Direct Learning Architectures for Digital Predistortion of Nonlinear Volterra Systems

Emad Abd-Elrady #1 , Bernard Mulgrew #2

Institute for Digital Communications
The University of Edinburgh, Edinburgh EH9 3JL, UK
[1] e.abdelrady@ed.ac.uk [2] b.mulgrew@ed.ac.uk

***Abstract*—Digital compensation of nonlinear distortion due to nonlinear characteristic of electronic or electromechanical device is becoming more and more important. This paper considers Direct Learning Architectures (DLAs) for predistortion of nonlinear systems described using Volterra series. The adaptive predistorter - which is connected in tandem with the nonlinear system - can be modeled as a Volterra filter or using linear and nonlinear FIR filters. Also, the coefficients of the adaptive predistorter are estimated in this paper using two approaches. The first approach is based on the Nonlinear Filtered-x Least Mean Squares (NFxLMS) algorithm. The second approach is based on using the Spectral Magnitude Matching (SMM) method that minimizes the sum squared error between spectral magnitudes of output signal of the nonlinear system and desired signal. The coefficients of the predistorter in this case are estimated recursively using the generalized Newton iterative algorithm. A comparative simulation study between these different architectures and approaches is given in this paper.**

I. Introduction

Recently, the problem of linearizing Power Amplifiers (PAs) used in wireless sensor networks and DSL systems attracts the attention of many researchers, see [1, 2]. More examples can be found in HiFi systems, speech processing and control engineering, see [3, 4]. The Volterra series provides a general way to model nonlinear dynamic systems. It is a multidimensional combination of a linear convolution and a nonlinear power series that has the ability to capture memory effects. Therefore, it can be employed to characterize a nonlinear PA with memory effects [5, 6].

Two kinds of adaptive compensation techniques can be used to reduce nonlinear distortion, which are adaptive postdistortion, also named as adaptive equalization, and adaptive predistortion. Although postdistortion is an effective way to compensate nonlinear distortion, predistortion is more efficient and necessary in many other situations, such as compensation of nonlinear distortion for power amplifiers in communication systems and active noise cancellation for loudspeakers.

Predistortion of nonlinear Volterra systems based on the Direct Learning Architecture (DLA) approach and using the Nonlinear Filtered-x Least Mean Squares (NFxLMS) algorithm has been considered in [4, 7]. The idea in [4, 7] is to connect a nonlinear Volterra predistorter in tandem, as shown in Fig. 1, with the nonlinear Volterra system and adaptively adjusting the coefficients of the predistorter in order to minimize the mean square distortion. These coefficients were estimated recursively using the NFxLMS algorithm. The

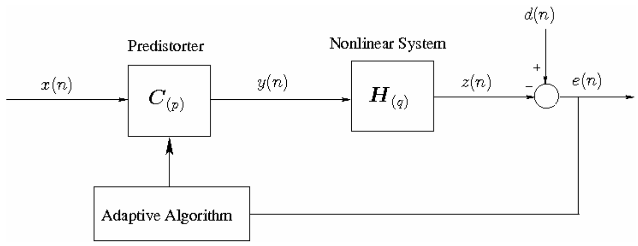

Fig. 1. Adaptive predistortion of nonlinear systems.

problems usually encountered while using the NFxLMS algorithm are slow convergence and need of accurate identification of the nonlinear Volterra system.

In [8], a method was introduced for estimating telephone handset nonlinearity by matching the spectral magnitude of the distorted signal to the output of a nonlinear channel model. In this paper, the same approach of [8] is used for the purpose of direct predistortion of nonlinear Volterra systems. The coefficients of the predistorter are estimated recursively using the generalized Newton iteration algorithm to minimize the sum squared error between spectral magnitudes of output signal of the nonlinear Volterra system and desired signal. The suggested Spectral Magnitude Matching (SMM) approach presented in this paper does not require the identification of the nonlinear Volterra system as in [4, 7].

The predistorter can be modeled as a nonlinear Volterra filter or using the approach in [3]. In [3], a linearization scheme was proposed that uses a nonlinear predistorter constructed as shown in Fig. 2. Since the nonlinear physical system in [3] is described using Volterra series, it can be regarded as a nonlinear system consisting of two subsystems, one is the purely linear subsystem $\boldsymbol{H}_L$ and the other is the purely nonlinear subsystem $\boldsymbol{H}_N$. The predistorter is constructed using $\boldsymbol{H}_N$, inverse of the linear subsystem $\boldsymbol{H}_L^{-1}$ and delayed input signal. The subsystems $\boldsymbol{H}_L^{-1}$ and $\boldsymbol{H}_N$ are modeled using linear and nonlinear FIR filters [9], respectively. In this paper, the coefficients of these filters are directly estimated using the NFxLMS algorithm and the SMM method.

The paper is organized as follows. In Sec. II, the DLAs based on the NFxLMS algorithm are discussed. The DLAs using the SMM method are presented in Sec. III. In Sec. IV, a comparative simulation study between these DLAs is given. Conclusions are presented in Sec. V.

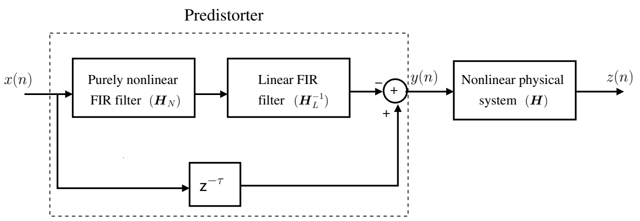

Fig. 2. The suggested linearization scheme using linear and nonlinear FIR filters.

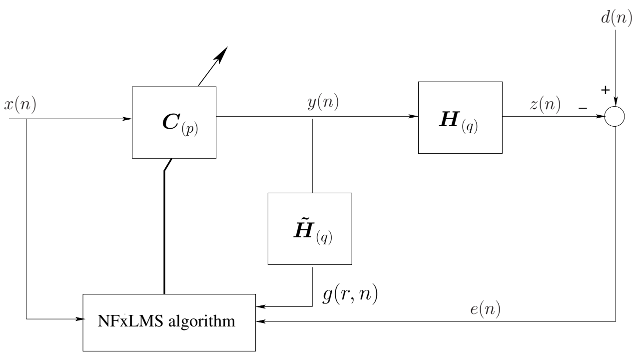

Fig. 3. DLA using adaptive Volterra predistorter and the NFxLMS algorithm.

II. Direct Learning Architectures Based on the NFxLMS Algorithm

A. Using a Nonlinear Volterra Filter

The DLA approach using a nonlinear Volterra filter, see Fig. 3, assumes that the nonlinear system $\boldsymbol{H}_{(q)}$ to be compensated is a discrete-time invariant causal system. Also, the system $\boldsymbol{H}_{(q)}$ with input and output signals $y(n)$ and $z(n)$ can be modeled by qth-order Volterra series with M-tap memories. Hence, the output $z(n)$ is given by

$$z(n) = \sum_{k=1}^{q} \left(\sum_{i_1=0}^{M-1} .. \sum_{i_k=0}^{M-1} h_k(i_1, .., i_k) y(n-i_1)..y(n-i_k) \right) \tag{1}$$

where $h_k(i_1, \cdots, i_k)$ are the kth-order kernels.

Similarly, the relation between the input and output of the adaptive Volterra predistorter $\boldsymbol{C}_{(p)}$ is given by

$$y(n) = \sum_{k=1}^{p} \left(\sum_{i_1=0}^{N-1} .. \sum_{i_k=0}^{N-1} \theta_k(i_1, .., i_k; n) x(n-i_1)..x(n-i_k) \right) \tag{2}$$

where N is the number of memories and $\theta_k(i_1, \cdots, i_k; n)$ are the kth-order kernels of this predistorter. According to the pth-order Volterra theorem [10], the Volterra filter $\boldsymbol{C}_{(p)}$ can remove nonlinearities up to pth-order provided that the inverse of the first-order Volterra system is causal and stable.

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

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_1^T & \cdots & \boldsymbol{\theta}_p^T \end{pmatrix}^T, \tag{3}$$

where $\boldsymbol{\theta}_k$ is given by

$$\boldsymbol{\theta}_k = \begin{pmatrix} \theta_k(0, \cdots, 0) & \cdots & \theta_k(N-1, \cdots, N-1) \end{pmatrix}^T. \tag{4}$$

Also, assume that the desired signal $d(n)$ is given as

$$d(n) = x(n-T) + v(n), \tag{5}$$

where T is the time delay necessary to have a causal Volterra predistorter and $v(n)$ is zero-mean additive white Gaussian noise (AWGN).

Remark 1: The time delay T equals zero in case the system to be compensated is minimum phase [4].

The main goal of digital predistortion is to estimate the parameter vector $\boldsymbol{\theta}$ such that the output signal $z(n)$ becomes very close to the desired signal $d(n)$. The kernels of the adaptive Volterra filter were estimated in [4], see Fig. 3, by minimizing the mean square distortion defined as

$$\mathrm{E}\{e^2(n)\} = \mathrm{E}\{[d(n) - z(n)]^2\} \tag{6}$$

where E denotes the expectation and $d(n)$ is the desired signal defined in Eq. (5).

The NFxLMS algorithm of [4] was obtained by applying the stochastic gradient algorithm [11]:

$$\boldsymbol{\theta}_k(n+1) = \boldsymbol{\theta}_k(n) - \frac{\mu_k}{2} \, \boldsymbol{\Delta}_k(n) \tag{7}$$

where μ_k is the step-size parameter. Also, the gradient vector $\boldsymbol{\Delta}_k(n)$ is defined as

$$\boldsymbol{\Delta}_k(n) = \begin{pmatrix} \frac{\partial e^2(n)}{\partial \theta_k(0, \cdots, 0; n)} & \cdots & \frac{\partial e^2(n)}{\partial \theta_k(N-1, \cdots, N-1; n)} \end{pmatrix}^T. \tag{8}$$

Taking into consideration that (*cf.* Eq. (6))

$$\frac{\partial e^2(n)}{\partial \theta_k(i_1, \cdots, i_k; n)} = -2e(n) \frac{\partial z(n)}{\partial \theta_k(i_1, \cdots, i_k; n)} \tag{9}$$

where $\frac{\partial z(n)}{\partial \theta_k(i_1, \cdots, i_k; n)}$ can be written as

$$\frac{\partial z(n)}{\partial \theta_k(i_1, \cdots, i_k; n)} = \sum_{r=0}^{M-1} g(r; n) \frac{\partial y(n-r)}{\partial \theta_k(i_1, \cdots, i_k; n)}. \tag{10}$$

Here $g(r; n)$ is given as

$$\begin{aligned} g(r; n) = \frac{\partial z(n)}{\partial y(n-r)} &= h_1(r) + 2 \sum_{i=0}^{M-1} h_2(r, i) y(n-i) \\ &+ 3 \sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} h_3(r, i_1, i_2) y(n-i_1) y(n-i_2) + \cdots \end{aligned} \tag{11}$$

Assuming that μ_k is chosen sufficiently small, $\frac{\partial y(n-r)}{\partial \theta_k(i_1, \cdots, i_k; n)}$ can be approximated as (*cf.* Eq. (2))

$$\begin{aligned} \frac{\partial y(n-r)}{\partial \theta_k(i_1, \cdots, i_k; n)} &\approx \frac{\partial y(n-r)}{\partial \theta_k(i_1, \cdots, i_k; n-r)} \\ &= x(n-r-i_1) \cdots x(n-r-i_k). \end{aligned} \tag{12}$$

Substituting by Eqs. (10)-(12) in Eq. (9), we have

$$\begin{aligned} &\frac{\partial e^2(n)}{\partial \theta_k(i_1, .., i_k; n)} = \\ &-2e(n) \sum_{r=0}^{M-1} g(r; n) x(n-r-i_1) \cdots x(n-r-i_k). \end{aligned} \tag{13}$$

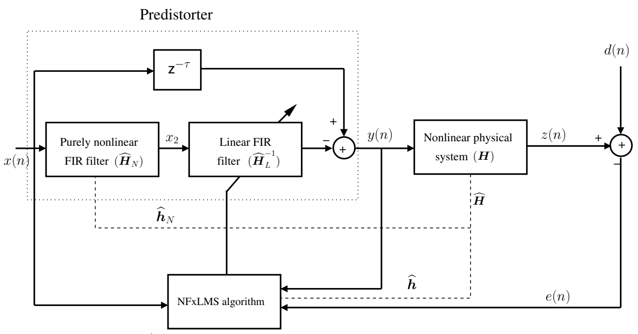

Fig. 4. DLA based on adaptive FIR filters and the NFxLMS algorithm.

Hence, the gradient vector $\mathbf{\Delta}_k(n)$ (*cf.* Eq. (8)) can be evaluated and the parameter vector $\boldsymbol{\theta}_k$ can be estimated using Eq. (7).

Remark 2: In Eq. (11), it is assumed that the correct kernels of the nonlinear system $\boldsymbol{H}_{(q)}$ are known or have been estimated.

B. Using Linear and Nonlinear FIR Filters

The DLA based on using adaptive FIR filters and the NFxLMS algorithm is given in Fig. 4. The output of the nonlinear physical system $\boldsymbol{H}$ can be written as

$$\begin{aligned} z(n) &= \boldsymbol{H}[y(n)] = \boldsymbol{H}_L[y(n)] + \boldsymbol{H}_N[y(n)] \\ &= \boldsymbol{h}^T\boldsymbol{y}(n) = \boldsymbol{h}_L^T\boldsymbol{y}_L(n) + \boldsymbol{h}_N^T\boldsymbol{y}_N(n) \end{aligned} \tag{14}$$

where

$$\boldsymbol{h} = \begin{pmatrix} \boldsymbol{h}_L^T & \boldsymbol{h}_N^T \end{pmatrix}^T \tag{15}$$

$$\boldsymbol{h}_L = \begin{pmatrix} h_0 & h_1 & \cdots & h_{M-1} \end{pmatrix}^T \tag{16}$$

$$\boldsymbol{h}_N = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{M-1,\cdots,M-1} \end{pmatrix}^T \tag{17}$$

and

$$\boldsymbol{y}(n) = \begin{pmatrix} \boldsymbol{y}_L^T(n) & \boldsymbol{y}_N^T(n) \end{pmatrix}^T \tag{18}$$

$$\boldsymbol{y}_L(n) = \begin{pmatrix} y(n) & y(n-1) & \cdots & y(n-M+1) \end{pmatrix} \tag{19}$$

$$\boldsymbol{y}_N(n) = \begin{pmatrix} y^2(n) & y(n)y(n-1) & \cdots & y^q(n-M+1) \end{pmatrix}. \tag{20}$$

Due to the fact that the NFxLMS algorithm requires the nonlinear Volterra system to be first identified, *i.e.* the nonlinear subsystem $\boldsymbol{H}_N$ is already estimated, it remains to adaptively estimate the inverse of the linear subsystem $\boldsymbol{H}_L$. This inverse is denoted as $\boldsymbol{H}_L^{-1}$ and modeled as a linear FIR filter with $\widehat{N}$-tap memory and a parameter vector $\boldsymbol{\theta}_l$ defined as

$$\boldsymbol{\theta}_l(n) = \begin{pmatrix} \theta_{l,0}(n) & \theta_{l,1}(n) & \cdots & \theta_{l,\widehat{N}-1}(n) \end{pmatrix}^T. \tag{21}$$

Similarly to Eq. (7), the NFxLMS algorithm for estimating $\boldsymbol{\theta}_l$ follows as:

$$\boldsymbol{\theta}_l(n+1) = \boldsymbol{\theta}_l(n) - \frac{\mu_l}{2}\mathbf{\Delta}^T(n) \tag{22}$$

and in this case the gradient vector $\mathbf{\Delta}(n)$ is defined as

$$\mathbf{\Delta}(n) = \frac{\mathrm{d}e^2(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} = -2e(n)\frac{\mathrm{d}z(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)}. \tag{23}$$

Using Eqs. (14)-(20), $\frac{\mathrm{d}z(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)}$ can be written as

$$\frac{\mathrm{d}z(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} = \frac{\mathrm{d}(\boldsymbol{h}^T\boldsymbol{y}(n))}{\mathrm{d}\boldsymbol{\theta}_l(n)} = \boldsymbol{h}^T \begin{pmatrix} \frac{\mathrm{d}\boldsymbol{y}_L(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \\ \frac{\mathrm{d}\boldsymbol{y}_N(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \end{pmatrix}. \tag{24}$$

Assuming that the parameter vector $\boldsymbol{\theta}_l$ is changing slowly [4, 12], we have

$$\frac{\mathrm{d}\boldsymbol{y}_L(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} = \begin{pmatrix} \frac{\mathrm{d}y(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \\ \vdots \\ \frac{\mathrm{d}y(n-M+1)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \end{pmatrix} \approx \begin{pmatrix} \frac{\mathrm{d}y(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \\ \vdots \\ \frac{\mathrm{d}y(n-M+1)}{\mathrm{d}\boldsymbol{\theta}_l(n-M+1)} \end{pmatrix}. \tag{25}$$

Now, since we have

$$y(n) = x(t-\tau) - \boldsymbol{\theta}_l^T(n)\boldsymbol{x}_2(n) \tag{26}$$

where τ is the time delay satisfying $\boldsymbol{H}_L^{-1}\boldsymbol{H}_L = \mathsf{z}^{-\tau}$ [3],

$$\begin{aligned} \boldsymbol{x}_2(n) &= \begin{pmatrix} x_2(n) & \cdots & x_2(n-\widehat{N}+1) \end{pmatrix}^T \\ &= \begin{pmatrix} \boldsymbol{h}_N^T\boldsymbol{x}(n) & \cdots & \boldsymbol{h}_N^T\boldsymbol{x}(n-\widehat{N}+1) \end{pmatrix}^T \end{aligned} \tag{27}$$

and

$$\boldsymbol{x}(n) = \begin{pmatrix} x^2(n) & x(n)x(n-1) & \cdots & x^q(n-M+1) \end{pmatrix}^T \tag{28}$$

Eq. (25) becomes

$$\frac{\mathrm{d}\boldsymbol{y}_L(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \approx - \begin{pmatrix} \boldsymbol{x}_2^T(n) \\ \boldsymbol{x}_2^T(n-1) \\ \vdots \\ \boldsymbol{x}_2^T(n-M+1) \end{pmatrix}. \tag{29}$$

Similarly, $\frac{\mathrm{d}\boldsymbol{y}_N(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)}$ can be evaluated as

$$\frac{\mathrm{d}\boldsymbol{y}_N(n)}{\mathrm{d}\boldsymbol{\theta}_l(n)} \approx - \begin{pmatrix} 2y(n)\boldsymbol{x}_2^T(n) \\ y(n)\boldsymbol{x}_2^T(n-1) + y(n-1)\boldsymbol{x}_2^T(n) \\ \vdots \\ qy^{q-1}(n-M+1)\boldsymbol{x}_2^T(n-M+1) \end{pmatrix}. \tag{30}$$

Substituting by Eqs. (29)-(30) in Eq. (24), the gradient vector $\mathbf{\Delta}(n)$ in Eq. (23) can be evaluated and the parameter vector $\boldsymbol{\theta}_l$ is estimated using Eq. (22).

III. Direct Learning Architectures Based on the SMM Method

A. Using a Nonlinear Volterra Filter

The suggested SMM approach shown in Fig. 5 minimizes the error between spectral magnitudes of the desired signal $d(n)$ and the output signal $z(n)$ through the following cost function, see [8, 13]:

$$V_{\boldsymbol{\theta}} = \sum_{k=0}^{K-1}\sum_{l=0}^{L-1}\left[|D(\omega_l;k)| - |Z(\omega_l;k;\boldsymbol{\theta})|\right]^2 \tag{31}$$

where the parameter vector $\boldsymbol{\theta}$ is defined in Eqs. (3)-(4), $D(\omega_l;k)$ and $Z(\omega_l;k;\boldsymbol{\theta})$ are the short-time DFT of the desired and output signals, respectively. Here, K is the number of

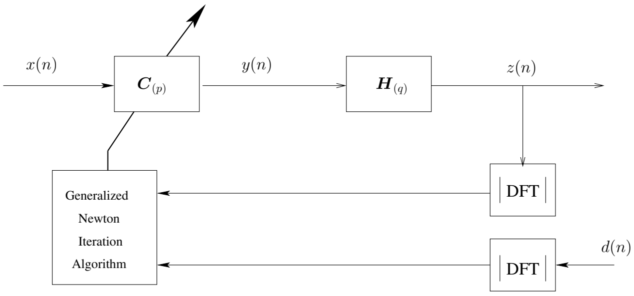

Fig. 5. DLA using adaptive Volterra predistorter and the SMM method.

uniformly-spaced short-time frames and L is the DFT length. The cost function in Eq. (31) can be written as

$$V_{\boldsymbol{\theta}} = \boldsymbol{\Gamma}_{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}_{\boldsymbol{\theta}} \tag{32}$$

where

$$\boldsymbol{\Gamma}_{\boldsymbol{\theta}} = \begin{pmatrix} \boldsymbol{\gamma}_0^T(\boldsymbol{\theta}) & \boldsymbol{\gamma}_1^T(\boldsymbol{\theta}) & \cdots & \boldsymbol{\gamma}_{K-1}^T(\boldsymbol{\theta}) \end{pmatrix}^T \tag{33}$$

and

$$\boldsymbol{\gamma}_k(\boldsymbol{\theta}) = \begin{pmatrix} |D(\omega_0;k)| - |Z(\omega_0;k;\boldsymbol{\theta})| \\ \vdots \\ |D(\omega_{L-1};k)| - |Z(\omega_{L-1};k;\boldsymbol{\theta})| \end{pmatrix}. \tag{34}$$

The parameter vector $\boldsymbol{\theta}$ that minimizes the cost function $V_{\boldsymbol{\theta}}$ can be estimated similarly as in [8] using the generalized Newton iteration algorithm, see [11, 14, 15]. Hence, the estimate of the parameter vector follows as

$$\boldsymbol{\theta}(m+1) = \boldsymbol{\theta}(m) + \alpha\ \boldsymbol{\Delta}(m) \tag{35}$$

where m is the iteration index, α is the adaptation gain, and the gradient $\boldsymbol{\Delta}(m)$ is given by

$$\begin{aligned} \boldsymbol{\Delta}(m) &= -\left[\frac{\mathrm{d}^2 V_{\boldsymbol{\theta}}}{\mathrm{d}\boldsymbol{\theta}^2}\right]^{-1} \left[\frac{\mathrm{d}V_{\boldsymbol{\theta}}}{\mathrm{d}\boldsymbol{\theta}}\right] \\ &= -\left(\boldsymbol{J}^T(m)\boldsymbol{J}(m)\right)^{-1} \boldsymbol{J}^T(m)\boldsymbol{\Gamma}_{\boldsymbol{\theta}}\ |_{\boldsymbol{\theta}=\boldsymbol{\theta}(m)}\ . \end{aligned} \tag{36}$$

Here $\boldsymbol{J}(m)$ is the Jacobian matrix of first derivative of $\boldsymbol{\Gamma}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}(m)$, *i.e.*

$$\boldsymbol{J}(m) = \frac{\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{\theta}}}{\mathrm{d}\boldsymbol{\theta}}\ |_{\boldsymbol{\theta}=\boldsymbol{\theta}(m)} = \begin{pmatrix} \boldsymbol{J}_0^T(m) & \cdots & \boldsymbol{J}_{K-1}^T(m) \end{pmatrix}^T \tag{37}$$

where

$$\boldsymbol{J}_k(m) = \frac{\mathrm{d}\boldsymbol{\gamma}_k(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{\theta}}\ |_{\boldsymbol{\theta}=\boldsymbol{\theta}(m)} = -\begin{pmatrix} \left[\frac{d|Z(\omega_0;k;\boldsymbol{\theta})|}{d\boldsymbol{\theta}}\right]^T \\ \vdots \\ \left[\frac{d|Z(\omega_{L-1};k;\boldsymbol{\theta})|}{d\boldsymbol{\theta}}\right]^T \end{pmatrix}. \tag{38}$$

Due to the fact that there is no close form expression for the gradient $\boldsymbol{\Delta}(m)$, an approximate gradient was evaluated in [8, 13] by finite element approximation. The same approach is considered in this paper. The approximation follows the following lines:

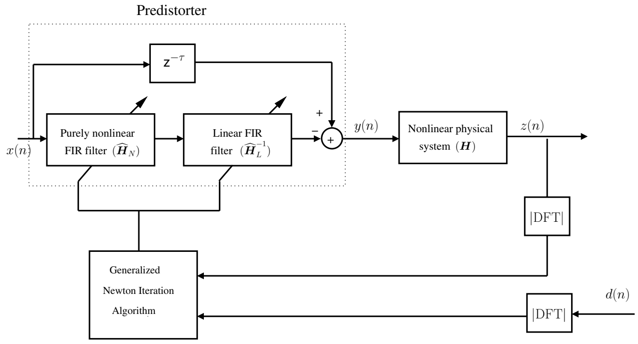

Fig. 6. DLA based on adaptive FIR filters and the SMM method.

1. Initiate with a parameter vector $\boldsymbol{\theta}(0)$ and compute the DFT magnitude $|D(\omega_l;k)|$.
2. Compute the DFT magnitude $|Z(\omega_l;k;\boldsymbol{\theta})|$ based on the current value of the parameter vector $\boldsymbol{\theta}(m)$ and form $\boldsymbol{\Gamma}_{\boldsymbol{\theta}}$.
3. Recalculate $z(n;\boldsymbol{\theta})$ for each perturbed component of $\boldsymbol{\theta}(m)$ and then compute its DFT magnitude. The (i,j) element of the matrix element $\boldsymbol{J}^k(m)$ denoted as $J_{i,j}^k(m)$ is evaluated using a first backward difference for each element of $\boldsymbol{\theta}(m)$ as

$$\begin{aligned} J_{i,j}^k(m) &= \frac{\partial \gamma_i^k(\boldsymbol{\theta};m)}{\partial \theta_j(m)} \approx -\frac{1}{\varepsilon_m} \times \\ &\Big(|Z(\omega_i;k;\theta_1(m),\cdots,\theta_j(m)+\varepsilon_m,\cdots)| \\ &\quad - |Z(\omega_i;k;\theta_1(m),\cdots,\theta_j(m),\cdots)|\Big) \end{aligned} \tag{39}$$

where $\gamma_i^k(\boldsymbol{\theta};m)$ is the ith element of $\boldsymbol{\gamma}^k(\boldsymbol{\theta})$, $\theta_j(m)$ is the jth element of the parameter vector $\boldsymbol{\theta}(m)$ and ε_m is a small adaptive perturbation evaluated as

$$\varepsilon_m = \frac{V_{\boldsymbol{\theta}}(m)}{V_{\boldsymbol{\theta}}(0)}\varepsilon_0 \tag{40}$$

where ε_0 is the initial perturbation, $V_{\boldsymbol{\theta}}(0)$ is the initial value of $V_{\boldsymbol{\theta}}$, and $V_{\boldsymbol{\theta}}(m)$ is the value of $V_{\boldsymbol{\theta}}$ at iteration m. This means that the perturbation decreases proportionally with the error.
4. Finally, evaluate the correction term $\boldsymbol{\Delta}(m)$ from Eq. (36) and update the parameter vector $\boldsymbol{\theta}$ using Eq. (35).

Remark 3: Regardless the fact that the SMM approach does not require the nonlinear system to be identified like the NFxLMS algorithm, its computation complexity is quite high. Future research will consider the possibility of reducing the computation complexity of the SMM method.

B. Using Linear and Nonlinear FIR Filters

The suggested SMM approach in this case is shown in Fig. 6. The parameter vector $\boldsymbol{\theta}$ is defined here as

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_l^T & \boldsymbol{\theta}_n^T \end{pmatrix}^T \tag{41}$$

where $\boldsymbol{\theta}_l$ is given by Eq. (21) and $\boldsymbol{\theta}_n$ is defined as

$$\boldsymbol{\theta}_n = \begin{pmatrix} \theta_{n,0,0} & \theta_{n,0,1} & \cdots & \theta_{n,\widehat{M}-1,..,\widehat{M}-1} \end{pmatrix}^T. \tag{42}$$

Similarly as done in Sec. III-A, the SMM method can also applied here in order to estimate the parameter vector $\boldsymbol{\theta}$ defined by Eq. (41).

IV. Simulation Study

In order to investigate the performance of the suggested DLAs in this paper for digital predistortion of nonlinear Volterra systems, the following simulations were performed.

The nonlinear system $\boldsymbol{H}_{(q)}$ is assumed to be a known second-order Volterra system. The input-output relation of this system is given by

$$z(n) = \boldsymbol{h}_L^T \boldsymbol{y}_L(n) + \boldsymbol{h}_N^T \boldsymbol{y}_N(n) \tag{43}$$

where $\boldsymbol{y}_L(n)$ and $\boldsymbol{y}_N(n)$ are defined in Eqs.(19)-(20), and

$$\boldsymbol{h}_L = \begin{pmatrix} 0.5625 & 0.4810 & 0.1124 & -0.1669 \end{pmatrix}^T \tag{44}$$

$$\boldsymbol{h}_N = \begin{pmatrix} 0.0175 & 0 & 0 & 0 & 0 & -0.0088 & 0 & -0.0088 & 0 \end{pmatrix}^T. \tag{45}$$

The adaptive predistorter $\boldsymbol{C}_{(p)}$ in Fig. 3 and Fig. 5 is also assumed to be a second-order Volterra filter, *i.e.* $q = p = 2$. Also, the number of memories in the adaptive Volterra predistorter was chosen as $N = 4$. For the adaptive predistorter in Fig. 4 and Fig. 6, the memory lengths of the parameter vectors $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_n$ are chosen as $\widehat{N} = 8$ and $\widehat{M} = 4$.

The input signal to the predistorter is chosen to be a random signal with uniform distribution over $(-1, 1)$ and the frequency band is limited to prevent aliasing [4, 10].

In the simulations, the NFxLMS algorithm used a data length of 5×10^5 with step sizes $\mu_1 = \mu_2 = 0.03$ and $\mu_l = 0.1$. For the SMM algorithm, 20 short-time frames were used with 128 samples and 128 DFT frequency components, *i.e.* $K = 20$ and $L = 128$. Also the adaptation gain was chosen as $\alpha = 0.05$.

Figure 7 shows power spectral densities (PSDs) of the output signals of the nonlinear Volterra system with and without the predistorter. The performance of the suggested DLAs using the NFxLMS algorithm and the SMM method are given for signal to noise ratio (SNR) of 60 dB. From this figure, we can see that the suggested architectures in this paper can reduce nonlinear distortion and suppress spectral regrowth. The DLAs using the SMM method performs much better than the DLAs using the NFxLMS algorithm. This is due to the slow convergence problem of the NFxLMS algorithm. The best performance achieved with the DLA using nonlinear Volterra predistorter and the SMM method.

V. Conclusions

Different DLAs for adaptive predistortion of nonlinear Volterra systems have been considered in this paper. Two models for the nonlinear predistorter using Volterra and FIR filters, and two approaches for estimating the coefficients of the predistorter based on the NFxLMS algorithm and the SMM method have been discussed. Simulation results show that the SMM approach can significantly suppress spectral regrowth and achieves better results than the NFxLMS algorithm especially when the predistorter is modeled using nonlinear Volterra filter. The drawback of the suggested SMM approach is high computational complexity. Future research will focus on reducing computation complexity of the SMM method.

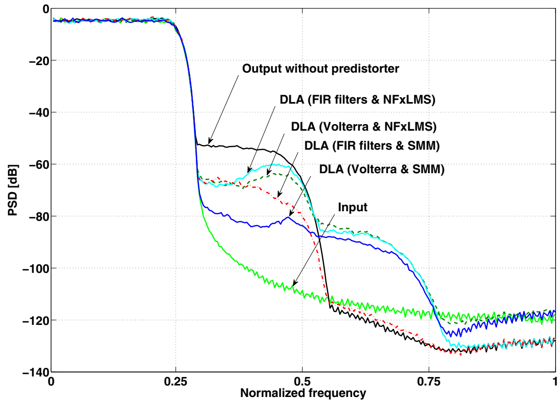

Fig. 7. PSDs for input and output signals.

References

[1] D. Aksin, S. Gregori, and F. Maloberti, “High-efficiency power amplifier for wireless sensor networks,” in *Proc. of the IEEE Symposium on Circuits and Systems (ISCAS’05)*, vol. 6, Kobe, Japan, 2005, pp. 5898–5901.

[2] L. Gan, “Adaptive digital predistortion of nonlinear systems,” Ph.D. Thesis, Faculty of Electrical and Information Engineering, Graz University of Technology, Graz, Austria, 2009.

[3] 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*, vol. 1, Calgary, Canada, 1990, pp. 9–12.

[4] 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 Signal Processing*, vol. 46, no. 6, pp. 1726–1730, 1998.

[5] A. Zhu, J. C. Pedro, and T. R. Cunha, “Pruning the volterra series for behavioral modeling of power amplifiers using physical knowledge,” *IEEE Trans. on Microwave Theory and Techniques*, vol. 55, no. 5, pp. 813–821, 2007.

[6] Y. Wan, T. J. Dodd, and R. F. Harrison, “Modeling of power amplifier nonlinearities using volterra series,” in *Proc. of the 16th IFAC Congress on Automatic Control*, Prague, Czech Republic, 2005, pp. 785–790.

[7] 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.

[8] T. F. Quatieri, D. A. Reynolds, and G. C. O’Leary, “Estimation of handset nonlinearity with application to speaker recognition,” *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 5, pp. 567–584, 2000.

[9] G. L. Sicuranza and G. Ramponi, “Adaptive nonlinear digital filters using distributed arithmetic,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp. 518–526, 1986.

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

[11] T. Söderström and P. Stoica, *System Identification*. Hemel Hempstead, United Kingdom: Prentice-Hall International, 1989.

[12] L. Gan and E. Abd-Elrady, “Adaptive predistortion of IIR Hammerstein system using the nonlinear filtered-x LMS algorithm,” in *Proc. of the IEEE CSNDSP*, Graz, Austria, 2008, pp. 702–705.

[13] E. Abd-Elrady and L. Gan, “Identification of Hammerstein and Wiener models using spectral magnitude matching,” in *Proc. of the 17th IFAC Congress on Automatic Control*, Seoul, Korea, 2008, pp. 6440–6445.

[14] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1985.

[15] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, Massachusetts: Addison-Wesley, 1973.