SUBSPACE BASED BLIND IDENTIFICATION AND EQUALIZATION OF LINEAR FIR MIMO SYSTEMS AND FIR SIMO VOLterra SYSTEMS

D. Kotoulas, P. Koukoulos, N. Kalouptsidis

Department of Informatics and Telecommunications, Communications and Signal Processing Division, University of Athens, Panepistimiopolis, 157 84 Ilissia, Athens, GREECE.
phone: +30 210 6111113, fax: +30 210 6118214, email: dkotoula@otenet.gr

ABSTRACT

In this paper a new algorithm based on subspace projections is developed for the blind equalization and kernel identification of linear FIR Multiple Input Multiple Output (MIMO) as well as nonlinear FIR SIMO Volterra systems. Simulations in the context of blind channel equalization show good performance in comparison to existing schemes.

1. INTRODUCTION

Blind methods are of great importance in digital signal communication systems as they allow channel identification/equalization at the receiver without the use of training signals. The topic of blind identification/equalization of linear time invariant (LTI) channels, both SIMO and MIMO, has drawn considerable attention over the past years and several algorithms have been developed (see [1, 2, 3] and references there in). Initially, focus on this area was centered in linear FIR SIMO systems as they allow channel identification/equalization at the receiver without the use of training signals. The topic of blind channel equalization show good performance in comparison to existing schemes.

Finally, the proposed algorithm can be used to compute equalizers for all possible delays. Compared to the algorithm of [7] the proposed method does not impose the positive definiteness condition on the input covariance matrix.

With regard to linear FIR MIMO systems, the main advantage of the proposed method is that it does not require a-priori knowledge of the channel lengths. As already mentioned, these parameters are internally estimated by the algorithm.

2. PREREQUISITES

The notation employed in this paper is standard. Signals are discrete-time and complex in general. Upper- and lower-case bold letters denote matrices and vectors respectively. (·)′ and (·)̃ are transpose and Hermitian operations. \( O_{m \times n} \) stands for the \( m \times n \) zero matrix. If \( x, X \) are a vector and a matrix respectively and \( A \) is a vector space, then \( x_M \) is the orthogonal projection of \( x \) onto \( A \) and \( X_A \) is the orthogonal projection of the rows of \( X \) onto \( A \). We shall deploy Blind Source Separation (BSS) techniques [9, 10, 11, 14, 15, 16, 17]. Specifically, we assume that we are given at the receiver some linear mixtures \( x_i(n) \) of a number of source signals \( s_j(t) \) that obey equation (1):

\[
x_i(n) = \sum_{j=1}^{p} a_{ij} s_j(n)
\]

or in matrix form,

\[
X = AS
\]

Matrix \( X \) has \( Q \) columns \( x(1), x(2), \ldots, x(Q) \). It consists of successive output vector observations, while the \( n \) column of \( X \) is \( x(n) = [x_1(n), \ldots, x_M(n)] \). \( A \) has elements \( a_{ij} \) and the matrix of source signals \( S \) has column vectors \( s(n) = [s_1(n), \ldots, s_P(n)] \). The goal of BSS is to recover \( S \) from the observed mixtures \( X \). Specifically, we seek a matrix \( V \), such that \( VA = PD \). As a result, the estimated signal \( U = VX \) will resemble the original source signals \( S \) up to multiplication with a diagonal matrix \( P \) and a permutation matrix \( P' \):

\[
U = VX = PDS
\]

Two popular approaches for the solution of the BSS problem are the Independent Component Analysis (ICA) and the (Approximate) Joint Diagonalization (JAD) [9, 10, 11]. In the proposed method, we consider the JADE algorithm [17] and the Extended Matrix Pencil algorithm [9, 10].

3. MODEL SPECIFICATION

3.1 The Volterra Case

We assume that our system is described by a discrete SIMO Volterra system, produced by oversampling a continuous SISO Volterra system at a rate \( M \). The Volterra system under consideration, could be
either passband or baseband. The passband system is described by the equation:

\[ y(n) = \sum_{k=1}^{p} \sum_{l_k=0}^{L_k} h_k(l_1, \cdots, l_k)s(n-l_1)\cdots s(n-l_k) \]

while, the baseband system is described by the equation:

\[ y(n) = \sum_{k=1}^{p} \sum_{l_k=0}^{L_k+1} h_{2k+1}(l_1, \cdots, l_{2k+1})s(n-l_1)\cdots s(n-l_{2k+1}) \]

\[ s^*(n-l_{2k+2})\cdots s^*(n-l_{2k+1}) \]

In the above equations \( y(n) \) denotes the \( M \) dimensional system output and \( s(n) \) denotes the system input. \( P \) specifies the order of nonlinearity while \( L_k \) denotes the memory length of each subsystem, involved in the model. The output signal \( y(n) \) is an \( M \times 1 \) dimensional vector. The following assumptions are made:

A1) The input sequence \( s(n) \) is i.i.d., zero mean and totally circular [18]. Furthermore, the input symbols are drawn from a discrete alphabet, following a uniform probability distribution. PKS falls in this category.

A2) The input sequence \( s(n) \), takes at least \( P + 1 \) distinct non-zero values. Such assumption is satisfied by PSK and QAM inputs of order greater than \( P + 1 \).

By appropriately redefining the inputs, the above systems can take the form

\[ y(k) = \sum_{i=1}^{B} \sum_{j=0}^{D_i} G_i(j)u_i(k-j) \quad (4) \]

or, equivalently,

\[ y(k) = |G(z)|u(k) \quad (5) \]

where

\[ G(z) = \sum_{i=0}^{D_i} G_i z^{-i} \quad (6) \]

is the system transfer function and \( u(k) = [u_1(k) \cdots u_B(k)]^T \). The orders of the \( B \) subsystems are given by the integers \( D_1, D_2, \cdots, D_B \). If \( D_1, D_2, \cdots, D_B \) are grouped into \( r \) distinct numbers \( J_1, J_2, \cdots, J_r \) such that \( J_1 < J_2 < \cdots < J_r \), then for all \( i, 1 \leq i \leq r \), we denote by \( m_i \) the number of subsystems that have order \( J_i \).

A3) \( M > B \). This oversampling assumption, limits the practical applicability to Volterra systems of relatively low order \( P \leq 5 \). \( B \) may also increase, depending on the initial channel lengths.

A4) \( G(z) \) is assumed to be irreducible and column reduced.

3.2 The FIR MIMO Case

We assume that the system is described by (4), where the number of inputs is \( B \), while the number of output channels is \( M \). The assumptions for the linear FIR MIMO case are the following:

B1) The input sequences are zero mean and i.i.d.

B2) \( M > B \).

B3) \( G(z) \) is assumed to be irreducible and column reduced.

4. THE PROPOSED ALGORITHM

In this section we describe the proposed algorithm. Matrix notations are the same as the ones defined in [8]. A detailed account is given in [19].

The steps taken by the algorithm are the following:

- Apply the method described in [8] to identify the smallest channel length \( J_1 \) and the number \( n_1 \) of the subsystems that attain it. The data projection matrix \( \mathbf{E}_{k,J_1} \), the kernel matrix \( \mathbf{G}_{J_1}^{\mathbf{F}_1} \), and the input sources matrix \( \hat{\mathbf{U}}_{k,0} \) satisfy the equation:

\[ \mathbf{E}_{k,J_1} = \mathbf{G}_{J_1}^{\mathbf{F}_1}(\mathbf{U}_{k,0}^{1}\hat{\mathbf{U}}_{k,0}^{1}) \quad (7) \]

- Apply the JADE algorithm or the Extended Matrix Pencil Algorithm to blindly separate the sources \( u_{1}, u_{2}, \cdots, u_{n_1} \) belonging to subsystems of order \( J_1 \) and identify the corresponding kernels, as columns of the matrix \( \mathbf{G}_{J_1}^{\mathbf{F}_1} \).

- Having equalized the first group of sources, use the method defined in [8] to identify the next channel length \( J_2 \) and the corresponding number of subsystems \( m_2 \). In this case, the data projection matrix \( \mathbf{E}_{k,J_2} \), the corresponding kernel matrix \( \mathbf{G}_{J_2}^{\mathbf{F}_2} \) and the input sources matrix \( \hat{\mathbf{U}}_{k,0} \) satisfy the equation:

\[ \mathbf{E}_{k,J_2} - \mathbf{G}_{J_2}^{\mathbf{F}_2}(\mathbf{U}_{k,J_2-1}\hat{\mathbf{U}}_{k,J_2-1}) = \mathbf{G}_{J_2}^{\mathbf{F}_2}(\mathbf{U}_{k,J_2-1}\hat{\mathbf{U}}_{k,J_2-1}) \quad (8) \]

- Use the JADE algorithm or the Extended Matrix Pencil Algorithm to blindly separate the sources \( u_{1}, u_{2}, \cdots, u_{m_2} \) belonging to subsystems of order \( J_2 \) and identify the corresponding kernels, as columns of the matrix \( \mathbf{G}_{J_2}^{\mathbf{F}_2} \).

- Having computed orders \( J_1, J_2, \cdots, J_r \), compute \( J_{n_1} \) and \( m_{n_1} \). Separate the next group of sources and identify the corresponding kernels using the equation:

\[ \mathbf{E}_{k,J_{n_1}} - \sum_{i=1}^{n_2} \mathbf{G}_{J_{n_1+i}}^{\mathbf{F}_1}(\mathbf{U}_{k,J_{n_1+i-1}\hat{\mathbf{U}}_{k,J_{n_1+i-1}}^{n_1}} = \mathbf{G}_{J_{n_1+i-1}}^{\mathbf{F}_1}(\mathbf{U}_{k,J_{n_1+i-1}}^{n_1}) \hat{\mathbf{U}}_{k,0}^{n_1-1} \quad (9) \]

- Repeat until all sources have been blindly separated and the corresponding kernels are identified as columns of the matrices \( \mathbf{G}_{J_i}^{\mathbf{F}_i} \), for \( 1 \leq i \leq r \).

For FIR linear MIMO systems, both BSS methods can be used. They allow input equalization as well as kernel identification up to multiplication with a diagonal and a permutation matrix.

For SIMO Volterra systems, however, only the Extended Matrix Pencil Method can be used for source separation, since the inputs computed are nonlinear combinations of the original signals. In this case, the applicability of the Extended Matrix Pencil Method is guaranteed by output cyclostationarity, that allows the use of several distinct output correlation matrices computed at different time lags. If the Volterra system under consideration is passband, then both input equalization and kernel identification can be performed. In case of a baseband Volterra system, even though input equalization is achieved, the identified kernels consist of sums of the original system kernels.

The performance of the above method is checked against existing algorithms by simulation.

5. SIMULATION EXAMPLES

We present the simulation experiments conducted for linear MIMO case as well as the SIMO FIR Volterra case. To compute the distinct subsystem orders we used 2000 input samples. In all runs PSK inputs were used.

5.1 MIMO Systems

Simulations entailing two out of three different systems presented in [8] have been performed. First, the JADE algorithm is used. In this case we compare the proposed algorithm to the one presented in [2]. Then, the Extended Matrix Pencil algorithm is used. In this case, the proposed method is compared to the SSUB algorithm, given in [3]. 100 independent Monte-Carlo runs were obtained per system and SNR value.
Example 1. The first of the three systems presented in [8] is considered. The performance of the proposed method is checked against the algorithm presented in [2]. Simulation results for kernel identification are presented in Tables 1 and 2. We see that the proposed method achieves a slight performance gain.

Example 2. The second of the three systems presented in [8] is considered. The performance of the proposed method is checked against the algorithm presented in [3]. Simulation results for kernel identification are presented in Tables 3 and 4. We see that the proposed method achieves significant performance gain.

5.2 Volterra Systems

Simulations of SIMO FIR Volterra systems are performed. The proposed method is compared with the methods presented in [6, 7] using the Extended Matrix Pencil algorithm.

Example 3. We apply the proposed algorithm to the system described in Example 2 of [6]. The proposed algorithm is checked against the methods presented in [6, 7] for two different SNR values, namely 15dB and 20dB. SER is calculated in both cases. Results are presented in Table 5. We see that the proposed method achieves significant performance gain over the other two methods.

6. CONCLUSIONS

A new algorithm for identifying system kernels as well as equalizing inputs is proposed. The proposed method employs the results of [8]. The algorithm allows stepwise input equalization using the JADE algorithm and the Extended Matrix Pencil Algorithm. Sources are equalized in groups, depending on the memory of the subsystems they belong to. For linear FIR MIMO systems as well as passband Volterra systems, channel identification is also performed. For baseband Volterra systems, the identified kernels consist of sums of the original system kernels. The performance of the algorithm is quite satisfactory, compared to existing algorithms for linear MIMO and SIMO Volterra equalization and identification [2, 3, 6, 7].

REFERENCES


Table 1: Example 1. MIMO Systems: Channel 1 Estimation

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Proposed Algorithm</th>
<th>Ding’s Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>-12</td>
<td>-10</td>
</tr>
<tr>
<td>25</td>
<td>-13</td>
<td>-11</td>
</tr>
<tr>
<td>30</td>
<td>-22</td>
<td>-20</td>
</tr>
<tr>
<td>35</td>
<td>-30</td>
<td>-28</td>
</tr>
<tr>
<td>40</td>
<td>-35</td>
<td>-32</td>
</tr>
<tr>
<td>45</td>
<td>-42</td>
<td>-40</td>
</tr>
<tr>
<td>50</td>
<td>-43</td>
<td>-41</td>
</tr>
</tbody>
</table>

Table 2: Example 1. MIMO Systems: Channel 2 Estimation

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Proposed Algorithm</th>
<th>Ding’s Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>-12</td>
<td>-10</td>
</tr>
<tr>
<td>25</td>
<td>-14</td>
<td>-12</td>
</tr>
<tr>
<td>30</td>
<td>-22</td>
<td>-19</td>
</tr>
<tr>
<td>35</td>
<td>-30</td>
<td>-27</td>
</tr>
<tr>
<td>40</td>
<td>-34</td>
<td>-31</td>
</tr>
<tr>
<td>45</td>
<td>-44</td>
<td>-38</td>
</tr>
<tr>
<td>50</td>
<td>-44</td>
<td>-40</td>
</tr>
</tbody>
</table>
### Table 3: Example 2. MIMO Systems: Channel 1 Estimation

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Proposed Algorithm</th>
<th>SSUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>−19</td>
<td>−15</td>
</tr>
<tr>
<td>25</td>
<td>−23</td>
<td>−17</td>
</tr>
<tr>
<td>30</td>
<td>−30</td>
<td>−25</td>
</tr>
<tr>
<td>35</td>
<td>−32</td>
<td>−26</td>
</tr>
<tr>
<td>40</td>
<td>−35</td>
<td>−27</td>
</tr>
</tbody>
</table>

### Table 4: Example 2. MIMO Systems: Channel 2 Estimation

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Proposed Algorithm</th>
<th>SSUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>−18</td>
<td>−14</td>
</tr>
<tr>
<td>25</td>
<td>−24</td>
<td>−19</td>
</tr>
<tr>
<td>30</td>
<td>−27</td>
<td>−26</td>
</tr>
<tr>
<td>35</td>
<td>−31</td>
<td>−28</td>
</tr>
<tr>
<td>40</td>
<td>−34</td>
<td>−30</td>
</tr>
</tbody>
</table>

### Table 5: Example 3. SIMO Volterra Systems: SER Calculation

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Proposed Algorithm</th>
<th>Valcarce et al</th>
<th>Giannakis et al</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(4 \times 10^{-3})</td>
<td>(2 \times 10^{-2})</td>
<td>(4 \times 10^{-1})</td>
</tr>
<tr>
<td>20</td>
<td>(2 \times 10^{-6})</td>
<td>(3 \times 10^{-5})</td>
<td>(7 \times 10^{-4})</td>
</tr>
</tbody>
</table>