

Error correcting least-squares Subspace algorithm for blind identification and equalization[☆]

Balaji Sampath, K.J. Ray Liu^{*}, Y. Goeffrey Li¹

Electrical and Computer Engineering Department and Institute for Systems Research University of Maryland, College Park, MD 20742, USA

Received 1 November 2000; received in revised form 30 April 2001

Abstract

A subspace based blind channel identification algorithm using only the fact that the received signal can be oversampled is proposed. No direct use is made in this algorithm of either the statistics of the input sequence or even of the fact that the symbols are from a finite set and therefore this algorithm can be used to identify even channels in which arbitrary symbols are sent. Using this algorithm as a base and using the extra information which becomes available when the transmitted symbols are from a known finite set, the EC-LS-Subspace algorithm is derived. The EC-LS-Subspace algorithm operates directly on the data domain and therefore avoids the problems associated with other algorithms which use the statistical information contained in the received signal directly. In the noiseless case, if some conditions are met, it is possible for the proposed Basic Subspace algorithm to identify the channel exactly using an observation interval of just $(J+2)T$, if the length of the impulse response of a channel is JT , T being the symbol interval. In the noisy case, simulations have shown that the channel can be identified accurately by using a very small observation interval (comparable to $(J+2)T$). © 2001 Elsevier Science B.V. All rights reserved.

1. Introduction

Most digital communication systems are subject to intersymbol interference (ISI). In many cases this is so severe that the correct reception of the transmitted sequence is hindered unless specific equalization procedures are adopted. Classically, these procedures are based on the knowledge of the channel which is obtained by sending a known training sequence. But, when the channel is varying, even slowly, the training sequence has to be sent periodically so that the channel estimates can be updated and this reduces the effective channel rate.

In contrast to the classical channel identification methods, blind channel identification methods are very attractive since they do not require training sequences. Most existing blind identification methods use the statistics of the transmitted sequence instead of the explicit knowledge of the sequence itself.

[☆] The work was supported in part by the NSF NYI Award MIP9457397.

^{*} Corresponding author. Tel.: +1-301-405-6619; fax: +1-301-405-6707.

E-mail address: kjrlui@eng.umd.edu (K.J. Ray Liu).

¹ Now with Electrical Engineering Department, Georgia Institute of Technology, Atlanta, GA 30332, USA.

Since communication channels are very likely to be nonminimum phase, most of the existing blind channel identification algorithms have used higher order statistics [2–6,9,10]. Elegant solutions using this approach have been given but all of them need a very large number of symbols (typically, much more than several hundred symbols).

The discovery by Tong et al. [1], of an algorithm which allows the blind identification of channels using only second-order statistics is considered to be a breakthrough. Their algorithm relies on the cyclo-stationary of communication signals and makes explicit use of the second-order cyclo-stationary statistics of the oversampled channel output. A number of algorithms, which improve on the results in [1] and which make explicit use of second-order cyclo-stationary statistics, have been proposed, [1,8,12,16,17].

Most of these algorithms do not make explicit use of the signal structure inherent in the oversampled channel output. The oversampled output has a lot of structure and this can be exploited in different ways. Recently, Xu et al. [16], and Liu and Xu [17] have come up with algorithms which exploit the inherent structure of the oversampled channel output in different ways.

In this paper we present a new algorithm which exploits the structure of the oversampled output in yet another way. But the approach that we use to exploit this structure is quite different from the other methods. We have used the following philosophy in deriving this algorithm—since a lot of the output structure comes from the shifts in the input sequence, it should be possible to look at the inverse channel and hence exploit the shifts in the input sequence instead of directly making use of the output structure. The idea is that visually explicit shifts in the input sequence can be exploited much more easily and more thoroughly than the implicit structure embedded in the oversampled output.

It must be noted here that instead of oversampling, one could also use multiple physical receivers or sensors. The basic idea involved here is the exploitation of the different instantiations of the same input signal by multiple FIR channels.

Although the Basic Subspace algorithm does not use any property of the symbol set and gives exact results in the noiseless case, in the noisy case a much better algorithm can be obtained by making use of the known properties of the transmitted sequence. We have therefore modified the Basic Subspace algorithm by incorporating least-square techniques. This modified algorithm, EC-LS-Subspace algorithm, has a number of desirable properties. In particular, it provides very accurate channel estimates, needs a small observation interval and is quite robust to channel noise.

The received baseband signal $x(\cdot)$ can be written as

$$r(t) = \sum_{k=-\infty}^{\infty} s_k h(t - kT) \quad (1)$$

$$x(t) = r(t) + n(t) \quad (2)$$

where s_k is an information symbol in a signal constellation \mathcal{S} (\mathcal{S} may be an infinite set), $h(\cdot)$ is the discrete-time channel impulse response, T is the symbol interval and $n(\cdot)$ is the additive noise. In the sequel we will assume that the impulse response $h(\cdot)$ has finite support, i.e. $h(t) = 0$ for $t \geq JT$, $J \in \mathcal{N}$.

The channel identification problem requires us to estimate the channel impulse response $h(t)$ (or at least samples of $h(t)$). Classical channel identification procedures using a training sequence have the knowledge of both $x(t)$ as well as the transmitted symbols $\{s_k\}$. A blind channel identification algorithm on the other hand has to estimate the channel response $h(\cdot)$ given only the received signal $x(\cdot)$.

In this paper we derive and study a new blind channel identification algorithm which needs very few symbols² to estimate the channel. The organization of the paper is as follows—in Section 2, we derive

² Under certain conditions, just $J + 3$ symbols are enough to obtain the exact impulse response in the noiseless case.

the Basic Subspace algorithm and this forms the backbone of our approach. In the next section we derive the conditions for the identifiability of the channels using this method. In Section 4, we modify the Basic Subspace algorithm by incorporating least-square techniques and obtain a robust algorithm which works very well in practical situations. Finally we discuss a simulation example and the conclusions that can be drawn from it.

2. The Basic Subspace algorithm

In this section we develop the Basic Subspace algorithm. In order to simplify the presentation we will ignore the noise for the moment. We will first discuss the case when the impulse response is of length $2T$, i.e. $J=2$ and then we will present the more general case.

2.1. The Basic Subspace algorithm for the $J=2$ case

We sample the signal at twice the baud rate, i.e., at $t=nT+\delta_1$ and $t=nT+\delta_2$, $0\leq\delta_1,\delta_2\leq T$. Using the fact that the impulse response has length $2T$, it can be easily seen that the following $2M$ equations are true:

$$y(2j-2)=s_jh_0+s_{j-1}h_2, \quad y(2j-1)=s_jh_1+s_{j-1}h_3, \quad 1\leq j\leq M,$$

where $y(2n+i-3)=x(nT+\delta_i)$, $h_{2n+i-1}=h(nT+\delta_i)$, for $i=1,2$.

An important fact to note here is that while there are $2M$ equations there are $M+5$ unknown variables (i.e., h_i and s_i). But these equations are homogeneous and hence h_i (or alternately s_i) can be fixed only upto a multiplying constant. This implies that we can arbitrarily fix the value of one of the variables (assuming that it is not zero) to be 1. Therefore, there are $M+4$ unknown variables and $2M$ equations. This means that when $M<4$ we certainly cannot solve for the unknown variables, but if $M\geq 4$ we may be able to do so. Therefore the minimum length of the observation interval required to find $\{h_i\}$ is no less than $4T$. We will now show that under certain conditions we can in fact find the exact impulse response with an observation interval of just $4T$. Define

$$\mathbf{x}_0=[y(0) \quad y(2) \quad y(4) \quad \cdots \quad y(2M-2)]^T, \quad \mathbf{x}_1=[y(1) \quad y(3) \quad y(5) \quad \cdots \quad y(2M-1)]^T,$$

$$\mathbf{s}_0=[s_0 \quad s_1 \quad s_2 \quad \cdots \quad s_{M-1}]^T, \quad \mathbf{s}_1=[s_1 \quad s_2 \quad s_3 \quad \cdots \quad s_M]^T,$$

$$\mathbf{H}=\begin{bmatrix} h_2 & h_3 \\ h_0 & h_1 \end{bmatrix},$$

then it is clear that

$$[\mathbf{x}_0 \quad \mathbf{x}_1]=[\mathbf{s}_0 \quad \mathbf{s}_1]\mathbf{H}. \tag{3}$$

Assuming that \mathbf{H} is invertible, i.e.

$$\mathbf{\Lambda}=\begin{bmatrix} \lambda_0 & \lambda_2 \\ \lambda_1 & \lambda_3 \end{bmatrix}=\mathbf{H}^{-1} \tag{4}$$

exists, we see by post-multiplying by $\mathbf{\Lambda}$ in (3), that the vectors \mathbf{s}_0 and \mathbf{s}_1 lie in the span of the vectors \mathbf{x}_0 and \mathbf{x}_1 and we have

$$\mathbf{s}_0=\lambda_0\mathbf{x}_0+\lambda_1\mathbf{x}_1, \tag{5}$$

$$\mathbf{s}_1=\lambda_2\mathbf{x}_0+\lambda_3\mathbf{x}_1. \tag{6}$$

But \mathbf{s}_0 and \mathbf{s}_1 are not just any two vectors in the span of \mathbf{x}_0 and \mathbf{x}_1 , they have a very special structure: The bottom $M - 1$ elements of \mathbf{s}_0 are the same as the top $M - 1$ elements of \mathbf{s}_1 . We will now exploit this relationship. Partition the vectors \mathbf{x}_0 and \mathbf{x}_1 as below

$$[\mathbf{x}_0 \quad \mathbf{x}_1] = \begin{bmatrix} y(0) & y(1) \\ \xi_0 & \xi_1 \end{bmatrix} = \begin{bmatrix} \eta_0 & \eta_1 \\ y(2M-2) & y(2M-1) \end{bmatrix}, \quad (7)$$

Where η_i and ξ_i are the top and bottom $M - 1$ elements of \mathbf{x}_i . The constraint that the bottom $M - 1$ elements of \mathbf{s}_0 are the same as the top $M - 1$ elements of \mathbf{s}_1 , translates into the following relation

$$\lambda_0 \xi_0 + \lambda_1 \xi_1 = \lambda_2 \eta_0 + \lambda_3 \eta_1. \quad (8)$$

This in turn implies that $\lambda = [\lambda_0 \quad \lambda_1 \quad \lambda_2 \quad \lambda_3]^T$ is in the null space of Φ where

$$\Phi = [\xi_0 \quad \xi_1 \quad -\eta_0 \quad -\eta_1].$$

If we make the further assumption that Φ has only a single dimensional null space,³ we can then find λ uniquely (up to a multiplication factor) and therefore $\mathbf{H} = \mathbf{\Lambda}^{-1}$ is also determined up to a multiplication factor. Note that this method needs an observation interval of only $4T$ to identify the channel, if the corresponding Φ matrix has a one-dimensional null space. We will show in Section 3 that under certain conditions, it is possible for the Φ matrix to have a one-dimensional null space even with an observation interval of just $4T$. (Note that with an observation interval of $3T$, this method will not work. This can be seen from the fact that Φ matrix will then be a 2×4 matrix and so the null space will have dimension at least two.)

2.2. The Basic Subspace algorithm for the arbitrary J case

We will now present the main steps of the algorithm for the identification of an arbitrary FIR channel. We assume that the impulse response has length JT and that the received signal is not corrupted by noise. In the noiseless case, this method can be used to identify the channel exactly⁴ with an observation length of just $(J + 2)T$. We use the same notations as for the $J = 2$ case. Here the received signal is sampled at J times the baud rate. It is easy to see that we obtain the following equations for $0 \leq i \leq J - 1$:

$$\begin{aligned} y(i) &= s_{J-1}h_i + s_{J-2}h_{J+i} + \cdots + s_0h_{(J-1)J+i}, \\ y(J+i) &= s_Jh_i + s_{J-1}h_{J+i} + \cdots + s_1h_{(J-1)J+i} \\ &\vdots \\ &\vdots \\ y(JM - J + i) &= s_{M+J-2}h_i + s_{M+J-3}h_{J+i} + \cdots + s_{M-1}h_{(J-1)J+i}, \end{aligned}$$

where $y(Jn - J + j - 1) = x(nT + \delta_j)$, $h_{Jn+j-1} = h(nT + \delta_j)$, $1 \leq j \leq J$.

Following the same procedure as in the $J = 2$ case, we now define the vectors \mathbf{x}_i and \mathbf{s}_i for $0 \leq i \leq J - 1$ as below

$$\begin{aligned} \mathbf{x}_i &= [y(i) \quad y(J+i) \quad y(2J+i) \quad \cdots \quad y((M-1)J+i)]^T, \\ \mathbf{s}_i &= [s_i \quad s_{i+1} \quad s_{i+2} \quad \cdots \quad s_{M+i-1}]^T. \end{aligned}$$

As before we obtain the relation

$$\mathbf{X} = \mathbf{S}\mathbf{H}, \quad (9)$$

where

$$\mathbf{X} = [\mathbf{x}_0 \quad \mathbf{x}_1 \quad \cdots \quad \mathbf{x}_{J-2} \quad \mathbf{x}_{J-1}], \quad \mathbf{S} = [\mathbf{s}_0 \quad \mathbf{s}_1 \quad \cdots \quad \mathbf{s}_{J-2} \quad \mathbf{s}_{J-1}]$$

³ That this indeed happens with an extremely high probability will be proved in Section 3.

⁴ Provided the corresponding Φ matrix in (11) has a one-dimensional null space.

and the k th column of \mathbf{H} is \mathbf{h}_k , $1 \leq k \leq J$, where

$$\mathbf{h}_k = [h_{J(J-1)+k-1} \quad h_{J(J-2)+k-1} \quad \cdots \quad h_{J+k-1} \quad h_{k-1}]^T.$$

Inverting this relation we see that $\mathbf{s}_i \in \text{span}\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{J-1}\}$. Therefore we have for $0 \leq i \leq J - 1$,

$$\mathbf{s}_i = \lambda_0^{(i)} \mathbf{x}_0 + \lambda_1^{(i)} \mathbf{x}_1 + \cdots + \lambda_{J-1}^{(i)} \mathbf{x}_{J-1}, \tag{10}$$

where the $\{\lambda_j^{(i)}\}$, taken appropriately, form the matrix $\mathbf{\Lambda} = \mathbf{H}^{-1}$.

Let ξ_i and η_i be the bottom $M - 1$ and top $M - 1$ elements of \mathbf{x}_i , respectively. Then the constraint imposed by the structure of \mathbf{s}_i (i.e. bottom $M - 1$ elements of \mathbf{s}_i is the same as the top $M - 1$ elements of \mathbf{s}_{i+1}) translates into the following relation:

$$\mathbf{\Phi} \lambda = 0, \tag{11}$$

where

$$\lambda = [\lambda_0^{(0)}, \dots, \lambda_{J-1}^{(0)}, \lambda_0^{(1)}, \dots, \lambda_{J-1}^{(1)}, \dots, \lambda_0^{(J-1)}, \dots, \lambda_{J-1}^{(J-1)}]^T,$$

$$\mathbf{\Phi} = \begin{bmatrix} \xi & \eta & 0 & 0 & \cdots & 0 & 0 \\ 0 & \xi & \eta & 0 & \cdots & 0 & 0 \\ 0 & 0 & \xi & \eta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \xi & \eta \end{bmatrix}$$

with the number of block-columns being J , $\mathbf{0}$ representing a matrix of zeros and

$$\xi = [\xi_0 \quad \xi_1 \quad \cdots \quad \xi_{J-1}], \quad \eta = -[\eta_0 \quad \eta_1 \quad \cdots \quad \eta_{J-1}]. \tag{12}$$

Therefore we know that $\lambda \in \text{Null}(\mathbf{\Phi})$, and again if we know that $\mathbf{\Phi}$ has a one-dimensional null space, then we can find λ and therefore \mathbf{H} and $\{s_k\}$.

In the next section, we will prove that the probability that $\mathbf{\Phi}$ does have one-dimensional null-space tends to one. We still have to substantiate our claim that under certain conditions this method will work with as small an observation interval as $(J + 2)T$. In the next section we show that under certain conditions, $\mathbf{\Phi}$ can have a one-dimensional null space with an observation interval of $(J + 2)T$. This implies that under certain conditions this algorithm can identify the channel with as small an observation interval as $(J + 2)T$. Note that it cannot work with a smaller observation interval. The number of equations will then become less than the number of unknown variables and so the solution cannot be uniquely determined. Another way to understand this is by looking at the dimensions of the matrix $\mathbf{\Phi}$ —if the observation interval is $(J + 1)T$, then the dimension of $\mathbf{\Phi}$ is $(J^2 - J) \times J^2$. Since, $J > 1$, the null space of $\mathbf{\Phi}$ has dimension greater than one.

3. Identifiability

The two assumptions under which the Basic Subspace algorithm will work are:

- \mathbf{H} is invertible (or in the noisy case \mathbf{H} should also be well-conditioned).
- $\mathbf{\Phi}$ has a one-dimensional null space.

\mathbf{H} being invertible is not a big problem but unfortunately it is often ill-conditioned and this causes problems in the noisy case. In the next section we will discuss ways to overcome this problem for many practical channels. For many channels, the modifications of the Basic Subspace algorithm derived in Section 4, works quite well. In our simulations we use an ill-conditioned \mathbf{H} matrix and the modified

algorithm and show that the estimates obtained by using the modified algorithm are in fact very close to the actual result.

If we assume that the \mathbf{H} matrix is indeed invertible, it can be shown that the second assumption is almost never violated. We now prove the following theorem.

Theorem 3.1. *For every non-trivial⁵ communication system, if \mathbf{H} is invertible then the probability that Φ has a one-dimensional null space tends to 1 as the observation interval increases. The rate of convergence is at least exponential if the symbols are independent.*

Theorem 3.1 ensures that as the observation interval increases, the probability of the violation of the second assumption tends to zero very fast and therefore even with a very small observation interval we can use the Basic Subspace algorithm to estimate the channel. Similar conditions based on persistently exciting sequences for the identifiability of a FIR channel have been discussed in [7].

We will now prove Theorem 3.1. In proving this theorem, we will also discover a very elegant approach to the Basic Subspace algorithm.

Let \mathbf{s}_{ξ_i} and \mathbf{s}_{η_i} be the bottom $M - 1$ and top $M - 1$ elements of the vector \mathbf{s}_i . Then we know that

$$\mathbf{s}_{\xi_0} = \mathbf{s}_{\eta_1}, \mathbf{s}_{\xi_1} = \mathbf{s}_{\eta_2}, \dots, \mathbf{s}_{\xi_{J-3}} = \mathbf{s}_{\eta_{J-2}}, \mathbf{s}_{\xi_{J-2}} = \mathbf{s}_{\eta_{J-1}}.$$

It is therefore clear that of the $2J$ vectors \mathbf{s}_{ξ_i} and \mathbf{s}_{η_i} , we only have $J+1$ different vectors $\mathbf{s}_{\eta_0}, \mathbf{s}_{\xi_0}, \mathbf{s}_{\xi_1}, \dots, \mathbf{s}_{\xi_{J-1}}$ and for convenience we will call them $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_J$, respectively.

Lemma 3.1. *If \mathbf{H} is invertible and the $J + 1$ vectors $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_J$, are linearly independent, then the matrix Φ has a one-dimensional null space.*

Proof. From (9) and (12) we get

$$\xi = [\mathcal{B}_1 \ \mathcal{B}_2 \ \dots \ \mathcal{B}_J] \mathbf{H}, \quad \eta = -[\mathcal{B}_0 \ \mathcal{B}_1 \ \dots \ \mathcal{B}_{J-1}] \mathbf{H}.$$

Using this relation in Φ , we get

$$\Phi = \Psi \tilde{\mathbf{H}}, \tag{13}$$

where

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{H} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{H} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{H} \end{bmatrix},$$

and

$$\Psi = \begin{bmatrix} \mathcal{B}_1 \mathcal{B}_2 \dots \mathcal{B}_J & \mathcal{B}_0 \mathcal{B}_1 \dots \mathcal{B}_{J-1} & 0 & \dots & 0 & 0 \\ 0 & \mathcal{B}_1 \mathcal{B}_2 \dots \mathcal{B}_J & \mathcal{B}_0 \mathcal{B}_1 \dots \mathcal{B}_{J-1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \mathcal{B}_1 \mathcal{B}_2 \dots \mathcal{B}_J & \mathcal{B}_0 \mathcal{B}_1 \dots \mathcal{B}_{J-1} \end{bmatrix}.$$

Since \mathbf{H} is invertible, $\tilde{\mathbf{H}}$ is also invertible. Therefore Φ will have a one-dimensional null space if Ψ has a one-dimensional null space. Lemma 3.2 confirms that Ψ does have a one-dimensional null space under the hypothesis of Lemma 3.1 and therefore we have proved Lemma 3.1. \square

⁵ A trivial communication system is one in which all the future transmitted symbols are decided by a finite number of transmitted symbols and this is clearly not a very useful communication system!

Lemma 3.2. *If the $J + 1$ vectors $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_J$ are linearly independent, then the matrix Ψ has a one-dimensional null space and*

$$\mathcal{N}(\Psi) = \{c[e_1 \ e_2 \ e_3 \ \dots \ e_J]^T : c \in \mathfrak{R}\},$$

where $e_i = [0 \ 0 \ \dots \ 1 \ 0 \ \dots \ 0]$ (1 is in the i th position).

Proof. Let the vector β be a vector in the null space of Ψ where

$$\beta = [\beta_{1,1}, \beta_{2,1}, \dots, \beta_{J,1}, \beta_{1,2}, \beta_{2,2}, \dots, \beta_{J,2}, \dots, \beta_{1,J}, \beta_{2,J}, \dots, \beta_{J,J}]^T.$$

Then

$$\Psi\beta = 0.$$

From this equation, and, using the fact that since the vectors $\mathcal{B}_0, \mathcal{B}_1, \dots, \mathcal{B}_J$ are linearly independent, a linear combination of them cannot be zero unless all the corresponding coefficients are zero, we get the following equations:

$$\begin{aligned} \beta_{1,i} &= 0 \quad \text{for all } i = 2, 3, \dots, J - 1, J, \\ \beta_{1,i} - \beta_{2,i+1} &= 0 \quad \text{for all } i = 1, 2, 3, \dots, J - 1, \\ \beta_{2,i} - \beta_{3,i+1} &= 0 \quad \text{for all } i = 1, 2, 3, \dots, J - 1 \\ &\vdots \quad \quad \quad \vdots \\ \beta_{J-1,i} - \beta_{J,i+1} &= 0 \quad \text{for all } i = 1, 2, 3, \dots, J - 1, \\ \beta_{J,i} &= 0 \quad \text{for all } i = 1, 2, 3, \dots, J - 1. \end{aligned}$$

From these equations it can be concluded that $\beta_{i,j}$ for $i \neq j$ are all zero and $\beta_{1,1} = \beta_{2,2} = \dots = \beta_{J,J}$. This means that the vector β is fixed up to a multiplication factor and since β is an arbitrary vector in the null space of Ψ , we infer that Ψ has a one-dimensional null space generated by the vector $[e_1 \ e_2 \ e_3 \ \dots \ e_J]^T$. □

Therefore we come to the conclusion that if the \mathbf{H} matrix is invertible, Φ does not have a one-dimensional null space only if the $J + 1$ vectors $\mathbf{s}_{\eta_0}, \mathbf{s}_{\xi_0}, \mathbf{s}_{\xi_1}, \dots, \mathbf{s}_{\xi_{J-1}}$ are linearly dependent. This condition forces the following matrix to have a non-trivial null space:

$$\mathbf{Q} = [\mathbf{s}_{\eta_0} \ \mathbf{s}_{\xi_0} \ \mathbf{s}_{\xi_1} \ \dots \ \mathbf{s}_{\xi_{J-1}}].$$

This in turn trivially implies the following necessary condition.

Lemma 3.3. *If \mathbf{H} is invertible, then Φ does not have a one-dimensional null space only if*

$$\det \begin{pmatrix} s_i & s_{i+1} & \dots & s_{i+J} \\ s_j & s_{j+1} & \dots & s_{j+J} \\ \vdots & \vdots & \vdots & \vdots \\ s_k & s_{k+1} & \dots & s_{k+J} \end{pmatrix} = 0 \quad \forall i, j, \dots, k: i \neq j \neq \dots \neq k.$$

Let the first $2J$ symbols $\{s_0, s_1, \dots, s_{2J-1}\}$ take on a set of arbitrary values. Now we will look at the only sequences for which it may be possible for the necessary condition to hold.

When we use the necessary condition for $(i, j, \dots, k) = (0, 1, \dots, J)$ we fix the value of s_{2J} . When we then apply the necessary condition for $(i, j, \dots, k) = (1, 2, \dots, J + 1)$ we have no choice but to fix the

value of s_{2J+1} . Similarly applying the necessary condition successively we will fix the values of all the symbols $\{s_{2J}, \dots, s_M\}$. Therefore there is only one sequence of length M with its first $2J$ elements specified that can *possibly* satisfy the necessary condition. It may also happen that for certain choices of the $2J$ elements no sequence may satisfy the necessary condition in its entirety. (Note moreover the fact that this is only a necessary condition and not a necessary and sufficient condition.)

In any rich communication system the probability that a particular sequence of length M is transmitted given that the first $2J$ elements of the sequence take on a specified values tends to zero as M increases. In particular if the symbols are from a finite set of size q then the number of possible values the first $2J$ elements can take is q^{2J} and the total number of sequences of length M are q^M . If the transmitted symbols are independent then the probability that the transmitted sequence satisfies the necessary condition is less than or equal to q^{2J-M} . Therefore we come to the conclusion that if \mathbf{H} is invertible then

$$\text{Prob}(\Phi \text{ has a one-dimensional null space}) \geq 1 - \frac{1}{q^{M-2J}}.$$

This means that as the observation interval M increases, the probability that the Basic Subspace algorithm fails tends to zero exponentially and if the symbols are not from a finite set, then clearly the probability of failure is zero if $M > 2J$ and we have therefore proved Theorem 3.1. \square

The proof of Theorem 3.1 also opens up a new way of deriving the Basic Subspace algorithm. From (13) we know that vectors in the null space of Φ , when pre-multiplied by $\tilde{\mathbf{H}}$, produce vectors in the null space of Ψ . Conversely, vectors in the null space of Ψ , when pre-multiplied by $\tilde{\mathbf{H}}^{-1}$, produce vectors in the null space of Φ . This combined with the fact that the only vector (disregarding multiplication factors) in the null space of Ψ is $[e_1, e_2, e_3, \dots, e_J]^T$ implies that if λ is a vector in the null space of Φ , then $\mathbf{H}\lambda = \mathbf{A}^{-1}$, where the matrix \mathbf{A} is constructed by appropriately arranging the elements of the vector λ and then scaling it suitably. That this is indeed the case can be easily verified by noting that the vector $[e_1 \ e_2 \ e_3 \ \dots \ e_J]^T$ picks out the columns of the matrix \mathbf{H}^{-1} from the matrix $\tilde{\mathbf{H}}^{-1}$ and arranges them one below the other to form the vector λ .

We still have to show that it is possible (under certain conditions) to use this algorithm to identify a general FIR channel of length JT exactly (in the noiseless situation) with an observation length of just $(J+2)T$. To show this, all we need to show is that when the observation length is $(J+2)T$, for certain input sequences, Φ has a one-dimensional null space. But from the proof of Theorem 3.1, we know that Φ will have one-dimensional null space, whenever the $J+1$ vectors $\mathbf{s}_{\eta_0}, \mathbf{s}_{\xi_0}, \mathbf{s}_{\xi_1}, \dots, \mathbf{s}_{\xi_{J-1}}$ are linearly independent. Since the observation interval is $J+2$, the dimension of each of these $J+1$ vectors is $J+1$. So we have $(J+1)$ $(J+1)$ -dimension vectors. But these vectors can be chosen to be linearly independent even with a constellation size of two. Therefore, under the condition that the $J+1$ vectors $\mathbf{s}_{\eta_0}, \mathbf{s}_{\xi_0}, \mathbf{s}_{\xi_1}, \dots, \mathbf{s}_{\xi_{J-1}}$ are linearly independent, it is possible to exactly identify (in the noiseless case) the channel using this algorithm with an observation interval of just $(J+2)T$.

In the noisy situation, the same algorithm can be applied with the small modification that in solving for λ , instead of finding the null space of Φ , we find the singular vector corresponding to the smallest singular value of Φ .

4. The least-squares Subspace algorithm

In this section we will attack the important problem of the \mathbf{H} matrix being ill-conditioned. Before that we first look at the Exhaustive LS-Search algorithm, a *conceptual algorithm* for solving the blind identification problem. In the classical identification procedures, which use a training sequence, the problem

is often solved in this way: Assuming that the received signal is sampled at the baud rate, we have the following matrix equation:

$$\mathbf{S}\mathbf{h} + \mathbf{n} = \mathbf{y}, \tag{14}$$

where

$$\mathbf{S} = \begin{bmatrix} s_1 & s_2 & \cdots & s_{J-1} & s_J \\ s_2 & s_3 & \cdots & s_J & s_{J+1} \\ s_3 & s_4 & \cdots & s_{J+1} & s_{J+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{M-J+1} & s_{M-J+2} & \cdots & s_{M-1} & s_M \end{bmatrix},$$

$$\mathbf{h} = [h_{J-1}, h_{J-2}, \dots, h_1, h_0]^T, \quad h_i = h(iT),$$

$$\mathbf{n} = [n_J, n_{J+1}, n_{J+2}, \dots, n_M]^T, \quad n_i = n(iT),$$

$$\mathbf{y} = [y_J, y_{J+1}, y_{J+2}, \dots, y_M]^T, \quad y_i = x(iT).$$

Since we know the \mathbf{S} matrix, we can find the least-squares solution for the vector \mathbf{h} which minimizes the norm-square error $\|\mathbf{S}\mathbf{h} - \mathbf{y}\|^2$. In the blind identification problem, we do not know the matrix \mathbf{S} , because we do not know the transmitted sequence. But if the transmitted symbols are from a finite set (of say size 2, i.e. $s_i = \pm 1$) then we do know that the symbols $\{s_i\}_{i=1}^M$ can take one of 2^M possibilities. For each of these possibilities we will have a \mathbf{S} matrix. For each of these possible \mathbf{S} matrices we can find the least-squares solution for \mathbf{h} and also the corresponding error $\|\mathbf{S}\mathbf{h} - \mathbf{y}\|^2$. We can then choose as our ‘best’ \mathbf{S} , the one which minimizes the error and therefore our estimate of \mathbf{h} will be the least-squares \mathbf{h} corresponding to the ‘best’ \mathbf{S} . It seems very likely that since in most cases the ‘best’ \mathbf{S} that we obtain will be close to the actual transmitted \mathbf{S} , our estimates of the channel impulse response \mathbf{h} will be almost the same as that obtained by sending a training sequence and so will provide us with an excellent estimate of the channel impulse response. We performed a few simulation runs to check this fact and they confirmed this.

But the main problem with this exhaustive-search approach is that we will have to perform the least-squares estimate 2^M times and this makes the practical implementation of this algorithm impossible. But if we could somehow get a reasonable estimate of the actual transmitted sequence, then we would need to search for the ‘best’ \mathbf{S} only in the vicinity of the estimated \mathbf{S} . This will considerably reduce the computational cost and if the estimate is very good (as it is when we use the Basic Subspace algorithm) then the computational cost may even be much lesser than the cost for other standard blind identification algorithms which make use of the statistics of the received signal.

4.1. The LS-Subspace algorithm

Since the Basic Subspace algorithm provides a way of directly finding the transmitted sequence, we will use it to estimate \mathbf{S} . But the problem is that \mathbf{H} is ill-conditioned and so we will not be able to use the Basic Subspace algorithm. But for many practical channels there is a way of circumventing this problem as we now illustrate.

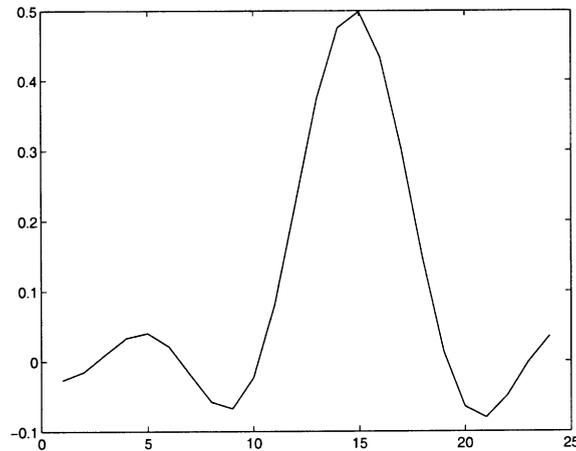


Fig. 1. A three-ray multipath channel impulse response.

Why is it that in many channels the \mathbf{H} matrix is ill-conditioned? Fig. 1 shows a typical channel response. The tail of the response is very long and has a very small magnitude. The k th column of \mathbf{H} is \mathbf{h}_k , $1 \leq k \leq J$, where

$$\mathbf{h}_k = [h_{J(J-1)+k-1} \quad h_{J(J-2)+k-1} \quad \cdots \quad h_{J+k-1} \quad h_{k-1}]^T$$

where $h_i = h(iT/J)$. Near the tail ends the magnitude of the response is very small and so all the h_i which are obtained by sampling the impulse response near its tails are very close to zero and hence to each other. As can be seen these h_i 's occupy the ends of the vector \mathbf{h}_i . Two vectors become very "similar" to each other if the number of almost-equal-elements becomes much more than the number of different elements and therefore a matrix composed of these vectors becomes ill-conditioned. It is therefore because of these end-elements that the \mathbf{H} matrix becomes ill-conditioned. If we can get rid of the end-elements then we could use the Basic Subspace algorithm (for a small \mathbf{H} matrix).

We now note that the tail of the impulse response contributes very little to the actual received signal and so we can neglect its contribution for our purpose (though it will be used in finding the transmitted sequence once the full impulse response is estimated). If we neglect the tail, the total length of the impulse becomes much less than it was before. Therefore the *effective length* of the impulse response is reduced from the original JT to the smaller $J'T$. We therefore need to sample the received signal at J' times the baud rate and perform the Basic Subspace algorithm under the assumption that the length of the response is only $J'T$. This will give us an estimate of the transmitted sequence \hat{s}_k and an estimate of the shortened impulse response $\hat{\mathbf{h}}_s$. We can then use the estimate of the transmitted sequence \hat{s}_k to obtain a very good estimate of the full (unshortened) impulse response by solving the matrix least squares estimation problem of minimizing $\|\mathbf{S}\mathbf{H} - \mathbf{Y}\|$ over all matrices \mathbf{H} . We could either solve for \mathbf{H} by using the estimated \mathbf{S} matrix directly or by adopting one of following search-approaches to improve the estimate of the \mathbf{S} matrix and using the 'best' \mathbf{S} matrix thus obtained.

Below we present the EC-LS-Subspace algorithm, a modification of the Basic Subspace algorithm, which makes use of the ideas discussed above. This modification allows us to use the Basic Subspace algorithm even in cases when the \mathbf{H} matrix is ill-conditioned, but when the impulse response can be shortened and the \mathbf{H} matrix for the shortened impulse response is well-conditioned.

4.2. Error correcting LS-Subspace algorithm

Once we have a first estimate of the transmitted symbols using the shortened channel, we can then improve the estimate of s_k and hence obtain a very good estimate of the complete impulse response \mathbf{h} . We propose the following method for improving the estimate.

Using the shortened impulse response \mathbf{h}_s , we can find the estimates $\hat{\mathbf{s}}_i$ of the vectors \mathbf{s}_i . These vectors are then thrown into the symbols set \mathcal{S} by hard-limiting. But since the vectors \mathbf{s}_i have overlapping elements, we have the problem of choosing the right value in case the overlapping elements of $\hat{\mathbf{s}}_i$ do not coincide.

To illustrate this point let us look at the following example in which $\mathcal{S} = \{\pm 1\}$ and we have the two ‘transmitted vectors’

$$\mathbf{s}_0 = [1, 1, -1, -1, 1, 1, -1, 1], \quad \mathbf{s}_1 = [1, -1, -1, 1, 1, -1, 1, -1]$$

and also the two ‘received’ estimates

$$\hat{\mathbf{s}}_0 = [0.9, 0.8, -0.5, -0.4, 0.5, 0.2, -0.1, 0.4], \quad \hat{\mathbf{s}}_1 = [0.3, -0.4, -0.7, 0.8, -0.1, -0.3, 0.1, -0.8],$$

which hard-limit to

$$\tilde{\mathbf{s}}_0 = [1, 1, -1, -1, 1, 1, -1, 1], \quad \tilde{\mathbf{s}}_1 = [1, -1, -1, 1, -1, -1, 1, -1].$$

As can be seen $\tilde{\mathbf{s}}_1$ has an error in the 5th bit, but the corresponding bit in $\tilde{\mathbf{s}}_0$ (i.e. the 6th bit) is correct. So there is clearly a contradiction and this signals the fact that there is an error in one of the estimates. If we use the arithmetic average of the two estimates $\tilde{\mathbf{s}}_0$ and $\tilde{\mathbf{s}}_1$ we will get zeros in places where the two estimates do not match. If there are d places in which we get zeros, then the correct estimate (at least in those d positions) is clearly one of the 2^d possible sequences (with ± 1 in the zero-positions). We can therefore apply the LS-search algorithm to this smaller search area and since we would expect d to be quite small, the LS-search algorithm is quite practical. For instance in our simulations d was in most cases 1 or 2, and in rare cases went up to maximum of 6 and therefore in all cases the LS-search added only a small computational cost. Therefore the first step in all the versions in the sequel is compensating for these inherently self-detected errors. These represent something analogous to receiving an erasure—we know that there is an error in that position and so it does not take too much effort to correct it. Let the estimate after correcting for the erasures be the sequence $\{\hat{s}_k\}$. Now we can incorporate different levels of sophistication into the algorithm depending on the purpose for which it is needed. The following are some of the levels of sophistication that we propose:

- Level 1:* We need not make any correction at all. Just use the arithmetic average of the J estimates for $\{s_k\}$ to construct the matrix \mathbf{S} and find the LS estimate for \mathbf{h} . We do not search for the ‘best’ LS estimate. We just use the \mathbf{S} matrix that we get. The advantage is that this has a small computational cost. This will be useful in applications in which computational cost is a much bigger criteria as compared to the accuracy. But at any rate even here the accuracy of the estimate is much better than the conventional algorithms.
- Level 2:* We can just correct the erasures by the procedure already discussed. This will increase the computational cost a little but increases the accuracy of the estimate. We will hence-forth refer to this algorithm as the *Erasure-correcting LS-Subspace algorithm* or the *ERC-LS-Subspace algorithm*.
- Level 3:* We can introduce one more level of correction by correcting for single errors. For this we search for the LS-minimum over the space of all matrices \mathbf{S} for which the corresponding sequence $\{s_k\}$ differs from the estimate $\{\hat{s}_k\}$ in at most one position. By this procedure we can find the best LS estimate if it lies within a distance of a single bit from the erasure-corrected estimate.

Level 4: We can generalize Level 3 by correcting for n -errors. For this we will have to expand the search area to cover all the matrices \mathbf{S} for which the corresponding sequence $\{s_k\}$ differs from the estimate $\{\hat{s}_k\}$ in at most n positions.

The problem with the last method is that the computational cost increases almost exponentially as n increases, but the corresponding increase in accuracy is not so significant since in most cases there are very few errors. Even with one-error or two-error correcting procedure the estimate becomes very close to the exhaustive search LS algorithm, which is asymptotically the ‘best’ possible estimate we can have. We can avoid the increasing computational cost with increasing n by adopting the following method which is by far the best method we have come across both in maintaining a low computational cost as well as in achieving an estimate almost as good as the exhaustive search LS algorithm. By ‘first estimate’ we shall mean the sequence obtained by correcting for the erasures i.e. the sequence obtained at the end of Level 2.

The EC-LS-Subspace algorithm

1. First let the search area be all the sequences which differ from the ‘first sequence’ in one position.
2. Find the ‘best’ sequence in this area.
3. If this sequence is the same as the ‘first sequence’ then stop and decide that this is the ‘best’ estimate.
4. If this sequence is not the same as the ‘first sequence’ then replace the ‘first sequence’ by this sequence and start all over again from step 1.

Why will this method work? The reason is that in most cases if the actual ‘best’ sequence differs from the estimated sequence in n positions then the ‘best’ sequence can be reached from the estimated sequence in steps and the corresponding sequences in the intermediate steps will be ‘best’ sequences in their respective local areas. It is very clear that this method reduces the computational complexity since at each stage we only test the sequences which differ in one position (if the number of symbols is m then it means m LS-computations) and we will have in most cases, n stages if there are n positions which have errors. This means we will totally have mn LS-computations in contrast to

$$\sum_{i=0}^n \binom{m}{i}$$

LS-computations if we use a search area with an ability to correct n errors. We will henceforth call this algorithm the *Error-correcting-LS-Subspace algorithm* or the *EC-LS-Subspace algorithm*.

An important point to note here is that we have used the intuitively clear fact that in most cases if the starting point is quite close to the global minimum, then the global minimum of a set can be reached by finding the successive local minimums in smaller subsets (where the successive subsets are chosen properly depending on the previous local minimum). This point is very well illustrated by the Fig. 2 where each search area is a circle centered at the local minimum of the previous search area. The stopping condition of the algorithm is reached when the local minimum lies in the interior of its subset—this condition is analogous to the condition in elementary calculus, that, when a function attains its minimum or maximum, the derivative of the function is zero. The great reduction in the computational cost results because of a two-fold effects:

- In the n -error correcting algorithm the original area is of size

$$\sum_{i=0}^n \binom{m}{i}$$

as compared to a maximum search area for the EC-LS-Subspace algorithm (i.e. mn).

- The n -error correcting algorithm is designed for the worst case (i.e. n errors) and even if there are fewer errors the whole area will have to be searched whereas the EC-LS-Subspace algorithm is designed to be iterative and so when there are fewer errors (as is most often the case) it searches only a correspondingly smaller area.

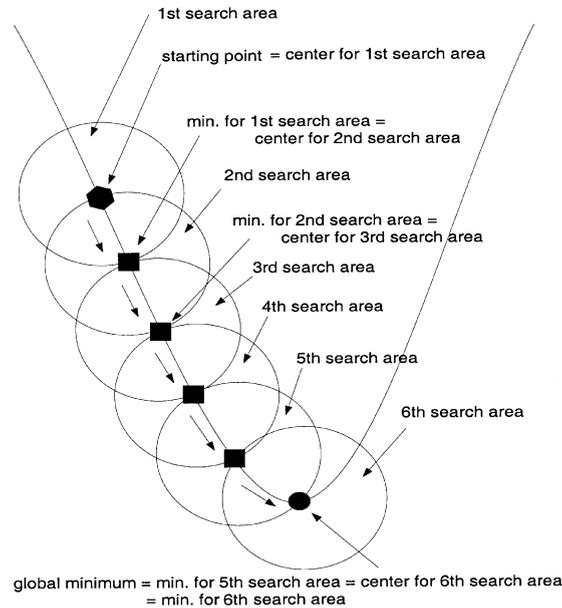


Fig. 2. Graphical demonstration of the fact that the global (interior) minimum can often be reached through successive local (boundary) minimums.

But since this is an iterative method it may be argued that it will have the associated problems of convergence and will not be able to find the global minimum if one of the local minimums lies in the interior of its subset. It may either need too many iterations or may not converge if the global minimum cannot be reached by moving through successive local minimums. But our simulation studies have shown that in almost all the cases (in fact in all the cases in our simulations) this method converges without any problems. We can take care of even the rare cases when there are convergence problems, by specifying an upper limit to the number of iterations, and, if the algorithm has not ended before the upper limit is reached, we can either discard the result obtained and look for a fresh set of samples, or, simply use the original estimate without corrections for finding \mathbf{h} . But these are merely technical issues, since, as our simulation results in the next section show, this algorithm is very good on all counts—low computational cost, high accuracy of estimate, and small observation interval.

5. Simulation results

We did extensive simulations using the EC-LS-Subspace algorithm, the algorithm by Tong et al. [1], the algorithm by Moulines et al. [7], the algorithm by Xu et al. [16] and the algorithm by Liu and Xu [17], and made a comparative study. Here we present one simulation example which uses the same channel as used in the simulation example of [1]. The impulse response is shown in Fig. 1 and in Table 1. The source symbols were drawn from a BPSK signal constellation with a uniform distribution.

A simulation of 100 independent trials was conducted for each algorithm under the same simulation scenario. Figs. 3a and b show 100 estimates of the channel obtained by using the EC-LS-Subspace algorithm with 30 and 100 symbols, respectively. In both cases the SNR is 30 dB. Similarly Figs. 4a and b, 5a and b, 6a and b and 7a and b show 100 estimates of the channel obtained by using the

Table 1
Channel impulse response

n	1	2	3	4	5	6
$h(n)$	-0.02788	-0.01556	0.009773	0.0343	0.04142	0.0216
n	7	8	9	10	11	12
$h(n)$	-0.01959	-0.06035	-0.07025	-0.0241	0.08427	0.2351
n	13	14	15	16	17	18
$h(n)$	0.3874	0.4931	0.5167	0.4494	0.3132	0.152
n	19	20	21	22	23	24
$h(n)$	0.01383	-0.06754	-0.08374	-0.05137	-0.001258	0.03679

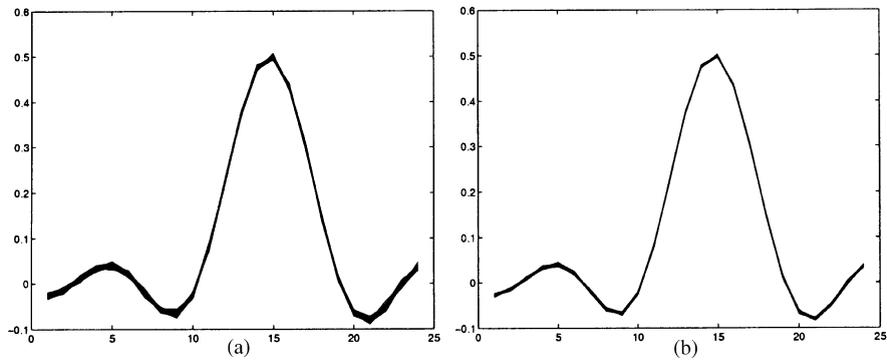


Fig. 3. 100 estimates, EC-LS-Subspace algorithm, 30 and 100 symbols, SNR= 30 dB.

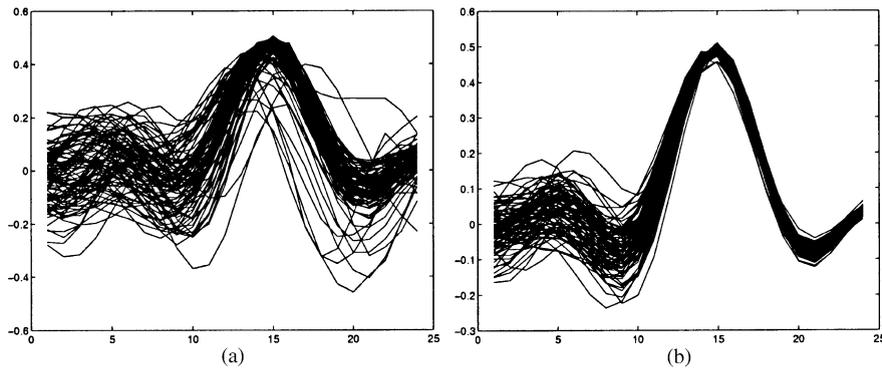


Fig. 4. 100 estimates, algorithm by Tong et al. [1], 30 and 100 symbols, SNR=30 dB.

algorithms in [1,7,16,17], respectively (the SNR is 30 dB and 30 symbols are used to obtain plots (a) and 100 symbols are used for obtaining plots (b)).

Figs. 8a–d show the 100 estimates for the EC-LS-Subspace algorithm for SNR = 20, 15, 10 and 5 dB, respectively. Simulations showed that even the few aberrations that occur in Figs. 8b and d vanish if the number of symbols used in the estimation is increased slightly. These aberrations occur because the EC-LS-Subspace algorithm is not able to converge to the correct sequence as the starting sequence was very different from the correct sequence. The starting sequence is very different from the correct sequence because the Φ matrix does not have a one-dimensional null space. We proved in Section 3, that the

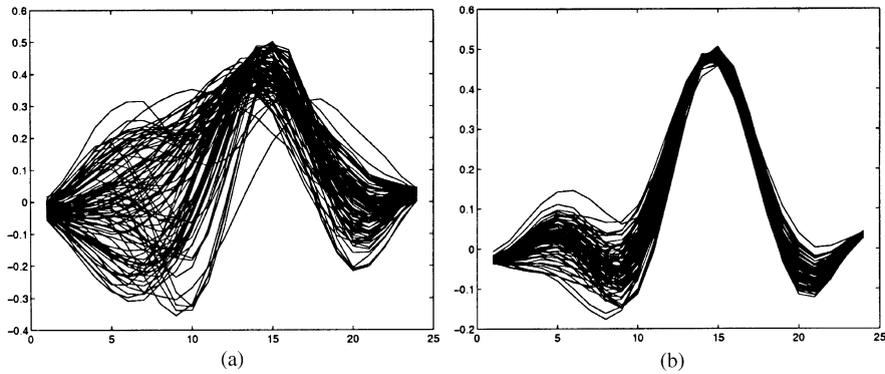


Fig. 5. 100 estimates, algorithm by Moulines et al. [7], 30 and 100 symbols, SNR= 30 dB.

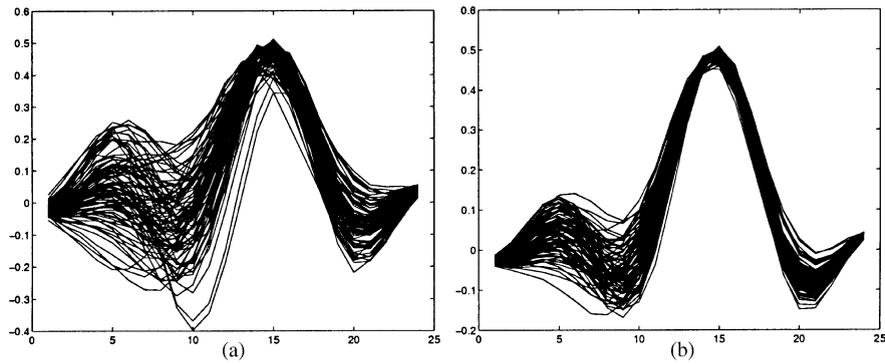


Fig. 6. 100 estimates, algorithm by Xu et al. [16], 30 and 100 symbols, SNR= 30 dB.

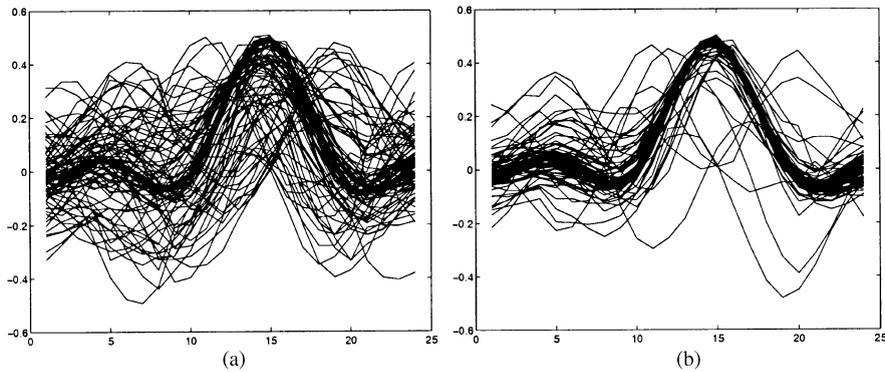


Fig. 7. 100 estimates, algorithm by Liu and Xu [17], 30 and 100 symbols, SNR=30 dB.

probability that the Φ matrix does not have a one-dimensional null space decreases exponentially with an increase in the number of symbols. Therefore if the number of symbols increases even a little, the probability that the starting sequence is very different from the correct one decreases rapidly and hence the aberrations vanish.

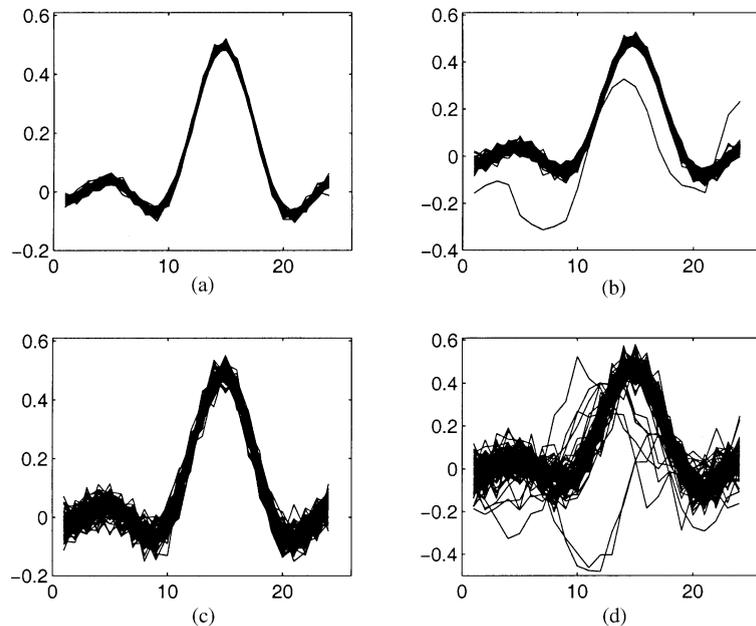


Fig. 8. 100 estimates of the channel for (a) SNR= 20 dB (b) SNR= 15 dB (c) SNR= 10 dB and (d) SNR= 5 dB for the EC-LS-Subspace algorithm. In all cases the number of symbols used for each estimate is 30.

To obtain a performance measure of the channel estimation, the normalized root-mean-square error (NRMSE) of the estimator is defined by

$$\text{NRMSE} = \frac{1}{\|\mathbf{h}\|} \sqrt{\frac{1}{M} \sum_{i=1}^M \|\hat{\mathbf{h}}_{(i)} - \mathbf{h}\|^2}$$

where M is the number for independent trials (100 in our case), and $\hat{\mathbf{h}}_{(i)}$ is the estimate of the channel from the i th trial. Fig. 9 shows the NRMSE's of the different algorithms versus SNR in a series of 100 independent runs using 35 symbols for each estimate. It can be seen that the EC-LS-Subspace algorithm converges to a much smaller NRMSE than the algorithm in [16,17] or [1]. The asymptotic performance of the algorithm by Liu and Xu [17], seems to be quite close to that of the EC-LS-Subspace algorithm, although at lower SNR, the EC-LS-Subspace algorithm performs much better than the algorithm by Liu and Xu [17].

We also compared the bit error rate (BER) against the SNR for each of the algorithms under consideration. Fig. 10 shows the average bit error rate vs. SNR curves for the EC-LS-Subspace algorithm and for the algorithms in [1,7,16,17]. These curves were obtained by first using 30 symbols for channel identification and then evaluating the bit error rate using the identified channel. The average bit error rate was obtained by averaging the BERs thus obtained over several channel estimates.

Fig. 11 describes the computational complexity of the different algorithms under consideration. The figure shows the plot of the number of floating point operations (FLOPS) used in the MATLAB program as a function of the number of symbols used for the estimate for each of the algorithms. These plots give us a general idea of the complexity of each algorithm. The algorithms were not optimized for the

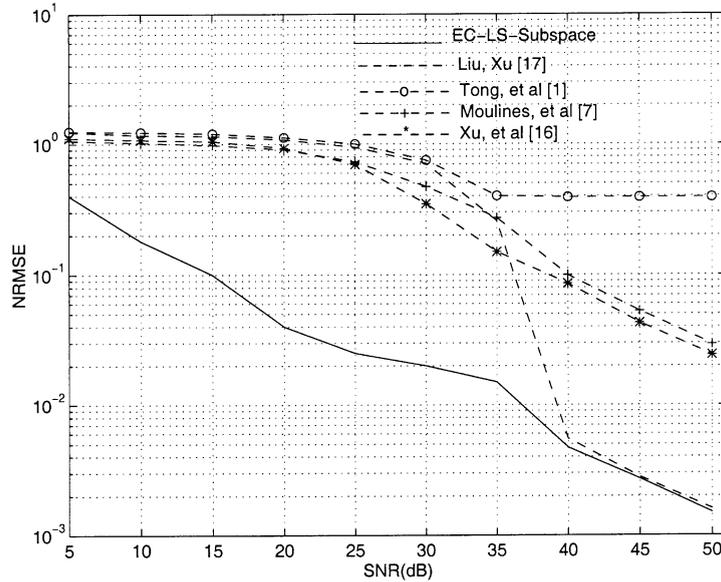


Fig. 9. NRMSE versus SNR, 100 independent runs were used for the estimates and each estimate of the channel used 35 symbols.

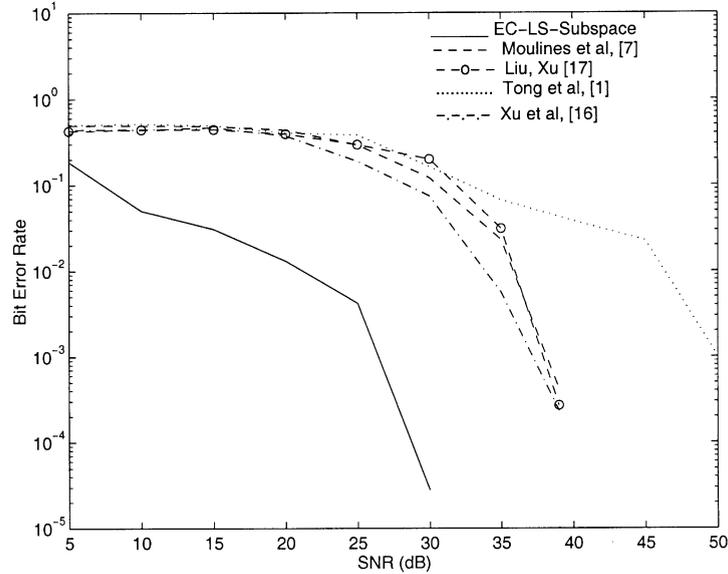


Fig. 10. Bit error rate versus SNR. 30 symbols were used to estimate the channels.

number of operations and so the estimates shown are slightly more than the actual complexity. A glance at the plots shows that at small observation intervals (which will be the running condition in practice) the complexity of the EC-LS-Subspace algorithm is even less than Tong’s algorithm.

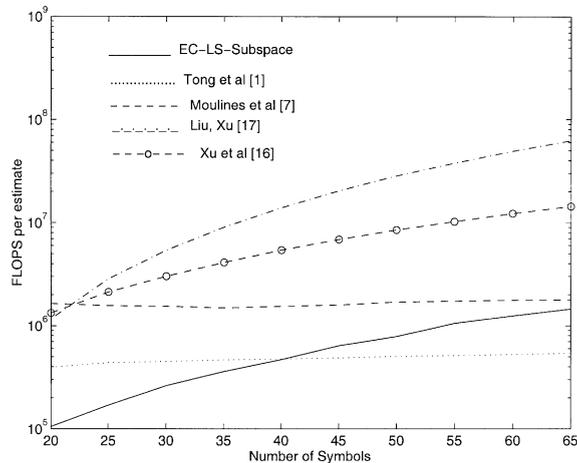


Fig. 11. Number of FLOPS per estimate versus number of symbols used for each estimate. SNR= 30 dB.

6. Conclusions

In this paper, we have proposed a new method for blind identification using subspace and least-square techniques. By exploiting the inherent structure in the received signal and using the ability to correct errors we are able to obtain a fast identification procedure which uses very few symbols. The proposed method leads to a very accurate estimate of the impulse response with a small observation interval. Also, the algorithm is very robust, has a low complexity and performs quite well even under low SNR. So it can be used in practical applications which use rapidly varying channels with very low SNR's.

7. Uncited references

[11,13,14,15]

References

- [1] L. Tong, G. Xu, T. Kailath, Blind identification and equalization based on second-order statistics: a time domain approach, *IEEE Trans. Inform. Theory* 40 (March 1994) 340–350.
- [2] W.A. Gardner, W.A. Brown, Frequency-shift filtering theory for adaptive co-channel interference removal, *Proceedings of the 23rd Asilomar Conference on Signals, Systems and Computing*, Pacific Grove, CA, October 1989, pp. 562–567.
- [3] G. Giannakis, Y. Inouye, J. Mendel, Cumulant-based identification of multichannel moving average models, *IEEE Trans. Commun.* 34 (1989) 783–787.
- [4] G. Giannakis, J. Mendel, Identification of non-minimum phase systems using higher-order statistics, *IEEE Trans. Acoust. Speech, Signal Process.* 37 (1989) 360–377.
- [5] F. Gustafsson, Blind equalization by direct examination of the input sequences, *IEEE Trans. Commun.* 43 (July 1995) 2213–2222.
- [6] D. Hatzinakos, Nonminimum phase channel deconvolution using the complex cepstrum of the cyclic autocorrelation, *IEEE Trans. Signal Process.* 42 (November 1994) 3026–3042.
- [7] E. Moulines, P. Duhamel, J. Cardoso, S. Mayrargue, Subspace methods for the blind identification of multichannel FIR filters, *IEEE Trans. Signal Process.* 43 (February 1994) 516–525.
- [8] Y. Li, Z. Ding, Blind channel identification based on second order cyclostationary statistics, *Proceedings of the International Conference on Acoustics Speech, Signal Process.*, April 1993, pp. IV: 81–84.

- [9] J. Mendel, Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications, *Proc. IEEE* 79 (March 1991) 278–305.
- [10] C. Nikias, Blind deconvolution using higher-order statistics, *Proceedings of the 2nd International Conference Higher-Order Statistics Elsevier, Amsterdam, 1992*, pp. 49–56.
- [11] Sampath, *Subspace Approaches for Blind Equalization and Identification*, Ph.D. Dissertation, University of Maryland, College Park, 1997.
- [12] Sampath, Y. Li, K.J.R. Liu, A Subspace Based Blind Identification and Equalization Algorithm, *Proceedings of the IEEE International Conference on Communications (ICC), Dallas, June 1996*, pp. 1010–1014.
- [13] D.T.M. Slock, Blind fractionally spaced equalization, perfect-reconstruction filter banks and multichannel linear prediction, *Proceedings of the IEEE ICASSP, 1994*, pp. IV:573–576.
- [14] S. Talwar, M. Viberg, A. Paulraj, Blind estimation of multiple co-channel digital signals using antenna array, *IEEE Signal Process. Lett.* 1 (February 1994) 29–31.
- [15] D. Yellin, B. Porat, Blind identification of FIR systems excited by discrete-alphabet inputs, *IEEE Trans. Signal Process.* 41 (March 1993) 1331–1339.
- [16] G. Xu, H. Liu, L. Tong, T. Kailath, A least-squares approach to blind channel equalization, *IEEE Trans. Signal Process.* 43 (December 1995) 2982–2993.
- [17] H. Liu, G. Xu, Closed-form blind symbol estimation in digital communications, *IEEE Trans. Signal Process.* 43 (November 1995) 2714–2723.