

# Blind Equalization and Identification of Nonlinear and IIR Systems—A Least Squares Approach

Gil M. Raz and Barry D. Van Veen, *Senior Member, IEEE*

**Abstract**—A deterministic approach to blind nonlinear channel equalization and identification is presented. This approach applies to nonlinear channels that can be approximately linearized by either finite memory, finite-order Volterra filters, or by a finite number of finite memory nonpolynomial nonlinearities. Both the nonlinear equalizers and the linearized channels are identified. This method also applies to blind identification of linear IIR channels. General conditions for existence and uniqueness are discussed, and numerical examples are given.

**Index Terms**—Blind deconvolution, communications channels, equalization, IIR systems, least squares, linearization, nonlinear systems, Volterra, .

## I. INTRODUCTION

**I**DENTIFICATION and equalization of nonlinear systems are problems of practical interest in many cases where the systems cannot be accurately modeled as linear. Blind linear identification and equalization problems have been studied by many researchers; however, relatively little work has been done in the field of blind nonlinear system identification and equalization. Blind approaches have been proposed for restricted classes of nonlinearities such as linear-zero memory nonlinearity-linear systems [7] or strict Volterra models of nonlinearity [3]. The work in [3] describes a very elegant approach for finding linear FIR equalizers for nonlinear channels. However, the results in [3] only apply to nonlinear channels that can be exactly described by Volterra filters. Another class of approaches are based on assuming a finite alphabet [11]; these suffer from ambiguity in the solution.

In this paper, we consider blind equalization of a very general class of nonlinear systems: those that can be approximately linearized by finite order, finite memory Volterra filters. We generalize the approach in [15] to show that under rather general conditions, it is possible to blindly determine  $P$ th-order linearizing equalizers and the linearized channels represented by the cascade of the nonlinear channels and the equalizers. We note that the equalizers derived here are not necessarily the so-called  $P$ th-order inverse equalizers [9]. Rather, they are

$P$ th-order Volterra filters that provide the optimal equalization in the least-squares sense. That is, the purpose of the equalizers is to optimally compensate the nonlinear behavior of the channels. The  $P$ th-order inverse approach eliminates the second — through  $P$ th-order components of that nonlinearity, which is a result that does not necessarily reduce the overall nonlinear behavior of the channels.

As a special case, we apply this approach to blind identification of linear IIR channels. We also generalize the results to design blind equalizers consisting of nonpolynomial forms of nonlinearities. Use of nonpolynomial equalizers may offer improved performance and reduced computational complexity in specific situations.

This paper is organized as follows. Section II introduces notation and develops the problem statement. The equations needed for linearization and identification are derived in Section III. In Section IV, rudimentary conditions for identifiability and linearizability are given. Orthogonal expansions and nonpolynomial expansions are discussed in Section V. Examples illustrating the effectiveness of our approach are given in Section VI.

## II. PRELIMINARIES

### A. Notation

Boldface lowercase and uppercase letters denote vectors and matrices, respectively. The symbol  $(\cdot)^T$  denotes the transposition operator, the symbol  $*$  represents convolution, and the symbol  $\otimes$  represents the Kronecker (tensor) product [6]. The symbol  $\mathbf{T}_L\{\mathbf{x}\}$  denotes a Toeplitz matrix with  $L + 1$  columns constructed from the vector  $\mathbf{x}[x(0), \dots, x(N)]^T$  as follows:

$$\mathbf{T}_L\{\mathbf{x}\} = \begin{bmatrix} x(0) & 0 & 0 & \dots & 0 \\ x(1) & x(0) & 0 & \dots & 0 \\ \vdots & & & \ddots & \\ x(L) & \dots & & & x(0) \\ \vdots & & & & \vdots \\ x(N) & \dots & & & x(N-L) \end{bmatrix}. \quad (1)$$

The memory of the FIR filter with impulse response vector  $\mathbf{h}$  is denoted by  $L_{\mathbf{h}}$ . Hence, we may write the convolution  $y(l) = (x * \mathbf{h})(l)$  in vector form as  $\mathbf{y} = \mathbf{x} * \mathbf{h} = \mathbf{T}_{L_{\mathbf{h}}}\{\mathbf{x}\}\mathbf{h}$ . The notation  $h(z)$  denotes the polynomial  $\sum_{l=0}^{L_{\mathbf{h}}} h(l)z^{-l}$ . Finally, a parenthesized superscript  $x^{(m)}$  indicates the value  $x$  belonging to the  $m$ th channel.

Manuscript received April 3, 1998; revised February 15, 1999. This work was supported in part by the Wisconsin Alumni Research Foundation and by the United States Air Force under Contract F19628-95-C-002. The associate editor coordinating the review of this paper and approving it for publication was Prof. José R. Casar.

G. M. Raz is with the Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02420 USA.

B. D. Van Veen is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: vanveen@engr.wisc.edu).

Publisher Item Identifier S 1053-587X(00)00102-1.

### B. Diagonal Coordinate System

The Volterra filter [10] is used in this paper to implement a nonlinear equalizer. In order to facilitate the use of linear algebraic methods, we express the Volterra filter output as a parallel combination of linear filters applied to nonlinear combinations of the input. We call this a diagonal coordinate system (DCS) representation since the linear filter coefficients are obtained from the diagonals of the sampled hypercube defined by the Volterra kernels.

Let  $x(l)$  be the input to a Volterra filter  $G$  of order  $P$  and memory  $L$ . The output sequence  $y(l)$  is commonly written as

$$\begin{aligned} y(l) &= G\{x(l)\} \\ &= \sum_{p=1}^P \sum_{l_1=0}^L \cdots \sum_{l_p=0}^L G_p(l_1, \dots, l_p) \\ &\quad \cdot x(l-l_1) \cdots x(l-l_p). \end{aligned} \quad (2)$$

Without loss of generality, the kernels  $G_p(l_1, \dots, l_p)$  are assumed to be symmetric under permutation of the variables  $l_1, \dots, l_p$ , and hence, (2) is rewritten as

$$\begin{aligned} y(l) &= \sum_{p=1}^P \sum_{0 \leq l_1 \leq \dots \leq l_p \leq L} G_p^{\text{symm}}(l_1, \dots, l_p) \\ &\quad \cdot x(l-l_1) \cdots x(l-l_p). \end{aligned} \quad (3)$$

There are  $D_p = D(p, L)(p+L-1)!/(p-1)!L!$  possible combinations of the parameters  $l_1, \dots, l_p$  for a given polynomial order  $p$  [8]. We introduce the DCS change of coordinates  $l_1 = l$  and  $l_{\alpha+1} = l + r_\alpha$  for  $\alpha = 1, \dots, p-1$  and the notation

$$x_{p,i}(l) = x(l) \prod_{\alpha=1}^{p-1} x(l - r_\alpha^i) \quad (4)$$

$$g_{p,i}(l) = G_p^{\text{symm}}(l, l + r_1^i, \dots, l + r_{p-1}^i) \quad (5)$$

where the superscript  $i$  denotes a given combination of the  $r_\alpha$ , and  $i = 1, \dots, D_p$ . Throughout this paper, we use a further simplification of this notation, using a single index  $j = 1, \dots, \mathcal{D}$ , where  $\mathcal{D} = \sum_{p=1}^P D_p$ , such that any  $j$  in the above range corresponds to a unique combination of  $p, i$  in their proper ranges. Using this notation, we write the Volterra filter output (3) in DCS format as

$$y(l) = \sum_{p=1}^P \sum_{i=1}^{D_p} (x_{p,i} * g_{p,i})(l) = \sum_{j=1}^{\mathcal{D}} (x_j * g_j)(l). \quad (6)$$

Thus, the Volterra filter output is given by a sum of linear filters acting on the various  $p$ th-order products of the input with itself as illustrated in Fig. 1.

Note that we may rewrite (6) in vector form as

$$\begin{aligned} \mathbf{y} &= [\mathbf{T}_{L_{g_1}}\{\mathbf{x}_1\} \cdots \mathbf{T}_{L_{g_{\mathcal{D}}}}\{\mathbf{x}_{\mathcal{D}}\}] \mathbf{g} \\ &= \mathbf{T}\{\mathbf{x}\} \mathbf{g} \end{aligned} \quad (7)$$

where  $\mathbf{g}$  is the vector of concatenated impulse responses  $\mathbf{g} = [g_1^T \cdots g_{\mathcal{D}}^T]^T$ , and  $\mathbf{T}\{\mathbf{x}\} = [\mathbf{T}_{L_{g_1}}\{\mathbf{x}_1\} \cdots \mathbf{T}_{L_{g_{\mathcal{D}}}}\{\mathbf{x}_{\mathcal{D}}\}]$ .

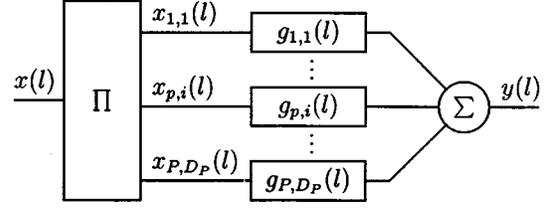


Fig. 1. Volterra filter in diagonal coordinate system.

### C. Problem Statement

Assume that oversampling or a sensor array is used to generate  $M$  outputs representing  $M$  nonlinear channels  $H^{(m)}\{\cdot\}$ ,  $m = 1, \dots, M$  applied to an input  $s(l)$ . Further, suppose the following is true:

*Assumption 2.1:* For all channels represented by the operators  $H^{(m)}\{\cdot\}$ ,  $m = 1, \dots, M$ , there are integers  $P, L$  such that there exist  $P$ th-order Volterra filters  $G^{(m)}\{\cdot\}$  of maximum memory  $L$  whose cascade connection with  $H^{(m)}\{\cdot\}$  create linear systems  $Q^{(m)}\{\cdot\}$ . That is,  $Q^{(m)}\{\cdot\} = G^{(m)}\{H^{(m)}\{\cdot\}\}$  has negligible nonlinear behavior.  $\square$

The blind identification and equalization problem is stated as follows: Given only the observations of the channel outputs, determine the Volterra filter equalizers and the linear channels that result from the cascade connection of the channels with their respective equalizers. Further, recover the input signals  $s(l)$ .

## III. IDENTIFICATION EQUATIONS

### A. Identification of General Nonlinear Channels

The output of the overall channel  $Q^{(m)}[\cdot]$  depicted in Fig. 2 is expressed as

$$\mathbf{y}^{(m)} = \mathbf{s} * \mathbf{q}^{(m)} = G^{(m)}\{\mathbf{x}^{(m)}\} \quad (8)$$

where  $G^{(m)}\{\mathbf{x}^{(m)}\} = \mathcal{T}\{\mathbf{x}^{(m)}\} \mathbf{g}^{(m)}$  is defined analogously to (7).

We now proceed with a derivation resembling the one given in [15]. Consider any two different channels  $m$  and  $n$ . Equation (8) implies

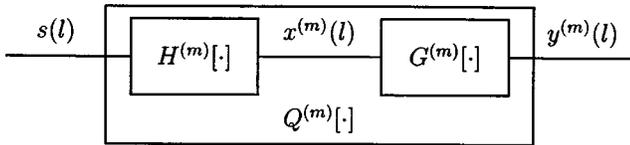
$$\mathbf{y}^{(m)} * \mathbf{q}^{(n)} = \mathbf{s} * \mathbf{q}^{(m)} * \mathbf{q}^{(n)} = \mathbf{s} * \mathbf{q}^{(n)} * \mathbf{q}^{(m)} = \mathbf{y}^{(n)} * \mathbf{q}^{(m)}. \quad (9)$$

Note that if the linearizing filters  $\mathbf{g}_j^{(m)}$  were known, then the results from [15] could be directly applied to identify the linearized channels represented by  $\mathbf{q}^{(m)}$ . In the problem at hand, we need to find not only the vectors  $\mathbf{q}^{(m)}$  but the vectors  $\mathbf{g}^{(m)}$  as well.

Write (8) in matrix form as

$$\begin{aligned} \mathbf{y}^{(m)} &= \mathbf{T}_{L_{q^{(m)}}}\{\mathbf{s}\} \mathbf{q}^{(m)} \\ &= \mathcal{T}\{\mathbf{x}^{(m)}\} \mathbf{g}^{(m)} \\ &= \sum_{j=1}^{\mathcal{D}} \mathbf{x}_j^{(m)} * \mathbf{g}_j^{(m)}. \end{aligned} \quad (10)$$

We note that the number of columns in  $\mathcal{T}\{\mathbf{x}^{(m)}\}$  depends on the lengths of the vectors  $\mathbf{g}_j^{(m)}$  and, hence, may vary between the channels. By using (10) in conjunction with (9), we find

Fig. 2. Channel  $m$  with linearizing equalizer.

identification equations that define the vectors  $\mathbf{g}^{(m)}$ . Denote  $\mathbf{q}_j^{(m,n)} = \mathbf{g}_j^{(m)} * \mathbf{q}^{(n)}$ , and expand the left-hand side of (9) by substituting (10) for  $\mathbf{y}^{(m)}$  to obtain

$$\begin{aligned} \mathbf{y}^{(m)} * \mathbf{q}^{(n)} &= \left( \sum_{j=1}^{\mathcal{D}} \mathbf{x}_j^{(m)} * \mathbf{g}_j^{(m)} \right) * \mathbf{q}^{(n)} \\ &= \sum_{j=1}^{\mathcal{D}} \mathbf{x}_j^{(m)} * \mathbf{q}_j^{(m,n)} \\ &= \mathcal{T}\{\mathbf{x}^{(m)}\} \mathbf{q}^{(m,n)} \end{aligned} \quad (11)$$

where

$$\mathbf{q}^{(m,n)} = \left[ \left( \mathbf{q}_1^{(m,n)} \right)^T \cdots \left( \mathbf{q}_{\mathcal{D}}^{(m,n)} \right)^T \right]^T.$$

We can now rewrite (9) as

$$\mathcal{T}\{\mathbf{x}^{(m)}\} \mathbf{q}^{(m,n)} = \mathcal{T}\{\mathbf{x}^{(n)}\} \mathbf{q}^{(n,m)}. \quad (12)$$

Let  $\mathcal{X}^{(m,n)} = [\mathcal{T}\{\mathbf{x}^{(m)}\} - \mathcal{T}\{\mathbf{x}^{(n)}\}]$ . Equation (12) implies that the vector  $[(\mathbf{q}^{(m,n)})^T (\mathbf{q}^{(n,m)})^T]^T$  is in the null space of  $\mathcal{X}^{(m,n)}$ .

If the null space of each  $\mathcal{X}^{(m,n)}$  is rank 1, then we can determine all the vectors  $\mathbf{q}^{(m,n)}$  up to a constant multiplicative factor. As we shall see, a necessary condition for a rank 1 null space is that the polynomials  $Q_j^{(m,n)}(z)$  have no common zeros over  $m, n$ . Assuming for the moment that this is indeed the case, we can write the polynomial  $Q_j^{(m,n)}(z)$  as  $Q_j^{(m,n)}(z) = G_j^{(m)}(z) Q_j^{(n)}(z)$ . In order to perform this factorization of  $Q^{(m,n)}(z)$ , we take  $G_j^{(m)}(z)$  as the common factor of the polynomials  $Q_j^{(m,1)}(z)$  through  $Q_j^{(m,M)}(z)$ . Similarly, we find  $Q_j^{(n)}(z)$  by finding the common factor of  $Q_1^{(1,n)}(z)$  through  $Q_{\mathcal{D}}^{(M,n)}(z)$ .

### B. Identification of Linear IIR Channels

The procedure developed here also applies to identification of linear IIR channels. The special case of a Volterra filter with  $P = 1$  and  $D_P = 1$  is a linear FIR filter with impulse response vector  $\mathbf{g}_1$ . Since Fig. 2 implies that  $H^{(m)}(z)G_1^{(m)}(z) = Q^{(m)}(z)$ , where  $Q^{(m)}(z)$  is also FIR, we conclude that

$$H^{(m)}(z) = \frac{Q^{(m)}(z)}{G_1^{(m)}(z)}. \quad (13)$$

That is, the zeros of  $G_1^{(m)}(z)$  are the poles of the channel, whereas the zeros of  $Q^{(m)}(z)$  are the zeros of the channel.

## IV. IDENTIFIABILITY CONDITIONS

### A. Existence of Finite-Order Polynomial Equalizers

One of the first questions regarding the approach presented in this paper is the following: What are the requirements on the channel operators  $H^{(m)}\{\cdot\}$  and the input  $s(k)$  so that equalizers  $G^{(m)}\{\cdot\}$  as specified in Assumption 2.1 exist?

An obvious requirement on the channels is the following assumption.

*Assumption 4.1:* The channels  $H^{(m)}\{\cdot\}$  are invertible over the range of the input  $s(k)$ .  $\square$

This assumption, however, does not assure that the equalizers can be implemented using finite-order, finite-memory polynomial operators. The inverses of  $H^{(m)}\{\cdot\}$  must be sufficiently smooth and have a bounded input range to be well approximated by a finite-order Volterra filter. For all channels considered here, the input  $s(k)$  may safely be assumed to be bounded.

The input to many digital communication channels is generated from a finite alphabet. In the absence of noise, the range of the output of a finite memory channel in response to a finite alphabet input is also a finite set. Hence, under Assumption 4.1, the equalizer can be implemented, *exactly*, using finite-order polynomials since it exists and is a mapping from one finite set (the range of the output of the channels) to another finite set (a finite linear combination of the input range: possibly the input itself). If the channels are not of finite memory but can be approximated to any desired accuracy as such, then the previous argument still holds within the given accuracy. Notice that in the case of finite alphabet input, no requirements, other than Assumption 4.1, are placed on the channels for the existence of a Volterra filter equalizer.

If the input is not generated from a finite alphabet, then conditions on the channels are required to assure proper smoothness of the inverses. Since we seek  $P$ th-order Volterra filter equalizers that are asymptotically better or equal in a least squares sense to  $P$ th-order inverses, it suffices to impose conditions that assure that  $P$ th order inverses exist. (Such conditions are discussed in [9].) The Volterra filter equalizers presented here are better than  $P$ th-order inverses since  $P$ th-order inverses merely assure that the equalized system has second- through  $P$ th-order kernels equal to zero; this does not mean, in general, that the remaining kernels of order  $P + 1$  and higher are limited in any way. On the other hand, the equalizers we seek are least squares optimal. Hence, conditions on the channels that guarantee good behavior of  $P$ th-order inverses may be overly restrictive for our purposes.

Notice that since, in practice, we consider only a finite length of input and output streams, we can always assume finite alphabet input. However, the whole discussion in this subsection is merely to establish the existence of equalizers. If we were to actually implement exact Volterra equalizers based on this argument, they would likely be of a very high polynomial order and long memory. For this reason, we seek least squares optimal equalizers of limited memory and polynomial order.

### B. Uniqueness

The results given here are an extension of those in [15].

*Theorem 4.1:* The multichannel system  $H^{(m)}\{\cdot\}$  for  $M \geq 2$  can be uniquely linearized and identified by solving the equations

$$\mathcal{X}^{(m,n)} \left[ \left( \mathbf{q}^{(m,n)} \right)^T \left( \mathbf{q}^{(n,m)} \right)^T \right]^T = 0 \quad (1 \leq m \leq n \leq M) \quad (14)$$

iff we have the following.

- 1) Assumption 2.1 holds.
- 2) The matrices  $\mathcal{X}^{(m,n)}$  are rank  $(\sum_{j=1}^{\mathcal{D}} (L_j + 1)) - 1$  (i.e., the  $\mathcal{X}^{(m,n)}$  have rank 1 null space).
- 3) There is a subset of  $G_j^{(m)}(z)$  of size greater or equal to 2 that is coprime (or have no common roots) with respect to  $j$  or  $m$ .  $\square$

*Proof:* By construction,  $[(\mathbf{q}^{(m,n)})^T (\mathbf{q}^{(n,m)})^T]^T$  is in the null space of  $\mathcal{X}^{(m,n)}$ . The rank condition assures uniqueness up to a common multiplicative constant. Hence, the only problem left is that of uniquely factoring  $Q_j^{(m,n)}(z)$  in the form of the product  $Q_j^{(m,n)}(z) = G_j^{(m)}(z)Q^{(n)}(z)$ . A unique factorization is guaranteed by the third condition. The common roots of  $Q_j^{(m,n)}(z)$  corresponding to  $m$  and  $j$  of the subset in condition 3) are exactly the roots of  $Q^{(n)}(z)$ .  $\square$

Note that the ability to uniquely factorize  $Q_j^{(m,n)}(z)$  stems from both the redundancy introduced by having multiple channels, which is represented by  $m$ , and the redundancy introduced by the DCS representation of the equalizer, which is represented by  $p$  and  $i$ . The latter represents a new form of redundancy in the context of blind equalization. Clearly, in the special case of blind linear IIR channel identification, the redundancy is introduced solely by the multiple channel construction.

Theorem 4.1 gives the general conditions for unique channel and equalizer identifiability. More explicit theorems, in the spirit of [15], are needed to provide further insight into the requirements on the channels and the input signal. One such theorem is stated next without proof. The proof is similar in spirit to that for the equivalent theorem in [15].

*Theorem 4.2:* The channels cannot be uniquely identified if, given  $j$ , there is a common root shared by  $\mathbf{q}_j^{(m,n)}$  for  $1 \leq m, n \leq M$ .  $\square$

This implies that a necessary condition for unique identifiability is that there be no common root for  $\mathbf{q}^{(m)}$  over  $1 \leq m \leq M$ . Moreover, Theorem 4.2 in conjunction with the third condition of Theorem 4.1 implies that no common roots may exist for  $\mathbf{g}_j^{(m)}$  over  $1 \leq m \leq M$ .

Note that while the conditions for identifiability are stated in terms of the  $\mathbf{g}_j^{(m)}$  and  $\mathbf{q}^{(m)}$ , these actually represent conditions on the nonlinear channels  $H^{(m)}\{\cdot\}$ . This follows from the fact that  $\mathbf{g}_j^{(m)}$  and  $\mathbf{q}^{(m)}$  define the inverse of the channel operators  $H^{(m)}\{\cdot\}$ .

## V. OTHER EQUALIZER REPRESENTATIONS

### A. Tensor Product (TP) Representation

Our choice of using the DCS representation of a Volterra filter for the equalizer is by no means a necessary or optimal

choice. Other linear combinations of nonlinear functions could serve the same purpose. One example is the tensor product (TP) implementation of the Volterra filter. Let  $\bar{\mathbf{x}}(l) = [x(l) \ x(l-1) \ \cdots \ x(l-L) \ 1]$  be the input to the Volterra filter augmented by the constant 1. Define

$$\mathbf{z}(l) = \underbrace{\bar{\mathbf{x}}(l) \otimes \cdots \otimes \bar{\mathbf{x}}(l)}_{P \text{ times}}.$$

Thus,  $\mathbf{z}(l)$  contains all the products of the elements of the input up to order  $P$ . Define the corresponding vector  $\mathbf{f}$  from the elements of the Volterra kernels  $G_p(l_1, \dots, l_p)$  ( $p = 1, \dots, P$ ). We can now rewrite (2) as  $y(l) = \mathbf{z}(l)^T \mathbf{f}$ . Using this notation, we write a derivation similar to the one in Section III.

$$\begin{aligned} \mathbf{y}^{(m)} &= \mathbf{T}_{L, \mathbf{q}^{(m)}} \{ \mathbf{s} \} \mathbf{q}^{(m)} \\ &= \mathbf{Z}^{(m)} \mathbf{f}^{(m)} \end{aligned} \quad (15)$$

where  $\mathbf{Z}^{(m)} = [\mathbf{z}^{(m)}(0) \ \cdots \ \mathbf{z}^{(m)}(N)]^T$ . Let  $\mathbf{z}_i^{(m)}$  be the  $i$ th column of the matrix  $\mathbf{Z}^{(m)}$ . Hence

$$\begin{aligned} \mathbf{Z}^{(m)} \mathbf{f}^{(m)} * \mathbf{q}^{(n)} &= \sum_i \mathbf{z}_i^{(m)} f^{(m)}(i) * \mathbf{q}^{(n)} \\ &= \bar{\mathbf{T}} \{ \mathbf{x}^{(m)} \} (\mathbf{f}^{(m)} \otimes \mathbf{q}^{(n)}) \end{aligned} \quad (16)$$

where  $\bar{\mathbf{T}} \{ \mathbf{x}^{(m)} \} = [\cdots \mathbf{T} \mathbf{q} \{ \mathbf{z}_{i-1}^{(m)} \} \mathbf{T} \mathbf{q} \{ \mathbf{z}_i^{(m)} \} \mathbf{T} \mathbf{q} \{ \mathbf{z}_{i+1}^{(m)} \} \cdots]$ . We use the overbar notation  $\bar{\mathbf{T}} \{ \mathbf{x}^{(m)} \}$  to signify that in general, the number of columns of  $\mathbf{Z}^{(m)}$  indexed by  $i$  is not equal to the number of diagonals  $\mathcal{D}$  we get when using the DCS representation. It is, in fact, greater than or equal to  $\mathcal{D}$ . Using the general identification equation (9), we find that the vector  $[(\mathbf{f}^{(m)} \otimes \mathbf{q}^{(n)})^T : (\mathbf{f}^{(n)} \otimes \mathbf{q}^{(m)})^T]^T$  is in the null space of  $\bar{\mathcal{X}}^{(m,n)} = [\bar{\mathbf{T}} \{ \mathbf{x}^{(m)} \} : -\bar{\mathbf{T}} \{ \mathbf{x}^{(n)} \}]$ , and hence, the equalizer coefficients  $\mathbf{f}^{(m)}$  and the linearized channel coefficients  $\mathbf{q}^{(m)}$  can be found under the proper uniqueness conditions.

Note that in the derivation above, we did not take advantage of the redundancy introduced by the symmetry of the Volterra kernels for sake of simplicity. We may easily remove the repeated entries in  $\mathbf{f}$  and the corresponding entries in  $\mathbf{Z}$ . However, the matrix  $\bar{\mathcal{X}}^{(m,n)}$  is potentially much larger than  $\mathcal{X}^{(m,n)}$  in Section III, even after accounting for the kernel redundancy. This is due to the fact that the DCS representation uses the convolution operator rather than the inner product used in the TP implementation, and hence, the matrices used there are already in Toeplitz form. Nonetheless, the above derivation may be useful in the special case when the channel outputs  $\mathbf{x}^{(m)}$  are known *a priori* to lie in a low-rank linear space. This may occur, for example, if the receiving end of the channel is of known limited bandwidth. In this case, we may reduce the complexity of the equalization by using the tensor product basis approximation [6]. The tensor product basis approximation reduces the dimension of both  $\mathbf{Z}^{(m)}(l)$  and  $\mathbf{f}^{(m)}$ . Similarly, we may implement a low-rank approximation for the equalizer using the diagonal coordinate system approximation [13] when the input of the channels is known to lie in a low-rank subspace since the outputs of the equalizers are in the same linear space as the input to the channels.

### B. Orthogonal Representations

We may also represent the nonlinear channels using series of orthogonal functions [1], [2], [4], [5], [14], [16] and derive equalizers based on orthogonal inverses [12]. Throughout the literature, these orthogonal functions are usually polynomials. However, since we are using least squares methods, the equalizers we derive would not be changed by the choice of expansion, except possibly due to numerical stability effects. Furthermore, orthogonal expansions are usually derived based on the statistical behavior of the input sequence. In this paper, we use a deterministic approach, that is, we do not assume an *a priori* statistical model for the input. Hence, use of orthogonal expansions would be more appropriate with statistical methods for blind identification and equalization of nonlinear systems.

### C. Nonpolynomial Representations

The algorithm derived here is applicable to nonpolynomial equalizers represented as a linear combination of predetermined nonlinear functions of the channel outputs. That is, the equalizer operator is of the form

$$\overline{G}^{(m)}\{\mathbf{x}^{(m)}\} = \sum_{j \in \text{some finite set}} \overline{\mathbf{g}}_j^{(m)} * v_j(\mathbf{x}^{(m)}) \quad (17)$$

where  $v_j(\mathbf{x}^{(m)})$  is a general, possibly nonpolynomial, nonlinear function of  $\mathbf{x}^{(m)}$ . The DCS representation corresponds to the special case where  $v_j(\mathbf{x}^{(m)}) = \mathbf{x}_j^{(m)}$ . The algorithm presented in previous sections is employed by substituting  $v_j(\mathbf{x}^{(m)})$  for  $\mathbf{x}_j^{(m)}$  to obtain the linear combinations  $\overline{\mathbf{g}}_j^{(m)}$  and the linear filters  $\mathbf{q}^{(m)}$ , formed by the cascade connection of the channels with the equalizers. In Section VI, a very simple example is given using such a nonpolynomial approach.

There are two reasons for using specialized nonpolynomial equalizers. One is that polynomial equalizers may simply not produce good results with a given finite polynomial order. Another reason is reduction of the computation complexity involved in determining the equalizer coefficients. While nonpolynomial nonlinearities may have a higher numerical overhead for calculating the  $\mathcal{T}$  matrices, this may be offset by reduction in the size of the  $\mathcal{T}$  matrices and the consequent computational complexity reduction in the singular value decomposition (SVD) used to find the null space. We note, however, that FIR Volterra filters can approximate arbitrarily well any bounded equalizer function with bounded input. Other classes of nonlinear functions are not guaranteed, in general, to have that quality.

## VI. NUMERICAL RESULTS

### A. Example 1: Nonlinear Channels with Exact Polynomial Inverses

To illustrate the proposed algorithm, we solve a simple example of nonlinear channel linearization and identification. Throughout this example, we use zero and pole information to describe linear filters since the linearization and identification are accurate up to a common multiplicative factor.

Assume there are three channels of the form depicted in Fig. 3. The zeros, poles, and nonlinearity for each channel are specified in Table I. The input sequence represents equiprobable symbols from the alphabet  $[-5, -3, -1, 1, 3, 5]$ .

It is obvious by inspection that there exists a Volterra filter of polynomial order 3 and memory 2 that serves as an exact linearizer. Specifically, the Volterra filter is constructed by the cascade connection of the memoryless nonlinearity  $f(u) = u + u^3$  followed by a linear filter of memory 2, whose zeros are equal to the poles in the channel. Hence, there are two vectors  $\mathbf{g}_j^{(m)}$  per equalizer (and, hence, per channel). Let  $j = 1$  correspond to the linear part of the equalizer (i.e.,  $p = 1$ ), and  $j = 2$  corresponds to the cubic term ( $p = 3$ ). Since the channel nonlinearity is memoryless, we have only one diagonal per polynomial order, that is,  $i = 1$  in (6) for all  $p$ . Indeed, when implementing the identification algorithm for this example with  $L_{\mathbf{q}_j^{(m)}} = 3$  for all  $j$ , we find that the zeros of the filters  $\mathbf{q}_j^{(m,n)}$  correspond to the expected zeros and poles of the “unknown” channel, as summarized in Table II. An input of less than 20 symbols is more than sufficient to identify the channels in this noiseless case. By appropriately grouping the zeros in Table II, we determine the DCS filters  $\mathbf{g}_j^{(m)}$  of the linearizer and the linear FIR filters  $\mathbf{q}^{(n)}$  of the linearized channels. For example, the only zeros common to rows corresponding to  $m = 2$ ,  $p = 3$ , and  $i = 1$  (or  $j = 2$ ) are  $-0.25 \pm 0.3i$ . These are the zeros of  $G_2^{(2)}(z)$ . We find that in this case,  $G_1^{(2)}(z) = G_2^{(2)}(z)$ , and indeed

$$X_1^{(2)}(z)G_1^{(2)}(z) + X_2^{(2)}(z)G_2^{(2)}(z) = S(z)Q^{(2)}(z) \quad (18)$$

where  $Q^{(2)}(z)$  is the first-order polynomial with zero at 0.4. We notice that  $Q^{(2)}(z)$  corresponds to the unique common zero in all the rows where  $n = 2$ , as expected.

### B. Example 2: Linear IIR Channels

In this example, we identify the poles and zeros of linear IIR channels as depicted in Fig. 4. The channel poles and zeros are the same as in Example 1, and the channel outputs are corrupted by a white noise with a signal-to-noise ratio (SNR) of 50 dB. We note that since the channels are linear, there is only one diagonal per channel, that is,  $\mathcal{D} = 1$  for all three channels. Using 200 equiprobable symbols from the alphabet  $[-5, -3, -1, 1, 3, 5]$ , we identify the zeros of the filters  $\mathbf{q}_1^{(m,n)}$ . The results are given in Table III.

The next step in the algorithm is to determine, for each row in the table, which of the zeros correspond to the poles of channel  $m$  and which to the poles of channel  $n$ . Unlike the previous example, however, corresponding zeros are not equal due to the presence of noise. For instance, when comparing the first row, which corresponds to  $m = 1$  and  $n = 2$ , with the third row, which corresponds to  $m = 1$  and  $n = 3$ , we note that no zeros match exactly. Therefore, a threshold level is set for the distance between zero locations. Any two zero locations that are closer than the threshold are considered equal. In this case, a threshold level of 0.005 is sufficient.

The threshold level also serves as an indicator of the accuracy of the identification algorithm. If no threshold that suffices to

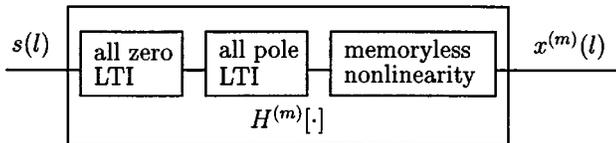


Fig. 3. Block diagram of unknown nonlinear channels.

 TABLE I  
 CHANNEL PARAMETERS  
 IN EXAMPLE 1

$m$	zeros	poles	nonlinearity
1	0.5	$-0.25 \pm 0.37i$	Inverse of operator $(\cdot) + (\cdot)^3$
2	0.4	$-0.25 \pm 0.3i$	
3	0.35	$-0.24 \pm 0.25i$	

 TABLE II  
 ZEROS OF  $Q_{p,i}^{(m,n)}(z)$  FOR EXAMPLE 1

$m$	$n$	$p$	$i$	zeros		
1	2	1	1	0.4	$-0.25 - 0.37i$	$-0.25 + 0.37i$
1	2	3	1	0.4	$-0.25 - 0.37i$	$-0.25 + 0.37i$
2	1	1	1	0.5	$-0.25 - 0.3i$	$-0.25 + 0.3i$
2	1	3	1	0.5	$-0.25 - 0.3i$	$-0.25 + 0.3i$
1	3	1	1	0.35	$-0.25 - 0.37i$	$-0.25 + 0.37i$
1	3	3	1	0.35	$-0.25 - 0.37i$	$-0.25 + 0.37i$
3	1	1	1	0.5	$-0.24 - 0.25i$	$-0.24 + 0.25i$
3	1	3	1	0.5	$-0.24 - 0.25i$	$-0.24 + 0.25i$
2	3	1	1	0.35	$-0.25 - 0.3i$	$-0.25 + 0.3i$
2	3	3	1	0.35	$-0.25 - 0.3i$	$-0.25 + 0.3i$
3	2	1	1	0.4	$-0.24 - 0.25i$	$-0.24 + 0.25i$
3	2	3	1	0.4	$-0.24 - 0.25i$	$-0.24 + 0.25i$

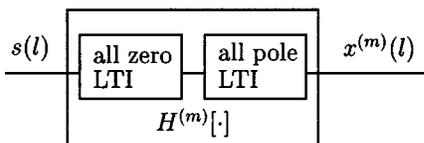


Fig. 4. Block diagram of unknown linear IIR channels.

 TABLE III  
 ZEROS OF  $Q_1^{(m,n)}(z)$  FOR EXAMPLE 2

$m$	$n$	zeros		
1	2	0.405	$-0.245 - 0.369i$	$-0.245 + 0.369i$
2	1	0.503	$-0.245 - 0.297i$	$-0.245 + 0.297i$
1	3	0.355	$-0.249 - 0.370i$	$-0.249 + 0.370i$
3	1	0.504	$-0.239 - 0.250i$	$-0.239 + 0.250i$
2	3	0.365	$-0.250 - 0.303i$	$-0.250 + 0.303i$
3	2	0.414	$-0.240 - 0.254i$	$-0.240 + 0.254i$

### C. Example 3: Nonlinear Channels without Exact Polynomial Inverses in the Presence of Noise

To illustrate the robustness of the algorithm, we add noise and consider nonlinear channels that cannot be exactly equalized by a finite-order Volterra filter. The channel poles and zeros are the same as in Example 1, and the memoryless nonlinearity is  $w(u) = u - 0.1u^3$ . By inspection, it is clear that the inverse operator has infinite polynomial order. However, we will approximately equalize the channels using third-order Volterra filters, as in Example 1. The channel outputs are further corrupted by independent white Gaussian noises that are independent of the input. The SNR is 50 dB. In this example, the input symbols are complex and equiprobable from the alphabet  $[-5, -3, -1, 1, 3, 5] \times i[-5, -3, -1, 1, 3, 5]$ .

In order to factorize  $Q_j^{(m,n)}(z)$ , we first round the estimated roots to the second decimal place and then search for the common roots.

The results of equalization using 200 input symbols are shown in Fig. 5. We also show the results of equalization using a standard  $P$ th-order inverse, where  $P = 3$ . In order to find the third-order inverse, the nonlinear channels must be assumed to be known, and thus, the third-order equalizer is not blind. Note that the least squares blind third-order equalizer performs almost identically to the nonblind third-order inverse. If we project each equalized symbol to the closest symbol in the alphabet, then the equalized symbols exactly match the input symbols.

### D. Example 4: Nonlinear Channels with Discontinuities

This example illustrates a nonlinearity that consists of discontinuity in the function and its derivative. The input consists of 200 equiprobable real symbols chosen from the alphabet  $[-1, -0.6, -0.2, 0.2, 0.6, 1]$ . The channels are again similar to those in Example 1, except that the nonlinearity is now of the form

$$w(u) = \begin{cases} u, & |u| \leq M/2 \\ \text{sign}(u)(2|u| - 0.95M/2), & |u| > M/2 \end{cases} \quad (19)$$

where  $M = \max(|u|)$ .

separate the zeros from the poles can be found, then a longer input sequence may be required due to the noise level.

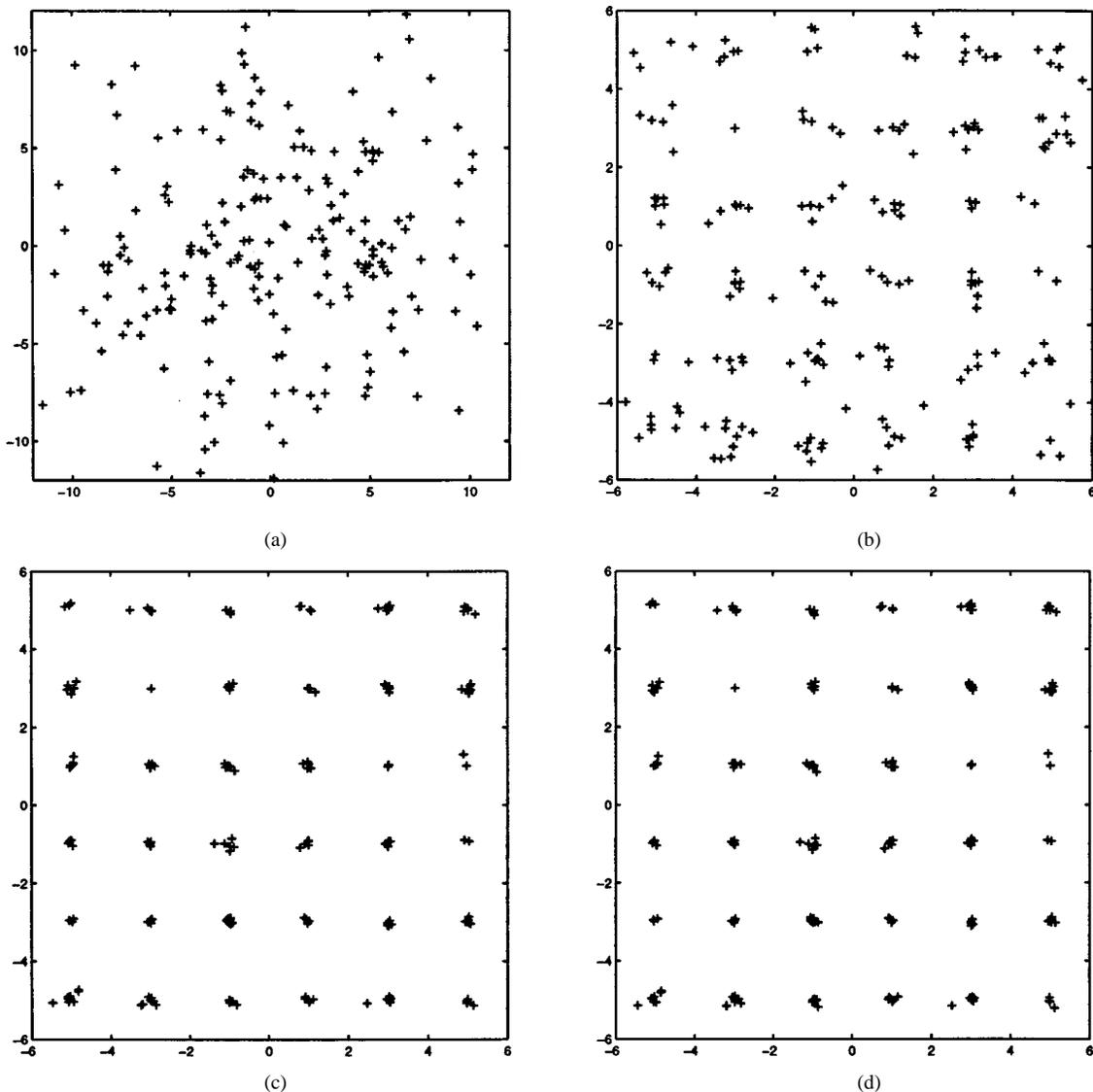


Fig. 5. Constellations of output for Example 3. (a) Before equalization. (b) After linear equalization. (c) After third-order equalization. (d) After  $P$ th-order inverse with  $P = 3$ . (Not blind).

Notice that although this nonlinearity is invertible, it is not a good candidate for an equalization approach based on the  $P$ th-order inverse because of the discontinuities. However, satisfactory equalization results are obtained with a polynomial least squares equalizer. The results of the equalization are depicted in Fig. 6. We show the eye diagrams of the unequalized output and linear, third-order, and fifth-order equalized output. After projecting the 200 output symbols onto the nearest symbol in the alphabet, we find the following.

- The unequalized output has 121 errors.
- The linearly equalized output has 31 errors.
- The cubic equalizer yields 19 errors.
- The fifth-order equalizer produces no errors.

#### E. Example 5: Nonpolynomial Equalizers

Here, we demonstrate use of a nonpolynomial equalizer. The input consists of 200 equiprobable real symbols chosen from the

alphabet  $[-1, -0.6, -0.2, 0.2, 0.6, 1]$ . The channels are once again similar to those in Example 1, except that the nonlinearity is of the form  $w(u) = \exp(u)$ . This nonlinearity is contrived; however, it serves the purpose of showing the utilization of a nonpolynomial equalizer based on *a priori* knowledge of the type of nonlinearities corrupting the data.

We compare the equalization results using polynomial equalizers with a logarithmic equalizer. The logarithmic approach is implemented by replacing  $\mathcal{T}\{\mathbf{x}^{(m)}\}$  as per (12) with  $\mathbf{T}_{L_q}\{\log(\mathbf{x}^{(m)})\}$ . That is, the equalizer has the form  $\overline{\mathbf{G}}^{(m)}\{\mathbf{x}^{(m)}\} = \overline{\mathbf{g}}^{(m)} * \log(\mathbf{x}^{(m)})$ . By using a logarithmic equalizer for this simple example, the problem degenerates into a linear channel with poles and zeros.

In Fig. 7, we depict the eye diagrams of the unequalized channel; eye diagrams of equalized channels using linear, quadratic, and cubic polynomial equalizers; and the eye diagram of equalized channels using a logarithmic equalizer. It is apparent that the polynomial equalizers are ill suited for

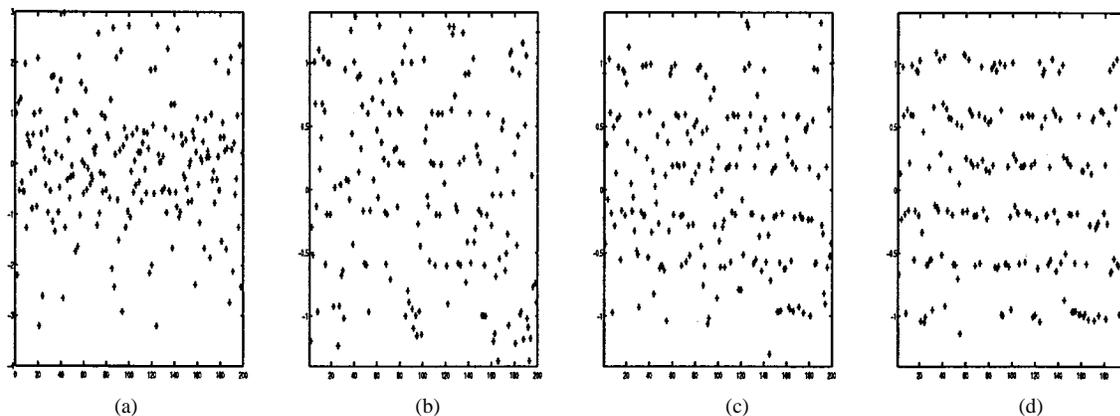


Fig. 6. Eye diagram of output for Example 4: Sample number versus sample value. (a) Before equalization. (b) After linear equalization. (c) After third-order equalization. (d) After fifth-order equalization.

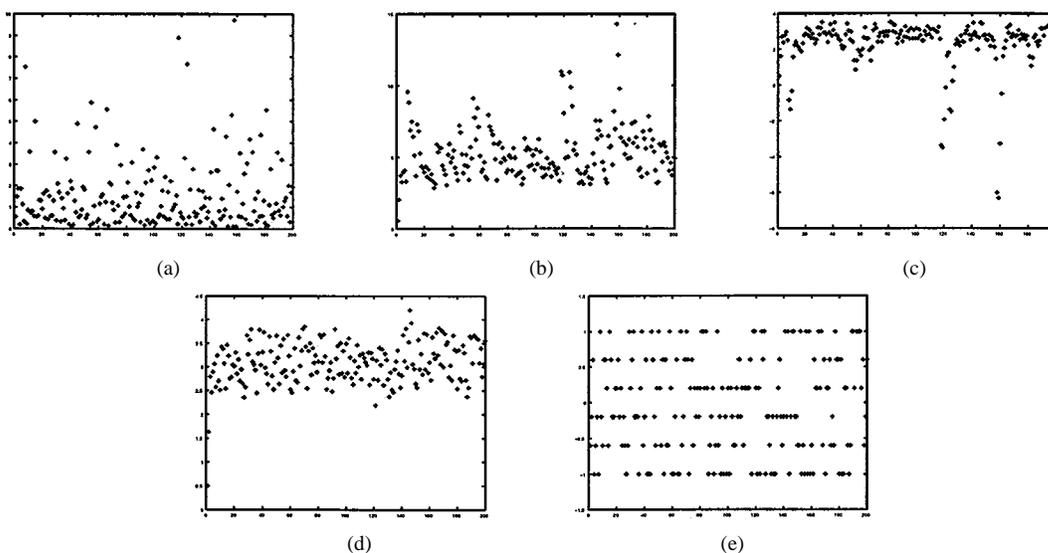


Fig. 7. Eye diagram of output for Example 5. Sample number versus sample value. (a) Before equalization. (b) After linear equalization. (c) After second-order equalization. (d) After third-order equalization. (e) After logarithmic equalization.

the task, whereas the logarithmic equalizer provides perfect equalization.

The two primary reasons for using specialized nonpolynomial equalizers are illustrated in this example. The first and most obvious reason is that the polynomial equalizers do not produce reasonable results. The second reason is computation complexity reduction. The logarithmic nonlinearity has a higher computational overhead initially, that is, calculating a logarithm of the channel outputs is more costly than their multiplication. However, the size of the  $\mathcal{X}$  matrices is significantly lower, and hence, the computational complexity of the SVD used to find the null space is lower. In this example, the third-order polynomial equalizer had a  $\mathcal{X}$  matrix with dimensions  $200 \times 24$ , whereas the logarithmic equalizer had a  $\mathcal{X}$  matrix with dimensions  $200 \times 8$ .

## VII. SUMMARY

A new method for blind linearization and identification of a very general class of nonlinear systems is presented. This method is based on the least squares identification of FIR Volterra filters that linearize the channels and consequent

identification of the impulse response of the cascade connection of the nonlinear channels and equalizers. As a special case, we show how to blindly identify IIR channels. General conditions for identifiability are given. Several generalizations of the equalizer representations are discussed, including the tensor product implementation, orthogonal, and nonpolynomial equalizers. Several numerical simulations are presented to illustrate the effectiveness of the algorithm presented in the paper. Further work is needed to establish the effect of noise on the accuracy of the algorithm. Possible extensions of this work include adaptation of the algorithm to systems with inputs of known statistics. In this case, orthogonal representations of the unknown channels and the equalizer are likely to be of value.

## REFERENCES

- [1] N. M. Blachman, "The signal X signal, noise X noise and signal X noise output of a nonlinearity," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 21–27, Jan. 1968.
- [2] —, "The uncorrelated output components of a nonlinearity," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 250–255, Feb. 1968.
- [3] G. B. Giannakis and E. Serpedin, "Linear multichannel blind equalizers of nonlinear FIR volterra channels," *IEEE Trans. Signal Processing*, vol. 45, pp. 67–81, Jan. 1997.

- [4] K. I. Kim and E. J. Powers, "A digital method of modeling quadratically nonlinear systems with a general input," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1785–1769, Nov. 1988.
- [5] M. Korenburg, S. Bruder, and P. McIlroy, "Exact orthogonal kernel estimation from finite data records: Extending Wiener's identification of nonlinear systems," *Ann. Biomed. Eng.*, vol. 6, pp. 201–214, 1988.
- [6] R. D. Nowak and B. D. Van Veen, "Tensor product basis approximation for volterra filters," *IEEE Trans. Signal Processing*, vol. 44, pp. 36–50, Jan. 1996.
- [7] S. Prakriya and D. Hatzinakos, "Blind identification of LTI-ZMNL-LTI nonlinear channel models," *IEEE Trans. Signal Processing*, vol. 43, pp. 3007–3013, Dec. 1995.
- [8] G. M. Raz and B. D. Van Veen, "Baseband volterra filters for implementing carrier based nonlinearities," *IEEE Trans. Signal Processing*, vol. 46, pp. 103–115, Jan. 1998.
- [9] M. Schetzen, "Theory of  $p$ th-order inverses of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 285–291, 1976.
- [10] —, *The Volterra and Wiener Theories of Nonlinear Systems*, New York: Wiley, 1980.
- [11] M. K. Tsatsanis and H. A. Cirpan, "Blind identification of nonlinear channels excited by discrete alphabet inputs," in *Proc. SSAP Eighth IEEE Signal Process. Workshop Stat. Signal Array Process.*, Jun. 1996, pp. 176–179.
- [12] J. Tsimbinos and K. V. Lever, "Nonlinear compensation using orthogonal inverses: Broadband inputs and robustness issues," in *Proc. IEEE Reg. 10 Conf. Digital Signal Process. Appl.*, Perth, Australia, Nov. 27–29, 1996, pp. 816–821.
- [13] B. D. Van Veen, R. D. Nowak, and G. M. Raz, "Diagonal coordinate system approximations for volterra filters," *Signal Process.*, to be published.
- [14] N. Wiener, *Nonlinear Problems in Random Theory*, New York: Wiley, 1958.
- [15] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Processing*, vol. 43, Dec. 1995.
- [16] A. M. Zoubir, "Identification of quadratic volterra systems driven by non Gaussian processes," *IEEE Trans. Signal Processing*, vol. 43, pp. 1302–1306, May 1995.

**Gil M. Raz** received the B. S. degree (with honors) in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 1989 and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin, Madison, in 1998.

He served as a Project Manager in a research and development unit in the Israeli military from 1989 to 1993. From 1993 to 1994, he worked as a Software Engineer at National Semiconductors (NSTA). He is currently a Staff Member at Lincoln Laboratory, Massachusetts Institute of Technology, Lexington.

**Barry D. Van Veen** (S'81–M'86–SM'97) was born in Green Bay, WI. He received the B.S. degree from Michigan Technological University, Houghton, in 1983 and the Ph.D. degree from the University of Colorado, Boulder, in 1986, both in electrical engineering.

In the spring of 1987, he was with the Department of Electrical and Computer Engineering, University of Colorado. Since August 1987, he has been with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, and currently holds the rank of Professor. His research interests include filtering, wireless communications, and biomedical applications of signal processing.

Dr. Van Veen was an ONR Fellow while pursuing the Ph.D. degree. He was a recipient of a 1989 Presidential Young Investigator Award from the National Science Foundation and a 1990 IEEE Signal Processing Society Paper Award. He is currently as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and served on the IEEE Signal Processing Society's Technical Committee on Statistical Signal and Array Processing from 1991 to 1997. He received the Holdridge Teaching Excellence Award from the Department of Electrical and Computer Engineering from the University of Wisconsin in 1997.