{erf}\left(\sqrt{\frac{\beta_i^2}{2\sigma_x^2}}\right)\right] \\ &= \sqrt{\frac{\pi\sigma_x^2}{2}}\left[\operatorname{erf}\left(\frac{\beta_\sigma}{\sigma_x\sqrt{2}}\right) - \operatorname{erf}\left(\frac{\beta_i}{\sigma_x\sqrt{2}}\right)\right] \\ &= 2\sqrt{\frac{\pi\sigma_x^2}{2}}\left[\Phi_{0,\sigma_x^2}(\beta_\sigma) - \Phi_{0,\sigma_x^2}(\beta_i)\right], \qquad (42) \end{aligned}$$

where $\Phi_{0,\sigma_x^2}(\beta_i)$ is the normal cumulative distribution function for $\beta_i \in \mathbb{R}$.

REFERENCES

[1] "Green base station—The benefits of going green," *Mobile Eur. Mag.*, pp. 38–41, Apr. 2008.

[2] M. Helaoui and F. M. Ghannouchi, "Linearization of power amplifiers using the reverse MM-LINC technique," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 6–10, Jan. 2010.

[3] H. H. Boo, S. W. Chung, and J. L. Dawson, "Adaptive predistorter using a $\Delta\Sigma$ modulator for automatic inversion of power amplifier nonlinearity," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 56, no. 12, pp. 901–905, Dec. 2009.

[4] M. Y. Cheong, S. Werner, J. E. Cousséau, and T. I. Laakso, "Predistorter identification using the simplicial canonical piecewise linear function," in *Proc. 12th Int. Conf. Telecommun. (ICT)*, Cape Town, South Africa, May 2005.

[5] M. Seo, S. Jeon, and S. Im, "Compensation for nonlinear distortion in OFDM system using a digital predistorter based on the SCPWL model," in *Proc. IEEE 6th Int. Conf. Wireless Mobile Commun.*, Valencia, Spain, Sep. 2010, pp. 27–32.

[6] Y. Ding, L. Sun, and A. Sano, "Adaptive nonlinear predistortion schemes with application to OFDM system," in *Proc. IEEE 2003 Conf. Control Appl.*, Istanbul, Turkey, Jun. 2003, vol. 1, pp. 1130–1135.

[7] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[8] L. Ding, Z. Ma, D. R. Morgan, M. Zierdt, and J. Pastalan, "A least-squares/Newton method for digital predistortion of wideband signals," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 833–840, May 2006.

[9] D. R. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, "A generalized memory polynomial for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[10] A. Zhu, P. Draxler, J. Yan, T. Brazil, D. Kimball, and P. Asbeck, "Open-loop digital predistorter for RF power amplifiers using dynamic deviation reduction-based Volterra series," *IEEE Trans. Microw. Theory Tech.*, vol. 56, no. 7, pp. 1524–1534, Jul. 2008.

[11] H. Jiang and P. A. Wilford, "Digital predistortion for power amplifiers using separable functions," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4121–4130, Aug. 2010.

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

[13] H. W. Kang, Y. S. Cho, and D. H. Youn, "On compensating nonlinear distortions of an OFDM system using an efficient adaptive predistorter," *IEEE Trans. Commun.*, vol. 47, no. 4, pp. 522–526, Apr. 1999.

[14] G. T. Zhou, H. Qian, L. Ding, and R. Raich, "On the baseband representation of a bandpass nonlinearity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2953–2957, Aug. 2005.

[15] S. P. Stapleton and F. C. Costescu, "An adaptive predistorter for a power amplifier based in adjacent channel emissions," *IEEE Trans. Veh. Technol.*, vol. 41, no. 1, pp. 49–56, Feb. 1991.

[16] J. Kim and K. Konstantinou, "Digital predistorter of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[17] J. Tsimbinos and K. V. Lever, "Nonlinear system compensation based on orthogonal polynomial inverses," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 4, pp. 406–417, Apr. 2001.

[18] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1468–1479, Sep. 2004.

[19] C. Eun and E. J. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.

[20] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1726–1730, Jun. 1998.

[21] V. E. Zhou and D. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 120–133, Jan. 2007.

[22] D. Morgan, Z. Ma, and L. Ding, "Reducing measurement noise effects in digital predistortion of RF power amplifiers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2003, vol. 4, pp. 2436–2439.

[23] S. Choi, E.-R. Jeong, and Y. Lee, "Adaptive predistortion with direct learning based on piecewise linear approximation of amplifier nonlinearity," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 3, pp. 397–404, Jun. 2009.

[24] P. Julián, A. C. Desages, and O. E. Agamennoni, "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, pp. 463–480, Apr. 1999.

[25] M. Y. Cheong, S. Werner, J. E. Cousséau, and R. Wichman, "Spectral characteristics of a piecewise linear function in modeling power amplifier type nonlinearities," in *Proc. IEEE 21st Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2010, pp. 639–644.

[26] A. Hagenblad, "Aspects of the identification of Wiener models," Licentiate Thesis No. 793, Dept. Electr. Eng., Linköping University, Linköping, Sweden, Nov. 1999.

[27] M. Bruno, J. Cousseau, A. S. H. Ghadam, and M. Valkama, "On high linearity high efficiency RF amplifier design," in *Proc. Argentine School Micro-Nanoelectron., Technol., Appl. (EAMTA)*, Oct. 2010, pp. 97–102.

[28] T. Wigren, "Recursive identification based on the nonlinear Wiener model," Ph.D. Thesis, Acta Universitatis Upsaliensis, Uppsala University, Uppsala, Sweden, Dec. 1990.

[29] T. Wigren, "Recursive prediction error identification using the nonlinear Wiener model," *Automatica*, vol. 29, no. 4, pp. 1011–1025, Jul. 1993.

[30] A. D. Kalafatis, L. Wang, and W. R. Cluett, "Identification of Wiener-type nonlinear systems in noisy environment," *Int. J. Control*, vol. 66, pp. 923–941, Apr. 1997.

[31] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[32] P. L. Feintuch, N. J. Bershad, and A. K. Lo, "A frequency domain model for 'filtered' LMS algorithms-stability analysis, design, and elimination of the training mode," *IEEE Trans. Signal Process.*, vol. 41, no. 4, pp. 1518–1531, Apr. 1993.

[33] G. Chen, T. Sone, N. Sato, M. Abe, and S. Makino, "The stability and convergence characteristics of the delayed-x LMS algorithm in ANC systems," *J. Sound Vib.*, vol. 216, no. 4, pp. 637–648, Oct. 1998.

[34] D. R. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1980, vol. 5, pp. 457–461.

[35] O. J. Tobias and R. Seara, "Performance comparison of the FXLMS, nonlinear FXLMS and leaky FXLMS algorithms in nonlinear active control applications," in *Proc. 11th Eur. Signal Process. Conf. (EUSIPCO)*, Toulouse, France, Sep. 2002, vol. 1, pp. 155–158.

[36] C. Eun and E. J. Powers, "A predistorter design for a memory-less nonlinearity preceded by a dynamic linear system," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'95)*, vol. 1, pp. 152–156.

[37] L. Ding, R. Raich, and G. Zhou, "A Hammerstein predistortion linearization design based on the indirect learning architecture," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2002, vol. 3, pp. 2689–2692.

[38] E.-W. Bai, "An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems," *Automatica*, vol. 34, no. 3, pp. 333–338, Mar. 1998.

[39] J. Figueroa, J. Cousseau, and R. J. P. de Figueiredo, "A simplicial canonical piecewise linear adaptive filter," *Circuits Syst. Signal Process.*, vol. 5, no. 23, pp. 365–386, 2004.

[40] G. Gonzalez, *Microwave Transistor Amplifiers. Analysis and Design*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[41] E. Ngoya, "Envelope transient analysis: A new method for the transient and steady state analysis of microwave circuit and systems," in *IEEE MTT Symp. Dig.*, 1996, pp. 1365–1368.

Mei Yen Cheong received the M.Sc. and Licentiate of Science degrees from the Department of Electrical and Communication Engineering at the Helsinki University of Technology, Finland, in 2003 and 2006, respectively. She is currently working toward the Ph.D. degree in the Department of Signal Processing and Acoustics, Aalto University, Aalto, Finland.

Her research interests include predistortion linearizers and signal processing for communications.

Stefan Werner (SM'07) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology (KTH), 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, Aalto University, Aalto, Finland, where he is also appointed as a Docent. His research interests include adaptive signal processing, signal processing for communications, and statistical signal processing.

Dr. Werner is a Member of the Editorial Board for the *EURASIP Signal Processing Journal*.

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), Bahía Blanca, Argentina, in 2005. He is currently working toward the Ph.D. degree in digital linearization of radio frequency power amplifiers at UNS.

His research interests include the study of nonlinearities in broadband radio frequency power amplifiers and signal processing on high speed circuits.

Jose Luis Figueroa (M'98–SM'01) received the B.Sc. degree in electrical engineering and the Ph.D. degree in systems control from the Universidad Nacional del Sur, Bahía Blanca, 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.

Dr. Figueroa served as Chair of the Standing Committee on Chapter Activities of the IEEE Control Systems Society between 2002 and 2004.

Juan E. Cousseau (S'90–M'92–SM'00) received the B.Sc. degree from the Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina, in 1983 and the M.Sc. and Dr.Sc. degree from COPPE/Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1989 and 1993, respectively, 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 also a Visiting Professor at University of California, Irvine (1999), and the Signal Processing Laboratory, Aalto University School of Electrical Engineering (many times since 2004). His research interests include modern wireless communication systems, statistical signal processing, and broadband RF front-end impairments reduction.

Dr. Cousseau was IEEE Circuits and Systems Chair of the Argentine 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 participated in the IEEE Signal Processing Society Distinguished Lecturer Program 2007.

Risto Wichman received the 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, Aalto University School of Electrical Engineering, Aalto, Finland, where he has been a Professor since 2003. His major research interests include digital signal processing for applications in wireless communications.