Adaptive Predistortion of IIR Hammerstein Systems Using the Nonlinear Filtered-x LMS Algorithm

Li Gan and Emad Abd-Elrady
Christian Doppler Laboratory for Nonlinear Signal Processing
Institute of Signal Processing and Speech Communication
Graz University of Technology. Inffeldgasse 16b, A-8010 Graz, Austria
Email: {li.gan, emad.abd-elrady}@TUGraz.at

***Abstract*— Adaptive predistortion of nonlinear systems described using IIR Hammerstein models is introduced in this paper. The adaptive predistorter is modeled as an IIR Wiener system. The parameters of the linear and nonlinear blocks of the predistorter are estimated simultaneously using the Nonlinear Filtered-x Least Mean Squares (NFxLMS) algorithm. The NFxLMS algorithm is derived under the assumption that the parameters of the IIR Wiener predistorter are changing slowly during adaptation. Simulation study shows that the suggested predistorter using the NFxLMS algorithm can well compensate the nonlinear distortion of the IIR Hammerstein system.**

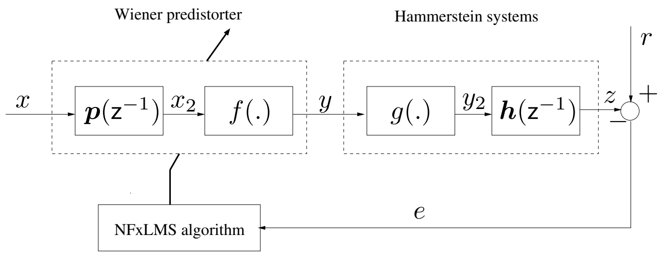

Fig. 1. Predistortion of IIR Hammerstein system.

I. Introduction

Cancelling or reducing nonlinear distortion due to nonlinearity characteristic of some electronic devices are essential requirement in many areas. Examples can be found in communication systems, speech processing and control engineering, see [1, 2].

Several adaptive predistortion techniques based on using Volterra series as a model for the nonlinear system have been proposed [1, 2]. However, since these techniques are based on using Volterra models, high computation complexity and slow convergence speed are expected problems during real-time implementation. Recently, an approach based on polyphase representation for Volterra filters that helps to reduce the computation complexity is introduced in [3].

Therefore, block-structured models such as the Wiener and Hammerstein model structures, see [4, 5], are considered in order to decrease the number of parameters to be estimated - hence decreasing computational complexity and convergence time. The Wiener model structure consists of a linear dynamic system followed by a static nonlinearity block. Wiener models arise in practice whenever a measurement device has nonlinear characteristic. For example, power amplifier can be modeled as a Wiener system [6, 7]. On the other hand, in the Hammerstein model structure, the static nonlinearity block precedes the linear dynamic system.

According to [7], power amplifier can be modeled as a FIR Wiener system or an IIR Hammerstein system. The authors also concluded that high power amplifiers are best modeled using IIR Hammerstein systems. The predistortion technique for the FIR Wiener system has been proposed in [8, 9]. The idea is to connect a FIR Hammerstein predistorter tandemly with the nonlinear system. Then the Nonlinear Filtered-x Least Mean Squares (NFxLMS) algorithm is used for adaptively adjusting the parameters of the predistorter. The approach of [8, 9] requires the estimation of the FIR Wiener system.

In this paper, the IIR Hammerstein model structure is considered as a model for the nonlinear system, hence the predistorter is modeled as an IIR Wiener system [10, 11]. The NFxLMS algorithm is developed here for estimating the parameters of the IIR Wiener predistorter. Also in this paper, the estimation of the IIR Hammerstein system is required.

This paper is organized as follows. In Section II, the structures of the IIR Hammerstein system and the IIR Wiener predistorter are defined. In Section III, the NFxLMS algorithm is derived for adaptively estimating the parameters of the predistorter. In Section IV, the validity of the proposed algorithm is demonstrated via computer simulations. Section V comes to conclusions.

II. Predistortion of Hammerstein IIR Systems

According to [10, 11], the IIR Hammerstein system shown in Fig. 1 is to be precompensated using an IIR Wiener predistorter. The output of the Hammerstein system is given by

$$\begin{aligned} z(n) &= \boldsymbol{h}(\mathsf{z}^{-1})y_2(n) = \frac{B(\mathsf{z}^{-1})}{1 - A(\mathsf{z}^{-1})}y_2(n) \\ &= \sum_{m=0}^{m_b} b_m y_2(n-m) + \sum_{m=1}^{m_a} a_m z(n-m) \quad (1) \end{aligned}$$

where $\boldsymbol{h}(\mathsf{z}^{-1}) = \frac{B(\mathsf{z}^{-1})}{1-A(\mathsf{z}^{-1})}$ and the polynomials $A(\mathsf{z}^{-1})$ and $B(\mathsf{z}^{-1})$ are defined as

$$A(\mathsf{z}^{-1}) = \sum_{m=1}^{m_a} a_m \mathsf{z}^{-m}$$
$$B(\mathsf{z}^{-1}) = \sum_{m=0}^{m_b} b_m \mathsf{z}^{-m}. \tag{2}$$

Here z^{-1} is the delay operator such that $\mathsf{z}^{-m}x(n) = x(n-m)$. The intermediate signal $y_2(n)$ is defined as

$$\begin{aligned} y_2(n) &= g_1 y(n) + g_2 y^2(n) + \cdots + g_{m_g} y^{m_g}(n) \\ &= \boldsymbol{\theta}_g^T \boldsymbol{y}(n) \end{aligned} \tag{3}$$

where

$$\boldsymbol{\theta}_g = \begin{pmatrix} g_1 & g_2 & \cdots & g_{m_g} \end{pmatrix}^T \tag{4}$$

and

$$\boldsymbol{y}(n) = \begin{pmatrix} y(n) & y^2(n) & \cdots & y^{m_g}(n) \end{pmatrix}^T. \tag{5}$$

Similarly, the output of this predistorter is given as

$$\begin{aligned} y(n) &= f_1(n)x_2(n) + f_2(n)x_2^2(n) + \cdots + f_{m_f} x_2^{m_f}(n) \\ &= \boldsymbol{\theta}_f^T(n)\boldsymbol{x}_2(n) \end{aligned} \tag{6}$$

where

$$\boldsymbol{\theta}_f(n) = \begin{pmatrix} f_1(n) & f_2(n) & \cdots & f_{m_f}(n) \end{pmatrix}^T \tag{7}$$

and

$$\boldsymbol{x}_2(n) = \begin{pmatrix} x_2(n) & x_2^2(n) & \cdots & x_2^{m_f}(n) \end{pmatrix}^T. \tag{8}$$

The intermediate signal $x_2(n)$ is given by

$$\begin{aligned} x_2(n) &= \boldsymbol{p}(n,\mathsf{z}^{-1})x(n) = \frac{D(n,\mathsf{z}^{-1})}{1-C(n,\mathsf{z}^{-1})}x(n) \\ &= \sum_{m=0}^{m_d} d_m(n)x(n-m) + \\ &\quad \sum_{m=1}^{m_c} c_m(n)x_2(n-m). \end{aligned} \tag{9}$$

where $\boldsymbol{p}(n,\mathsf{z}^{-1}) = \frac{D(n,\mathsf{z}^{-1})}{1-C(n,\mathsf{z}^{-1})}$ and the polynomials $C(n,\mathsf{z}^{-1})$ and $D(n,\mathsf{z}^{-1})$ are defined as

$$C(n,\mathsf{z}^{-1}) = \sum_{m=1}^{m_c} c_m(n)\mathsf{z}^{-m}$$
$$D(n,\mathsf{z}^{-1}) = \sum_{m=0}^{m_d} d_m(n)\mathsf{z}^{-m}. \tag{10}$$

Let us define the parameter vector $\boldsymbol{\theta}$ of the predistorter as follows

$$\begin{aligned} \boldsymbol{\theta} &= \begin{pmatrix} \boldsymbol{\theta}_f^T & \boldsymbol{\theta}_d^T & \boldsymbol{\theta}_c^T \end{pmatrix}^T \\ \boldsymbol{\theta}_f &= \begin{pmatrix} f_1 & f_2 & \cdots & f_{m_f} \end{pmatrix}^T \\ \boldsymbol{\theta}_d &= \begin{pmatrix} d_0 & d_1 & \cdots & d_{m_d} \end{pmatrix}^T \\ \boldsymbol{\theta}_c &= \begin{pmatrix} c_1 & c_2 & \cdots & c_{m_c} \end{pmatrix}^T. \end{aligned} \tag{11}$$

The goal of this paper is to estimate the parameter vector $\boldsymbol{\theta}$ by minimizing the mean square difference given by

$$E\{e^2(n)\} = E\{(r(n) - z(n))^2\} \tag{12}$$

where $E\{.\}$ denotes the Expectation and $r(n)$ is the reference signal which is defined in [2] and given by

$$r(n) = x(n-\tau) + \upsilon(n). \tag{13}$$

Here τ is the delay time and $\upsilon(n)$ is zero-mean Additive White Gaussian Noise (AWGN).

Remark 1: The delay time τ equals to zero in case the system to be compensated is minimum phase [2].

The NFxLMS algorithm can be developed to estimate the parameter vector $\boldsymbol{\theta}$. This algorithm is introduced in the next section.

III. The NFxLMS Algorithm

The NFxLMS algorithm is obtained by applying the stochastic gradient algorithm [2]:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \frac{\mu}{2}\boldsymbol{\Delta}^T(n) \tag{14}$$

where μ is a positive constant with value less than 1 and usually defined as the *step-size* parameter. Also, $\boldsymbol{\Delta}(n)$ is the gradient vector which is defined as

$$\boldsymbol{\Delta}(n) = \frac{de^2(n)}{d\boldsymbol{\theta}(n)} = -2e(n)\frac{dz(n)}{d\boldsymbol{\theta}(n)}. \tag{15}$$

Using Eq. (1), $\frac{dz(n)}{d\boldsymbol{\theta}(n)}$ can be derived as

$$\frac{dz(n)}{d\boldsymbol{\theta}(n)} = \sum_{m=0}^{m_b} b_m \frac{dy_2(n-m)}{d\boldsymbol{\theta}(n)} + \sum_{m=1}^{m_a} a_m \frac{dz(n-m)}{d\boldsymbol{\theta}(n)}. \tag{16}$$

If the step-size μ is sufficiently small so that the parameter vector $\boldsymbol{\theta}$ changes slowly [2, 12, 13], the following approximations can be made:

$$\begin{aligned} \frac{dy_2(n-m)}{d\boldsymbol{\theta}(n)} &\approx \frac{dy_2(n-m)}{d\boldsymbol{\theta}(n-m)}, \quad m = 0,1,\cdots,m_b \\ \frac{dz(n-m)}{d\boldsymbol{\theta}(n)} &\approx \frac{dz(n-m)}{d\boldsymbol{\theta}(n-m)}, \quad m = 1,2,\cdots,m_a. \end{aligned} \tag{17}$$

Consequently, Eq. (16) can be written as

$$\begin{aligned} \frac{dz(n)}{d\boldsymbol{\theta}(n)} &\approx \sum_{m=0}^{m_b} b_m \frac{dy_2(n-m)}{d\boldsymbol{\theta}(n-m)} + \sum_{m=1}^{m_a} a_m \frac{dz(n-m)}{d\boldsymbol{\theta}(n-m)} \\ &= \frac{B(\mathsf{z}^{-1})}{1-A(\mathsf{z}^{-1})}\frac{dy_2(n)}{d\boldsymbol{\theta}(n)} = \boldsymbol{h}(\mathsf{z}^{-1})\frac{dy_2(n)}{d\boldsymbol{\theta}(n)} \\ &\approx \widehat{\boldsymbol{h}}(\mathsf{z}^{-1})\frac{dy_2(n)}{d\boldsymbol{\theta}(n)} \end{aligned} \tag{18}$$

where $\widehat{\boldsymbol{h}}(\mathsf{z}^{-1})$ is the estimate of $\boldsymbol{h}(\mathsf{z}^{-1})$. From Eqs. (3)-(5), we have

$$\begin{aligned} \frac{dy_2(n)}{d\boldsymbol{\theta}(n)} &= \boldsymbol{\theta}_g^T \frac{d\boldsymbol{y}(n)}{d\boldsymbol{\theta}(n)} \\ &= \boldsymbol{\theta}_g^T \frac{d\boldsymbol{y}(n)}{dy(n)}\frac{dy(n)}{d\boldsymbol{\theta}(n)} \\ &\approx s_1(n)\frac{dy(n)}{d\boldsymbol{\theta}(n)} \end{aligned} \tag{19}$$

where

$$s_1(n) = \widehat{\boldsymbol{\theta}}_g^T \frac{d\boldsymbol{y}(n)}{dy(n)} = \widehat{\boldsymbol{\theta}}_g^T \begin{pmatrix} 1 \\ 2y(n) \\ \vdots \\ m_g y^{m_g-1}(n) \end{pmatrix}. \tag{20}$$

Here $\widehat{\boldsymbol{\theta}}_g$ is the estimate of $\boldsymbol{\theta}_g$. Using Eqs. (11) and (18), Eq. (19) becomes

$$\begin{aligned}\frac{dz(n)}{d\boldsymbol{\theta}(n)} &\approx \widehat{\boldsymbol{h}}(\mathsf{z}^{-1})s_1(n)\frac{dy(n)}{d\boldsymbol{\theta}(n)}\\ &= \widehat{\boldsymbol{h}}(\mathsf{z}^{-1})s_1(n)\left(\frac{\partial y(n)}{\partial\boldsymbol{\theta}_f(n)}\ \frac{\partial y(n)}{\partial\boldsymbol{\theta}_d(n)}\ \frac{\partial y(n)}{\partial\boldsymbol{\theta}_c(n)}\right).\end{aligned} \tag{21}$$

Considering Eqs. (6), (8) and (9), $\frac{\partial y(n)}{\partial\boldsymbol{\theta}_f(n)}$ can be derived as

$$\begin{aligned}\frac{\partial y(n)}{\partial\boldsymbol{\theta}_f(n)} &= \frac{\partial\boldsymbol{\theta}_f^T(n)\boldsymbol{x}_2(n)}{\partial\boldsymbol{\theta}_f(n)} = \boldsymbol{x}_2^T(n)\\ &= \begin{pmatrix}\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\\ \left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]^2\\ \vdots\\ \left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]^{m_f}\end{pmatrix}^T.\end{aligned} \tag{22}$$

Note that the intermediate signal $x_2(n)$ should be estimated since it is usually not measurable. Again, using Eqs. (6), (8) and (9), $\frac{\partial y(n)}{\partial\boldsymbol{\theta}_d(n)}$ and $\frac{\partial y(n)}{\partial\boldsymbol{\theta}_c(n)}$ can be derived as

$$\begin{aligned}\frac{\partial y(n)}{\partial\boldsymbol{\theta}_d(n)} &= \boldsymbol{\theta}_f^T(n)\frac{\partial\boldsymbol{x}_2(n)}{\partial x_2(n)}\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_d(n)} = s_2(n)\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_d(n)}\\ \frac{\partial y(n)}{\partial\boldsymbol{\theta}_c(n)} &= \boldsymbol{\theta}_f^T(n)\frac{\partial\boldsymbol{x}_2(n)}{\partial x_2(n)}\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_c(n)} = s_2(n)\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_c(n)}\end{aligned} \tag{23}$$

where

$$\begin{aligned}s_2(n) &= \boldsymbol{\theta}_f^T(n)\frac{\partial\boldsymbol{x}_2(n)}{\partial x_2(n)} = \boldsymbol{\theta}_f^T(n)\begin{pmatrix}1\\ 2x_2(n)\\ \vdots\\ m_f x_2^{m_f-1}(n)\end{pmatrix}\\ &= \boldsymbol{\theta}_f^T(n)\begin{pmatrix}1\\ 2\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]\\ \vdots\\ m_f\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]^{m_f-1}\end{pmatrix}.\end{aligned} \tag{24}$$

Now, it remains to derive $\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_d(n)}$ and $\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_c(n)}$. Differentiating both sides of Eq. (9) with respect to $d_k(n)$ and $c_k(n)$ gives

$$\begin{aligned}\frac{\partial x_2(n)}{\partial d_k(n)} &= x(n-k) + \sum_{m=1}^{m_c} c_m(n)\frac{\partial x_2(n-m)}{\partial d_k(n)}\\ \frac{\partial x_2(n)}{\partial c_k(n)} &= x_2(n-k) + \sum_{m=1}^{m_c} c_m(n)\frac{\partial x_2(n-m)}{\partial c_k(n)}.\end{aligned} \tag{25}$$

Since the parameter vector $\boldsymbol{\theta}$ is assumed to be changing slowly, we can write

$$\begin{aligned}\frac{\partial x_2(n-m)}{\partial d_k(n)} &\approx \frac{\partial x_2(n-m)}{\partial d_k(n-m)}, m = 1,2,\cdots,m_c\\ \frac{\partial x_2(n-m)}{\partial c_k(n)} &\approx \frac{\partial x_2(n-m)}{\partial c_k(n-m)}, m = 1,2,\cdots,m_c.\end{aligned} \tag{26}$$

Hence, Eq. (25) can be rewritten as

$$\begin{aligned}\frac{\partial x_2(n)}{\partial d_k(n)} &\approx x(n-k) + \sum_{m=1}^{m_c} c_m(n)\frac{\partial x_2(n-m)}{\partial d_k(n-m)}\\ \frac{\partial x_2(n)}{\partial c_k(n)} &\approx x_2(n-k) + \sum_{m=1}^{m_c} c_m(n)\frac{\partial x_2(n-m)}{\partial c_k(n-m)}\end{aligned} \tag{27}$$

or

$$\begin{aligned}\frac{\partial x_2(n)}{\partial d_k(n)} &\approx \frac{\mathsf{z}^{-k}}{1-C(n,\mathsf{z}^{-1})}x(n), k = 0,1,\cdots,m_d\\ \frac{\partial x_2(n)}{\partial c_k(n)} &\approx \frac{\mathsf{z}^{-k}}{1-C(n,\mathsf{z}^{-1})}x_2(n)\\ &= \frac{\mathsf{z}^{-k}}{1-C(n,\mathsf{z}^{-1})}\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right],\\ &\quad k = 1,\cdots,m_c.\end{aligned} \tag{28}$$

Therefore, we have

$$\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_d(n)} = \begin{pmatrix}\frac{\partial x_2(n)}{\partial d_0(n)}\\ \frac{\partial x_2(n)}{\partial d_1(n)}\\ \vdots\\ \frac{\partial x_2(n)}{\partial d_{m_d}(n)}\end{pmatrix}^T \approx \begin{pmatrix}\frac{1}{1-C(n,\mathsf{z}^{-1})}x(n)\\ \frac{\mathsf{z}^{-1}}{1-C(n,\mathsf{z}^{-1})}x(n)\\ \vdots\\ \frac{\mathsf{z}^{-m_d}}{1-C(n,\mathsf{z}^{-1})}x(n)\end{pmatrix}^T \tag{29}$$

and

$$\begin{aligned}\frac{\partial x_2(n)}{\partial\boldsymbol{\theta}_c(n)} &= \begin{pmatrix}\frac{\partial x_2(n)}{\partial c_1(n)}\\ \frac{\partial x_2(n)}{\partial c_2(n)}\\ \vdots\\ \frac{\partial x_2(n)}{\partial c_{m_c}(n)}\end{pmatrix}^T\\ &\approx \begin{pmatrix}\frac{\mathsf{z}^{-1}}{1-C(n,\mathsf{z}^{-1})}\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]\\ \frac{\mathsf{z}^{-2}}{1-C(n,\mathsf{z}^{-1})}\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]\\ \vdots\\ \frac{\mathsf{z}^{-m_c}}{1-C(n,\mathsf{z}^{-1})}\left[\boldsymbol{p}(n,\mathsf{z}^{-1})x(n)\right]\end{pmatrix}^T\end{aligned} \tag{30}$$

Now, we have completely derived the components of $\frac{dz(n)}{d\boldsymbol{\theta}(n)}$ in Eq. (21) and hence the gradient vector $\boldsymbol{\Delta}(n)$.

IV. SIMULATION STUDY

In this simulation study, the following IIR Hammerstein system was considered:

$$\begin{aligned}z(n) &= \frac{0.26+0.77\mathsf{z}^{-1}+0.77\mathsf{z}^{-2}+0.26\mathsf{z}^{-3}}{1+0.58\mathsf{z}^{-1}+0.42\mathsf{z}^{-2}+0.06\mathsf{z}^{-3}}y_2(n)\\ y_2(n) &= y(n)+0.5y^2(n)+0.25y^3(n).\end{aligned} \tag{31}$$

Remark 2: In this simulation study we assumed that the system is already well identified, *i.e.* $\widehat{\boldsymbol{h}}(\mathsf{z}^{-1}) = \boldsymbol{h}(\mathsf{z}^{-1})$ and $\widehat{\boldsymbol{\theta}}_g = \boldsymbol{\theta}_g$, respectively.

The order of the linear and nonlinear blocks of the IIR Wiener predistorter were chosen as $m_c = 3$, $m_d = 3$ and $m_f = 9$, respectively. The input signal was a random signal with uniform distribution over $(-1,1)$ with data length of 10^6 samples. The bandwidth of the input signal was limited in order to prevent aliasing [14].

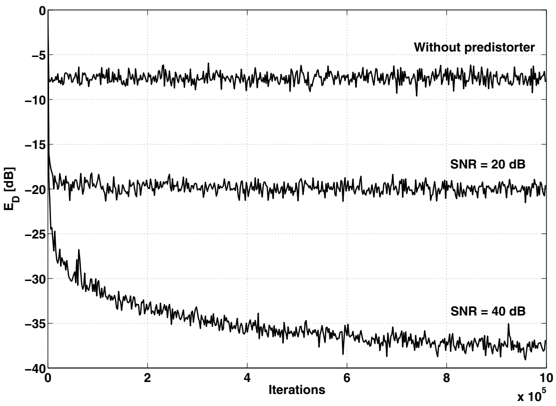

Fig. 2. E_D for different SNRs for the NFxLMS algorithm.

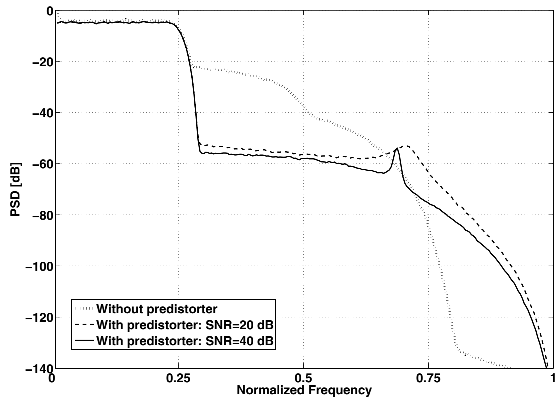

Fig. 3. Power spectral density for different SNRs.

As a performance measure, the normalized mean-square distortion E_D of the system containing the predistorter and the nonlinear system has been evaluated. E_D is defined as

$$E_D(n) = 10\log_{10}\left(\frac{\widehat{E}\{e^2(n)\}}{\widehat{E}\{r^2(n)\}}\right) \tag{32}$$

where $\widehat{E}\{.\}$ is the mean over 200 independent realizations.

The parameter vectors were initialized as

$$\begin{aligned} \boldsymbol{\theta}_f(0) &= \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^T \\ \boldsymbol{\theta}_d(0) &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T \\ \boldsymbol{\theta}_c(0) &= \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T. \end{aligned} \tag{33}$$

Figure 2 shows E_D of the NFxLMS algorithm for different values of signal to noise ratios (SNRs). The step size was set as $\mu = 0.1$. On average, the values achieved for E_D were -19.98 dB and -37.72 dB for SNR=20 dB and 40 dB, respectively. The value of E_D without the predistorter in noise-free scenario was -7.59 dB.

Figure 3 shows power spectral densities (PSDs) of the output signals of the IIR Hammerstein system with and without predistorter. From this figure, we can see that the IIR Wiener predistorter using the NFxLMS algorithm can effectively reduce spectral regrowth and it is quite robust against measurement noise.

V. CONCLUSIONS

Adaptive predistortion of nonlinear systems described using IIR Hammerstein models is considered in this paper. The NFxLMS algorithm has been derived for the estimation of the parameters of the predistorter. This is done under the assumption that the parameters of the predistorter are changing slowly during the adaptation process. The simulation results show that the suggested IIR Wiener predistorter using the NFxLMS algorithm can effectively compensate the nonlinear distortion of the IIR Hammerstein system.

ACKNOWLEDGMENT

This work was supported by Infineon Technologies Austria AG. The authors would also like to thank Prof. G. Kubin, Dr. D. Schwingshackl, and Dr. G. Paoli for valuable discussions during this work.

REFERENCES

[1] X. Y. Gao and W. M. Snelgrove, "Adaptive linearization schemes for weakly nonlinear systems using adaptive linear and nonlinear FIR filters," in *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*, 1990, vol. 1, pp. 9–12.

[2] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Tran. on Sig. Proc.*, vol. 46, no. 6, 1998.

[3] D. Schwingshackl and G. Kubin, "Polyphase representation of multirate nonlinear filters and its applications," *IEEE Trans. on Signal Processing*, vol. 55, no. 5, pp. 2145–2157, 2007.

[4] E. Abd-Elrady, "A nonlinear approach to harmonic signal modeling," *Signal Processing*, vol. 84, no. 1, pp. 163–195, 2004.

[5] E. Abd-Elrady, *Nonlinear Approaches to Periodic Signal Modeling*, Ph.d. thesis, Department of Information Technology, Uppsala University, Uppsala, Sweden, 2005, PDF file at http://publications.uu.se/abstract.xsql?dbid=4644.

[6] L. Ding, R. Raich, and G. T. Zhou, "A Hammerstein predistortion linearization design based on the indirect learning architecture," in *Proc. of The IEEE ICASSP*, Orlando, Florida, Canada, 2002, vol. 3, pp. 2689–2692.

[7] P. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *Proc. of APMC*, Suzhou, China, 2005, vol. 2.

[8] H. W. Kang, Y. S. Cho, and D. H. Youn, "Adaptive precompensation of Wiener systems," *IEEE Trans. on Signal Processing*, vol. 46, no. 10, 1998.

[9] 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. on Comm.*, vol. 47, no. 4, 1999.

[10] M. Ibnkahla, "Natural gradient learning neural networks for adaptive inversion of Hammerstein systems," *IEEE Signal Processing Letters*, vol. 9, no. 10, pp. 315–317, 2002.

[11] K. Lashkari and A. Puranik, "Exact linearization of Wiener and Hammerstein systems," in *Proc. of The 5th ICICS*, Bangkok, Thailand, 2005, pp. 917–920.

[12] C. R. Johnson Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 237–250, 1984.

[13] D. Y. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Trans. on Signal Processing*, vol. 55, no. 1, 2007.

[14] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, R. E. Krieger, Florida, USA, 1989.