BLIND CHANNEL IDENTIFICATION BY SUBSPACE TRACKING AND SUCCESSIVE CANCELLATION

Xiaohua Li

Department of EE
SUNY at Binghamton
Binghamton, NY 13902
xli@binghamton.edu

H. Howard Fan

Department of ECECS
University of Cincinnati
Cincinnati, OH 45221
h.fan@uc.edu

ABSTRACT

Traditional subspace methods (SS) for blind channel identification require accurate rank estimation with a computational complexity of $O(m^3)$, where m is the data vector length. In this paper, we introduce new adaptive subspace algorithms using ULV updating and successive cancellation techniques. In addition to reduce the computational complexity to $O(m^2)$, the new algorithms do not need to estimate the subspace rank. Channel length can be over-estimated during the subspace tracking and channel vector optimization steps. It can then be recovered at the end by a successive cancellation procedure. Simulation shows that the new algorithms outperform the traditional SS methods in case the subspace rank is difficult to estimate.

1. INTRODUCTION

Many digital communication channels suffer from the problem of intersymbol interference (ISI) caused by multipath propagation. Blind channel identification and equalization are effective for reducing ISI and improving system throughput. Blind channel identification can also assist equalization. Among all blind methods the subspace method (SS) [1] is well known for its good performance. However, the batch algorithm [1] requires singular value decomposition (SVD). It is inconvenient for adaptive implementation. Furthermore, it requires accurate channel length estimation and/or accurate rank estimation of the correlation matrix, which are not easy in an inherently noisy environment.

Many subspace tracking algorithms were developed to compute the subspace adaptively with computational complexity around $O(m^2)$, or $O(md)$ where d is the dimension of the signal subspace, in each recursion [3], [5]. However, the total computation of the blind channel identification is still $O(m^3)$ due to the second step, i.e., using the estimated subspace vectors to optimize channel estimation. The second step can not be recursively implemented with reduced computations in an obvious manner.

In this paper we propose new adaptive algorithms for blind channel identification using ULV updating and successive cancellation techniques. We separate the channel length estimation and subspace rank estimation into two estimation problems. Our new approaches require no rank estimation. The channel length can be over-estimated during subspace tracking and optimization. It can then be recovered by a successive cancellation procedure. The second optimization step can be recursively implemented as another subspace tracking without rank estimation. Hence the total computations of the adaptive algorithm is reduced to $O(m^2)$.

In Section 2, we introduce the communication system model and the traditional subspace algorithm. In Section 3, we discuss the ULV updating for subspace tracking without rank estimation and derive a batch algorithm. Then an adaptive algorithm is given in Section 4. This is followed by simulation results in Section 5 and a conclusion in Section 6.

2. SUBSPACE METHOD FOR BLIND CHANNEL IDENTIFICATION

2.1. Problem Formulation

Consider a fractionally sampled communication system with fractional ratio L

$$x(n) = \sum_{k=-\infty}^{\infty} s_k h(n-kL) + v(n). \tag{1}$$

Define $x_i(n) = x(nL+i)$, $h_i(n) = h(nL+i)$, and $v_i(n) = v(nL+i)$. (1) has an equivalent single-input multiple-output description as $x_i(n) = \sum_{k=-\infty}^{\infty} s_k h_i(n-k) + v_i(n), \quad i = 0, \cdots, L-1$. Let $\mathbf{x}(n) = [\ x_0(n), \cdots, x_{L-1}(n)\ ]^T$ where $(\cdot)^T$ denotes transpose, $\mathbf{h}(n) = [\ h_0(n), \cdots, h_{L-1}(n)\ ]^T$, $\mathbf{v}(n) = [\ v_0(n), \cdots, v_{L-1}(n)\ ]^T$. Then

$$\mathbf{x}(n) = \sum_{k=-\infty}^{\infty} s_k \mathbf{h}(n-k) + \mathbf{v}(n). \tag{2}$$

If the channel is of order L_h, the system (2) can be represented in the matrix form

$$\mathcal{X}(n) = \mathcal{H}\mathbf{s}(n) + \mathcal{V}(n) \tag{3}$$

where $\mathcal{H}$ is a $NL \times (N+L_h-1)$ block Toeplitz matrix

$$\mathcal{H} = \begin{bmatrix} \mathbf{h}(0) & \cdots & \mathbf{h}(L_h-1) & & \\ & \ddots & & \ddots & \\ & & \mathbf{h}(0) & \cdots & \mathbf{h}(L_h-1) \end{bmatrix} \tag{4}$$

and $\mathcal{X}(n) = [\ \mathbf{x}^T(n), \cdots, \mathbf{x}^T(n-N+1)\ ]^T$, $\mathbf{s}(n) = [\ s_n, \cdots, s_{n-N-L_h+2}\ ]^T$, $\mathcal{V}(n) = [\ \mathbf{v}^T(n), \cdots, \mathbf{v}^T(n-N+1)\ ]^T$.

We assume throughout this paper that i) the input sequence s_k is stationary with zero mean and $E\{s_k s_l^*\} = \delta(k-l)$, ii) the noise v is stationary with zero mean and white with variance σ_v^2, iii) s & v are uncorrelated.

2.2. Subspace Method [1]

The subspace separation can be performed on the correlation matrix $\mathbf{R}_x \stackrel{\triangle}{=} E[\mathcal{X}(n)\mathcal{X}^H(n)]$, where $E[\cdot]$ denotes statistical expectation, by the eigenvalue decomposition (EVD)

$$\mathbf{R}_x = \begin{bmatrix} \mathbf{U}_s & \mathbf{U}_o \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma_o} \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^H \\ \mathbf{U}_o^H \end{bmatrix} \quad (5)$$

where $\mathbf{\Sigma_s}$ contains signal eigenvalues and is $d \times d$, the vectors in $\mathbf{U}_o$ are associated with the $NL-d$ noise eigenvalues $\mathbf{\Sigma_o}$ and span the noise subspace. Assume $\mathcal{H}$ is full column rank, the channel can be uniquely determined up to a constant factor by $\mathbf{U}_o$.

Define the $LL_h \times 1$ channel vector $\mathbf{h} \stackrel{\triangle}{=} [\,\mathbf{h}^H(0), \cdots, \mathbf{h}^H(L_h - 1)\,]^H$ where $(\cdot)^H$ stands for conjugate transpose. Let $\mathbf{U}_o = [\,\mathbf{u}_0, \cdots, \mathbf{u}_{NL-d-1}\,]$ where the ith column vector $\mathbf{u}_i$ can be written as $\mathbf{u}_i = [\,\mathbf{u}_i^H(0), \cdots, \mathbf{u}_i^H(N-1)\,]^H$ with $\mathbf{u}_i(j)$ being an $L \times 1$ vector. We further define an $LL_h \times (L_h + N - 1)$ matrix

$$\mathcal{U}_i \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{u}_i(0) & \cdots & \mathbf{u}_i(N-1) & & \\ & \ddots & & \ddots & \\ & & \mathbf{u}_i(0) & \cdots & \mathbf{u}_i(N-1) \end{bmatrix}. \quad (6)$$

Then the channel can be estimated by minimizing

$$J(\mathbf{h}) = \mathbf{h}^H \left(\sum_{i=0}^{NL-d-1} \mathcal{U}_i \mathcal{U}_i^H \right) \mathbf{h} \stackrel{\triangle}{=} \mathbf{h}^H \mathbf{Q} \mathbf{h} \quad s.t. \ \ \|\mathbf{h}\| = 1 \quad (7)$$

Hence the estimate of $\mathbf{h}$ is the singular vector corresponding to the minimum singular value of the LL_h dimensional matrix $\mathbf{Q}$.

If L_h is an over-estimated channel length, then the LL_h dimensional vector $\mathbf{h}$ becomes $\mathbf{h} = [\,\mathbf{h}^H(0), \cdots, \mathbf{h}^H(L_r - 1), \mathbf{0}^T\,]^H$ where L_r is the actual channel length. The channel estimation is then inconsistent [2] and will include a random polynomial factor of order $L_h - L_r + 1$

$$\hat{\mathbf{h}} = \begin{bmatrix} \mathbf{h}(0) & & \\ \vdots & \ddots & \\ \mathbf{h}(L_r - 1) & & \mathbf{h}(0) \\ & \ddots & \vdots \\ & & \mathbf{h}(L_r - 1) \end{bmatrix} \begin{bmatrix} r(0) \\ \vdots \\ r(L_h - L_r) \end{bmatrix}. \quad (8)$$

3. BLIND CHANNEL IDENTIFICATION WITHOUT RANK ESTIMATION

Both the SVD based and the subspace tracking based [3] methods for subspace separation require accurate rank estimation. Blind channel identification is thus performed by estimating and applying the entire subspace at the same time.

On the contrary, our basic idea is trying to calculate a *time-varying* subspace vector recursively by ULV updating. In each recursion, only one noise subspace vector is estimated. A series of such vectors will jointly span the entire noise subspace. Since only one vector is estimated, no rank estimation is needed. Furthermore, the channel length can be over estimated at the beginning.

3.1. ULV updating

The ULV decomposition [4]-[5] estimates the entire signal subspace and noise subspace, hence requires accurate rank estimation. However, we can modify the ULV updating to overcome this requirement by estimating only one noise subspace vector at a time.

At iteration n, define $\mathbf{A}_n \stackrel{\triangle}{=} [\,\mathcal{X}(1), \cdots, \mathcal{X}(n)\,]^H$. Assume the ULV decomposition is

$$\mathbf{A}_n = \mathbf{U}_n \begin{bmatrix} \mathbf{B}_n & \mathbf{0} \\ \mathbf{b}_n & b_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_n^H \\ \mathbf{v}_n^H \end{bmatrix} \quad (9)$$

where $\mathbf{B}_n$ is an $(NL-1) \times (NL-1)$ matrix, $\mathbf{b}_n$ is an $(NL-1) \times 1$ vector, b_n is a scalar, $\mathbf{V}_n^H$ is $(NL-1) \times NL$ and $\mathbf{v}_n^H$ is $1 \times NL$. $\begin{bmatrix} \mathbf{V}_n & \mathbf{v}_n \end{bmatrix}$ is orthonormal, and $\mathbf{v}_n$ lies in the noise subspace with b_n sufficiently small or 0. Note that we do not need to separate $\mathbf{V}_n$ into signal and noise subspaces explicitly, which is different from the traditional ULV decomposition.

At iteration $n+1$, our goal is to get the same ULV decomposition as (9) for the row appended matrix $\mathbf{A}_{n+1} = [\,\beta \mathbf{A}_n^H,\ \mathcal{X}(n+1)\,]^H$ where $\beta \in [0 \quad 1]$ is the forgetting factor. By a series of Givens rotations we can zero out the new appended row. Deleting this all zero row, we have

$$\mathbf{A}_{n+1} = \tilde{\mathbf{U}}_n \begin{bmatrix} \tilde{\mathbf{B}}_n & \mathbf{0} \\ \tilde{\mathbf{b}}_n & \tilde{b}_n \end{bmatrix} \tilde{\mathbf{V}}_n^H. \quad (10)$$

Although (10) is in lower triangular form, $\tilde{b}_n$ is not necessarily small enough to be a noise component. In order to restore the correct form, we first calculate a reliable condition estimator (*c.f.*, [4]) $\mathbf{p}_n$ such that

$$\left\| \begin{bmatrix} \tilde{\mathbf{B}}_n & \mathbf{0} \\ \tilde{\mathbf{b}}_n & \tilde{b}_n \end{bmatrix} \mathbf{p}_n \right\| \approx \sigma_{min}(n) \quad (11)$$

where $\sigma_{min}(n)$ is the smallest singular value or noise power of the matrix $\begin{bmatrix} \tilde{\mathbf{B}}_n & \mathbf{0} \\ \tilde{\mathbf{b}}_n & \tilde{b}_n \end{bmatrix}$. Applying $NL-1$ left Givens rotations to $\mathbf{p}_n$, we have $\mathbf{W}_{1n}\mathbf{p}_n = [\,0, \cdots, 0,\ 1\,]^T = \mathbf{e}_{NL}$ where $\mathbf{W}_{1n}$ is $NL \times NL$ and orthonormal. Applying $\mathbf{W}_{1n}$ and some other proper Givens rotations to (10), we get

$$\mathbf{A}_{n+1} \tilde{\mathbf{V}}_n \mathbf{W}_{1n}^H = \mathbf{U}_{n+1} \begin{bmatrix} \mathbf{B}_{n+1} & \mathbf{0} \\ \mathbf{b}_{n+1} & b_{n+1} \end{bmatrix} \quad (12)$$

where b_{n+1} is small as a noise component. This is because from (11) and (12) we get $|b_{n+1}| = \|\mathbf{A}_{n+1}\tilde{\mathbf{V}}_n \mathbf{W}_{1n}^H \mathbf{e}_{NL}\| \approx \sigma_{min}(n)$. Therefore the ULV decomposition at iteration $n+1$ is

$$\mathbf{A}_{n+1} = \mathbf{U}_{n+1} \begin{bmatrix} \mathbf{B}_{n+1} & \mathbf{0} \\ \mathbf{b}_{n+1} & b_{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{n+1}^H \\ \mathbf{v}_{n+1}^H \end{bmatrix} \quad (13)$$

To summarize, the total computation of the above ULV updating algorithm is $O((NL)^2)$. No rank estimation is required. The detailed procedures are similar to [4]-[5].

In order to examine the relation of $\mathbf{v}_n$ and $\mathbf{U}_o$, we assume the channel is either time invariant or slowly time-variant.

Proposition 1: Suppose (11) holds with an exact equality. Then under mild conditions $span\{\mathbf{v}_n^H, n = 0, 1, \cdots\} = span\{\mathbf{U}_o\}$ with probability one.

Proof: First, from (11)-(13), $\mathbf{v}_n \in \mathbf{U}_o$ for all n. Then, since $\begin{bmatrix} \tilde{\mathbf{B}}_n & \mathbf{0} \\ \tilde{\mathbf{b}}_n & \tilde{b}_n \end{bmatrix}$ has rank d, it has at least $NL-d$ entries with small random values on its diagonal due to noise. Thus $\mathbf{p}_n$ contains correspondingly $NL-d$ random entries. From (12) and (13) we can easily obtain $\mathbf{v}_{n+1} = \tilde{\mathbf{V}}_n \mathbf{p}_n$. Since $\tilde{\mathbf{V}}_n$ is orthonormal, its columns are independent (orthonormal) and span the entire NL dimensional vector space. The $NL-d$ random entries in $\mathbf{p}_n$ then form random linear combinations of the columns of $\tilde{\mathbf{V}}_n$ by the product $\mathbf{v}_{n+1} = \tilde{\mathbf{V}}_n \mathbf{p}_n$. Since for each n there are $NL-d$ linearly independent vectors involved, their linear combination at each n will result in a random vector in the $NL-d$ dimensional subspace. Hence the rank of $[\ \mathbf{v}_1, \cdots, \mathbf{v}_n, \cdots\ ]$ is no less than $NL-d$, which is the rank of $\mathbf{U}_o$, with probability one. For details of the proof, see [6]. □

3.2. Successive Cancellation for Channel Estimation

Because of Proposition 1, we can use the recursively updated noise subspace vectors $\mathbf{v}_n$ to optimize channel estimation according to (7). We may not need all the $\mathbf{v}_n$ vectors. Instead, a small subset may be enough because we only require that the matrix $\mathbf{Q}$ in (7) is rank-one deficient so that the channel estimation is unique.

If the channel length L_h is over-estimated in the ULV updating and the optimization step (7), we need to recover the correct channel $\mathbf{h}$ and length L_r from (8). In this case, the $\mathbf{Q}$ in (7) is rank $L_h - L_r + 1$ deficient. Let its null space be an $LL_h \times (L_h - L_r + 1)$ matrix $\mathbf{Q}_0 = [\ \mathbf{q}_0, \cdots, \mathbf{q}_{L_h - L_r}\ ]$. The vector $\mathbf{q}_i = [\ \mathbf{q}_i^H(0), \cdots, \mathbf{q}_i^H(L_h-1)\ ]^H$ has the same form as (8). From (8) we find that the last L entries of $\mathbf{q}_i$, i.e., $\mathbf{q}_i(L_h-1)$, are just $\mathbf{h}(L_r-1)$ with a multiplicative factor $r_i(L_h-L_r)$. For all the vectors $\mathbf{q}_i$ we can nullify their last L entries by minimizing

$$\left\| [\mathbf{q}_0(L_h-1)\ \cdots\ \mathbf{q}_{L_h-L_r-1}(L_h-1)] - \mathbf{q}_{L_h-L_r}(L_h-1)\mathbf{f}^H \right\| \tag{14}$$

where $\mathbf{f}^H$ is a $1 \times (L_h - L_r)$ vector. The solution to (14) is

$$\mathbf{f}_{opt}^H = \frac{\mathbf{q}^H_{L_h-L_r}(L_h-1)[\mathbf{q}_0(L_h-1), \cdots, \mathbf{q}_{L_h-L_r-1}(L_h-1)]}{\mathbf{q}^H_{L_h-L_r}(L_h-1)\mathbf{q}_{L_h-L_r}(L_h-1)} \tag{15}$$

Then the matrix

$$\mathbf{Q}_1 = [\ \mathbf{q}_0 \quad \cdots \quad \mathbf{q}_{L_h-L_r-1}\ ] - \mathbf{q}_{L_h-L_r}\mathbf{f}_{opt}^H \tag{16}$$

is similar in form as $\mathbf{Q}_0$ (omit the L all zero rows). The last L entries in each column of $\mathbf{Q}_1$ are again proportional to $\mathbf{h}(L_r-1)$. Hence the procedure (15)-(16) can be applied recursively until we get

$$\mathbf{Q}_{L_h-L_r} = \begin{bmatrix} \mathbf{h}(0) \\ \vdots \\ \mathbf{h}(L_r-1) \end{bmatrix} \tilde{r}_0 \tag{17}$$

where $\tilde{r}_0$ is an unknown scalar. At this step both the channel $\mathbf{h}$ and its length L_r is obtained from (17). If $\mathbf{Q}_0$ contains more than $L_h - L_r + 1$ columns, then the signal subspace vector in $\mathbf{Q}$ is involved, whose last L elements are not proportional to $\mathbf{h}(L_r-1)$ or the corresponding part of $\mathbf{Q}_{L_h-L_r}$. The successive cancellation will not result in zero or small enough value for (14). Therefore the effective channel length L_r can be determined.

It is computationally more convenient to put the channel length determination to the last stage. We can choose an over estimated length for subspace tracking and optimization. Then by the successive cancellation procedure channel estimations with all proposed lengths can be obtained simultaneously without computational overhead. The most suitable one can be determined from (14) or the SVD of $\mathbf{Q}$, or by some other verifying methods. The traditional SS algorithm [1], however, requires forming several different matrices and the corresponding SVD's.

4. ADAPTIVE BLIND CHANNEL IDENTIFICATION

The method in Section 3 still requires SVD on $\mathbf{Q}$. In order to avoid SVD, we try to optimize (7) without explicitly calculating the matrix $\mathbf{Q}$. (7) is equivalent to $J(\mathbf{h}) = E\{\|\mathbf{h}^H \mathcal{U}_n\|^2\}$. Instead of performing optimization upon matrices, we transform the matrix $\mathcal{U}_n$ into a vector by right multiplying a vector $\mathbf{g}_n$

$$\mathbf{y}_n = \mathcal{U}_n \mathbf{g}_n, \quad \forall\ n \tag{18}$$

where the $\mathbf{g}_n$ is randomly chosen to satisfy $E\{\mathbf{g}_n \mathbf{g}_n^H\} = \mathbf{I}$. Then the problem becomes

$$\min J(\mathbf{h}) = E\{\|\mathbf{h}^H \mathbf{y}_n\|^2\} \tag{19}$$

Proposition 2: $E\{\|\mathbf{h}^H \mathcal{U}_n\|^2\} = 0$ iff $E\{\|\mathbf{h}^H \mathbf{y}_n\|^2\} = 0$.

Proof: First, if $E\{\|\mathbf{h}^H \mathcal{U}_n\|^2\} = 0$, then $\mathbf{h}^H \mathcal{U}_n = 0$ for any n with probability 1. Then $\mathbf{h}^H \mathcal{U}_n \mathbf{g}_n = 0$, hence $\mathbf{h}^H \mathbf{y}_n = 0$, which leads to $E\{\|\mathbf{h}^H \mathbf{y}_n\|^2\} = 0$. On the other hand, if $E\{\|\mathbf{h}^H \mathbf{y}_n\|^2\} = 0$, we have

$$\begin{aligned} 0 &= E\{\|\mathbf{h}^H \mathcal{U}_n \mathbf{g}_n\|^2\} \\ &= E\{\mathrm{tr}[\mathcal{U}_n^H \mathbf{h}\mathbf{h}^H \mathcal{U}_n \mathbf{g}_n \mathbf{g}_n^H]\} \\ &= \mathrm{tr}[E\{\mathcal{U}_n^H \mathbf{h}\mathbf{h}^H \mathcal{U}_n\} E\{\mathbf{g}_n \mathbf{g}_n^H\}] \\ &= E\{\|\mathbf{h}^H \mathcal{U}_n\|^2\} \end{aligned}$$

Hence the proposition is proved. □

From Proposition 2, the channel can be estimated from the null space of $\{\mathbf{y}_n,\ n = 0, 1, \cdots\}$. We apply the ULV updating of Section 3.1 again to get a series of null space vectors $\mathbf{v}_y(n)$ such that $\|[\ \mathbf{y}_0, \cdots, \mathbf{y}_n\ ]^H \mathbf{v}_y(n)\| \approx 0$ and $\mathbf{v}_y(n), n = 0, 1, \cdots$, span the entire null space of $\{\mathbf{y}_n\}$ for all n.

If the channel length is over estimated, $\mathbf{v}_y(n)$ is in a form similar to (8). Therefore the successive cancellation introduced in Section 3.2 can be used for channel estimation and length recovery. Because we have a series of null space vectors in this case, the procedure is illustrated in Figure 1. First, as illustrated in Layer 0,

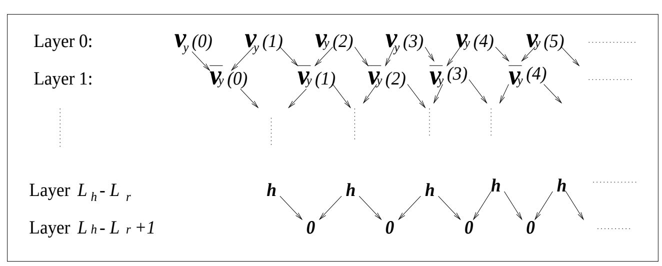

Fig. 1. Successive cancellation for channel estimation.

we use the neighboring vector $\mathbf{v}_y(n+1)$ to cancel the last L entries of $\mathbf{v}_y(n)$, similar to (15)-(16). We get a series of new vectors

$\bar{\mathbf{v}}_y(n)$ as listed in the Layer 1. Then we perform successive cancellation upon $\bar{\mathbf{v}}_y(n)$. This recursive procedure is repeated until we get a series of vectors that are estimates of true channel coefficients $\mathbf{h} = [\mathbf{h}^H(0), \cdots, \mathbf{h}^H(L_r - 1)]^H$ up to some multiplicative factors. Any further cancellations will completely cancel $\mathbf{h}$ and will result in (theoretically) zero vectors. Hence the channel and its length are recovered.

5. SIMULATIONS

In this section, we use simulations to study the performance of our proposed subspace algorithms. We denote our batch algorithm in Section 3 as SS-ULV, and the adaptive algorithm in Section 4 as SS-ADAP. We compare our algorithms with the traditional SVD based subspace algorithm [1], which is denoted as SS-SVD, and with the subspace tracking algorithm PASTd in [3], which we denote as SS-PASTd.

First, we use the same channel as in [1]. $L = 4$, $L_h = 5$, $N = 3$, and rank $d = 7$. Figure 2 compares the four algorithms assuming that the exact channel length and rank are known. SS-SVD has the best performance whereas SS-ULV performs very closely. SS-ADAP also has good performance. However, SS-PASTd fails.

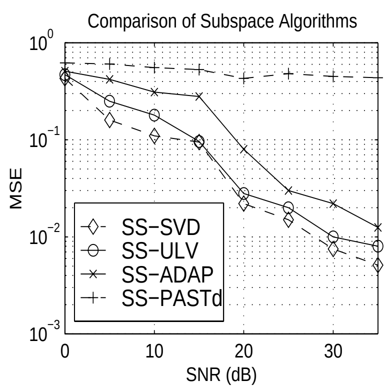

Fig. 2. Subspaces algorithms for Channel 1 [1]. 500 samples.

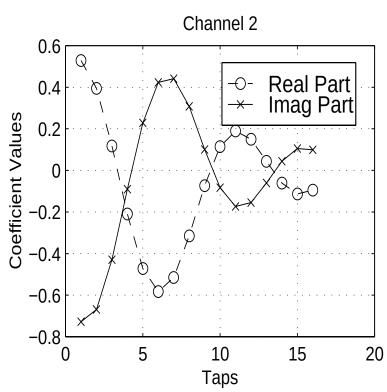

Fig. 3. Coefficients of Channel 2

In the second example, the channel impulse response is $h_c(t) = e^{-2\pi(0.15)j} c(t - 0.25T, 0.11)W(t - 0.25T) + 0.8 * e^{-2\pi(0.6)j} c(t - T, 0.11)W(t - T)$ where $c(t, 0.11)$ is a raised cosine pulse with the roll-off factor 0.11, and $W(t)$ is a rectangular truncation window spanning $[0,\ 3.99T]$. $L = 4$. Fig. 3 shows the channel and Fig. 4 shows the performance of the subspace algorithms. For SS-SVD, we assume that the true channel length $L_h = 4$ and the rank are known. For SS-ULV and SS-ADAP, we choose $L_h = 5$. We see that the SS-SVD can not identify this channel, even if the channel length is known. However, our proposed algorithms still have good performance.

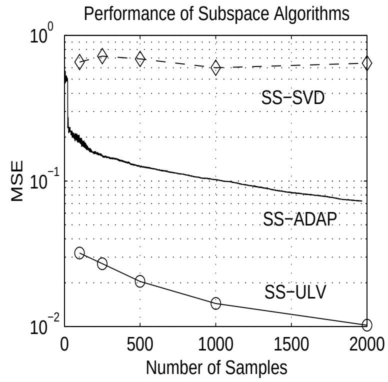

Fig. 4. Subspace algorithms Channel 2. SNR=30dB.

6. CONCLUSION

The existing subspace algorithms for blind channel identification require accurate rank estimation and $O(m^3)$ computations, where m is the dimension of data vector. We proposed new algorithms that do not require rank estimation, and channel length can be over estimated in the beginning and recovered in the end. Our algorithms are based on ULV updating and successive cancellation. The computation is greatly reduced, to $O(m^2)$ for the adaptive algorithm.

7. REFERENCES

[1] E. Moulines, P. Duhamel, J. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. on Signal Processing*, vol. 43, no. 2, 516-525, Feb. 1995.

[2] K. Abed-Meraim, P. Loubaton and E. Moulines, "A subspace algorithm for certain blind identification problems," *IEEE Trans. on Information Theory*, vol. 43, no. 2, 499-511, Feb. 1997.

[3] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. on Signal Processing*, vol. 43, no. 1, 95-107, January 1995.

[4] G. Golub and C. Loan, *Matrix Computations.* Third Edition, The Johns Hopkins Univ. Press, Baltimore, MD, 1996.

[5] G.W. Stewart, "Updating a rank-revealing ULV decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 14, 494-499, 1993.

[6] X. Li and H. Fan, "Blind channel identification: subspace tracking method without rank estimation," submitted to *IEEE Trans. on Signal Processing*.