

Novel Adaptive Nonlinear Predistorters Based on the Direct Learning Algorithm

Dayong Zhou, *Member, IEEE*, and Victor E. DeBrunner, *Senior Member, IEEE*

Abstract—The adaptive nonlinear predistorter is an effective technique to compensate for the nonlinear distortion existing in digital communication and control systems. However, available adaptive nonlinear predistorters using indirect learning are sensitive to measurement noise and do not perform optimally. Other available types are either slow to converge, complicated in structure and computationally expensive, or do not consider the memory effects in nonlinear systems such as a high power amplifier (HPA). In this paper, we first propose several novel adaptive nonlinear predistorters based on direct learning algorithms—the nonlinear filtered-x RLS (NFXRLS) algorithm, the nonlinear adjoint LMS (NALMS) algorithm, and the nonlinear adjoint RLS (NARLS) algorithm. Using these new learning algorithms, we design adaptive nonlinear predistorters for an HPA with memory effects or for an HPA following a linear system. Because of the direct learning algorithm, these novel adaptive predistorters outperform nonlinear predistorters that are based on the indirect learning method in the sense of normalized mean square error (NMSE), bit error rate (BER), and spectral regrowth. Moreover, our developed adaptive nonlinear predistorters are computationally efficient and/or converge rapidly when compared to other adaptive nonlinear predistorters that use direct learning, and furthermore can be easily implemented. We further simplify our proposed algorithms by exploring the robustness of our proposed algorithm as well as by examining the statistical properties of what we call the “instantaneous equivalent linear” (IEL) filter. Simulation results confirm the effectiveness of our proposed algorithms.

Index Terms—Adaptive filters, adaptive nonlinear filter, communication system nonlinearities, nonlinear distortion, power amplifiers, predistortion, spectral regrowth.

I. INTRODUCTION

THE use of nonlinear devices in signal transmission for digital communication systems such as code-division multiple access (CDMA) and frequency-division multiple access (FDMA) is well-known. One example is high power amplifiers (HPAs) [1]–[4]—in order to increase the efficiency of the HPAs, these amplifiers are often driven into their nonlinear region. Another example can be found for laser diodes used in optical

transmission fiber links, as applied in CDMA and FDMA to connect base stations to control stations [5].

Some researchers study the effects of these nonlinearities in digital communication systems, e.g., see [1]–[4]. These results show that the nonlinearity warps the signal constellation, spreads the frequency range, and distorts the pulse shape. All of these effects generate intersymbol interference (ISI) and interchannel interference (ICI), both of which increase the bit error rate (BER). These effects become more prominent as the communication speed increases. As a result, we must compensate for the nonlinearity if we want to improve the performance of these digital communication systems.

Currently, there are two major nonlinear compensation techniques, the predistorter and the nonlinear equalizer (the postdistorter), which put a nonlinear inverse function either on the transmitter side or on the receiver side, respectively. There are several advantages of the predistorter over the nonlinear equalizer. Although both of them can reduce the ISI, the predistorter can effectively suppress the out-of-band emission, which is the main cause of ICI and is strictly limited by regulatory agencies. Moreover, the absence of noise makes the predistorter perform better than the postdistorter for a given complexity [4], [6]. Depending on whether a predistorter modifies the transmitted data constellation or the transmitted signal shape, the predistorter techniques can be grouped into two main classes: the data predistorter and the signal predistorter [6]–[8]. The configurations of a typical data predistorter and a typical signal predistorter for a HPA are shown in Fig. 1(a) and (b), respectively. Usually, a data predistorter only samples at symbol rate, however, the fractional spaced data predistorter has been proposed to improve the predistorter performance [6], [8]. The sample rate of a signal predistorter depends on both the bandwidth of the signals to be processed and generated and the power consumption of the digital processor [6].

In this paper, we will introduce several new predistorters which could work for both the data and the signal predistorter configurations to compensate the nonlinear distortion introduced by HPAs. Many research results have been proposed to compensate the nonlinear distortion caused by HPAs. Some of these techniques are not adaptive, such as the p th-order inverse techniques [9], [10], which can only remove the first p orders of nonlinear distortion. Usually, an adaptive predistorter is required for an HPA because of variations in the HPA due to changes over time and temperature, or when the input signal power changes. Other previously proposed adaptive predistorters (e.g. [11] and [12]) for HPAs did not consider the existence of memory effects in the HPA. However, recent research results have indicated that the memory effects in

Manuscript received November 24, 2004; revised January 24, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ioan Tabus.

D. Zhou is with the School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK 73069 USA (e-mail: dayong@ou.edu).

W. E. DeBrunner was with the School of Electrical and Computer Engineering, University of Oklahoma, Norman, OK 73069 USA. He is now with Florida State University, Tallahassee, FL 32306 USA (e-mail: victor.debrunner@eng.fsu.edu).

Color versions of Figs. 12–15 are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2006.882058

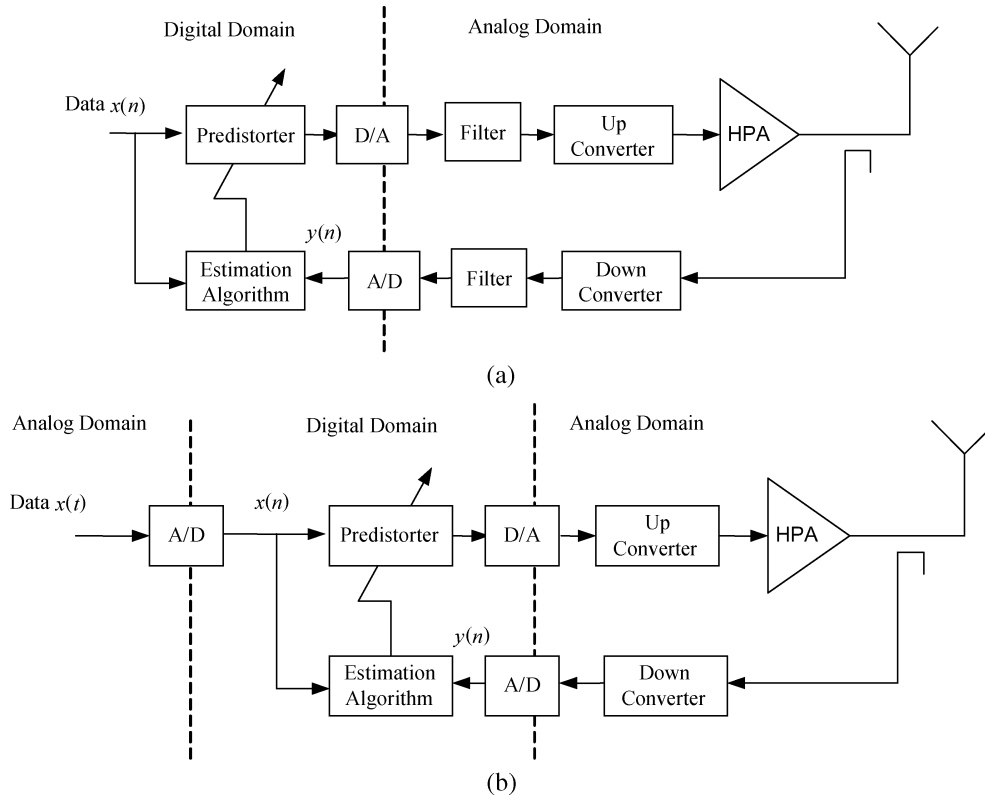


Fig. 1. Typical configurations of (a) data predistorter and (b) signal predistorter.

an HPA could seriously affect the performance of a wireless communication system [13]–[15]. Unfortunately, the updating techniques for a memoryless predistorter [11], [12] cannot be directly extended to update a nonlinear predistorter for an HPA with memory. Most available adaptive predistorters for HPAs with memory effects are based on the indirect learning method [16], [17], whose configuration is shown in Fig. 2(a). This system structure was first proposed by Gao *et al.* [18] to remove the nonlinearity in loudspeakers. The method is called the indirect learning method because the post-inverse filter coefficients are first identified, and are then copied to work as a predistorter. There are two main drawbacks that affect the performance of these indirect learning methods [19]. First, the measurement of $y(n)$ in Fig. 2(a) will be noisy, making the adaptive filter converge to biased values. Second, theoretically speaking, the nonlinear filters cannot be permuted. In essence, placing a copy of the post-inverse filter in front of the nonlinear device cannot guarantee good performance in removing the nonlinear distortion—the order of the cascade affects performance.

Several researchers have noticed these drawbacks and have recently proposed techniques to improve the performance of these indirect methods [20], [21]. Morgan *et al.* [20] tried to solve the first drawback of indirect learning—the noisy measurement problem—by changing the configuration of these methods. While their proposed solution yields adequate performance, it greatly increases the computational complexity. Furthermore, its good performance significantly depends on the quality of the nonlinear system identification. Moreover, that method does not address the second problem of cascade

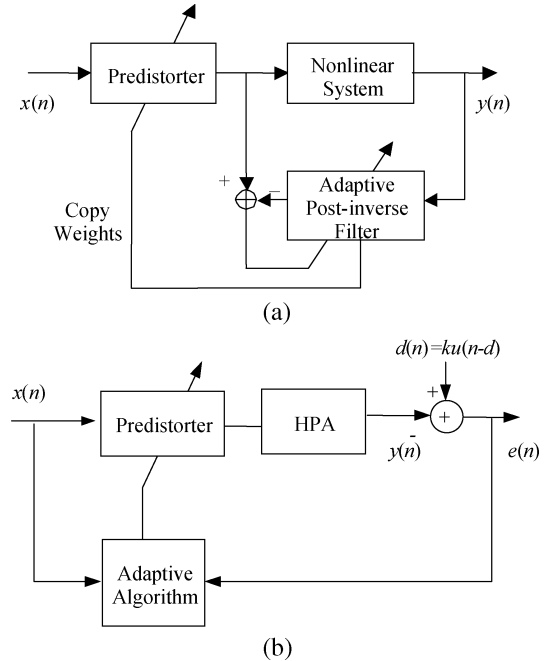


Fig. 2. Adaptive predistorters based on (a) indirect learning [17] and (b) direct learning.

order. Qian *et al.* [21] introduced a new nonlinear predistortion method that updates the predistorter coefficients directly and successfully addresses this second concern. However, the method increases the computational complexity by a very

significant factor. Maybe even more importantly, any errors incurred while estimating the nonlinear post-inverse filter ($\hat{P}$ in [21]) will cause the overall system performance to deteriorate. An adaptive predistorter based on a direct learning architecture will remedy both of these problems. In direct learning, the coefficients of the predistorter are updated directly based on its input and reference error. In Fig. 2(b), a block diagram of the predistorter based on direct learning is given; our new algorithms studied in this paper are based on this structure, where it is tacitly assumed that a model of the nonlinear system has already been obtained using standard estimation techniques. Currently, there are several available adaptive predistorters for HPAs with memory and based on the direct learning methods [6], [22], [23], which are slow to convergence, computationally expensive, and complicated in structure. This is true even for the method proposed in [22], where they use a special nonlinear structure (the Wiener nonlinear model) in an attempt to reduce the computational complexity.

In this paper, we first introduce a new concept—the instantaneous equivalent linear (IEL) filter—to explain the “nonlinear filtered-x LMS algorithm” given in [23]. We also use the IEL filter to simplify the structure of the nonlinear filtered-x LMS (NFXLMS) algorithm. Then, using this new concept and the simple structure resulting from its employment, we introduce several novel adaptive nonlinear predistorters based on direct learning algorithms: The nonlinear filtered-x RLS (NFXRLS), the nonlinear adjoint LMS (NALMS), and the nonlinear adjoint RLS (NARLS) algorithms. These novel adaptive nonlinear predistorters are easily implemented by computer hardware and software. The NFXRLS algorithm greatly increases the adaptive filter convergence speed as compared to that of the NFXLMS algorithm. The NALMS and NARLS algorithms are introduced to substantially reduce the computational burden and memory requirements for the NFXLMS and the NFXRLS algorithms, respectively. Then we apply the developed techniques to design an adaptive Hammerstein nonlinear predistorter for a Wiener model HPA system. By resorting to a block-oriented nonlinear model, the computational complexity of our proposed adaptive nonlinear predistorter is greatly reduced. The computational complexity and implementation cost can be further reduced by exploring the robustness of the proposed algorithms along with the statistical properties of the IEL filter for a Wiener nonlinear system and a “weakly” nonlinear system under certain conditions.

We have organized this paper as follows. A general polynomial nonlinear model is introduced in Section II-A and we extend the “nonlinear filtered-x LMS” method [23] to this general polynomial nonlinear model in Section II-B. With the introduction of our IEL filter concept, we unify this nonlinear filtered-x LMS algorithm and the linear filtered-x LMS algorithm, and represent this algorithm using a simple structure in Section II-C. Using our IEL filter and a weak nonlinearity assumption, we develop the quickly convergent NFXRLS algorithm in Section II-D. In Section III, the computationally efficient NALMS and NARLS algorithms are developed. Then, in Section IV, we apply the developed algorithm to an HPA and design the adaptive Hammerstein nonlinear predistorter. We discuss further reductions in computation and the corresponding

reduction in implementation costs by exploring the robustness of our proposed algorithms in Section V. The computational complexity analysis presented in Section VI and the simulation results given in Section VII show the effectiveness of our novel predistorters. Finally, we state conclusions and discuss some future work in Section VIII.

In our notation, the superscripts $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote, respectively, the conjugate, the transpose, and the conjugate transpose.

II. NONLINEAR POLYNOMIAL PREDISTORTER BASED ON THE DIRECT LEARNING ALGORITHM

Although the (direct learning algorithm) nonlinear filtered-x LMS introduced in [23] used the most general Volterra nonlinear model, that algorithm is directly extendible to other polynomial models and simple specific cases, such as the recently developed memory polynomial model [15], [16] and the orthogonal nonlinear polynomial model [24]. In this section, we define a nonlinear polynomial model that is general enough to incorporate each of these specific cases along with other nonlinear models, and could incorporate the nonlinear filtered-x LMS algorithm. Then we introduce the instantaneous equivalent linear (IEL) filter concept. Using this, we give a novel explanation of the nonlinear filtered-x LMS algorithm, and use the weak nonlinearity assumption to derive a new nonlinear filtered-x RLS algorithm that converges quickly.

A. General Nonlinear Polynomial Model

The general nonlinear polynomial model representation is given as

$$y(n) = \sum_{i=1}^Q w_i(n) \phi_i[\bar{x}(n)] \quad (1)$$

where $\bar{x}(n)$ represents the input sequence to a nonlinear device [i.e., $x(n), x(n-1), \dots$], $y(n)$ is the output, $w_i(n)$ is the i th complex coefficient at time n , Q is the total number of terms, and $\phi_i[\bar{x}(n)]$ is a function of $\bar{x}(n)$, being differentiable with respect to $\bar{x}(n)$ everywhere except possibly at a few points. Note that this general polynomial nonlinear model is linear with respect to the filter coefficients $w_i(n)$.

Using this general polynomial model to represent the Volterra series nonlinear model as in [17] yields

$$\phi_i[\bar{x}(n)] = \underbrace{x(n-i)x(n-k)\cdots x^*(n-d)}_{P \text{ terms}}. \quad (2)$$

Here, $i, k, \dots, d$ are arbitrary integers representing delay. The maximum delay determines the memory size, and p represents the order of the nonlinearity. The conjugate sign is optional and could be at any term or terms, which depends on the specific model that users choose. Since the Wiener nonlinear model and the Hammerstein nonlinear model in [22] can be regarded as special cases of the Volterra model, they can also be represented by the general polynomial nonlinear model. The recently developed memory polynomial model [15], [16] and the orthogonal polynomial model [24] for high power amplifiers (HPAs)

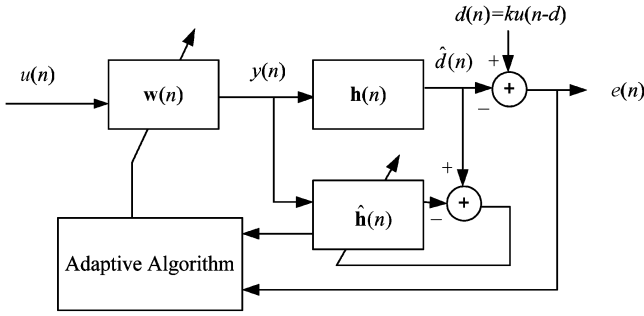


Fig. 3. Schematic diagram for compensation of nonlinear distortion using the nonlinear filtered-x LMS algorithm.

can also be expressed using this general polynomial nonlinear model with $\phi_i[\bar{x}(n)]$ being, respectively

$$\phi_i[\bar{x}(n)] = |x(n-q)|^{K-1} x(n-q) \quad (3a)$$

or

$$\phi_i[\bar{x}(n)] = \sum_{l=1}^K \frac{(-1)^{l+K} (K+l)!}{(l-1)!(l+1)!(K-l)!} |x(n-i)|^{l-1} x(n-i). \quad (3b)$$

In this, K represents the order of the nonlinearity and q is a non-negative integer representing the delay. However, the two $\phi_i[\bar{x}(n)]$ given in (3) are not differentiable with respect to $x(n)$, which conflicts with the definition of our general nonlinear polynomial. To alleviate this problem, we write $|x(n-q)|^2$ as $x(n-q)x^*(n-q)$ and treat $x(n-q)$ and $x^*(n-q)$ as independent variables as in [25]. We can represent (1) in vector form

$$y(n) = \mathbf{x}^T(n) \mathbf{w}(n) \quad (4)$$

where

$$\mathbf{x}(n) = [\phi_1[\bar{x}(n)] \quad \phi_2[\bar{x}(n)] \quad \cdots \quad \phi_Q[\bar{x}(n)]]^T \quad (5)$$

$$\mathbf{w}(n) = [w_1(n) \quad w_2(n) \quad \cdots \quad w_Q(n)]^T. \quad (6)$$

B. The Nonlinear Filtered-x LMS Algorithm

The block diagram for the direct learning method, called the *nonlinear filtered-x LMS algorithm* in [23], is shown in Fig. 3. We derive their algorithm using our more general nonlinear polynomial model.

In Fig. 3, $\mathbf{w}(n)$ and $\hat{\mathbf{h}}(n)$ are adaptive nonlinear polynomial filters whose input and output relationships can be expressed by the general nonlinear polynomial as discussed in Section II-A. We use $\mathbf{h}(n)$ to represent a nonlinear device. The filter $\hat{\mathbf{h}}(n)$ is an adaptive polynomial filter that is used to identify and model the nonlinear device. The adaptive identification method is not a relevant issue in this paper, and a wide variety of methods can be found in the literature such as in [19]. Consequently, we do not cover this issue here. The adaptive predistorter $\mathbf{w}(n)$ is a nonlinear filter based on the general polynomial model. Its

output $y(n)$ is fed into the nonlinear device $\mathbf{h}(n)$. We want the output $\hat{d}(n)$ of the nonlinear device to be as close as possible to the desired signal $d(n)$, which is a delayed and amplified version of the overall input $u(n)$.

Similar to the equations in Section II-A, we define the vectors

$$\mathbf{u}(n) = [\phi_1[\bar{u}(n)] \quad \phi_2[\bar{u}(n)] \quad \cdots \quad \phi_Q[\bar{u}(n)]]^T$$

$$\mathbf{w}(n) = [w_1(n) \quad w_2(n) \quad \cdots \quad w_Q(n)]^T$$

$$\mathbf{y}(n) = [\phi_1[\bar{y}(n)] \quad \phi_2[\bar{y}(n)] \quad \cdots \quad \phi_R[\bar{y}(n)]]^T$$

$$\hat{\mathbf{h}}(n) = [\hat{h}_1(n) \quad \hat{h}_2(n) \quad \cdots \quad \hat{h}_R(n)]^T$$

where Q and R are the total number of terms in the nonlinear polynomials that express the input and output relationships of $\mathbf{w}(n)$ and $\hat{\mathbf{h}}(n)$. So, we have

$$y(n) = \mathbf{u}^T(n) \mathbf{w}(n) \quad (7)$$

$$\hat{d}(n) = \mathbf{y}^T(n) \hat{\mathbf{h}}(n) \quad (8)$$

with error

$$e(n) = d(n) - \hat{d}(n). \quad (9)$$

The objective function is the mean-square error (MSE) defined as

$$J(\mathbf{w}) = E\{|e(n)|^2\} = E\{|d(n) - \hat{d}(n)|^2\}. \quad (10)$$

Like the traditional LMS algorithm, we can use a one-point sample mean to replace (10)

$$J(\mathbf{w}) = |d(n) - \hat{d}(n)|^2. \quad (11)$$

The weight update equation based on the steepest descent method is then

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2} \boldsymbol{\mu} \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{w}(n) + \boldsymbol{\mu} e^*(n) \frac{\partial \hat{d}(n)}{\partial \mathbf{w}(n)} \quad (12)$$

where $\boldsymbol{\mu}$ is the step size that controls the stability and convergence speed of the adaptive nonlinear filter. The step size $\boldsymbol{\mu}$ is a scalar if all coefficients are updated uniformly; otherwise, it is a matrix. In this paper, the step size takes the same value for all coefficients of the same nonlinear order, and so $\boldsymbol{\mu}$ is a diagonal matrix with P different diagonal elements. Now

$$\frac{\partial \hat{d}(n)}{\partial \mathbf{w}(n)} = \sum_{r=0}^{M-1} \frac{\partial \hat{d}(n)}{\partial y(n-r)} \frac{\partial y(n-r)}{\partial \mathbf{w}(n)} \quad (13)$$

where M is the memory size of the nonlinear device $\mathbf{h}(n)$. From (7), we have

$$y(n-r) = \mathbf{u}^T(n-r) \mathbf{w}(n-r). \quad (14)$$

We assume $\mathbf{w}(n) \approx \mathbf{w}(n-r)$ as in [23], so that

$$\frac{\partial y(n-r)}{\partial \mathbf{w}(n)} \approx \mathbf{u}(n-r). \quad (15)$$

If we define

$$g(r, n) = \frac{\partial \hat{d}(n)}{\partial y(n-r)} \quad (16)$$

we can combine (16) with the general polynomial model defined in (1), yielding

$$\begin{aligned} g(r, n) &= \frac{\partial \hat{d}(n)}{\partial y(n-r)} \\ &= \frac{\partial \sum_{i=1}^Q w_i \phi_i[\bar{y}(n)]}{\partial y(n-r)} \\ &= \sum_{i=1}^Q w_i \frac{\partial \phi_i[\bar{y}(n)]}{\partial y(n-r)}. \end{aligned} \quad (17)$$

Combining (12), (15), and (17) yields the new update equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{e}^*(n) \sum_{r=0}^{M-1} g(r, n) \mathbf{u}(n-r). \quad (18)$$

Note that in our derivation, we have assumed that the nonlinear polynomial model is both linear with respect to individual coefficients and differentiable with regard to its input. The definition of our general nonlinear polynomial model in (1) meets these two assumptions. Equations (17) and (18) form the direct learning algorithm for the adaptive nonlinear polynomial predistorter. Because this update method directly finds the preinverse filter and not the post-inverse filter as in the indirect learning method in Fig. 2(a), we expect the method to perform better for the same nonlinear model. These improvements are, as indicated in the conclusions of [23], achieved at the cost of a significant increase in computational complexity and a more complicated implementation structure.

C. The Instantaneous Equivalent Linear (IEL) Filter

We notice that in (18), $\sum_{r=0}^{M-1} g(r, n) \mathbf{u}(n-r)$ is a filtered version of $\mathbf{u}(n)$. This filter is a linear time-varying filter whose coefficient vector $\tilde{\mathbf{h}}(n)$ can be obtained from (17)

$$\tilde{\mathbf{h}}(n) = [g(0, n) \quad g(1, n) \quad \cdots \quad g(M-1, n)]^T. \quad (19)$$

We call this filter the “instantaneous equivalent linear (IEL) filter” $\tilde{\mathbf{h}}(n)$ because, for a weak nonlinearity, the output of the nonlinear system $\hat{d}(n)$ at time n can be approximated by

$$\hat{d}(n) \approx [y(n) \quad y(n-1) \quad \cdots \quad y(n-M+1)] \tilde{\mathbf{h}}(n). \quad (20)$$

This approximation will be used next to develop the RLS based nonlinear predistorter. Based on this IEL filter concept, we define the filtered reference signal as

$$\mathbf{u}_f(n) = \sum_{r=0}^{M-1} g(r, n) \mathbf{u}(n-r) = \mathbf{u}(n) * \tilde{\mathbf{h}}^T(n). \quad (21)$$

We have the adaptive filter update function

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{e}^*(n) \mathbf{u}_f(n). \quad (22)$$

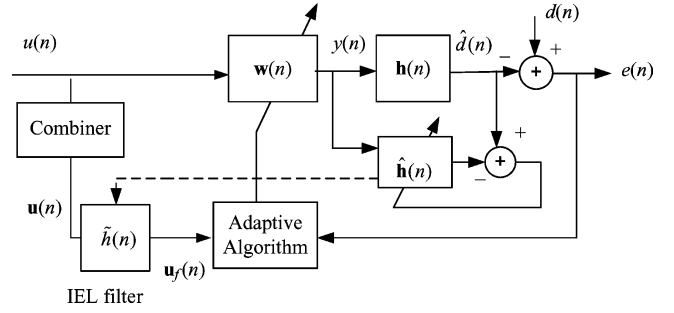


Fig. 4. Adaptive nonlinear predistorter based on the nonlinear filtered-x LMS.

Using (19)–(22), we derive a new representation of the NFXLMS algorithm whose block diagram representation is given in Fig. 4. The combiner generates the nonlinear elements in vector $\mathbf{u}(n)$. This representation is simple and easy to implement with DSP hardware. Furthermore, if $\mathbf{h}(n)$ and $\mathbf{w}(n)$ are linear filters, then $\tilde{\mathbf{h}}(n)$ will be equal to $\mathbf{h}(n)$ and so our algorithm is reduced to the filtered-x LMS algorithm, which is widely used in adaptive feed-forward control [26]–[28]. With this view, it is apparent that our algorithm is a generalization of the linear filtered-x LMS. Thus, available analysis for the linear filtered-x LMS algorithm can be extended to the nonlinear case, as we do in Section V. Note, however, that the computational complexity is not affected by this new structure.

D. Nonlinear Filtered-x Recursive-Least-Square (NFXRLS) Algorithm

It is well known that the nonlinear LMS based adaptive algorithm can produce slow convergence and poor performance [19], especially if the nonlinear terms in $\mathbf{u}(n)$ are correlated. It is also well known that the RLS-based algorithm can greatly increase the convergence speed and reduce the power of the residual error. The linear filtered-x RLS algorithm was discussed in [27]. However, due to the nonlinear effect, its extension from the linear case to the nonlinear case is not straightforward. Nevertheless, we circumvent this difficulty by the weak nonlinearity assumption and the IEL filter concept. The weak nonlinearity assumption is used only in the development of NFXRLS. We will prove that NFXRLS will also works for a system with relatively heavy nonlinear distortion.

We first define our cost function as the exponentially weighted error (the same as in the linear case)

$$\xi(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (23)$$

where $0 < \lambda < 1$ is an exponential forgetting factor. To derive the nonlinear RLS algorithm we need to solve $(\partial \xi(n) / \partial \mathbf{w}(n)) = 0$ (see [25] and [27]), i.e.,

$$\frac{\partial \xi(n)}{\partial \mathbf{w}(n)} = - \sum_{i=0}^n 2\lambda^{n-i} \mathbf{u}_f(i) e^*(i). \quad (24)$$

Representing $e(i)$ using the IEL filter concept and the weak nonlinearity assumption,

$$e(i) \approx d(i) - [y(i) \ y(i-1) \ \cdots \ y(i-M+1)] \tilde{\mathbf{h}}(i). \quad (25)$$

Using the fact that $y(i-r) = \mathbf{w}^T(i-r)\mathbf{u}(i-r)$ combined with the assumption that the weight vector $\mathbf{w}(n)$ is constant during the time interval $(1, n+M)$, we have

$$\begin{aligned} e(i) &\approx d(i) - \mathbf{w}^T(n) [\mathbf{u}(i) \ \mathbf{u}(i-1) \ \cdots \ \mathbf{u}(i-M+1)] \tilde{\mathbf{h}}(i) \\ &\approx d(i) - \mathbf{w}^T(n) \mathbf{u}_f(i) \end{aligned} \quad (26)$$

for $1 \leq i \leq n$. Combining (24) and (26), we get

$$\begin{aligned} \frac{\partial \xi(n)}{\partial \mathbf{w}(n)} &\approx - \sum_{i=0}^n 2\lambda^{n-i} \mathbf{u}_f(i) [d(i) - \mathbf{w}^T(n) \mathbf{u}_f(i)]^* \\ &= - \sum_{i=0}^n 2\lambda^{n-i} \mathbf{u}_f(i) d^*(i) \\ &\quad + \sum_{i=0}^n 2\lambda^{n-i} \mathbf{u}_f(i) \mathbf{u}_f^H(i) \mathbf{w}^*(n). \end{aligned} \quad (27)$$

Setting the right-hand side of (27) equal to zero yields

$$\mathbf{R}_{\mathbf{u}_f}(n) \mathbf{w}(n) = \mathbf{r}_{d\mathbf{u}_f}(n) \quad (28)$$

where

$$\mathbf{R}_{\mathbf{u}_f}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_f(i) \mathbf{u}_f^H(i) \quad (29)$$

and

$$\mathbf{r}_{d\mathbf{u}_f}(n) = \sum_{i=1}^n \lambda^{n-i} d(i) \mathbf{u}_f^*(i). \quad (30)$$

We can obtain the Kalman gain as in [22]

$$\mathbf{k}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{u}_f(n)}{1 + \lambda^{-1} \mathbf{u}_f^H(n) \mathbf{P}(n-1) \mathbf{u}_f(n)} \quad (31)$$

where we have used

$$\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}_f^H(n) \mathbf{P}(n-1). \quad (32)$$

The resulting update is then, simply

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{k}(n) e^*(n). \quad (33)$$

Here, the step size μ is a scalar and takes a value less than 1, depending on the length of delays and the degree of the nonlinear distortion. For a large delay length and severe nonlinear distortion, μ should take relatively small value to guarantee the stability of the adaptive algorithm. We use (31)–(33) to derive

our proposed NFXRLS algorithm, which has the same block diagram as NFXLMS shown in Fig. 4.

Although our development of this algorithm assumes the weak nonlinearity condition, the NFXRLS algorithm also works for relatively large nonlinearities. If we combine (31) and (33), the NFXRLS algorithm can be regarded as the NFXLMS algorithm with a variable step size (VSS) matrix; and the assumption of weak nonlinearity only affects the estimation of this VSS matrix. As a result, if the nonlinear distortion is considerable, we can reduce the step size μ in (33) so that the nonlinear filtered-x RLS algorithm will work, albeit with a small sacrifice in the convergence speed (though the convergence will still be much faster than that of NFXLMS).

III. ADJOINT POLYNOMIAL LMS AND RLS ALGORITHM

As we discussed earlier, the better performance obtained from the nonlinear predistorter based on direct learning (over that of indirect learning) is gained at the cost of significantly increasing the computational burden and memory requirements. There are three main factors that contribute to this cost. The first is the procedure to identify $\hat{\mathbf{h}}(n)$, which is not a concern of this paper. Many such methods exist for this identification, none of which are affected by the algorithms in this paper. The second is the filtering, i.e., calculation of the output of the predistorter $y(n) = \mathbf{u}^T(n) \mathbf{w}(n)$. The computational complexity and memory requirements for this calculation are a direct consequence of the nonlinear model. This can be reduced by resorting to the block-oriented nonlinear model used in [22] and Section IV of this paper. The third cause is the weight update, which includes the calculation of the IEL filter $\tilde{\mathbf{h}}(n)$, the filtering of the reference signal $\mathbf{u}_f(n)$, and the actual calculation of the new weights. In the update, each term in $\mathbf{u}(n)$ must be filtered by $\tilde{\mathbf{h}}(n)$ in order to calculate $\mathbf{u}_f(n)$. Because the IEL filter $\tilde{\mathbf{h}}(n)$ is time-varying, we cannot take advantage of the delay relation in $\mathbf{u}(n)$. As a result, any increase in the order or memory size of the nonlinearity results in a correspondingly large increase in the number of terms in $\mathbf{u}(n)$. So, we find that the computational complexity and memory requirements of the adaptive algorithm can increase significantly. Our new algorithm discussed in this section is aimed at reducing the implementation cost of the direct learning algorithm by simplifying the weight update.

To significantly reduce the computational complexity, we propose a novel update method for the adaptive nonlinear predistorter. Rewrite (18) as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \Delta \mathbf{w} \quad (34)$$

with

$$\Delta \mathbf{w} = e^*(n) \sum_{r=0}^{M-1} \tilde{\mathbf{h}}(r, n) \mathbf{u}(n-r). \quad (35)$$

where $\tilde{\mathbf{h}}(r, n)$ represents the $(r-1)$ th coefficient in the IEL filter $\tilde{\mathbf{h}}(n)$; similar notation will be applied to other time-varying vectors later. Now, using the change of variable $k = n-r+M-$

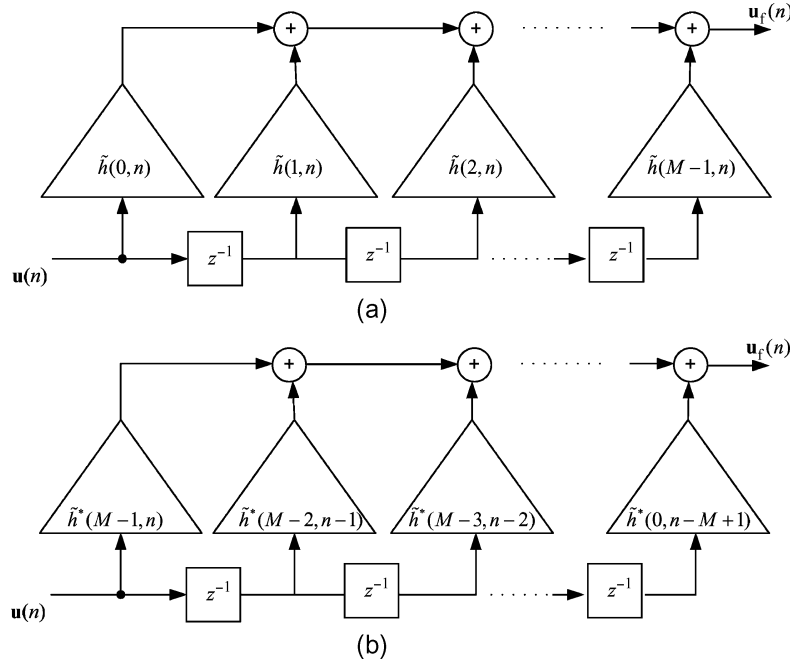


Fig. 5. (a) IEL filter $\tilde{h}(n)$. (b) Adjoint IEL filter $\tilde{h}_-(n)$.

1 (so that $n = k + r - M + 1$), we have the weight correction at each time

$$\Delta \mathbf{w} = \left[\sum_{r=0}^{M-1} e(k+r-M+1) \tilde{h}^*(r, k+r-M+1) \right]^* \times \mathbf{u}(k-M+1). \quad (36)$$

Define the filtered error

$$e_f(n) = \left[\sum_{r=0}^{M-1} e(n+r-M+1) \tilde{h}^*(r, n+r-M+1) \right]. \quad (37)$$

The physical meaning of signal $e_f(k)$ is passing the original error $e(k)$ through a filter called the adjoint IEL filter $\tilde{h}_-(n)$, with coefficient vector defined as

$$\tilde{\mathbf{h}}_-(n) = \begin{bmatrix} \tilde{h}^*(M-1, n) & \tilde{h}^*(M-2, n-1) & \tilde{h}^*(M-3, n-2) & \cdots & \tilde{h}^*(0, n-M+1) \end{bmatrix}^T. \quad (38)$$

The relationship of IEL filter $\tilde{h}(n)$ and adjoint IEL filter $\tilde{h}_-(n)$ is shown in Fig. 5. Since

$$e_f^*(n) = [e(n) * \tilde{\mathbf{h}}_-(n)]^* \quad (39)$$

we can write our novel nonlinear predistorter based on the direct learning algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_f^*(n) \mathbf{u}(n-M+1). \quad (40)$$

Thus, our novel update method for the polynomial predistorter based on the direct learning architecture is fully described by (17) and (38)–(40) (see Fig. 6 for a block diagram). When $\mathbf{h}(n)$ and $\mathbf{w}(n)$ are linear filters, $\tilde{\mathbf{h}}_-(n) = \tilde{\mathbf{h}}_-(n)$. Consequently, in

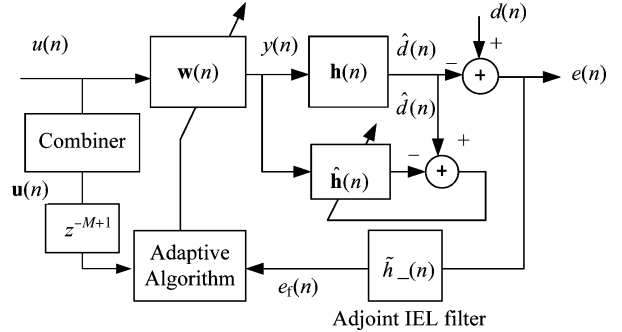


Fig. 6. Nonlinear adjoint LMS and RLS algorithm.

this case, our algorithm reduces to the adjoint filtered-x LMS algorithm discussed in [29], and so our algorithm can be regarded as a generalized version of the originally proposed linear algorithm. It is this relationship to the linear case that has prompted us to name our method the *nonlinear adjoint LMS algorithm* (NALMS). In the NALMS, one need only filter the error signal by the adjoint IEL filter instead of filtering every component of $\mathbf{u}(n)$. This modification greatly reduces the computational complexity and memory size requirements. A detailed analysis of the computational savings will be given in Section VI.

Similar to the development of the NFXRLS algorithm, we can derive the nonlinear adjoint RLS (NARLS) algorithm, which leads to the result

$$\mathbf{k}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{u}(n-M)}{1 + \lambda^{-1} \mathbf{u}^H(n-M) \mathbf{P}(n-1) \mathbf{u}(n-M)} \quad (41)$$

$$\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n-M) \mathbf{P}(n-1) \quad (42)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{k}(n) e_f^*(n). \quad (43)$$

The block diagram of NARLS is the same as that of NALMS algorithm, as shown in Fig. 6.

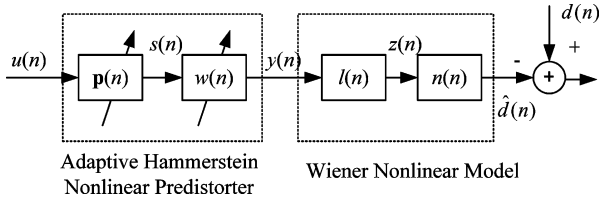


Fig. 7. Hammerstein nonlinear predistorter and Wiener nonlinear model of HPA system.

IV. ADAPTIVE HAMMERSTEIN NONLINEAR PREDISTORTER FOR HPA

Sometimes, the general polynomial nonlinear model may not be efficient for modeling a high power amplifier in digital communication systems like orthogonal frequency-division multiplexing (OFDM) and CDMA. For example, if an HPA only demonstrates memoryless nonlinear distortion like the AM/AM and AM/PM effects [19] for narrowband input signals, one would commonly use a block-oriented model to model the HPA preceded by a linear filter, such as a pulse-shaping filter, to reduce the number of coefficients and thus the computational complexity [22]. Recent research results show that HPAs exhibit some memory effects for a wideband input signal [15], [30]. These memory effects could also be absorbed into the linear filter model to reduce system complexity [30].

We use the Wiener nonlinear model to represent our HPA system in this paper. This model consists of the cascade of a N taps linear filter $l(n)$ with coefficient vector $\mathbf{l}(n)$ followed by a memoryless nonlinear system $g(n)$. Let $z(n)$ be the output of the linear filter $l(n)$ and input to the nonlinear system $n(n)$. The Wiener nonlinear model can be linearized by an adaptive Hammerstein nonlinear predistorter (see Fig. 7)—the cascade of a memoryless adaptive nonlinear filter $\mathbf{p}(n)$ and an M taps adaptive linear filter $w(n)$ [with coefficients vector $\mathbf{w}(n)$]. The method introduced in this paper can be easily extended to compensate either a Hammerstein or a linear-nonlinear-linear (LNL) block-oriented nonlinear model. From Section II, we know that in order to design the adaptive predistorter, we need to find the IEL filter and adjoint IEL filters of the path from the adaptive filter output to the summation point. We now examine this procedure.

From Fig. 7, we can see the path following adaptive filter $w(n)$ is a Wiener nonlinear model. Using (17) and (19) we find the coefficient vector $\tilde{\mathbf{v}}_1(n)$ of IEL filter $\tilde{v}_1(n)$ for the Wiener nonlinear model

$$\begin{aligned} \tilde{\mathbf{v}}_1(n) &= \left[\frac{\partial \hat{d}(n)}{\partial y(n)}, \frac{\partial \hat{d}(n)}{\partial y(n-1)}, \dots, \frac{\partial \hat{d}(n)}{\partial y(n-N+1)} \right]^T \\ &= \frac{\partial d(n)}{\partial z(n)} \left[\frac{\partial z(n)}{\partial y(n)}, \frac{\partial z(n)}{\partial y(n-1)}, \dots, \frac{\partial z(n)}{\partial y(n-N+1)} \right]^T \\ &= \frac{\partial d(n)}{\partial z(n)} \mathbf{l}(n) = g(n) \mathbf{l}(n) \end{aligned} \quad (44)$$

where $g(n)$ represents the derivative $\partial d(n)/\partial z(n)$, which can be easily obtained by first finding the relationship between $z(n)$ and $d(n)$. The physical meaning of (44) is a linear filter $l(n)$ followed by an amplifier with gain $g(n)$, as shown in Fig. 8(a),

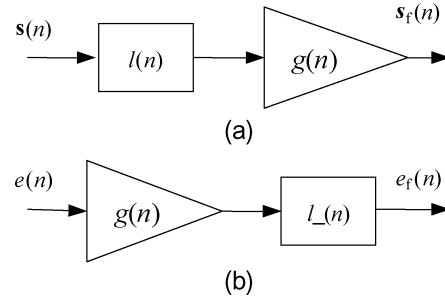


Fig. 8. Block diagram of (a) IEL filter and (b) adjoint IEL filter for the Wiener nonlinear model.

where the box is a linear filter and the triangle is a gain. The coefficients vector of the adjoint IEL filter $\tilde{\mathbf{v}}_1(n)$ can be obtained from (38)

$$\tilde{\mathbf{v}}_1(n) = [l(N-1, n)g(n), l(N-2, n)g(n-1), \dots, l(0, n)g(n-N+1)]^T. \quad (45)$$

This is shown in Fig. 8(b). Note that in this case the time-varying gain must be placed ahead of the linear FIR filter.

The path from the output of the adaptive filter $\mathbf{p}(n)$ to the summation point in Fig. 7 can also be regarded as a Wiener nonlinear model, yielding the IEL filter coefficients vector

$$\tilde{\mathbf{v}}_2(n) = g(n)[w(n) * l(n)]. \quad (46)$$

The corresponding adjoint IEL filter may be obtained as before. Similarly, we can define the linear and nonlinear state vectors $\mathbf{s}(n)$ and $\mathbf{u}(n)$, which represent the inputs to adaptive linear filter $w(n)$ and adaptive nonlinear filter $\mathbf{p}(n)$, respectively. Using this IEL filter and the update given in (22), we derive the update function for the NFXLMS-based adaptive filters in the Hammerstein predistorter

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e * (n) \mathbf{s}_f(n) \quad (47)$$

$$\mathbf{p}(n+1) = \mathbf{p}(n) + \mu_p e * (n) \mathbf{u}_f(n) \quad (48)$$

where $\mathbf{u}_f(n)$ and $\mathbf{s}_f(n)$ are the state vectors $\mathbf{u}(n)$ and $\mathbf{s}(n)$ filtered by their corresponding IEL filters. Note, due to the time-varying nature of the IEL filter coefficients vector $\tilde{\mathbf{v}}_1(n)$, all states in $\mathbf{s}(n)$ must be filtered by $\tilde{v}_1(n)$. The block diagram of the adaptive Hammerstein predistorter based on NFXLMS is shown in Fig. 9. It can be determined that (47) and (48) are identical to those introduced in [22]. However, our IEL filter configuration has a much simpler structure.

Using the adjoint IEL filters with the update function given in (40), we have the updates for the NALMS algorithm shown in Fig. 10

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e_{f1}^*(n) \mathbf{s}(n-N+1) \quad (49)$$

$$\mathbf{p}(n) = \mathbf{p}(n-1) + \mu_p e_{f2}^*(n) \mathbf{u}(n-N-M+2). \quad (50)$$

Note that the predistorter using the NALMS algorithm has a simpler structure that requires fewer computations than the one based on the NFXLMS. Later, in Section VI, we give the number of multiplications for each algorithm per iteration. Other adaptive Hammerstein nonlinear predistorters based on the NFXRLS

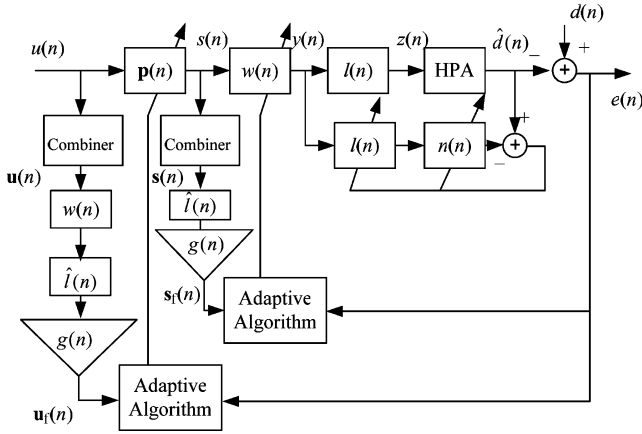


Fig. 9. Adaptive Hammerstein nonlinear predistorter based on NFXLMS.

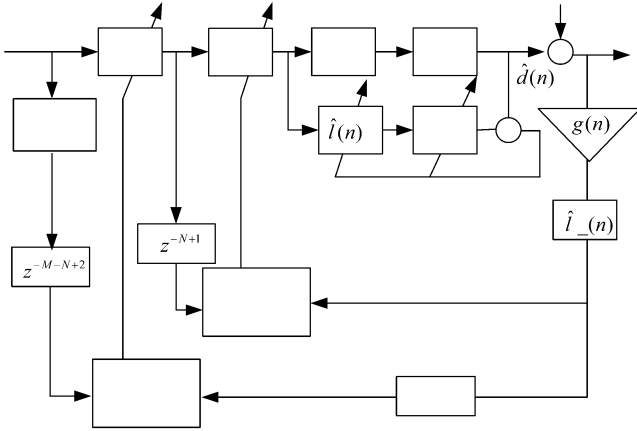


Fig. 10. Adaptive Hammerstein nonlinear predistorter based on NALMS.

and NARLS algorithms can be easily obtained via a method similar to our previous discussion.

V. FURTHER SIMPLIFICATION USING PHASE ROBUSTNESS

As we discussed earlier, the NFXLMS and NALMS algorithm cases can be regarded as extensions (or as generalizations) of the linear filtered-x LMS and adjoint LMS algorithms, respectively. The main idea behind NFXLMS and NALMS is common to their linear counterparts—to keep the reference signal and error signal aligned in time [26], [27]. The IEL filter and adjoint IEL filter in the two algorithms serve this purpose. However, to make the adaptive filter converge with a sufficiently small step size, we only need to filter the reference signal or the error signal with a filter that has the frequency response phase within $\pm 90^\circ$ of the actual phase response [28], [31]. As a result, we can say the NFXLMS and NALMS algorithms are robust and can tolerate some estimation errors of the nonlinear system. We now want to use this robustness and the statistical properties of the IEL filter to further simplify our proposed predistorters.

Authorized licensed use limited to: Suzanne Habib. Downloaded on February 20, 2025 at 15:05:46 UTC from IEEE Xplore. Restrictions apply.

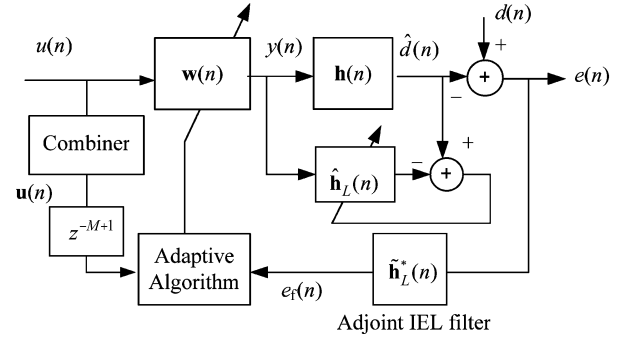


Fig. 11. Simplified nonlinear predistorter for a “weakly” nonlinear system.

A. Simplified Predistorter for a Wiener Nonlinear System

We closely examine the IEL filter of the Wiener model in Fig. 8. First, note that the mean value of $g(n)$ is a simple gain. Because a simple gain $g(n)$ does not introduce phase delay, it may be disregarded during the weight update. Using this approximation to further simplify the adaptive predistorter yields a structure similar to that shown in Figs. 9 and 10, but with the simple gain $g(n)$ removed. However, it is now relevant to pursue the effect of this approximation on the step size. Consider that the update of (49) is now

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu' e_{\text{fl}}^*(n) \mathbf{s}(n) \quad (51)$$

where the e_{fl}^* corresponds to e_{fl}^* in Fig. 10, except that it is not amplified by the gain $g(n)$. Defining $\mu' = \mu g(n)$ will equate the two cases as far as the step sizes are concerned, and so it is now clear that the approximation will affect the step size stability bound of the adaptive algorithm. If $E[g(n)] \approx 0$ (as is the case for saturation), then μ' should take a value much smaller than μ for convergence. This simplification can reduce the computational count and simplify the structure in the update of the adaptive nonlinear predistorter. Even more importantly, in the identification procedure, we find that we need only identify the linear part of the Wiener model if the memoryless nonlinear part will not significantly constrain the step size.

B. Simplified Predistorter for a “Weakly” Nonlinear System

A nonlinear system can also be modeled as the parallel connection of a linear part and a nonlinear part

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}_L(n) + \hat{\mathbf{h}}_N(n) \quad (52)$$

where $\hat{\mathbf{h}}(n)$ represents the nonlinear general polynomial, and $\hat{\mathbf{h}}_L(n)$ and $\hat{\mathbf{h}}_N(n)$ are the linear and nonlinear parts respectively. So the IEL filter of $\hat{\mathbf{h}}(n)$ can be written as

$$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}_L(n) + \tilde{\mathbf{h}}_N(n). \quad (53)$$

According to (19), the IEL filter for a linear filter is that linear filter itself, so we have

$$\tilde{\mathbf{h}}(n) = \hat{\mathbf{h}}_L(n) + \tilde{\mathbf{h}}_N(n). \quad (54)$$

TABLE I
COMPUTATIONAL COMPLEXITY BASED ON THE NUMBER OF MULTIPLICATIONS REQUIRED BY DIFFERENT UPDATING ALGORITHMS PER ITERATION

Operation	NFXLMS	NALMS	Simplified NALMS
$\mathbf{u}(n)$ (Generated by combiner)	$\frac{(P+N-1)!}{N!(P-1)!} - 1$	$\frac{(P+N-1)!}{N!(P-1)!} - 1$	$\frac{(P+N-1)!}{N!(P-1)!} - 1$
IEL filter $\tilde{\mathbf{h}}(n)$	$M \left[\frac{(P+N-2)!}{N!(P-2)!} - 1 \right]$	$M \left[\frac{(P+N-2)!}{N!(P-2)!} - 1 \right]$	None
$\mathbf{u}_f(n)$	$M \left[\frac{(P+N)!}{N!P!} - 1 \right]$	None	None
$e_f(n)$	None	M	M
$y(n)$	$\frac{(P+N)!}{N!P!} - 1$	$\frac{(P+N)!}{N!P!} - 1$	$\frac{(P+N)!}{N!P!} - 1$
$\mu_p e_f(n)$ or $\mu_p e(n)$	P	P	P
$[\mu_p e_f(n)]\mathbf{u}(n)$ or $[\mu_p e(n)]\mathbf{u}_f(n)$	$\frac{(P+N)!}{N!P!} - 1$	$\frac{(P+N)!}{N!P!} - 1$	$\frac{(P+N)!}{N!P!} - 1$
Total	C_{NFXLMS}	C_{NALMS}	C_{SNALMS}

If the phase response difference of $\tilde{\mathbf{h}}(n)$ and $\hat{\mathbf{h}}_L(n)$ is within $\pm 90^\circ$, which is the real meaning of “weakly,” then we can approximate the IEL filter of the nonlinear system using only its linear part

$$\tilde{\mathbf{h}}(n) \approx \hat{\mathbf{h}}_L(n). \quad (55)$$

Using this approximation allows us to design a simplified nonlinear predistorter based on the NFXLMS algorithm. It turns out that the resulting algorithm is the same as the “filtered-A LMS algorithm” introduced in [32]. However, our discussion provides a theoretical foundation and framework for this simplification. Also, the simplified predistorter based on the NALMS algorithm (see Fig. 11) is more efficient. Our version yields a significant savings in both the weight update and the nonlinear system identification procedure. However, the price paid for this simplification is a slower convergence speed; this is examined in more detail in our simulations.

VI. COMPUTATIONAL COMPLEXITY

In order to illustrate the reduced computational burden of our proposed algorithms, we compare the number of complex multiplications required to complete a single iteration for the different algorithms. As we discussed earlier, the complexity comes from filtering, updating the weights, and identification. In the following calculations, we do not consider the identification part

because it is required for most of the adaptive predistorters based on direct learning methods.

First, we apply the discussed algorithms (NFXLMS in Figs. 3 and 4, NALMS in Fig. 6, and simplified NALMS in Fig. 11) to a Volterra nonlinear predistorter with P th-order nonlinearity and a memory size of N . The nonlinear device $\mathbf{h}(n)$ is also modeled by a Volterra polynomial with order P and memory size M . The number of multiplications required for the different operations of each algorithm is shown in Table I, where the total are given by (56)–(58), as shown at the bottom of the page. These calculations can be obtained by referring to the discussion in [33]. Similarly, we can also determine that the predistorter based on the NALMS saves about $(M-1)(P+N-1)[N!(P-1)!]^{-1}$ memory elements when compared to the NFXLMS algorithm, since only the error signal needs to be filtered by the adjoint filter instead of all elements in $\mathbf{u}(n)$. The simplified predistorter also saves computations and reduces memory requirements by avoiding the identification of the nonlinear part of the nonlinear device.

In Table II, we give out the computational complexity of an adaptive Hammerstein nonlinear predistorter discussed in Section IV with different algorithms. In the table, M and N represent the order of $w(n)$ and $l(n)$ respectively, and P is the order of nonlinearities for both $n(n)$ and $\mathbf{p}(n)$ shown in Figs. 9 and 10.

From these results, we can see that the adaptive predistorter based on our proposed NALMS algorithm can greatly reduce

$$C_{\text{NFXLMS}} = \frac{(P+N-2)![(3P+2N+MP+MN)(P+N-1)+MP(P-1)]}{N!P!} - 3 + P - 2M \quad (56)$$

$$C_{\text{NALMS}} = \frac{(P+N-2)![(3P+2N)(P+N-1)+MP(P-1)]}{N!P!} - 3 + P \quad (57)$$

$$C_{\text{SNALMS}} = \frac{(P+N-1)!(3P+2N)}{N!P!} - 3 + M + P \quad (58)$$

TABLE II
COMPUTATIONAL COMPLEXITY BASED ON THE NUMBER OF MULTIPLICATIONS PER ITERATION REQUIRED BY DIFFERENT HAMMERSTEIN PREDISTORTERS

Operation	NFXLMS	NALMS	Simplified NALMS
$\mathbf{u}(n)$	$Q-1$	$Q-1$	$Q-1$
$g(n)$	$Q-2$	$Q-2$	None
$\mathbf{u}_f(n)$	$Q(M+N+1)$	None	None
$\mathbf{s}_f(n)$	$M(N+1)$	None	None
$e_{f1}(n)$	None	$N+1$	N
$e_{f2}(n)$	None	M	M
$\mu_p e_f(n)$ or $\mu_p e(n)$	Q	Q	Q
$y(n)$	$Q+M$	$Q+M$	$Q+M$
$[\mu_p e_f(n)]\mathbf{u}(n)$ or $[\mu_p e(n)]\mathbf{u}_f(n)$	Q	Q	Q
$[\mu e_f(n)]\mathbf{s}(n)$ or $[\mu e(n)]\mathbf{s}_f(n)$	$M+1$	$M+1$	$M+1$
Total	$M(Q+N+3)+QN+6Q-2$	$3M+6Q-1$	$3M+4Q+N$

the computational complexity and memory requirements. Comparing the results of Table I and Table II, the large computational savings obtained from using the block-oriented nonlinear model is clear. The computational savings of the NARLS over the NFXRLS can be calculated in a similar fashion. We do not give the details here because several efficient implementations of the RLS algorithms already exist, for instance the fast-transversal-filter (FTF) and the least-square-lattices (LSLs) algorithms [34]. These algorithms can be extended using our methods. Also, we can always apply the RLS algorithms to initialize the nonlinear predistorters, and then switch to the computation-saving LMS algorithms near convergence to track the slow variations typically found in nonlinear systems such as HPAs in our case.

VII. SIMULATIONS

Using computer simulations that are carried out under Matlab and Matlab Simulink, we compare the performance of the various algorithms in terms of normalized MSE (NMSE), BER, and the effectiveness in suppressing spectral regrowth. Before carrying out the simulations, we first give the definition of NMSE and output back-off (OBO). The NMSE is given by

$$\text{NMSE} = 10 \log_{10} \left(\frac{\sum_{k=1}^K |d_k - \hat{d}_k|^2}{\sum_{k=1}^K |d_k|^2} \right) \quad (59)$$

where K is the total number of test data samples, d_k is the desired output of the k th test datum, and $\hat{d}_k$ is the actual output. The OBO is given by

$$\text{OBO} = 10 \log_{10} \frac{P_{\max}}{P_o} \quad (60)$$

where $P_{\max}$ represents the maximum output power of the HPA at saturation and P_o denotes the mean output power of the signal at the HPA output. The smaller the OBO, the more the HPA is working in a nonlinear fashion.

Simulation 1: We simulate data predistorters with the configuration shown in Fig. 1(a). We generate a 54-subcarrier OFDM signal and pass it through the nonlinear Wiener system that is

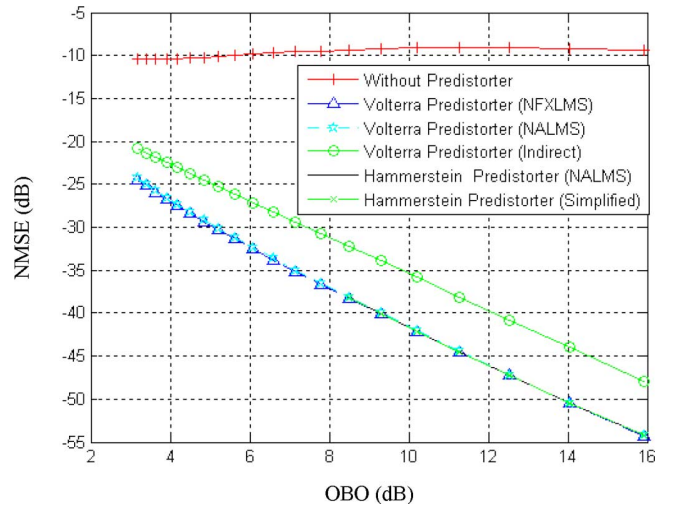


Fig. 12. NMSE versus OBO for different nonlinear predistorters.

used in [22] and [35]. This path is modeled with a causal linear FIR filter whose coefficients are $[0.7692 \ 0.1538 \ 0.0769]$, which precedes an AM/AM and AM/PM model of a memoryless HPA given in [36]. We implement a third-order adaptive Volterra predistorter with memory size 9, based on the indirect learning method shown in Fig. 2, as well as our direct learning algorithms. We also implement the p th-order inverse predistorter [10], a memoryless predistorter, and the adaptive Hammerstein predistorters (with the same order and memory size with the same structure as in Figs. 9 and 10). During our simulation, we tune the adaptive filter step size by the trial and error method to ensure that each algorithm achieves its best performance.

In Fig. 12, we show the performance improvement in terms of the NMSE of our proposed predistorters over the predistorter based on the indirect learning method. In this case, the adaptive Volterra and Hammerstein predistorters based on different direct learning algorithms have similar, and better, performance at different OBO than does the predistorter-based indirect learning. Thus, the direct learning curves are undistinguishable from each other in the figure. Another way of comparing these results is to note that for the same NMSE, the direct learning method will boost the output power about 5 dB over that provided by the

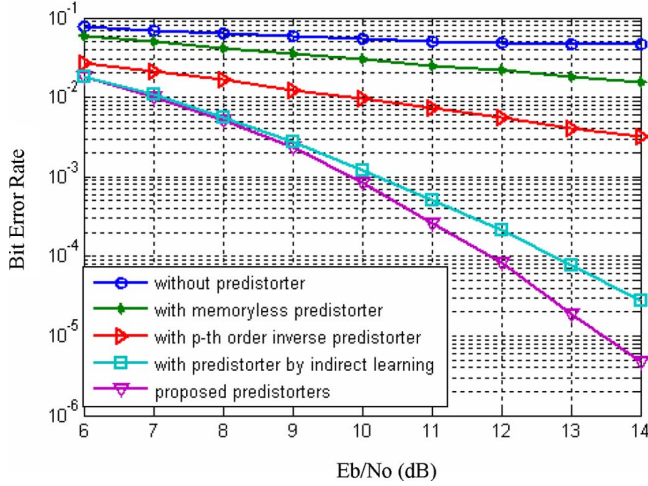


Fig. 13. Bit error rate versus Eb/No at OBO = 6.2.

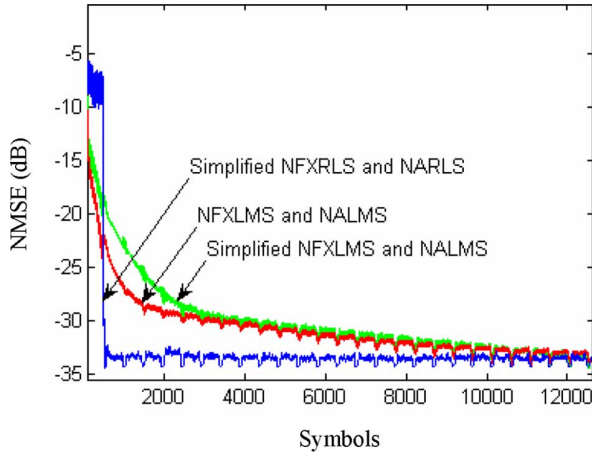


Fig. 14. The learning curve of different algorithms.

indirect methods. Consequently, the amplifier's peak power requirement could be reduced by about a factor of 3, resulting in a very significant cost advantage.

Fig. 13 shows the improvement of the different data predistorters on the BER. We find that the memoryless predistorter and the p -th order predistorter do not provide much improvement on the BER. This is because the memoryless predistorter cannot remove the ISI caused by memory effects, and the p -th order inverse can only remove the first p orders of the nonlinearity (similar results are reported in [17]). Our proposed predistorters successfully compensate the nonlinear distorter of the HPA and have the lowest BER.

The learning curves of adaptive Hammerstein predistorters based on different training algorithms are obtained by averaging 200 independent trials and are shown in Fig. 14. Comparing the NFXLMS and NALMS algorithms, and their simplified algorithms, we see that, under this situation, the computational complexity is reduced with minimal convergence rate cost. The simplified NFXRLS and NARLS algorithms converge much faster than do the LMS-based algorithms. Due to the short memory length of this HPA model, there is no discernable difference in

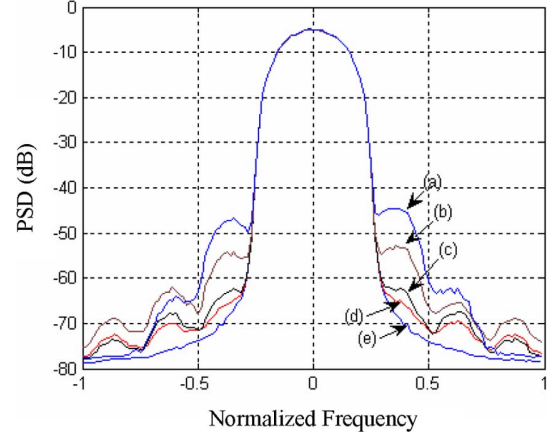


Fig. 15. Power spectral density of HPA with measurement SNR = 43.3 dB. (a) Output of HPA without predistorter. (b) Output with memoryless predistorter. (c) Output with predistorter trained by indirect learning. (d) Output with predistorter trained by NARLS and simplified NFXRLS algorithms. (e) Original input.

the convergence between the filtered-x algorithms and their corresponding adjoint algorithms. As a result, their learning curves are overlapped.

Simulation 2: In this simulation, we apply different algorithms to predistorters using the memory polynomial, which is claimed to be robust in [16]. The PA takes the same model as in [16], i.e.

$$y(n) = \sum_{k=1}^K \sum_{q=0}^Q c_{kq} z(n-q) |z(n-q)|^{k-1}. \quad (61)$$

We also use the same coefficients as those in Example 2 of [16]. Our simulation is set up as in Fig. 1(b). We generate a WCDMA baseband signal, up-sample it by a factor of 8, and then filter by a raised cosine filter with roll-off factor 0.2. The bandlimited complex signal is then passed through the nonlinear model. The predistorters also use the memory polynomial model with $Q = 6$ and $K = 5$ as in [16]. Due to the correlation of different nonlinear terms in (61), the LMS-based training algorithms converge very slowly and perform poorly. For this reason, the RLS-based training algorithms are implemented for the indirect and direct learning algorithms. Fig. 15 shows the spectral regrowth due to the nonlinear distortion with and without predistorters. From this figure and other simulations, we see that our proposed algorithms can effectively reduce the spectral regrowth. The measurement noise will affect the performance of all algorithms. However, our algorithms are less affected by the measurement noise. The effects of the noise on the indirect learning algorithm are caused by the biased estimation of the post-inverse function, and the noise effect on our direct learning algorithm is due to the adaptive filter misalignment. We also see that the memoryless predistorter cannot effectively suppress the spectral regrowth caused by the HPA.

VIII. CONCLUSION AND DISCUSSION

In this paper, we have first extended the nonlinear filtered-x LMS method to update a general nonlinear polynomial predistorter. With our proposed IEL filter concept, we were then able

to simplify the structure of the NFXLMS algorithm and unify it to a linear form. We also develop several novel direct learning algorithms: The NFXRLS, the NALMS, and the NARLS algorithms. These novel predistorters outperform predistorters based on the indirect learning algorithm in term of NMSE and spectral regrowth. Moreover, when compared with other adaptive nonlinear predistorters based on the direct learning architecture, our methods maintain good performance even though they enjoy a significantly lower computational burden, much lower memory needs, and/or a faster convergence rate.

We also applied the developed algorithm to design an adaptive Hammerstein nonlinear predistorter for memoryless and memory polynomial HPA models. We found that by using the block-oriented nonlinear model, we can greatly reduce the implementation cost of the adaptive nonlinear predistorter. Similar methods can be extended to other block-oriented nonlinear models such as the LNL model. Further simplifications were conducted by exploiting the phase robustness of the developed method.

Finally, although our discussions are based on digital communication systems, the proposed methods can also be used in other applications, such as control systems, to remove the nonlinear distortion of actuators. Thus, we believe our algorithm has broad application. However, the local minimum problem and the nonlinear effect on the convergence speed of our proposed algorithms are still under investigation.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the anonymous reviewers for their invaluable comments on the original and revised versions of this manuscript. The authors also would like to thank Dr. D. Morgan from Bell Laboratories for his encouragement and wonderful suggestions.

REFERENCES

- [1] C. Cripps, *RF Power Amplifiers for Wireless Communications*. Norwood, MA: Artech House, 1999.
- [2] L. Rugini, P. Banelli, and S. Cacopardi, "SER performance of linear multiuser detectors for DS-CDMA downlink with transmitter nonlinear distortions," *IEEE Trans. Veh. Technol.*, vol. 53, no. 7, pp. 992–1000, Jul. 2004.
- [3] A. Conti, D. Dardari, and V. Tralli, "An analytical framework for CDMA system with a nonlinear amplifier and AWGN," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1110–1120, Jul. 2002.
- [4] G. Pupolin, "Performance analysis of digital radio links with nonlinear transmit amplifiers," *IEEE J. Select. Areas Commun.*, vol. SAC-5, no. 4, pp. 534–546, Apr. 1987.
- [5] X. N. Fernando and A. B. Sesay, "Higher order adaptive filter based predistortion for nonlinear distortion compensation of radio over fiber links," in *Proc. IEEE ICC*, 2000, pp. 367–371.
- [6] F. J. Gonzalez-Serrano, J. J. Murillo-Fuentes, and A. Artes-Rodriguez, "GCMAC-based predistortion for digital modulations," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1679–1689, Sep. 2001.
- [7] G. Karam and H. Sari, "A data predistortion technique with memory for QAM radio systems," *IEEE Trans. Commun.*, vol. 39, no. 2, pp. 336–344, Feb. 1991.
- [8] —, "Data predistortion techniques using intersymbol interpolation," *IEEE Trans. Commun.*, vol. 38, no. 10, pp. 1716–1723, Oct. 1990.
- [9] M. Schetzen, "Theory of p th-order inverses of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 5, pp. 285–291, May 1976.
- [10] A. Sarti and S. Pupolin, "Recursive techniques for the synthesis of a p th-order inverse of a Volterra system," *Euro. Trans. Telecomm.*, vol. 3, no. 4, pp. 315–322, Jul./Aug. 1992.
- [11] J. K. Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans. Veh. Technol.*, vol. 39, no. 4, pp. 374–382, Nov. 1990.
- [12] M. Ghaderi, S. Kumar, and D. E. Dodds, "Fast adaptive polynomial I and Q predistorter with global optimization," *Inst. Elect. Eng. Proc. Commun.*, vol. 143, no. 2, pp. 78–86, Apr. 1996.
- [13] W. Böch and G. Gatti, "Measurement and simulation of memory effects in predistortion linearizers," *IEEE Trans. Microwave Theory Tech.*, vol. 37, no. 12, pp. 1885–1890, Dec. 1989.
- [14] J. H. K. Vuolevi, T. Rahkonen, and J. P. A. Manninen, "Measurement technique for characterizing memory effects in RF power amplifiers," *IEEE Trans. Microwave Theory Tech.*, vol. 49, no. 8, pp. 1383–1388, Aug. 2001.
- [15] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, pp. 1417–1418, Nov. 2001.
- [16] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, and J. S. Kenney, "A robust digital baseband predistorter constructed using memory polynomial," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.
- [17] C. Eun and E. J. Powers, "New Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.
- [18] F. X. Y. Gao and W. M. Snelgrove, "Adaptive linearization of a loudspeaker," in *Proc. IEEE ICASSP*, 1991, pp. 3589–3592.
- [19] J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York: Wiley, 2000.
- [20] D. R. Morgan, Z. Ma, and L. Ding, "Reducing measurement noise effects in digital predistortion of RF power amplifiers," in *Proc. IEEE ICC*, 2003, pp. 2436–2439.
- [21] Y. Qian and T. Yao, "Structure for adaptive predistortion suitable for efficient adaptive algorithm application," *Electron. Lett.*, vol. 38, pp. 1282–1283, Oct. 2002.
- [22] 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.
- [23] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "Adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1726–1730, Jun. 1998.
- [24] R. Raich, H. Qian, and T. Zhou, "Digital baseband predistortion of nonlinear power amplifiers using orthogonal polynomials," in *Proc. IEEE ICASSP*, 2003, pp. 689–692.
- [25] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: Wiley, 1998.
- [26] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Process. Mag.*, vol. 10, no. 10, pp. 12–35, Oct. 1993.
- [27] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems: Algorithm and DSP Implementations*. New York: Wiley, 1996.
- [28] D. R. Morgan, "Analysis of multiple correlation cancellation loops with a filter in the auxiliary path," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 454–467, Aug. 1980.
- [29] E. A. Wan, "Adjoint LMS: an efficient alternative to the filtered-X LMS and multiple error LMS algorithms," in *Proc. IEEE ICASSP*, 1996, pp. 1842–1845.
- [30] C. J. Clark, G. Chrisikos, M. S. Muha, A. A. Moulthrop, and C. P. Silva, "Time-domain envelope measurement technique with application to wideband power amplifier modeling," *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 12, pp. 2531–2540, Dec. 1998.
- [31] S. D. Snyder and C. H. Hansen, "Effect of transfer function estimation errors on the filtered-X LMS algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 4, pp. 950–953, Apr. 1994.
- [32] W. J. Klippel, "Adaptive inverse control of weakly nonlinear systems," in *Proc. IEEE ICASSP*, 1997, pp. 355–358.
- [33] L. Tan and J. Jiang, "Adaptive Volterra filters for active control of nonlinear noise processes," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1667–1676, Aug. 2001.
- [34] M. Bouchard and S. Quednau, "Multichannel recursive-least-squares algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 606–618, Sep. 2000.
- [35] C. Eun and E. J. Powers, "Predistorter design for a memory-less nonlinearity preceded by a dynamic linear system," in *Proc. IEEE GLOBECOM*, 1995, pp. 152–156.
- [36] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1715–1729, Nov. 1981.



Dayong Zhou (S'02–M'06) received the B.S. degree in electronic mechanical engineering from the University of Electronic Science and Technology, Chengdu, China, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of Oklahoma, Norman, in 2001 and 2005, respectively.

From 1995 to 1998, he worked as a Design Engineer and Group Leader in an institute of the Shanghai Aerospace Industry Bureau, Shanghai, China. In 1999, he worked as an Electrical Engineer in the Perkin Elmer Analytical Instruments,

Shanghai Office. Since May 2005, he has been a Postdoctoral Research Associate at the Dynamic Structures Sensing and Control (DySSC) Center, the University of Oklahoma, Norman. His current research interests include signal processing for wireless communication, system identification, nonlinear signal processing, noise and vibration mitigation, intelligent transportation infrastructure, and structural health monitoring.

Dr. Zhou has served as an active reviewer for several IEEE and other international journals. He is currently a program committee member of several international conferences on signal processing and intelligent systems.



Victor DeBrunner (S'82–M'90–SM'00) was born in Auburn, AL, on August 21, 1962. He received the B.E.E. degree from Auburn University in 1984, and the M.S. and Ph.D. degrees in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1986 and 1990, respectively.

He moved to Florida State University, Tallahassee, in August 2006, where he serves as Professor and Department Chair of the FAMU-FSU Department of Electrical and Computer Engineering. He was at the University of Oklahoma, Norman, from 1990 to

2006, where he left as the Kerr-McGee Presidential Professor in the School of Electrical and Computer Engineering and the Director of the Dynamic Structures Sensing and Control (DySSC) Center. His research interests include signal and image processing algorithms and implementations.

Dr. DeBrunner has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE SIGNAL PROCESSING LETTERS, and he is currently an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I. He is the Technical Chair of the 2006 IEEE DSP Workshop, as well as the 2006 Asilomar Conference on Signals, Systems, and Computers, where he is also a member of the steering committee. He will be the General Chair of the Asilomar Conference in 2007.