Efficient Nonlinear Wiener Model Identification Using a Complex-Valued Simplicial Canonical Piecewise Linear Filter

Juan E. Cousseau, *Senior Member, IEEE*, Jose Luis Figueroa, *Senior Member, IEEE*, Stefan Werner, *Member, IEEE*, and Timo I. Laakso, *Senior Member, IEEE*

***Abstract*—This paper proposes an efficient adaptive realization of the Wiener model for the identification of complex-valued nonlinear systems. Using a two-dimensional *simplicial canonical piecewise linear* filter for the complex-valued nonlinear mapping, we derive a realization of the Wiener model requiring fewer parameters than previous approaches. An adaptive implementation of the proposed Wiener model is derived, and local convergence analysis for the updating algorithm is presented. The tradeoff between computational complexity and modeling performance is discussed. Simulations of a system identification example show that the proposed algorithm can provide similar or better performance than other approaches in terms of computational complexity, convergence speed, and final mean-squared error (MSE).**

***Index Terms*—Adaptive estimation, adaptive filters, adaptive systems, identification, nonlinear filters, nonlinear systems, signal processing.**

I. Introduction

LINEAR adaptive filtering algorithms are useful in several applications where the signals can be modeled as Gaussian noise applied to linear systems [1]. However, requiring strict linearity of all the equipment is costly, and especially problematic in the power amplifiers of the transmitter, where high linearity often means low efficiency and thus trouble with power feed and heat generation [2]. This is a pressing problem in future wideband mobile communications systems, which employ the orthogonal-frequency-division multiplexing (OFDM) with particularly large dynamic range for the signal amplitude [3]. Instead, it may be more advantageous to increase the used dynamic range of the amplifiers in the nonlinear region. This means that nonlinear system models need to be able to accurately capture the system performance.

A major drawback of general functional series forms [4], [5] for describing an input–output nonlinear relation, like the Volterra and Wiener descriptions [6]–[8], is the excessive number of parameters involved due to the memory of the system and the degree of nonlinearity. Therefore, they are not very useful for online implementations due to the high computational demand they imply, especially for the case of higher order kernels. A simplified form of the general Wiener representation leads to what is known in literature as the *Wiener model*. The use of Wiener models has been treated in the literature in various contexts like chemical processes [9]–[12], biomedical engineering [13]–[15], and control [16]–[18]. The Wiener model is composed of a linear dynamic part followed by a static nonlinearity. Wiener models have been successfully used in many applications [7], [19], [20], and it is shown in [21] that the class of time-invariant systems with *fading memory* can be approximated arbitrarily well with a Wiener model.

For the parameterization of the Wiener model, various alternatives are possible. For example, the linear part can be represented by finite-impulse-response (FIR), rational [22], Laguerre [23], Kautz [24], or linear state-space models [25]. As for the nonlinear static part, power series, Chebyshev polynomials, neural networks, wavelets, and piecewise linear (PWL) functions have been proposed [22], [26], [27]. An interesting discussion to justify the tuning of the static nonlinearity was presented in [28]. The number of parameters used in the two parts of the Wiener model affects the final modeling capability. Often, reducing the number of parameters in the linear dynamic part will cause an increase in the number of parameters in the nonlinear static part, if similar modeling capability is to be maintained [25]. A complete characterization of recursive identification methods for a Wiener model consisting of a rational model followed by a piecewise linear nonlinearity was reported in [22], where also local convergence results were established.

In a series of detailed articles, the analysis of stochastic gradient algorithms, with baseband signals, and for different variants of the Wiener model were considered [29]–[31]. These results are similar to the analysis of stochastic algorithms for neural networks (see, for example, [32]–[34]).

In [29], a Wiener model for Gaussian inputs and with input and output noise was studied. The Wiener model considered was formed by a FIR linear part and a static nonlinearity with known shape but unknown input and output gains. Stochastic gradient adaptive identification algorithms were studied when the nonlinear system operates in a small region in the neighborhood of a bias point.

One of the alternatives considered there was a two-step scheme. In the first step of this alternative they obtain estimates

Manuscript received November 11, 2005; revised July 7, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kenneth E. Barner. This work was partially supported by the Academy of Finland, Smart and Novel Radios (SMARAD) Center of Excellence.

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

S. Werner and T. I. Laakso are with the Signal Processing Laboratory, Helsinki University of Technology, FIN-02015 HUT, Finland (e-mail: stefan.werner@hut.fi; timo.laakso@hut.fi).

Digital Object Identifier 10.1109/TSP.2006.890893

of the parameters using only the FIR model. The second step involves a Wiener model, with a linear part with the parameters obtained in the first step, looking to estimate the parameters of the static nonlinear part. As demonstrated using a usual small step-size approximation, the linear part is identified in the first step except for a time-varying scale factor. The input and output gains of the nonlinearity of a known shape are obtained in the second step. An alternative algorithm is to estimate the parameters of the linear and nonlinear part simultaneously. As a general result, the input affects both schemes in a similar way.

A complete analysis of the mean-squared error (MSE) behavior of the two-step scheme using stochastic gradient algorithms was presented in [30] (see also [35]). In that case, a polynomial static nonlinearity was considered and no input noise was assumed. Although not explicitly discussed there, using the small step-size approximation, they justify the approximated statistical independence between the procedure of two estimation steps. Using this assumption for the linear part, convergence in the mean and *in the MSE sense* to the minimum can be obtained. These stochastic gradient analysis results were further extended in [31], considering both the two-step and the simultaneous schemes when the static nonlinearity is time varying.

In this paper, we focus on implementations of the Wiener model for complex-valued signals, rather than the commonly treated case of real-valued signals. The motivation is that many recent nonlinear modeling applications, e.g., communications systems [36], require processing of complex-valued signals. The goal is to develop a low-complexity realization of the Wiener model suitable for implementation using efficient adaptive signal processing algorithms.

Considering an adaptive implementation of the complex Wiener model, the complex least-mean-square (CLMS) algorithm [37] is a possible choice for identifying the linear dynamic part. The problem of a suitable complex static nonlinearity description has been addressed with diverse techniques. Considering neural networks, when using the complex backpropagation algorithm [38], [39], the choice of a suitable fixed activation function [40] does not seem to have an evident solution. Recently, a Wiener filter with a fixed static activation function was proposed in [41]. Together with this structure, the authors derived a normalized nonlinear complex-valued gradient descent algorithm.

This paper proposes a novel complex Wiener adaptive filter consisting of an FIR linear part and a particular PWL representation of the nonlinearity, referred to as the *complex-valued simplicial canonical piecewise linear* (CS-CPWL) filter [42]. The use of CS-CPWL functions allows the representation of any arbitrary continuous memoryless mapping between two variables. This mapping implies more complete modeling capabilities than other representations of the static nonlinearity of similar computational complexity (see [43]–[45]). The proposed algorithm can be seen as an extension of [41] to the case of adaptive nonlinear gain.

The properties and characteristics of the CS-CPWL filter are presented as follows. General aspects of the piecewise linear representation are reviewed in Section II. The description of the CS-CPWL algorithm for complex signals and some modeling issues are presented in Section III. Section IV presents a local convergence study and discusses the implementation of the proposed algorithm. In Section V, a numerical simulation example is provided illustrating the performance of the proposed algorithm in a system identification application. Finally, Section IV draws the conclusions and makes the final observations.

II. High-Level PWL Representation

This section reviews the PWL function which is an attractive tool for nonlinear modeling. Extensive research has been carried out in the last three decades to find general, efficient, and complete PWL representations (see, for example, [46]–[52]). The basic idea of the PWL function is to approximate a nonlinear function by a series of linear functions defined in properly partitioned subregions of the original definition region.

Recently, a systematic way to generate the PWL representation for arbitrary (continuous) domains in a more compact and efficient form was introduced in [53] and [42]. This PWL representation uses the concept of *simplicial partitions* (simplex) of the domain of interest. In the general *simplicial* canonical piecewise linear (SCPWL) representation, the nonlinear mapping of $\boldsymbol{x} \in \mathbb{R}^{\aleph}$, $f(\boldsymbol{x})$, is given by the combination of $\aleph$ basis representations $\boldsymbol{\Lambda}_l \in \mathbb{R}^{q_l}$ as

$$f(\boldsymbol{x}) = \sum_{l=0}^{\aleph} \boldsymbol{c}_l^T \boldsymbol{\Lambda}_l(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{\Lambda}(\boldsymbol{x}) \tag{1}$$

where each component of $\boldsymbol{x}$ is defined in a rectangular compact domain $x_i \in [0,\ P_i\Delta]$ for $i = 1, \ldots, \aleph$, where P_i is the number of partition segments and Δ is the grid size. The vectors $\{\boldsymbol{\Lambda}, \boldsymbol{c}\} \in \mathbb{R}^M$ with $M = \sum_{l=0}^{\aleph} q_l$ contain the basis vectors $\boldsymbol{\Lambda}_l$ ordered according to their so-called *nesting level* [53] and the linear weighting coefficients $\boldsymbol{c}_k$

$$\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_0{}^T, \boldsymbol{\Lambda}_1{}^T, \ldots, \boldsymbol{\Lambda}_l{}^T, \ldots, \boldsymbol{\Lambda}_{\aleph}{}^T]^T$$
$$\boldsymbol{c} = [\boldsymbol{c}_0{}^T, \boldsymbol{c}_1{}^T, \ldots, \boldsymbol{c}_l{}^T, \ldots, \boldsymbol{c}_{\aleph}{}^T]^T.$$

An important parameter contributing to the computational complexity of (1) is the dimension of the basis representation for the nesting level l, i.e., q_l, where (as defined in [53]) $q_0 = 1$, and

$$q_l = \sum_{k_1=1}^{\aleph} P_{k_1} \sum_{k_2=k_1+1}^{\aleph} P_{k_2} \cdots \sum_{k_l=k_{l-1}+1}^{\aleph} P_{k_l}; l = 1, \ldots, \aleph. \tag{2}$$

In the particular case when all the dimensions in $\mathbb{R}^{\aleph}$ have the same number of partitions, i.e., $P_i = P$, (2) reduces to

$$q_l = P^l \binom{\aleph}{l}. \tag{3}$$

From (3), we see that problems requiring high dimension $\aleph$ and many partitions P for the implementation of the simplicial CPWL mapping result in a huge number of parameters, i.e., a high computational complexity. This is the main practical limitation of any PWL approximation. To further decrease the computational complexity, reduced-order model techniques can be used that incorporate knowledge of the nonlinearity to be modeled [42]. These techniques are based on the close relationship

between the general CPWL model description and a complete Hilbert space representation. Some simplified alternatives, contemplating nonlinear prediction, have been proposed in [26]. A solution regarding low computational complexity when only real-valued signals are involved is discussed in [27].

The rest of this paper is devoted to the particular PWL representation obtained in [42] for the $\mathbb{R}^2$ domain in order to derive an efficient realization of the Wiener model for complex-valued systems. We assume that each dimension has an equal number of partitions P. In this case, the number of parameters in $\boldsymbol{c}$ is

$$M = \sum_{l=0}^{2} q_l = 1 + 2P + P^2. \tag{4}$$

The interested reader is referred to [53] and [42] for a more detailed description of the general PWL mapping.

III. Reduced-Complexity Wiener Model

This section considers a reduced-complexity Wiener model consisting of an FIR linear part and a PWL description for the static nonlinearity. For this purpose, we first extend the work in [27] to the case of complex-valued signals and derive a CS-CPWL realization. Thereafter, a low-complexity adaptive implementation is derived for the considered Wiener model. The section ends discussing some issues concerning the proposed realization.

A. CS-CPWL Filter

For the derivation of the CS-CPWL filter representation, we need to express the input and output signals to the model as complex-valued quantities. For this purpose, we employ the representation frequently used for deriving complex Wiener filters, where $x(k) = x_R(k) + \mathrm{j}x_I(k)$ and $y(k) = y_R(k) + \mathrm{j}y_I(k)$ are the complex-valued input and output signals, respectively.[1]

The proposed Wiener model consists of a complex-coefficient linear FIR filter cascaded with a two-dimensional piecewise linear function. The complex-valued output $v(k)$ of the FIR filter $\boldsymbol{h} \in \mathbb{C}^N$ is given by

$$v(k) = \boldsymbol{h}^H \boldsymbol{x}(k) \tag{5}$$

where $\boldsymbol{x}(k) = [x(k) \cdots x(k-N+1)]^T$ is the input-signal vector and $(\cdot)^H$ denotes the Hermitian (transpose with complex conjugation). Since the complex-valued output from the linear dynamic filter $\boldsymbol{h}$ can be expressed using its in-phase and quadrature components, i.e., $v(k) = v_R(k) + \mathrm{j}v_I(k) \in \mathbb{C}$, we can form the complex-valued output $y(k)$ from our Wiener model using a two-dimensional simplicial CPWL function [42] (one dimension for $v_R(k)$ and one for $v_I(k)$). Using (1), we can build the mapping $f[\cdot] : \mathbb{C} \to \mathbb{C}$ as

$$y(k) = f[v(k)] = \boldsymbol{c}^H \boldsymbol{\Lambda}[v(k)] = \boldsymbol{c}^H \boldsymbol{\Lambda}\left[\begin{pmatrix} \mathrm{Re}\{v(k)\} \\ \mathrm{Im}\{v(k)\} \end{pmatrix}\right] \tag{6}$$

where $\boldsymbol{c} \in \mathbb{C}^M$ is a vector containing the parameters associated with the nonlinear static representation and $\boldsymbol{\Lambda} : \mathbb{C} \to \mathbb{R}^M$ is a vector function depending on the partition of the output $v(k)$ from the linear part of the Wiener model.

[1] R and I represent the real and imaginary parts, respectively

Each sector of the simplicial partition [42] is given by

$$\beta_{j-1} \le v_R \le \beta_j \qquad \beta_{i-1} \le v_I \le \beta_i$$

for $j, i = 1, 2, \ldots, P$. In this form, β_j divides each component of the domain in P partitions, with $\beta_0 \le \beta_1 \le \cdots \le \beta_P$.

Based on this description, $\boldsymbol{\Lambda}$ is defined by

$$\boldsymbol{\Lambda}(v(k)) = \begin{bmatrix} \Lambda_0 \\ \boldsymbol{\Lambda}_1[v(k)] \\ \boldsymbol{\Lambda}_2[v(k)] \end{bmatrix} \tag{7}$$

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

$$\boldsymbol{\Lambda}_1(v) = \begin{bmatrix} \boldsymbol{\Upsilon}^1[v_R(k)] \\ \boldsymbol{\Upsilon}^1[v_I(k)] \end{bmatrix} \tag{8}$$

is the first-order basis with $\boldsymbol{\Upsilon}^1 : \mathbb{R} \to \mathbb{R}^P$ whose ith entry is given by

$$\Upsilon_i^1(\nu) = \begin{cases} \frac{1}{2}\left(\nu - \beta_i + |\nu - \beta_i|\right) & \text{if } \nu \le \beta_P \\ \frac{1}{2}\left(\beta_P - \beta_i + |\beta_P - \beta_i|\right) & \text{if } \nu > \beta_P \end{cases} \tag{9}$$

and $\boldsymbol{\Lambda}_2[v] = \boldsymbol{\Upsilon}^2[v_R, v_I] : \mathbb{R}^2 \to \mathbb{R}^{P^2}$ is the second-order basis, whose $[(i-1)P + j]$th entry is defined by

$$\Upsilon^2_{(i-1)P+j}(\nu_1, \nu_2) = \begin{cases} \Upsilon_i^1(\nu_1) & \text{if } \Upsilon_i^1(\nu_1) \le \Upsilon_j^1(\nu_2) \\ \Upsilon_j^1(\nu_2) & \text{if } \Upsilon_i^1(\nu_1) > \Upsilon_j^1(\nu_2) \end{cases} \tag{10}$$

for $i, j = 1, \ldots, P$. Thus, (7) and the terms (9) and (10) define a second-order simplicial canonical PWL suitable for complex filtering.

Using this definition, each element of the basis will be bounded. To see this, note that $\Upsilon_i^1(\nu)$ tends to 0 when $\nu \to -\infty$ and to $\Upsilon_i^1(\beta_P)$ when $\nu \to \infty$. Similarly, Υ^2 is also bounded since it is defined in terms of Υ^1. From (7)–(10), note that if only $\boldsymbol{\Lambda}_0$ and $\boldsymbol{\Lambda}_1$ are used for the mapping, we obtain a solution that considers separate filtering of the real and imaginary parts, i.e., one real-valued SCPWL function for $v_R(k)$ and one for $v_I(k)$. This *split complex* approach will not provide the best solution [41]. It is therefore necessary to include $\boldsymbol{\Lambda}_2$ to yield a *fully complex* mapping.

Fig. 1 illustrates the approximation capabilities of each nesting level. From this picture, it is clear that nesting level 0 enables the representation of constant functions. Nesting level 1 allows the description of continuous functions that are linear on each squared partition of the domain. If we want to have higher resolution (approximation on each triangle), it is necessary to use the nesting-level-2 terms.

Obviously, the number of parameters increases with the nesting levels. Because the in-phase and quadrature components are divided in P sectors, the number of coefficients M in $\boldsymbol{c}$ is given by $M = \sum_{l=0}^{2} q_l = 1 + 2P + P^2$ (see (4)). In this way, the number of coefficients related with nesting level 0 is one, the number of terms for the nesting level 1 is $2P$, and the level two requires P^2 coefficients. For example, with $P = 5$ partitions we will have $M = 36$ coefficients.

Note that in this approach we used a basis in $\mathbb{R}^2$ to describe a complex function. Therefore, the real and imaginary components of the filter are obtained by taking the real and imaginary components of $\boldsymbol{c}$, respectively.

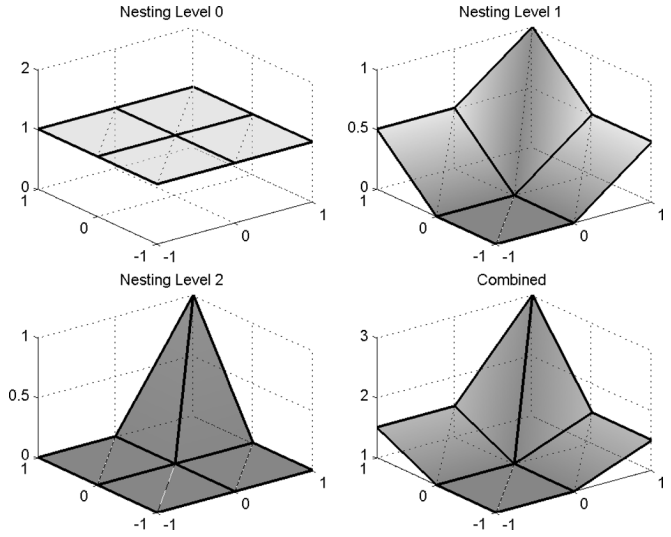


Fig. 1. Different nesting levels for the piecewise linear representation.

In addition and different from the case of the split complex approach where an entire region has no well-defined gradient [41], [60], the proposed fully complex PWL model has only a finite number of such points.

B. Adaptive Wiener Model Implementation

For the adaptive implementation, we need to derive updating recursions for both the nonlinear static part and the linear dynamic part of the Wiener model, i.e., for $\boldsymbol{c}$ and $\boldsymbol{h}$. In order to keep the computational complexity of the implementation low, a complex-valued stochastic gradient algorithm is derived that minimizes the mean-squared error given by

$$\begin{aligned} J[\boldsymbol{h},\boldsymbol{c}] &= E\left[|e(k)|^2\right] = E\left[|d(k)-y(k)|^2\right] \\ &= E\left[\left|d(k)-\boldsymbol{c}^H\boldsymbol{\Lambda}[v(k)]\right|^2\right] \end{aligned} \tag{11}$$

where $y(k) = y_R(k) + \mathrm{j}y_I(k)$ is the nonlinear filter output, estimating the desired model output $d(k) = d_R(k) + \mathrm{j}d_I(k)$.

In order to minimize $J[\boldsymbol{h},\boldsymbol{c}]$ of (11), it is possible to implement a steepest-descent algorithm to reduce the instantaneous error $e(k)$, yielding the following update equations:

$$\boldsymbol{h}(k+1) = \boldsymbol{h}(k) - \mu_h e^*(k)\boldsymbol{\psi}_h(k) \tag{12}$$

$$\boldsymbol{c}(k+1) = \boldsymbol{c}(k) - \mu_c e^*(k)\boldsymbol{\psi}_c(k) \tag{13}$$

where μ_h and μ_c are the step sizes controlling the convergence and final error of $\boldsymbol{h}$ and $\boldsymbol{c}$, respectively, and $\boldsymbol{\psi}_c(k)$ and $\boldsymbol{\psi}_h(k)$ represent the gradient estimates of the error function with respect to $\boldsymbol{c}$ and $\boldsymbol{h}$, respectively. Using differentiation techniques for complex-valued vectors [54], we get

$$\boldsymbol{\psi}_c(k) = -\boldsymbol{\Lambda}(v(k)) \tag{14}$$

$$\boldsymbol{\psi}_h(k) = -\boldsymbol{c}^H\left(\frac{\partial\boldsymbol{\Lambda}(v(k))}{\partial v_R} - \mathrm{j}\frac{\partial\boldsymbol{\Lambda}(v(k))}{\partial v_I}\right)\boldsymbol{x}(k) \tag{15}$$

where

$$\frac{\partial\boldsymbol{\Lambda}(v(k))}{\partial v_R} = \begin{bmatrix} 0 \\ \boldsymbol{\phi}^1(v_R(k)) \\ \boldsymbol{0}_{P\times 1} \\ \boldsymbol{\phi}_R^2(v_R(k), v_I(k)) \end{bmatrix} \text{ and}$$

$$\frac{\partial\boldsymbol{\Lambda}(v(k))}{\partial v_I} = \begin{bmatrix} 0 \\ \boldsymbol{0}_{P\times 1} \\ \boldsymbol{\phi}^1(v_I(k)) \\ \boldsymbol{\phi}_I^2(v_R(k), v_I(k)) \end{bmatrix} \tag{16}$$

with the following:

- $\boldsymbol{\phi}^1 : \mathbb{R} \to \mathbb{R}^P$ is the gradient of the first-order basis whose ith entry is defined by

$$\phi_i^1(\nu) = \begin{cases} \frac{1}{2}\left[1+\mathrm{sign}(\nu-\beta_i)\right] & \text{if } \nu \le \beta_P \\ 0 & \text{if } \nu > \beta_P \end{cases}; \tag{17}$$

- $\boldsymbol{\phi}_R^2 : \mathbb{R}^2 \to \mathbb{R}^{P^2}$ is the gradient of the second-order basis with respect to the in-phase component whose $(i-1)P+j$th entry is given by

$$\phi_{R,[(i-1)P+j]}^2(\nu_1,\nu_2) = \begin{cases} \frac{1}{2}\left[1+\mathrm{sign}(\nu_1-\beta_i)\right] & \text{if } \Upsilon_i^1(\nu_1) \le \Upsilon_j^1(\nu_2) \\ 0 & \text{if } \Upsilon_i^1(\nu_1) > \Upsilon_j^1(\nu_2) \end{cases} \tag{18}$$

  for $i, j = 1, \ldots, P$;

- $\boldsymbol{\phi}_I^2 : \mathbb{R}^2 \to \mathbb{R}^{P^2}$ is the gradient of the second-order basis with respect to the quadrature component whose $(i-1)P+j$th entry is given by

$$\phi_{I,[(i-1)P+j]}^2(\nu_1,\nu_2) = \begin{cases} 0 & \text{if } \Upsilon_i^1(\nu_1) \le \Upsilon_j^1(\nu_2) \\ \frac{1}{2}\left[1+\mathrm{sign}(\nu_2-\beta_j)\right] & \text{if } \Upsilon_i^1(\nu_1) > \Upsilon_j^1(\nu_2) \end{cases} \tag{19}$$

  for $i, j = 1, \ldots, P$.

In (17)–(19), $\mathrm{sign}(x)$ is defined as

$$\mathrm{sign}(x) = \begin{cases} -1, & x \le 0 \\ +1, & x > 0 \end{cases} \tag{20}$$

such that the gradient value at the edges of the partitions equals to zero.

To summarize, Tables I and II present the complete CS-CPWL algorithm. The simplest initial condition for the linear parameters $\boldsymbol{h}$ can be assumed to be a null vector. The nonlinear parameters $\boldsymbol{c}$ can be chosen to define a unit gain at the static nonlinearity, i.e., $\boldsymbol{c}_0 = \beta_1 - \mathrm{j}\beta_1$, $\boldsymbol{c}_1 = 1$, $\boldsymbol{c}_{P+1} = j$, and $\boldsymbol{c}_i = 0$ for $i \neq 0, 1, P+1$, for P sectors.

C. CS-CPWL Modeling Issues

Some aspects of the CS-CPWL algorithm modeling capabilities are discussed in the following.

- *Filter realization: user-defined segments.* The simplicial partition used in this paper differs from other PWL descriptions (e.g., [44] and [51]), since the parameters β_i are specified *a priori*. This user-defined specification leads to a reduction in the number of parameters to be estimated and

obviously simplifies the adaptation algorithm as the error function becomes linear in the set of parameters describing the nonlinearity.

- *Fixed nonlinear mapping (known* a priori*) is not necessarily invertible.* An important point in the identification of the nonlinear block of the Wiener structure is the existence of the inverse of the nonlinear function. This is an important limitation of most of the identification methods. However, if the nonlinear function is known *a priori*, and we need to identify only the linear block, the inverse of the nonlinear mapping can be relaxed [25]. Moreover, in some applications the inversion of the nonlinearity can lead to severe amplification of measurement noise.
- *CS-CPWL mapping versus other alternatives.* There are alternatives for the PWL mapping to describe the nonlinear block. The use of power series has the drawback that the approximation at points not included in the data set may be poor [25]. Another alternative is to use neural networks, but this choice makes the parameter estimation harder. We can also use hinging hyperplanes [55]. In fact, the PWL approach used here can be interpreted as a generalization of hinging hyperplanes.

IV. Convergence Analysis

In this section, a convergence analysis of the proposed algorithm is carried out using the ordinary differential equation (ODE) method [56].

Also other approaches for the study of convergence are possible. As referenced in the Introduction, several algorithms for a real-valued Wiener model were studied in detail in [29], [30] with an analysis of convergence in the mean and mean square error. Even when the results are limited to an FIR filter for the linear part of the Wiener model, convergence of parameters in the mean and MSE is characterized. To this purpose, two basic assumptions are used: the independence theory (not always realistic but useful in terms of theoretical analysis) and also the small step-size approximation.

Considering applications where the linear part of the Wiener model could be more general than an FIR filter (i.e., a filter with orthogonal rational transfer functions), we follow the ODE approach. The ODE approach, in addition to being a general proved analysis tool, gives results that seem to be easier to extend for future study. The cost of this extension possibility is that we can only prove local properties of the algorithms involved.

A. Static Gain Requirement

In order to proceed with the characterization of convergence properties of the CS-CPWL algorithm, we need to discuss the nonlinear Wiener model parameterization. Due to the cascade form of the Wiener filter, one specific partition of the PWL description of the static (differential) gain must be fixed for either the linear filter part or the static nonlinearity[2] [7]. In general, fixing the static gain avoids the existence of an infinite number of minima.

[2]Note that we can multiply the linear block by a constant ρ without changing the complete model if the nonlinear block is multiplied by a constant $1/\rho$.

Since the information to solve this problem is generally not available with input-output data, additional knowledge is required to fix the static gain. As discussed in [7] (see also [22]), anchoring the static gain of the linear part is not a good solution because the range of the output of the linear part $v(k)$ is not known *a priori*. As a consequence, with an inappropriate dynamic range for $v(k)$, the estimation of the static nonlinearity will not work properly. In addition, if a closed-loop application is considered, the behavior of the Wiener model is not adequate [6]. Therefore, we assume that the static gain is fixed at the static nonlinear part of the nonlinear filter for a particular simplex I_o.

B. Main Result

The ODE for the stochastic gradient-based algorithm proposed in this paper is related to the one obtained in [22], which also studied local parameter convergence for nonlinear Wiener models. Here, we extend the results of [22] to the case of complex-valued parameters and the particular case of a CS-CPWL description of the static nonlinearity.

A basic extension of the ODE analysis to complex-valued signals was presented in [57]. The extension can be summarized as basic regularity conditions that can be verified for the proposed CS-CPWL algorithm. These conditions are related to the differentiability and boundedness of the criterion and the average direction used. They are easily verified in general, since the only difference from the approach in [22] is the bounded static complex nonlinearity that is differentiable everywhere except at the partition limits. Due to space limitations, we refer the reader to [57] and [22] for the technical details of the ODE analysis.

By defining $\boldsymbol{\theta}_s = [\boldsymbol{h}^T \; \boldsymbol{c}^T]^T$ and $\boldsymbol{\psi}_s = [\boldsymbol{\psi}_h^T \; \boldsymbol{\psi}_c^T]^T$, the ODE associated with the CS-CPWL filter is

$$\frac{\partial \boldsymbol{\theta}_s}{\partial t} = -E\left\{\boldsymbol{\psi}_s e^*(k)\right\}. \tag{21}$$

Assuming that the true nonlinear model is described by the CS-CPWL filter except for the bounded zero-mean measurement noise $r(k)$ (assumed to be a stationary stochastic process not correlated with input $x(k)$), it can be easily verified that a stationary point $\overline{\boldsymbol{\theta}}_s$ of the proposed algorithm corresponds to the solution of $E\{\boldsymbol{\psi}_s e^*(k)\} = 0$. This rather restrictive assumption is usually made [44], [58].

With local convergence properties in mind, a linearization of (21) in the neighborhood of a stationary point $\overline{\boldsymbol{\theta}}_s$ leads to the following expression:

$$\frac{\partial \boldsymbol{\theta}_s}{\partial t} \cong -\left.\frac{\partial E\left\{\boldsymbol{\psi}_s e^*(k)\right\}}{\partial \boldsymbol{\theta}_s}\right|_{\overline{\boldsymbol{\theta}}_s} (\boldsymbol{\theta}_s - \overline{\boldsymbol{\theta}}_s). \tag{22}$$

In this way, using the definitions from (14) and (15), we obtain

$$\begin{aligned} \boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s) &= \left.\frac{\partial E\left\{\boldsymbol{\psi}_s e^*(k)\right\}}{\partial \boldsymbol{\theta}_s}\right|_{\overline{\boldsymbol{\theta}}_s} \\ &= E\left[\begin{matrix}\boldsymbol{c}^H\left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} - \mathrm{j}\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I}\right)\boldsymbol{x} \\ \boldsymbol{\Lambda}(v)\end{matrix}\right]\left[\begin{matrix}\boldsymbol{c}^H\left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} - \mathrm{j}\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I}\right)\boldsymbol{x} \\ \boldsymbol{\Lambda}(v)\end{matrix}\right]^H\Bigg|_{\overline{\boldsymbol{\theta}}_s}. \end{aligned} \tag{23}$$

Convergence can be guaranteed if the eigenvalues of $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ have positive real part. Note that, by construction, $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ is Hermitian and positive semidefinite. Therefore, convergence is automatically assured if $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ is positive definite.

In order to guarantee the positive definiteness of $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$, the static (differential) gain is fixed at some sector (simplex) of the static nonlinearity. Using the CS-CPWL description of the static nonlinearity, this means fixing an element of $\boldsymbol{c}$ for some partition interval $I_o \in (\beta_{I_o-1}, \beta_{I_o})$ such that

$$\left\| \boldsymbol{c}^H \left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} - \mathrm{j} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I} \right) \right\| = k_o > \delta_1 \quad v \in I_o \tag{24}$$

for some positive constant δ_1.

Following the convergence analysis, we can extract the contribution of $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ in the interval I_o. This can be performed using the *gate function* defined by

$$\mathtt{gate}(v) = \begin{cases} 1, & \text{if } v \in I_o \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

where v is the output of the linear part.

Convergence results related to the CS-CPWL algorithm are summarized in the following theorem.

Theorem 1: Consider the CS-CPWL algorithm described in Tables I and II. Assume the following.

1) $x(k)$ is a bounded, zero-mean stationary stochastic process persistently exciting of order M.
2) The bounded, zero-mean measurement noise $r(k)$ is a stationary stochastic process not correlated with input $x(k)$.
3) $x(k)$ is such that the probability density function of $v(k|\overline{\boldsymbol{\theta}}_s)$ (the output of the linear part of the Wiener model at a stationary point) fulfills $h_v(v) \geq \delta > 0$ in at least one partition I_i.
4) $E\left\{\mathtt{gate}(v)\boldsymbol{x}\boldsymbol{x}^H\right\} \geq \delta E\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}$, for some constant $\delta > 0$.
5) The Wiener system to be identified corresponds to the model set, i.e., the linear part is an FIR filter of order N and the static nonlinearity of the system is contained in the complex PWL model set.
6) $\left\|\boldsymbol{c}^H \left[\partial \boldsymbol{\Lambda}(v)/\partial v_R - \mathrm{j}\partial \boldsymbol{\Lambda}(v)/\partial v_I\right]\right\| = k_o$, for some positive constant k_o and for $v \in I_o$.

Then, the proposed algorithm is locally convergent to a stationary point $\overline{\boldsymbol{\theta}}_s$. ♢

Proof: See the Appendix.

Some comments regarding the convergence assumptions and properties are as follows.

- Assumptions 3 and 4 imply that the input signal $x(k)$ should have sufficient energy located at any arbitrary simplex I_i (Assumption 3) and also at the particular simplex I_o, where the static gain is fixed. This implies conditions for the amplitude distribution of the input signal.
- Different from other criteria than the mean-squared output error, no positive real condition is needed. This is coherent with the results of [22].
- Global convergence can be verified for real-valued signals using a simplified two-step scheme with a known static nonlinearity [29].

TABLE I
COMPLEX-VALUED SIMPLICIAL CPWL ADAPTIVE FILTER ALGORITHM

Definitions:

$v(k)$ output of the linear part

$\boldsymbol{\Lambda}[v]$ Simplicial piecewise linear function

Parameters:

N = number of $\boldsymbol{h}$ coefficients

β_i, $i = 1, \cdots, P$ = sectors of (v_R, v_I)

M = number of $\boldsymbol{c}$ coefficients $(= 1 + 2P + P^2)$

μ_h = step size related to $\boldsymbol{h}$ coefficients

μ_c = step size related to $\boldsymbol{c}$ coefficients

Data: $\boldsymbol{x}(k)$ input vector $(N \times 1)$.

$d(k)$ desired response at instant k.

Initialization

$\boldsymbol{h}(0) = \boldsymbol{0}$

$\boldsymbol{c}(0) = [(\beta_1 - j\beta_1),\ 1,\ 0,\ \cdots,\ 0]^H$

- The essential characteristic of the CS-CPWL algorithm is the improved modeling capability of the high-level representation of the complex static nonlinearity, i.e., the second-order nesting level of the CS-CPWL algorithm.

V. COMPUTATIONAL COMPLEXITY AND IMPLEMENTATION ISSUES

The steps of the CS-CPWL filter implementation are summarized in Tables I and II. The only nonstandard feature in the implementation is the evaluation of absolute values. It involves fewer parameters than necessary in [44] (which was presented for the real domain case). The main difference between the implementations is that for the CS-CPWL realization we only need a single FIR filter. Other adaptive filtering algorithms than a stochastic gradient-based algorithm can be used for estimating the parameters of proposed the CS-CPWL realization. The stochastic gradient version was used mostly to illustrate the characteristics and convergence of the CS-CPWL description.

Implementation issues of the CS-CPWL algorithm are discussed below.

A. Selection of PWL Segments

A key aspect of the proposed algorithm is the selection of the partition β_i for $i = 1, \ldots, P$. The CS-CPWL algorithm uses a fixed set of grid points, which is the same for the in-phase and quadrature components. The interval $[\beta_1, \beta_P]$ must coincide with the range of the signal $v_R(k)$ (and $v_I(k)$) keeping in mind that the linear part of the Wiener model may be time varying. This is solved by choosing the interval $[\beta_1, \beta_P]$ wide enough with respect to the variation of the linear filter parameters and the input-signal range. Choosing the interval $[\beta_1, \beta_P]$ too small is not expected to cause any problems for the practical case when the nonlinearity is bounded. After choosing $[\beta_1, \beta_P]$, the interior points need to be determined. If we have knowledge of the nonlinearity, a higher density of points can be used in the intervals where the slope of the nonlinearity changes significantly than in the intervals where it is close to linear. However, if we do not

TABLE II
COMPLEX SIMPLICIAL CPWL ADAPTIVE FILTER ALGORITHM (CONT. TABLE I)

For each $k = 1, 2, \cdots$

$v(k) = \boldsymbol{h}^H \boldsymbol{x}(k)$

For $i = 1$ to P,

$$\Upsilon_i^1 = \begin{cases} \frac{1}{2}\left(v_R - \beta_i + |v_R - \beta_i|\right) & \text{if } v_R \leq \beta_P \\ \frac{1}{2}\left(\beta_P - \beta_i + |\beta_P - \beta_i|\right) & \text{if } v_R > \beta_P \end{cases}$$

$$\overline{\Upsilon}_i^1 = \begin{cases} \frac{1}{2}\left(v_I - \beta_i + |v_I - \beta_i|\right) & \text{if } v_I \leq \beta_P \\ \frac{1}{2}\left(\beta_P - \beta_i + |\beta_P - \beta_i|\right) & \text{if } v_I > \beta_P \end{cases}$$

For $i = 1$ to P,

For $j = 1$ to P,

$$\Upsilon_{(i-1)P+j}^2 = \begin{cases} \Upsilon_i^1 & \text{if } \Upsilon_i^1 \leq \overline{\Upsilon}_j^1 \\ \overline{\Upsilon}_j^1 & \text{if } \Upsilon_i^1 > \overline{\Upsilon}_j^1 \end{cases}$$

$\boldsymbol{\Lambda} = [1 \;\; \boldsymbol{\Upsilon}^1 \;\; \overline{\boldsymbol{\Upsilon}}^1 \;\; \boldsymbol{\Upsilon}^2]^T$

$y(k) = \boldsymbol{c}^H(k)\boldsymbol{\Lambda}$

$e(k) = d(k) - y(k)$

$\boldsymbol{\psi}_c(k) = -\boldsymbol{\Lambda}$

For $i = 1$ to P,

$$\phi_i^1 = \begin{cases} \frac{1}{2}\left[1 + \text{sign}(v_R - \beta_i)\right] & \text{if } v_R \leq \beta_P \\ 0 & \text{if } v_R > \beta_P \end{cases}$$

$$\overline{\phi}_i^1 = \begin{cases} \frac{1}{2}\left[1 + \text{sign}(v_I - \beta_i)\right] & \text{if } v_I \leq \beta_P \\ 0 & \text{if } v_I > \beta_P \end{cases}$$

For $i = 1$ to P,

For $j = 1$ to P,

$$\phi_{(i-1)P+j}^2 = \begin{cases} \frac{1}{2}\left[1 + \text{sign}(v_R - \beta_i)\right] & \text{if } \Upsilon_i^1 \leq \overline{\Upsilon}_j^1 \\ 0 & \text{if } \Upsilon_i^1 > \overline{\Upsilon}_j^1 \end{cases}$$

$$\overline{\phi}_{(i-1)P+j}^2 = \begin{cases} 0 & \text{if } \Upsilon_i^1 \leq \overline{\Upsilon}_j^1 \\ \frac{1}{2}\left[1 + \text{sign}(v_I - \beta_j)\right] & \text{if } \Upsilon_i^1 > \overline{\Upsilon}_j^1 \end{cases}$$

$\boldsymbol{\phi}_R = [0 \;\; \boldsymbol{\phi}^1 \;\; \mathbf{0}_{P\times 1} \;\; \boldsymbol{\phi}^2]^T$

$\boldsymbol{\phi}_I = [0 \;\; \mathbf{0}_{P\times 1} \;\; \overline{\boldsymbol{\phi}}^1 \;\; \overline{\boldsymbol{\phi}}^2]^T$

$\boldsymbol{\psi}_h(k) = -\mathbf{c}^H(\boldsymbol{\phi}_R - \mathrm{j}\boldsymbol{\phi}_I)\boldsymbol{x}(k)$

$\mathbf{c}(k+1) = \mathbf{c}(k) - \mu_c e^*(k)\boldsymbol{\psi}_c(k)$

$\mathbf{h}(k+1) = \mathbf{h}(k) - \mu_h e^*(k)\boldsymbol{\psi}_h(k)$

To be consistent with Eq. (8), Υ_i^1 is equal to $\Upsilon_i^1[v_R(k)]$, and $\overline{\Upsilon}_i^1$ is equal to $\Upsilon_i^1[v_I(k)]$

To be consistent with Eq. (17), ϕ_i^1 is equal to $\phi_i^1[v_R(k)]$, and $\overline{\phi}_i^1$ is equal to $\phi_i^1[v_I(k)]$.

know anything about the nonlinearity, it is common practice to choose a uniform partition.

B. Tradeoff Between Flexibility of the Linear Part (FIR) and Modeling Capabilities

An FIR model is used for the linear part of the Wiener model. In general, this involves a large number of parameters. It is possible to reduce the complexity of this part of the model using the Kautz basis with fixed poles [59]. In order to improve the modeling capability of the PWL description, we should increase the number of sectors. This implies an increased number of parameters in $\boldsymbol{c}$. For example, if we duplicate the number of sectors (from P to $2P$), the number of parameters increases by $3P^2 + 2P$ (see (4)).

C. Step Sizes μ_h and μ_c

Following the ideas of [44], it is possible to bound the parameters μ_h and μ_c. In order to guarantee stability of (12), step size μ_h should be chosen as

$$0 < \mu_h(k) < \frac{1}{\lambda_{\mu_h}(k)} \tag{26}$$

where $\lambda_{\mu_h}(k)$ is the maximum eigenvalue of the matrix $E[\boldsymbol{\psi}_h(k)\boldsymbol{\psi}_h^H(k)]$. To obtain a useful bound on $\lambda_{\mu_h}(k)$, it can be noted that [see the equation at the bottom of the next page].

TABLE III
COMPUTATIONAL COMPLEXITY COMPARISON. N: DIMENSION OF THE INPUT-SIGNAL VECTOR, P: ORDER (IN THE CASE OF VOLTERRA AND WIENER FILTERS) AND NUMBER OF SECTORS IN THE CS-CPWL)

	Parameters	Filtering	Updating	Additional Functions
CLMS [54]	N	N	N	—
Bershad [29]	$N+4$	$N+4$	$N+4$	*norm cdf*
CLMS + *sin* [60]	N	N	N	*sin*
Volterra [61]	$\sum_{i=0}^{P} N^i$	$\sum_{i=0}^{P} N^i$	$\sum_{i=0}^{P} N^i$	—
Wiener (polynomial) [62]	$P+N+1$	$(P-1)+P+N$	$(P-2)+P+N$	—
CS-CPWL	$2P+P^2+N+1$	$2P+P^2+N+1$	$2P+2P^2+N$	Λ

In the last step, we use the fact that the entries of $(\partial\mathbf{\Lambda}[v(k)]/\partial v_R - \mathrm{j}\partial\mathbf{\Lambda}(v(k))/\partial v_I)$ only include terms 0, 1 or j. Therefore, a tighter bound for the step size μ_h is

$$0 < \mu_h(k) < \frac{1}{\left(\sum_{i=1}^{M} |\, \boldsymbol{c}_i\,|^2\right) E\left\{\boldsymbol{x}^H(k)\boldsymbol{x}(k)\right\}}. \tag{27}$$

Similarly, to ensure the stability of (13), step size μ_c is chosen as

$$0 < \mu_c(k) < \frac{1}{\lambda_{\mu_c}(k)} \tag{28}$$

where $\lambda_{\mu_c}(k)$ is the maximum eigenvalue of the matrix $E[\boldsymbol{\psi}_c(k)\boldsymbol{\psi}_c^H(k)]$. A practical bound for $\lambda_{\mu_c}(k)$ can be obtained by

$$\begin{aligned}\lambda_{\mu_c}(k) &\le tr\left\{\boldsymbol{\psi}_c(k)\boldsymbol{\psi}_c^H(k)\right\}\\ &= tr\left\{\mathbf{\Lambda}(v(k))\mathbf{\Lambda}^T(v(k))\right\} < \left(1+\sum_{i=1}^{P-1}(\beta_P-\beta_i)^2\right).\end{aligned}$$

In the last step, we used the fact that the first entry of $\mathbf{\Lambda}(v(k))$ is bounded by 1, and the rest by $(\beta_P - \beta_i)$. Then

$$0 < \mu_c < \frac{1}{\left(1+\sum_{i=1}^{P-1}(\beta_P-\beta_i)^2\right)}. \tag{29}$$

D. Computational Complexity

The computational complexity in terms of the number of multiplications (for filtering and coefficient updates) is shown in Table III for five different adaptive filtering strategies whose performances will be evaluated in the next section.

1) The complex LMS (CLMS) filter [54]. Included here to provide a benchmark for linear filtering techniques. The filter proposed in [29] is a CLMS filter followed by a pre/postscaled normal distribution function (norm cdf).
2) The CLMS filter followed by the elementary sin activation function [60] illustrating the capability of such fixed-shape function to represent nonlinear complex-valued functions.[3]
3) The Volterra filter approach used, e.g., in [61] for nonlinear equalization of digital satellite channels.
4) The Wiener model formed by a complex FIR filter and a polynomial, used, e.g., in [62] in the context of adaptive precompensation.
5) The Wiener model consisting of a complex FIR and the proposed CS-CPWL filter.

In Table III, N represents the number of taps at the input of the filter and P represents the order (in the case of Volterra and Wiener filters) and the number of sectors in the CS-CPWL. The filter proposed in [29] has a complexity similar to the CLMS algorithm (increased by four parameters in order to prescale and postscale the nonlinear gain). Note that this filter also requires the implementation of the normal distribution function. From this table, the Wiener model with a polynomial nonlinearity has a lower number of parameters than the proposed CS-CPWL implementation (for the same P). However, for the same level of approximation, P in the Wiener polynomial filter should be larger than the number of sectors in the CS-CPWL. For the CS-CPWL implementation, it is necessary to compute Λ, which only implies computing absolute values of known quantities.

VI. SIMULATION EXAMPLE

In this section, the modeling capability of the CS-CPWL filter is illustrated by identifying a nonlinear Wiener model whose linear part is given by [61]

$$v(k) = h_0x(k) + h_1x(k-1) + h_2x(k-2) + h_3x(k-3)$$

[3]Other elementary functions were tested (atan, asin, th). The best performance for the example in Section IV was with a sin.

$$\begin{aligned}\lambda_{\mu_h}(k) &\le \mathrm{tr}\left\{\boldsymbol{\psi}_h(k)\boldsymbol{\psi}_h^H(k)\right\} = \mathrm{tr}\left\{\left[\boldsymbol{c}^H\left(\frac{\partial\mathbf{\Lambda}[v(k)]}{\partial v_R} - \mathrm{j}\frac{\partial\mathbf{\Lambda}[v(k)]}{\partial v_I}\right)\boldsymbol{x}(k)\right]\left[\boldsymbol{c}^H\left(\frac{\partial\mathbf{\Lambda}[v(k)]}{\partial v_R} - \mathrm{j}\frac{\partial\mathbf{\Lambda}[v(k)]}{\partial v_I}\right)\boldsymbol{x}(k)\right]^H\right\}\\ &< \left(\sum_{i=1}^{M} |\, \boldsymbol{c}_i\,|^2\right) E\left\{\boldsymbol{x}^H(k)\boldsymbol{x}(k)\right\}.\end{aligned}$$

TABLE IV
SIMULATION PARAMETERS FOR DIFFERENT NONLINEAR FILTERS

Filter	Parameters (linear part)	Parameters (nonlinear part)
CLMS [54]	$N = 4$	—
CLMS + sin [60]	$N = 4$	sin function
Volterra [61]	$N = 4$	5th-order
Wiener (polynomial) [62]	$N = 4$	12th-order
CS-CPWL	$N = 4$	$\beta = [-2 \cdots 2]$ with $\Delta = 0.5$ $(P = 9)$

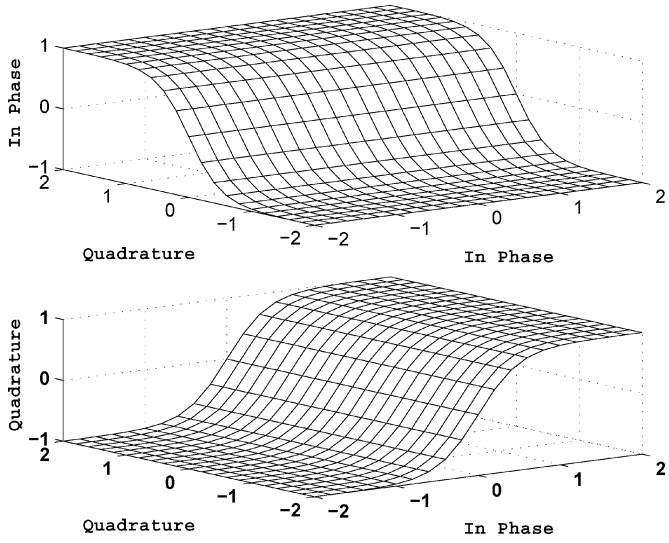


Fig. 2. Nonlinearity associated to the real and imaginary QAM signal components.

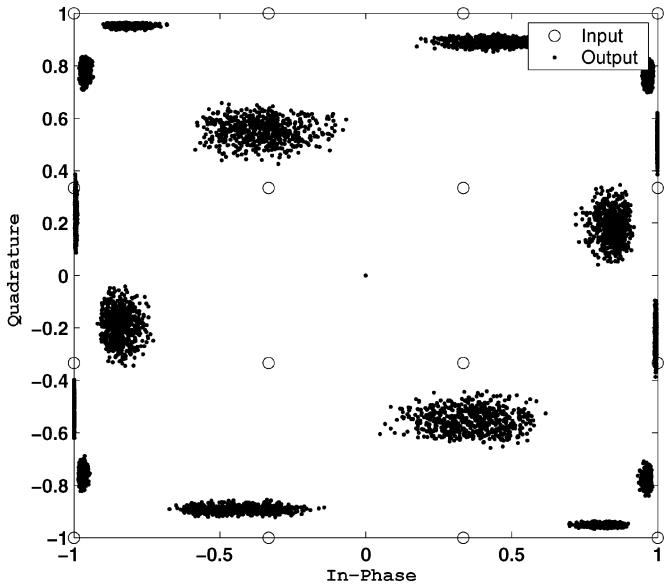


Fig. 3. Typical distorted 16-QAM signal constellation.

where $h_0 = 0.036 + \mathrm{j}0.031$, $h_1 = -0.024 - \mathrm{j}0.014$, $h_2 = 0.063 - \mathrm{j}0.001$, and $h_3 = 1.22 + \mathrm{j}0.646$. The static nonlinearity of the plant to be identified is given by [44]

$$y(k) = \frac{e^{v_R(k)/0.25} - 1}{e^{v_R(k)/0.25} + 1} + \mathrm{j}\frac{e^{v_I(k)/0.5} - 1}{e^{v_I(k)/0.5} + 1} + r(k)$$

where $r(k)$ is a complex, bounded, and zero-mean noise sequence and $x(k)$ is a non-constant-modulus QAM signal. Fig. 2 illustrates the nonlinearity for the real and imaginary components, and Fig. 3 depicts the 16-QAM constellation before and after passing through the specified nonlinear system. The signal-to-noise ratio is fixed to 40 dB.

The strategies in Table III were implemented in order to compare their modeling capability. The number of coefficients used with the linear part and the nonlinear part for each approach can be found in Table IV. The number of parameters used was 1364 with the Volterra, 24 with the polynomial, and 104 with the CS-CPWL approach. The number of parameters was selected to obtain a good steady-state error. The step sizes kept constant during the adaptation and were selected to obtain comparable convergence speeds (but ensuring the convergence). The CS-CPWL has a relatively high number of parameters but the complexity for their adaptation is low.

Fig. 4 depicts the MSE versus the number of iterations for the different strategies. The curves were obtained by averaging 100 independent realizations. The step sizes for the proposed CS-CPWL approach were chosen as $\mu_h = 10^{-3}$ and $\mu_c = 5 \cdot 10^{-3}$ to be in accordance with the bounds in (27) and (29). In the adaptation of the nonlinear parameters, the value of $\boldsymbol{c}(i_0)$ (where i_0 is the entry associated with the second-order basis element that contains $v = 0$) is extracted from the adaptation procedure and computed at each iteration to obtain the condition of (24) with $k_0 = 1$. The Wiener polynomial of order 12 was also adapted using an LMS algorithm, and the step sizes for the linear and static nonlinear parts were chosen as $\mu_h = 10^{-3}$ and $\mu_f = 2 \cdot 10^{-3}$, respectively.[4] The fifth-order Volterra filter was updated using an LMS algorithm with individual step sizes for each order, where $\mu_1 = 10^{-3}$, $\mu_2 = 7 \cdot 10^{-4}$, $\mu_3 = 10^{-4}$, $\mu_4 = 10^{-4}$, and $\mu_5 = 5 \cdot 10^{-5}$. The realization in [29] is also included in the comparison. For that purpose, the simultaneous adaptation of the linear part and the gains related to the known shape static nonlinearity is performed. The step sizes used in that case are $\mu_{\text{linear}} = 10^{-4}$ for the linear CLMS, and $\mu_s = 10^{-4}$, for the adaptation of the prescale and postscale parameters of the nonlinear gain. Note that since the filter in [29] is real-valued, a

[4]Simulations (not included) for several polynomial orders, using different step sizes, were performed. The step size in each case was chosen following a tradeoff between stability of the corresponding algorithm and mean-squared error. For orders higher than 30, the algorithm diverges. The best result was obtained for order 12, but the final error is still larger than that of the CS-CPWL. Modifications on the step sizes cannot reduce the error further.

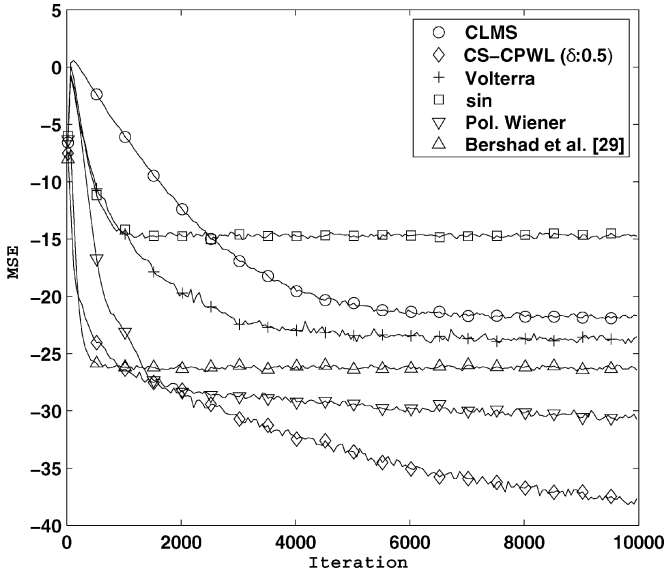


Fig. 4. Mean-squared error (MSE) for different nonlinear filtering schemes.

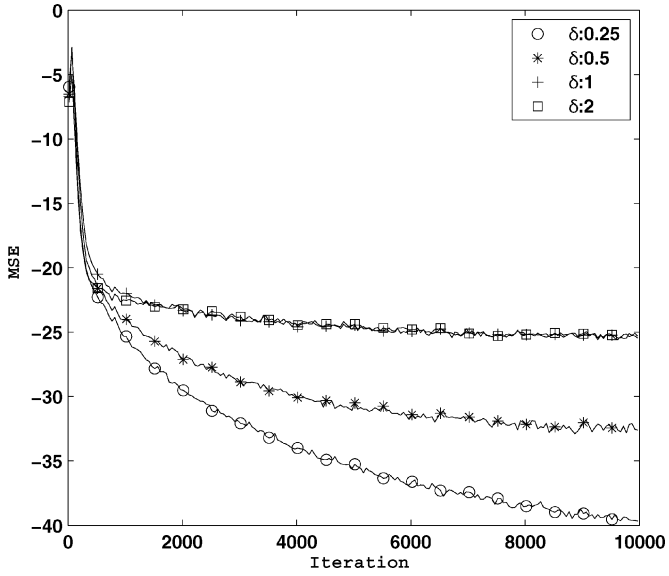


Fig. 5. Mean-squared error (MSE) for CS-CPWL algorithm with different segment sizes Δ.

split complex structure is required. Since this structure does not follow a full complex filtering, as in the case of the CS-CPWL filter, this result should be considered carefully.

As can be seen from the figure, the performance in terms of obtainable convergence speed is similar for almost all the studied structures. In particular, the CS-CPWL algorithm presents a good speed in the first portion of the curve but slows down close to steady state.

The linear CLMS filter shows a large final error. This is because the saturation effect of the nonlinearity is not modeled. The use of the `sin` function as nonlinearity does not improve this value. This is due to the fact that the `sin` function introduces saturation at the output but this saturation is not adapted to the real systems. In the original application [60], elementary functions (like `sin`) were used as part of a neural network realization.

The performance of the Volterra filter is inferior to that of the Wiener model with CS-CPWL and polynomial filters. This is coherent with the well-known sensitivity of the truncated Volterra model to the input data which leads to numerical problems in parameter estimation [4], [5]. To improve the results of the Volterra filter, more parameters will be required.

The Wiener model with a polynomial static nonlinearity presents a larger final error than the CS-CPWL. The use of smaller step sizes cannot reduce the error further (simulations not included). The problem with the polynomial model is that it does not provide a good basis for describing general nonlinearities because a high number of parameters is required. In order to improve numerical properties for high-order cases, it is possible to use orthogonal polynomials [63].

The low complexity realization of [29] presents better final errors than other alternatives that use fixed nonlinear functions. However, a better performance is obtained when it is possible to adapt the shape of the nonlinearity (not just the scale), like in the proposed CS-CPWL algorithm.

Fig. 5 shows the MSE versus the number of iterations for four different values of the segment size Δ of the CS-CPWL algorithm. The curve was obtained by averaging 50 realizations. As can be seen from the figure, a better approximation can be obtained when the number of parameters increases (the segment size gets smaller). However, the computational complexity will increase.

To summarize, the tradeoff that the CS-CPWL realization introduces in terms of computational complexity and MSE performance is better than for the Wiener filter with polynomial static nonlinearity, or other reduced complexity schemes. This is due to the increasing modeling capabilities of the CS-CPWL for high-order approximations.

VII. Conclusion

This paper proposed a low-complexity nonlinear Wiener model consisting of a complex-valued simplicial canonical piecewise linear (CS-CPWL) filter. The resulting structure needs fewer parameters than others found in the literature and converges fast to a low MSE level. An adaptive implementation was derived and a local convergence analysis was carried out using the ordinary differential equation (ODE) method. The low computational complexity is due to the special user-defined partitions of the static nonlinear function of the Wiener model. Furthermore, the *a priori* defined partitions make the error function linear in parameters describing the nonlinearity. This *a priori* information can usually be easily extracted in the modeling and linearization of nonlinear wideband power amplifiers in communications systems.

The performance of the proposed Wiener model using a CS-CPWL filter was tested in a system identification setup. The results show that using the proposed structure, an improvement in the MSE can be achieved as compared to the widely used Wiener model with a polynomial static nonlinearity. Furthermore, the polynomial model is also prone to unstable behavior for high orders.

APPENDIX
PROOF OF THEOREM 1

The proof follows similar steps as in [6] and [22], with some differences in the used static nonlinearity. Using the definition of the function $\mathtt{gate}(v)$ in (25), we can write

$$\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s) = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^H & \boldsymbol{D} \end{bmatrix}_{\overline{\boldsymbol{\theta}}_s} + \begin{bmatrix} \tilde{\boldsymbol{A}} & \boldsymbol{0} \\ \boldsymbol{0}^H & \boldsymbol{0} \end{bmatrix}_{\overline{\boldsymbol{\theta}}_s}$$

where [see the equation at the bottom of the page], where $\boldsymbol{\Lambda}(v)_{I_o}$ is the vector which concentrates the entries of the basis $\boldsymbol{\Lambda}$ related with the sector I_o. The ith entry of this vector is $[\boldsymbol{\Lambda}(v)_{I_0}]_i = [\boldsymbol{\Lambda}(v)]_i$ if $[\boldsymbol{\Lambda}(v_o)]_i \neq 0$ for $v_o \in I_0$ and $[\boldsymbol{\Lambda}(v)_{I_0}]_i = 0$ for the other entries.

Note that $[1-\mathtt{gate}(v)]^2 = [1-\mathtt{gate}(v)]$. Before proceeding, we consider the following.

Lemma A1 [6]: Consider the block matrix

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^H & \boldsymbol{D} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{A}} & \boldsymbol{0} \\ \boldsymbol{0}^H & \boldsymbol{0} \end{bmatrix}. \tag{30}$$

If the following holds:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^H & \boldsymbol{D} \end{bmatrix} \geq 0, \quad \tilde{\boldsymbol{A}} > 0, \quad \boldsymbol{D} > 0$$

then (30) is positive definite. □

By invoking Lemma A1, $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ is positive definite and local convergence of the CS-CPWL algorithm is established if $\tilde{\boldsymbol{A}} > 0$ and $\boldsymbol{D} > 0$, as shown in the following section.

Positive Definiteness of $\tilde{\boldsymbol{A}}$ and $\boldsymbol{D}$: Following the definition of $\tilde{\boldsymbol{A}}$, by fixing the static nonlinearity at the partition interval I_o means that

$$\tilde{\boldsymbol{A}} \geq E\left\{ \mathtt{gate}(v) \begin{bmatrix} \delta_1^2 \boldsymbol{x}\boldsymbol{x}^H & \delta_1 \boldsymbol{x} \\ \delta_1 \boldsymbol{x}^H & 1 \end{bmatrix} \right\}.$$

Assuming now that

$$E\left\{\mathtt{gate}(v)\boldsymbol{x}\boldsymbol{x}^H\right\} \geq \delta_2 E\left\{\boldsymbol{x}\boldsymbol{x}^H\right\}, \quad \delta_2 > 0 \tag{31}$$

and assuming a zero mean input, we finally obtain

$$\tilde{\boldsymbol{A}} \geq \delta_2 \begin{bmatrix} \delta_1^2 E\left\{\boldsymbol{x}\boldsymbol{x}^H\right\} & \delta_1 E\left\{\boldsymbol{x}\right\} \\ \delta_1 E\left\{\boldsymbol{x}\right\}^H & 1 \end{bmatrix} = \delta_2 \delta_1^2 \begin{bmatrix} E\left\{\boldsymbol{x}\boldsymbol{x}^H\right\} & \boldsymbol{0} \\ \boldsymbol{0} & 1 \end{bmatrix} > 0$$

if the input is persistently exciting of order N. The additional assumption (31) is a condition on the amplitude distribution of the input signal. In other words, the input signal must have enough energy in the interval I_o in order to guarantee this assumption.

On the other hand, the positive definiteness of $\boldsymbol{D}$ is verified using particular properties of the PWL description used in the proposed algorithm.

As shown in [42, Lemma 2], the high-level piecewise linear description used for the CS-CPWL algorithm for an arbitrary partition can be written at a stationary point as

$$\bar{\boldsymbol{\Gamma}}(v) = \boldsymbol{T}\boldsymbol{\Lambda}(v)$$

where $\boldsymbol{T}$ is a positive definite upper triangular matrix of suitable dimensions, and $\boldsymbol{\Gamma}(v) = [\gamma_0(v) \cdots \gamma_{M-1}(v)]^T$ is a vector whose nesting level components, in this case of order 0, 1, and 2, are orthonormal. Using this property, we get

$$\begin{aligned} \boldsymbol{D} &= E\left\{[1-\mathtt{gate}(v)]\boldsymbol{\Lambda}(v)\boldsymbol{\Lambda}(v)^T\right\} = E\left\{\boldsymbol{\Lambda}(v)\boldsymbol{\Lambda}(v)^T\right\} \\ &= \boldsymbol{T}^{-T} E\left\{\bar{\boldsymbol{\Gamma}}(v)\bar{\boldsymbol{\Gamma}}(v)^T\right\}\boldsymbol{T}^{-1} \end{aligned}$$

where the same property that leads to (30) is used, i.e., $\boldsymbol{\Lambda}(v)$ corresponds to the derivative with respect to $\boldsymbol{c}$ coefficients, that is defined zero at the partition I_0.

Then, if the probability density function $h_v(v)$ of v is such that

$$h_v(v) \geq \delta > 0$$

in at least one nonzero interval I_i, it follows that

$$\begin{aligned} E\left\{\boldsymbol{\Gamma}(v)\boldsymbol{\Gamma}(v)^T\right\} &= \sum_{i=0}^{M-1} E\left\{\gamma_i^2(v)\right\} \\ &= \sum_{i=0}^{M-1} \int_{I_i, I_i \neq I_0} \gamma_i^2(v) h_v(v) dv > 0 \end{aligned}$$

$$\begin{aligned} \boldsymbol{A} &= E\left\{ [1-\mathtt{gate}(v)] \begin{bmatrix} \boldsymbol{x}\boldsymbol{c}^H \left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R}^T + \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I}^T \right) \boldsymbol{c}\boldsymbol{x}^H & \boldsymbol{c}^H \left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} - \mathrm{j} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I} \right) \boldsymbol{\Lambda}(v)_{I_o}^T \\ \boldsymbol{\Lambda}(v)_{I_o} \left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R}^T + \mathrm{j} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I}^T \right) \boldsymbol{c} & \boldsymbol{\Lambda}(v)_{I_o} \boldsymbol{\Lambda}(v)_{I_o}^T \end{bmatrix} \right\} \\ \boldsymbol{B} &= E\left\{ [1-\mathtt{gate}(v)] \begin{bmatrix} \boldsymbol{c}^H \left(\frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_R} - \mathrm{j} \frac{\partial \boldsymbol{\Lambda}(v)}{\partial v_I} \right) \boldsymbol{\Lambda}(v)^T \\ \boldsymbol{\Lambda}(v)_{I_o} \boldsymbol{\Lambda}(v) \end{bmatrix} \right\} \\ \boldsymbol{D} &= E\left\{[1-\mathtt{gate}(v)]\boldsymbol{\Lambda}(v)\boldsymbol{\Lambda}(v)^T\right\} \\ \tilde{\boldsymbol{A}} &= E\left\{ \mathtt{gate}(v) \begin{bmatrix} k_o^2 \boldsymbol{x}\boldsymbol{x}^H & k_o \boldsymbol{x} \\ k_o \boldsymbol{x}^H & 1 \end{bmatrix} \right\}. \end{aligned}$$

in at least one nonzero partition I_i. The continuity of $\gamma_i(v)$ finally allows to conclude the same for any partition, and as a consequence $\boldsymbol{D} > 0$.

Finally, the positive definiteness of $\tilde{\boldsymbol{A}}$ and $\boldsymbol{D}$ now implies that $\boldsymbol{P}_s(\overline{\boldsymbol{\theta}}_s)$ is positive definite. This result is associated to the linearization of the ODE, (22), and implies the local convergence of the CS-CPWL algorithm.

REFERENCES

[1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Norwell, MA: Kluwer Academic, 1997.

[2] J. C. Pedro and N. B. Carvalho, *Intermodulation Distortion in Microwave and Wireless Circuits*. Boston, MA: Artech House, 2003.

[3] J. Terry and J. Heiskala, *OFDM Wireless LANs: A Theoretical and Practical Guide*. Indianapolis, IN: SAMS, 2001.

[4] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, S. Glorennec, H. Hjalmarsson, and A. Juditskys, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, pp. 1691–1724, Dec. 1995.

[5] R. Haber and L. Keviczky, "Identification of nonlinear dynamic systems," in *Proc. Int. Federat. Aut. Control Symp.*, 1976, pp. 62–112.

[6] T. Wigren, "Recursive identification based on the nonlinear Wiener model," Ph.D. dissertation, Technology Dept., Uppsala University, Uppsala, Sweden, 1990.

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

[8] A. E. Nordsjö and L. H. Zetterberg, "Identification of certain time-varying nonlinear Wiener and Hammerstein systems," *IEEE Trans. Signal Process.*, vol. 49, pp. 577–592, 2001.

[9] A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett, "A new approach to the identification of pH processes based on Wiener model," *Chem. Eng. Sci.*, vol. 50, no. 23, pp. 3693–3701, 1995.

[10] G. A. Pajunen, "Adaptive control of Wiener type nonlinear systems," *Automatica*, vol. 28, no. 4, pp. 781–785, 1999.

[11] R. K. Pearson and M. Pottmann, "Gray-box identification of block-oriented nonlinear models," *J. Process Control*, vol. 10, pp. 301–315, 2000.

[12] Y. Zhu, "Distillation column identification for control using Wiener model," in *Proc. Amer. Control Conf.*, San Diego, CA, 1999, vol. 5, pp. 3462–3466.

[13] J. C. Ralston, A. M. Zoubir, and B. Boashash, "Identification of a class of time-varying nonlinear system based on the Wiener model with application to automotive engineering," *IEICE Trans. Fundam. Electron., Commun., Comput. Sci.*, pp. 1192–1200, 1995.

[14] D. T. Westwick and R. E. Kearney, "Nonparametric identification of nonlinear biomedical systems. Part I. Theory," *Crit. Rev. Biomed. Eng.*, vol. 26, pp. 153–226, 1998.

[15] P. Celka and P. Colditz, "Nonlinear nonstationary Wiener model of infant EEG seizures," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 6, pp. 556–564, Jun. 2002.

[16] S. J. Norquay, A. Palazoglu, and J. A. Romagnoli, "Model predictive control based on Wiener models," *Chem. Eng. Sci.*, vol. 53, no. 1, pp. 75–84, 1998.

[17] S. Gerkšič, D. Juričic, S. Strmčnic, and D. Matko, "Wiener model based nonlinear predictive control," *Int. J. Syst. Sci.*, vol. 31, pp. 189–202, 2000.

[18] S. I. Biagiola, O. E. Agamennoni, and J. L. Figueroa, "H_∞ control of a Wiener type system," *Int. J. Control*, vol. 77, no. 6, pp. 572–583, 2004.

[19] S. A. Billings, "Identification of nonlinear systems – A survey," in *Proc. Inst. Elect. Eng.*, 1980, vol. 127, pp. 272–285.

[20] S. A. Billings and S. Y. Fakhouri, "Identification systems containing linear dynamics and static nonlinear elements," *Automatica*, vol. 18, pp. 15–26, 1982.

[21] S. Boyd and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Trans. Circuits Syst.*, vol. 32, no. 11, pp. 1150–1161, Nov. 1985.

[22] T. Wigren, "Recursive prediction error identification using the nonlinear Wiener model," *Automatica*, vol. 29, pp. 1011–1025, Apr. 1993.

[23] B. Wahlberg, "System identification using Laguerre models," *IEEE Trans. Autom. Control*, vol. 36, pp. 551–562, May 1991.

[24] ——, "System identification using Kautz models," *IEEE Trans. Autom. Contr.*, vol. 39, pp. 1276–1281, Jun. 1994.

[25] A. Hagenblad, "Aspects of the identification of Wiener models," M.Sc. thesis, Elect. Eng. Dept., Linköping Univ., Linköping, Sweden, 1999.

[26] J. L. Figueroa, J. E. Cousseau, and R. J. P. de Figueiredo, "A piecewise linear dynamical functional artificial neural network (PWL-DFANN) for nonlinear adaptive time series prediction," in *Proc. IEEE Symp. Circ. Syst. (ISCAS)*, 2002, pp. 29–32.

[27] J. Figueroa, J. Cousseau, and R. J. P. de Figueiredo, "A low complexity simplicial canonical piece-wise linear adaptive filter," *Circ., Syst., Signal Process. J.*, vol. 23, pp. 365–386, Oct. 2004.

[28] C. T. Cheng and W. D. Chang, "A feedforward neural network with function shape autotuning," *Neural Netw.*, vol. 9, pp. 627–641, 1996.

[29] N. J. Bershad, P. Celka, and J.-M. Vesin, "Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output," *IEEE Trans. Signal Process.*, vol. 47, no. 3, pp. 675–689, Mar. 1999.

[30] P. Celka, N. J. Bershad, and J.-M. Vesin, "Fluctuation analysis of stochastic gradient identification of polynomial Wiener systems," *IEEE Trans. Signal Process.*, vol. 48, no. 6, pp. 1820–1825, Jun. 2000.

[31] N. J. Bershad, P. Celka, and J.-M. Vesin, "Analysis of stochastic gradient tracking of time-varying polynomial Wiener systems," *IEEE Trans. Signal Process.*, vol. 48, no. 6, pp. 1676–1686, Jun. 2000.

[32] J. J. Shynk and N. J. Bershad, "Steady-state analysis of a single layer perceptron based on a system identification model with bias terms," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 1030–1042, Sep. 1991.

[33] N. J. Bershad, J. J. Shynk, and P. L. Feintuch, "Statistical analysis of a single-layer backpropagation algorithm, Part I: Mean weight behavior," *IEEE Trans. Signal Process.*, vol. 41, pp. 573–591, Feb. 1993.

[34] N. J. Bershad, M. Ibnkahla, and F. Castanie, "Statistical analysis of a two-layer backpropagation algorithm used for modeling memoryless channels: The single neuron case," *IEEE Trans. Signal Process.*, vol. 45, pp. 747–756, Mar. 1997.

[35] P. Celka, N. J. Bershad, and J.-M. Vesin, "Analysis of stochastic gradient identification of polynomial nonlinear systems with memory," in *Proc. 5th Int. Symp. Signal Process. Applic.*, Brisbane, Australia, Aug. 1999, pp. 1001–1004.

[36] 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, pp. 522–526, Apr. 1999.

[37] B. Widrow, J. MC-Cool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, vol. 63, pp. 719–720, 1975.

[38] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 967–969, Apr. 1992.

[39] A. Hirose, "Continuous complex-value backpropagation learning," *Electron. Lett.*, vol. 28, pp. 1854–1855, 1992.

[40] T. Kim and T. Adali, "Approximation by fully complex MLP using elementary transcendental activation functions," in *Proc. XI IEEE Workshop Neural Networks Signal Processing*, 2001, pp. 203–212.

[41] A. I. Hanna and D. P. Mandic, "A fully adaptive normalized nonlinear gradient descent algorithm for complex-valued nonlinear adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2540–2549, Sep. 2003.

[42] P. Julian, A. Desages, and B. D'Amico, "Orthonormal high-level canonical PWL functions with applications to model reduction," *IEEE Trans. Circuits Syst. I*, vol. 47, pp. 702–712, May 2000.

[43] T. Koh and E. J. Powers, "An adaptive nonlinear digital filter with lattice orthogonalization," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1983, pp. 37–40.

[44] J. N. Lin and R. Unbehauen, "Adaptive nonlinear digital filter with canonical piecewise-linear structure," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 347–353, 1990.

[45] F. L. Luo and R. Unbehauen, *Applied Neural Networks for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[46] L. O. Chua, "Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks," *IEEE Trans. Circuit Theory*, vol. 18, pp. 73–85, Jan. 1971.

[47] L. O. Chua and S. Kang, "Section wise piecewise-linear functions: Canonical representation, properties and applications," *Proc. IEEE*, vol. 65, pp. 915–929, Jun. 1977.

[48] L. O. Chua and A. C. Deng, "Canonical piecewise-linear representation," *IEEE Trans. Circuits Syst. I*, vol. 35, pp. 511–525, Jan. 1988.

[49] J. Katzenelson, "An algorithm for solving nonlinear networks," *Bell Syst. Tech. J.*, vol. 44, pp. 1605–1620, 1965.

[50] C. Kahlert and L. O. Chua, "A generalized canonical piecewise-linear representation," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 373–383, Mar. 1990.

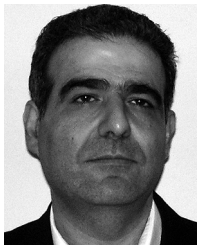
[51] J. N. Lin, H. Xu, and R. Unbehauen, "A generalization of canonical piecewise-linear functions," *IEEE Trans. Circuits Syst.*, vol. 41, pp. 345–347, Apr. 1994.

[52] M. Chien and E. Kuth, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Trans. Circuits Syst.*, vol. 24, pp. 305–317, Jun. 1997.
[53] P. Julian, M. Jordan, and A. Desages, "Canonical piecewise linear approximation of smooth functions," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 567–571, May 1996.
[54] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[55] L. Breiman, "Hinging hyperplanes for regression, classification and function approximation," *IEEE Trans. Inf. Theory*, vol. 39, pp. 999–1012, Mar. 1993.
[56] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.
[57] H. Fan and Y. Yang, "Convergence of a frequency-domain adaptive IIR filtering algorithm," in *Proc. Midwest Symp. Circuits Systems*, 1990, pp. 1134–1138.
[58] T. Wigren, "Convergence analysis of recursive identification algorithms based on the nonlinear Wiener model," *IEEE Trans. Autom. Control*, vol. 38, pp. 2191–2206, Nov. 1994.
[59] Z. Fejzo and H. Lev-Ari, "Adaptive Laguerre-lattice filters," *IEEE Trans. Signal Process.*, vol. 45, no. 12, pp. 3006–3016, Dec. 1997.
[60] T. Kim and T. Adali, "Complex backpropagation neural network using elementary transcendental activation functions," in *Proc. IEEE Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, 2001, pp. 1281–1284.
[61] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Sel. Areas Commun.*, vol. 1, pp. 57–62, Jan. 1983.
[62] H. W. Kang, Y. S. Chi, and D. Youn, "Adaptive precompensation of Wiener systems," *IEEE Trans. Signal Process.*, vol. 46, pp. 2825–2829, Oct. 1998.
[63] R. Raich, H. Quian, 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.

Juan E. Cousseau (S'90–M'92–SM'00) 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 has been involved in scientific and industry projects with research groups from Argentina, Brazil, Spain, and the United States. 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 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 is a participant of the IEEE Signal Processing Society Distinguished Lecturer Program 2006.



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, 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. Between 2002 and 2004, he served as Chair of the Standing Committee on Chapter Activities of the IEEE Control Systems Society.

Stefan Werner (M'03) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998 and the D.Sc. (E.E.) degree (with honors) from the Signal Processing Laboratory, Smart and Novel Radios (SMARAD) Center of Excellence, Helsinki University of Technology (HUT), Espoo, Finland, in 2002. He is currently a Senior Researcher with the Signal Processing Laboratory, HUT. His research interests are in multiuser communications and adaptive filtering.

Timo I. Laakso (SM'95) was born in Vantaa, Finland, on February 1, 1961. He received the Master's of Science in Technology, the Licentiate of Science in Technology, and the Doctor of Science in Technology degrees, all in electrical engineering, from the Department of Electrical Engineering at the Helsinki University of Technology, Finland, in 1987, 1990, and 1991, respectively.

After graduation, from 1992 to 1994, he was with the Nokia Research Center, Helsinki, Finland, working on the research of third-generation mobile communication systems. From 1994 to 1996, he was Lecturer at the University of Westminster, London, U.K. He is currently a Professor of signal processing in telecommunications at the Helsinki University of Technology. He has about 130 journal and conference publications, and he holds 16 patents.