A Subspace Based Blind Identification and Equalization Algorithm

Balaji Sampath, Ye Li and K. J. Ray Liu
Electrical Engineering Department and Institute for Systems Research
University of Maryland, College Park, MD 20742
E-mail: bsampath(liye, kjrliu)@src.umd.edu, Fax: (301) 314-9281

Abstract— **A subspace based blind channel identification algorithm is proposed here. This 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, this algorithm uses the least number of symbols that can possibly be used to identify the channel exactly. In the noisy case, simulations have shown that almost exact identification can be obtained by using a few more symbols than the theoretical minimum. This is orders of magnitude better than the other blind algorithms. Once the channel has been identified by using this procedure, any of the existing equalization techniques can be used along with it to obtain the symbols.**

I. Introduction

Most digital communication systems are subject to intersymbol interference (ISI)and therefore specific identification and equalization procedures are needed. Classically, the channel is identified 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 this, blind channel identification methods do not require training sequences. Existing blind identification methods use the statistics of the transmitted sequence instead of the explicit knowledge of the sequence itself. Since communication channels are very likely to be nonminimum phase, most of the existing blind channel identification algorithms have used higher order statistics [10]-[13]. A different approach by Tong, Xu, and Kailath [1] allows the blind identification of channels using only the second-order cyclo-stationary statistics. Recently a number of algorithms [3]-[9] have been proposed which also make use of this idea.

But none of these algorithms have used the signal structure inherent in the oversampled channel output. In our approach we make full use of this structure and therefore find that it is not necessary to explicitly use any statistics. For the noisy case a modification of this algorithm can be obtained by incorporating least-square techniques. By doing this we not only use the explicit knowledge of the transmitted symbols but also implicitly make use of the statistics of all orders and not just the second or third order statistics. This algorithm provides very good estimates and needs very few symbols for estimating the channel. Moreover, it is quite robust to noise and therfore it can be used in practical situations.

The received baseband signal x(·) can be written as

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

$$x(t) = r(t) + n(t) \tag{2}$$

where s_k is an information symbol in a signal constellation $\mathcal{S}$ ($\mathcal{S}$ may be an infinite set), h(·) 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 $x(\cdot)$.

The organization of the paper is as follows - in Section 2, we derive the Basic Subspace algorithm and this forms the backbone of our approach. In Section 3 we derive the conditons for the identifiability of the channels using this method. In Section 4, we modify the Basic Subspace algorithm and obtain a very robust algorithm which works very well in practical situations. Finally we discuss one simulation example and the conclusions that can be drawn from it.

II. 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 assume that the impulse response has length JT and that the received signal is not corrupted by noise. Here the received signal is sampled at J times the baud rate. It is easy to see that we obtain the following equations for $0 \leq i \leq J-1$:

$$\begin{aligned} y(i) &= s_{J-1}h_i + s_{J-2}h_{J+i} + \cdots + s_0 h_{(J-1)J+i} \\ y(J+i) &= s_J h_i + s_{J-1}h_{J+i} + \cdots + s_1 h_{(J-1)J+i} \\ \vdots \; &\vdots \; \vdots \\ y(JM-J+i) &= s_{M+J-2}h_i + \cdots + s_{M-1}h_{(J-1)J+i} \end{aligned}$$

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

Define the vectors $\mathbf{x}_i$ and $\mathbf{s}_i$ for $0 \leq i \leq J-1$ as below

$$\mathbf{x}_i = \begin{bmatrix} y(i) & y(J+i) & y(2J+i) & \cdots & y((M-1)J+i) \end{bmatrix}^T$$

$$\mathbf{s}_i = \begin{bmatrix} s_i & s_{i+1} & s_{i+2} & \cdots & s_{M+i-1} \end{bmatrix}^T.$$

It is easy to obtain the relation

$$\mathbf{X} = \mathbf{SH} \tag{3}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_{J-2} & \mathbf{x}_{J-1} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_0 & \mathbf{s}_1 & \cdots & \mathbf{s}_{J-2} & \mathbf{s}_{J-1} \end{bmatrix}$$

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

$$\mathbf{h}_k = \begin{bmatrix} h_{J(J-1)+k-1} & h_{J(J-2)+k-1} & \cdots & h_{J+k-1} & h_{k-1} \end{bmatrix}^T.$$

* The work was supported in part by the NSF grants MIP9309506 and MIP9457397.

0-7803-3250-4/96$5.00©1996 IEEE

Assuming that $\mathbf{H}$ is invertible we can invert this relation and we then see that $\mathbf{s}_i \in span\{\mathbf{x}_0, \mathbf{x}_0, \cdots, \mathbf{x}_{J-1}\}$. Therefore we have for $0 \leq i \leq J-1$,

$$\mathbf{s}_i = \lambda_0^{(i)}\mathbf{x}_0 + \lambda_1^{(i)}\mathbf{x}_1 + \cdots + \lambda_{J-1}^{(i)}\mathbf{x}_{J-1} \tag{4}$$

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

The vectors $\{\mathbf{s}_i\}$ have a very special structure : bottom $M-1$ elements of $\mathbf{s}_i$ is the same as the top $M-1$ elements of $\mathbf{s}_{i+1}$. 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 $\mathbf{s}_i$ translates into the following relation.

$$\mathbf{\Phi}\lambda = 0 \tag{5}$$

where

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

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

with the number of block-columns being J, 0 representing a matrix of zeros and

$$\xi = \begin{bmatrix} \xi_0 & \xi_1 & \cdots & \xi_{J-1} \end{bmatrix}, \quad \eta = -\begin{bmatrix} \eta_0 & \eta_1 & \cdots & \eta_{J-1} \end{bmatrix} \tag{6}$$

Therefore we know that $\lambda \in Null(\mathbf{\Phi})$, and if we know that $\mathbf{\Phi}$ has a one-dimensional null space, then we can find λ uniquely (upto a multiplication factor) and therefore $\mathbf{H} = \mathbf{\Lambda}^{-1}$ can be determined from it.

III. Identifiability

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

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

The first assumption is violated in systems where $\mathbf{H}$ is ill-conditioned. In Section 4, we modify the algorithm slightly to take care of this. The second condition can be easily taken care of since it is hardly ever violated. It can be shown that the following theorem is true.

Theorem 1: For every non-trivial [1] communication system, if $\mathbf{H}$ is invertible then the probability that $\mathbf{\Phi}$ has a one-dimensional null space tends to 1 as the observation interval increases. The rate of convergence is atleast exponential if the symbols are independant. Theorem 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 persistantly exciting sequences for the identifiability of a FIR channel have been discussed in [15].

IV. 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.

$$\mathbf{Sh} + \mathbf{n} = \mathbf{y} \tag{7}$$

where

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

$$\mathbf{h} = [h_{J-1}, h_{J-2}, \cdots, h_1, h_0]^T \quad , \quad h_i = h(iT)$$
$$\mathbf{n} = [n_J, n_{J+1}, n_{J+2}, \cdots, n_M]^T \quad , \quad n_i = n(iT)$$
$$\mathbf{y} = [y_J, y_{J+1}, y_{J+2}, \cdots, y_M]^T \quad , \quad y_i = x(iT)$$

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 $\|\mathbf{Sh}-\mathbf{y}\|^2$. 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 $\|\mathbf{Sh}-\mathbf{y}\|^2$. 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 is easy to show that the variance of the estimated $\mathbf{h}$ will converge to the Cramer-Rao bound for known signal and therefore this algorithm is asymptotically the 'best' blind equalization algorithm one can have [2].

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 direct use of the statistics of the received signal.

A. 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 there is a way of circumventing this problem as we now illustrate.

Why is it that in many channels the $\mathbf{H}$ matrix is ill-conditioned? The 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

$$\mathbf{h}_k = \begin{bmatrix} h_{J(J-1)+k-1} & h_{J(J-2)+k-1} & \cdots & h_{J+k-1} & h_{k-1} \end{bmatrix}^T$$

where $h_i = h(\frac{iT}{J})$. 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

[1] 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!

of these vectors becomes ill-conditioned. It is therefore because of these end-elements that the **H** matrix becomes ill-conditioned. If we can get rid of the end-elements then we could use the Basic Subspace algorithm (for a smaller **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 to $J'T$ from the original JT. For HF channel and mobile radio channels J' is usually very small - about 2 or 3. 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 shortend 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 $||\mathbf{SH} - \mathbf{Y}||$ over all matrices **H**. We could either solve of **H** by using the estimated **S** matrix directly or by adopting one of following search-approaches to improve the estimate of the **S** matrix and using the 'best' **S** matrix thus obtained.

B. Approaches for Performance Improvement

Once we have a first estimate of the transmitted symbols we can then use a number of methods to improve our estimate of s_k and hence obtain a very good estimate of the impulse response **h**. The following are some of the proposed methods.

B.1 LS-Subspace Algorithms with Error Correction

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 symbol 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. Assume that the symbol set has only two elements, ie. $\mathcal{S} = \pm 1$. If we use the arithmetic average of the estimates $\hat{\mathbf{s}}_i$ we will get zeros in places where the estimates do not match. If there are d such places then the correct estimate (atleast 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. 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 doesn't 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 sophistry that we propose.

Level 1 We need not make any correction at all. Just use the arithmetic average of the two estimates for $\{s_k\}$ to construct the matrix **S** and find the LS estimate for **h**. We do not search for the 'best' LS estimate. We just use the **S** matrix that we obtain [2]. 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 henceforth 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 **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 **S** for which the corresponding sequence $\{s_k\}$ differs from the estimate $\{\hat{s}_k\}$ in at most n positions.

We can avoid the high computational cost associated with the last method 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 in Level 2.

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.

From our simulations we have found that this algorithm works exceptionally well in practical situations. We have done done exhaustive simulations using this algorithm and we have found that in terms of accuracy, number of symbols needed for the estimate, robustness to noise and computational complexity this algorithm is very good. Our simulation example documents and supports this fact. We will call this algorithm the **Error-Correcting Least Square Subspace algorithm** or the **EC-LS-Subspace algorithm.**

B.2 The LS Iteration Algorithm

Once we have a sufficiently good estimate of the transmitted symbols, we can employ various error correcting mechanisms to improve the performance. Below we discuss another method which attempts to improve the performance of the LS-Subspace algorithm iteratively.

Note that we have to solve for the **S** and the **H** matrices and they are related by

$$\mathbf{SH} + \mathbf{N} = \mathbf{Y}$$

where **N** and **Y** represent the noise and received matrices. Our aim is therefore to minimize the error, $||\mathbf{SH} - \mathbf{Y}||$ over all the possible matrices **S** and **H**. It is therefore a joint minimization problem over a two-dimensional matrix space . It might therefore be possible to solve it iteratively, by finding the minimum over a one-dimensional matrix space at each step.

We have an estimate of the **S** matrix obtained from the Basic Subspace algorithm. The idea is that we can solve for the least squares **H** by using the estimated **S** and then obtain a 'better' estimate of the **S** matrix from the equation $\mathbf{S} = \mathbf{Y}\mathbf{H}^{\dagger}$. We can

[2] It may have a few zeros but that does not matter since there are very few zeros anyway!

then repeat this procedure iteratively untill a good estimate for **H** is obtained. A simulation study of this algorithm showed that this method is not as good as the EC-LS-Subspace algorithm.

V. A Simulation Example

For our simulation we used the same channel which was used in the simulation example of [1] and the impulse response is shown in the Fig. 1. The source symbols were drawn from a BPSK signal constellation with a uniform distribution. We implemented both - our algorithms and the algorithm by Tong et al [1] and made a comparative study.

A simulation of 100 independent trials was conducted for each algorithm under the same simulation scenario. Fig.2a, Fig.2b, Fig.2c and Fig.2d show the 100 estimates for the EC-LS-Subspace algorithm, ERC-LS-Subspace algorithm, the LS-Iteration algorithm and the algorithm by Tong et al. [1] each estimate using only 30 symbols. Fig.3a and Fig.3b show the 100 estimates for the EC-LS-Subspace algorithm and Tong's algorithm when each estimate is based on 100 symbols. These graphs show clearly that even if the number of symbols used for each estimate increases, the EC-LS-Subspace algorithm performs much better than Tong's algorithm. Fig.4a, Fig.4b, Fig.4c and Fig.4d show the 100 estimates for the EC-LS-Subspace algorithm for SNR = 20,15,10 and 5 dB respectively.

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

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

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.5a shows the NRMSE's of the different algorithms versus SNR in a series of 100 independent runs using 35 symbols for each estimate. Fig.5b shows the NRMSE's of the EC-LS-Subspace algorithm versus SNR in a series of 100 independent runs using different numbers of symbols for each set in the series. These plots show that unlike Tong's algorithm, the subspace algorithms do not have a cut-off SNR below which the performance decreases rapidly. Moreover it can be seen that the subspace algorithms converge to a much smaller NRMSE than Tong's algorithm. Initially, with an increase in the number of symbols, the performance of the EC-LS-Subspace algorithm improves but the rate of this increase decreases and beyond 35 symbols the improvement in performance is not very significant since it's performance is already very close to that of the conventional least squares algorithm.

Fig.6 describe the computational complexity of the different algorithms under different running conditions. Fig.6a 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. Fig.6b shows the same plot for the EC-LS-Subspace algorithm for different SNR's. These plots give us a general idea of the complexity of each algorithm. The algorithms were not optimized for the number of operations and so the estimates shown are well-above the actual complexity.

VI. Conclusions

Blind Identification and Equalization is very useful in many communication systems. Traditionally blind identification algorithms need a large observation interval to obtain a reasonable estimate of the channel. This is because they have used higher order statistics for making their estimates. Even a cyclo-stationary second order statistics based approach by Tong, Xu and Kailath [1] needs a very large obeservation interval as compared to identification algorithms which use training sequences. This has therefore prevented an effective use of blind equaliz^tion algorithms in communication systems.

We have therefore proposed a new method for blind identification using subspace and least-square techiniques. 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, slightly more than what is used in identification procedures with training sequences. The proposed method leads to a very accurate estimate (much better than any of the previous algorithms) of the impulse response with a much smaller sample size (comparable to estimation procedures using a training sequence) than any other blind identification algorithm proposed till now. Moreover, the computational complexity of this algorithm is also lower than other algorithms. This is especially true for smaller observation intervals and equivalent performance levels. Furthermore the algorithm performs very well even under small SNR's and so it can be used in practical applications which use rapidly varying channels with very low SNR. The main features of our algorithm are summarized below:

1. This algorithm needs very few symbols. In our simulation example for instance, we need 30 symbols as compared to a 100 symbols in Tong's algorithm [1] and infact our estimate of the impulse response with 30 symbols is much better than Tong's estimate with 100 symbols. This implies that it can be used in a wide range of applications which have rapidly changing channels.
2. The computational cost of the Basic Subspace algorithm is quite small and even with the extra additions for the least squares approach, the complexity is significantly lower than other blind equalization techniques. This is especially true for observation intervals of practical interest.
3. The algorithm provides a wide range of choices to the user and the user can choose between different levels of computational costs, accuracy levels and length of the estimation period depending on his or her need.
4. The performance of this algorithm is very close to the asymptotically 'best' algorithm, i.e. the least squares algorithm.
5. Since the received signal is oversampled, it has a better immunity to noise and interference [1], [14].
6. The algorithm works directly on the data domain and so the problems associated with other algorithms which are based on explicitly computing the statistics are not encountered.
7. The Basic Subspace algorithm (although not the LS-Subspace algorithm) can be used to estimate the channel even when nothing is known about the transmitted sequence. This will be useful in applications when the signal is received from an unknown transmitter.

References

[1] L. Tong, G. Xu and T. Kailath, "Blind identification and equalization based on second-order statistics: A time domain approach," *IEEE Trans. Inform. Theory,* vol. 40, pp. 340-350, Mar. 1994.

[2] D. Yellin and B. Porat, "Blind identification of FIR systems excited by discrete-alphabet inputs," *IEEE Trans. Signal Processing,* vol. 41, pp. 1331-1339, Mar. 1993.

[3] D. Hatzinakos, "Nonminimum phase channel deconvolution using the complex cepstrum of the cyclic autocorrelation," *IEEE Trans. Signal Processing,* vol. 42, pp. 3026-3042, Nov. 1994.

[4] Y.Li and Z.Ding, "Blind channel identification based on second order cyclostationary statistics," *Proc. Int. Conf. Acoust. Speech, Signal Processing.,* pp. IV:81-84, April 1993.

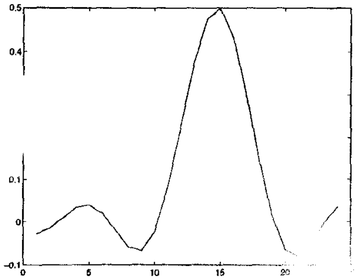


Fig. 1. A three-ray multipath channel impulse

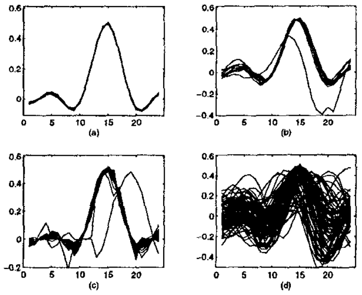


Fig. 2. 100 estimates of the channel using 30 symbols for each estimate for (a) EC-LS-Subspace Algorithm (b) ERC-LS-Subspace Algorithm (c) LS-Iteration Algorithm and (d) Tong's Algorithm. In all cases SNR=30dB.

[5] H. Liu, G. Xu and L. Tong, " A deterministic approach to blind equalization," *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, 1993.

[6] D. T. M. Slock, " Blind fractionally-spaced equalization, perfect-reconstruction filter banks and multichannel linear prediction," *Proc. IEEE ICASSP,* 1994, pp. IV:573-576.

[7] E. Moulines, P. Duhamel, J. Cardoso and S. Mayrargue, " Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. Signal Processing,* vol. 43, pp. 516-525, Feb. 1994.

[8] S. Talwar, M. Viberg and A. Paulraj, " Blind estimation of multiple co-channel digital signlas using antenna array," *IEEE Signal Processing Lett.*, vol. 1, pp. 29-31, Feb. 1994.

[9] A. J. van der Veen, S. Talwar and A. Paulraj, " Blind estimation of multiple digital signals transmitted over FIR channels," *IEEE Signal Processing Lett.*, vol. 2, pp. 99-102, May 1995.

[10] G. Giannakis, Y. Inouye and J. Mendel, "Cumulant-based identification of multichannel moving average models," *IEEE Trans. Commun.*, vol. 34, pp. 783-787, 1989.

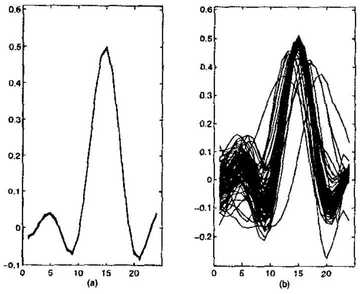
[11] G. Giannakis and J. Mendel, "Identification of non-minimum phase systems using higher-order statistics," *IEEE Trans. Acoust. Speech, Signal Processing.*, vol. 37, pp. 360-377, 1989.

[12] J. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications,"*Proc. IEEE,* vol. 79, pp. 278-305, Mar. 1991.

[13] C. Nikias, "Blind deconvolution using higher-order statistics,"*Proc. 2nd Int. Conf. Higher-Order Stat.* Elsevier, 1992, pp. 49-56.

[14] W. A. Gardner and W. A. Brown, "Frequency-shift filtering theory for adaptive co-channel interference removal,"*Proc. 23rd Asilomar Conf. Signals,Syst., and Comput.*, Pacific Grove, CA, Oct.1989, pp. 562-567.

[15] F. Gustafsson, "Blind equalization by direct examination of the input sequences," *IEEE Trans. Commun.*, vol. 43, pp. 2213-2222, July 1995.



[illegible]ig. 3. 1[illegible]timates of the channel using 100 symbols for each estimate [illegible]space Algorithm (b) Tong's Algorithm. In both [illegible]es SNR=[illegible]

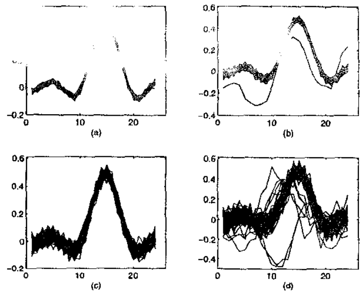


Fig. 4. 100 estimates of the channel for (a) SNR=20 dB (b) SNR=15 dB (c) SNR=10 dB and (d) SNR=5 dB for the EC-LS-Subspace Algorithm. In all cases the number of symbols used for each estimate=30.

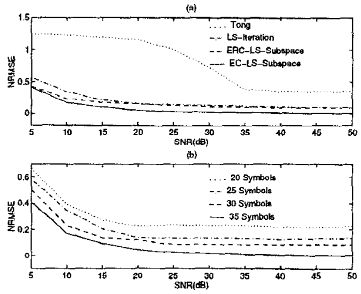


Fig. 5. NRMSE versus SNR. 100 independent runs were used for the estimates and (a) Each estimate of the channel used 35 symbols (b) All estimates are for the EC-LS-Subspace Algorithm

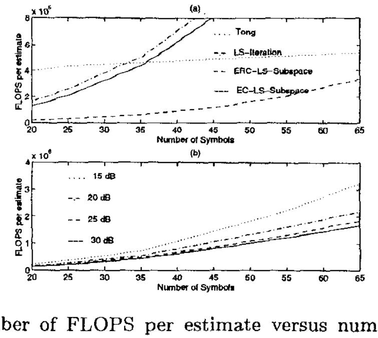


Fig. 6. Number of FLOPS per estimate versus number of symbols used for each estimate. (a) Different Algorithms, SNR=30 dB (b) Different SNR's for the EC-LS-Subspace Algorithm