ADAPTIVE BLIND EQUALIZATION FOR ASYNCHRONOUS DS-CDMA SYSTEMS BASED ON RLS

Geert Leus, Marc Moonen *
Katholieke Universiteit Leuven - ESAT
K. Mercierlaan 94, 3001 Heverlee - Belgium
tel. 32/16/32 19 24, fax 32/16/32 19 70
geert.leus@esat.kuleuven.ac.be, marc.moonen@esat.kuleuven.ac.be

ABSTRACT

This paper proposes a new adaptive blind equalization scheme for asynchronous direct-sequence code-division multiple-access (DS-CDMA) systems based on recursive least squares (RLS). Linear equalizer based blind direct symbol estimation is considered, which is related to subspace based blind direct symbol estimation. Compared with the subspace based algorithm, the proposed algorithm has increased dimensions but does not make use of the computationally demanding singular value decomposition (SVD). Furthermore, simulations show that the performance is more robust against multi-user interference (MUI).

1 INTRODUCTION

Asynchronous DS-CDMA communication offers simple transmitter structures at the expense of more complex receiver structures, for the receiver has to suppress both MUI and intersymbol interference (ISI).

To combat MUI and ISI in this work, we use linear equalizer based blind direct symbol estimation, which is related to subspace based blind direct symbol estimation [1]. These techniques use a two-fold spreading, together with some spatial oversampling.

The proposed block processing problem resembles the blind least squares (BLS) problem that was first presented in [2] and that later also appeared in [3], although there are some important differences: 1) In this paper we combine different linear equalizers for different delays instead of using smoothing[1]. This relaxes the condition on the window length, which results in a better tracking capability in the case of time-varying channels. 2) In this paper, not only spatial oversampling but also short code spreading is used to generate a multirate/multichannel structure. This relaxes the condition on the number of antennas at the cost of an increased bandwidth. 3) In this paper we focus on direct symbol estimation without an explicit computation of the linear equalizer. This leads to a simple adaptive implementation.

The adaptive processing problem is obtained by combining different block processing problems. When this problem is subjected to a linear constraint on the data symbols it can easily be solved by a RLS with fixed matrix dimensions. In this paper we also compare the proposed algorithm with the related algorithm of [1].

2 DATA MODEL

2.1 Multirate Framework

Consider an asynchronous J-user DS-CDMA system where every user j first spreads his data sequence with a distinct long code $c_j[n]$, nonzero for $n = 0, 1, \ldots, dN-1$, where N is the spreading factor, i.e. the number of code symbols per data symbol. Let $\{c_j[k;n]\}_{k=0}^{d-1}$ be a family of sequences defined as $c_j[k;n] = c_j[kN+n]$, nonzero for $n = 0, 1, \ldots, N-1$. Because k can fall outside $\{0, \ldots, d-1\}$ we define

$$\tilde{c}_j[k;n] = c_j[k \mod d; n].$$

The coded data sequence for the j^{th} user is then

$$x_j[n] = \sum_{k=-\infty}^{\infty} s_j[k]\tilde{c}_j[k; n-kN]$$
$$= s_j[k]\tilde{\boldsymbol{c}}_j[k](n-kN), \text{ with } k = \left\lfloor \frac{n}{N} \right\rfloor, \quad (1)$$

where $s_j[k]$ is the j^{th} user's data sequence with symbols in the finite alphabet Ω and

$$\tilde{\boldsymbol{c}}_j[k] = \begin{bmatrix} \tilde{c}_j[k;0] & \cdots & \tilde{c}_j[k;N-1] \end{bmatrix}^T.$$

In a second step every coded data sequence $x_j[n]$ is further spread with a distinct short code $d_j[m]$, nonzero for

*Geert Leus is a Research Assistant supported by the Fund for Scientific Research - Flanders (Belgium) (F.W.O.). Marc Moonen is a Research Associate of the F.W.O. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the framework of a Concerted Action Project of the Flemish Community, entitled *Model-based Information Processing Systems* (GOA/MIPS/95/99/3) as well as the IT-program of the I.W.T., *Integrating Signal Processing Systems* (ITA/GBO/T23) and was partially sponsored by IMEC (Flemish Interuniversity Microelectronics Center) and IUAP P4-02 (1997-2001): Modeling, Identification, Simulation and Control of Complex Systems. The scientific responsibility is assumed by its authors.

[1] We assume there is enough spatial diversity to make smoothing superfluous from a dimensional point of view.

$m = 0, 1, \ldots, P-1$ (P is the spreading factor of $d_j[m]$). The resulting sequence or chip sequence for the j^{th} user can be expressed as

$$u_j[m] = \sum_{n=-\infty}^{\infty} x_j[n] d_j[m - nP].$$

This sequence is then transmitted over a channel $g_j(t)$ at a rate of $1/T_c = NP/T$ (the chip rate), with T equal to the symbol period. After sampling the receiver input at the chip rate, the received sequence is

$$y[m] = \sum_{j=1}^{J} y_j[m] + v[m],$$

where $v[k]$ is additive noise and where

$$y_j[m] = \sum_{n=-\infty}^{\infty} x_j[n] h_j[m - nP], \qquad (2)$$

with $h_j[m]$ the convolution of $d_j[m]$ with $g_j[m] = g_j(mT_c)$. We assume the channel $g_j[m]$ has an order of L_j and a delay of m_j, which means that $g_j[m]$ is nonzero for $m = m_j, m_j+1, \ldots, m_j + L_j$.

2.2 Multichannel Framework

In order to realize a multichannel model we define

$$\boldsymbol{y}[n] = \begin{bmatrix} y[nP+p] & \cdots & y[(n+1)P - 1 + p] \end{bmatrix}^T,$$

with offset $p \in \{0, 1, \ldots, P-1\}$. In the same way we also define $\boldsymbol{y}_j[n]$, $\boldsymbol{v}[n]$ and $\boldsymbol{h}_j[n]$. This leads to

$$\boldsymbol{y}[n] = \sum_{j=1}^{J} \boldsymbol{y}_j[n] + \boldsymbol{v}[n],$$

with $\boldsymbol{y}_j[n] = \sum_{l=-\infty}^{\infty} x_j[l] \boldsymbol{h}_j[n-l]$. We can also write

$$\boldsymbol{y}[n] = H\boldsymbol{x}[n] + \boldsymbol{v}[n],$$

$$\text{where } \begin{cases} H &= \begin{bmatrix} H_1 & H_2 & \cdots & H_J \end{bmatrix} \\ \boldsymbol{x}[n] &= \begin{bmatrix} \boldsymbol{x}_1^T[n] & \boldsymbol{x}_2^T[n] & \cdots & \boldsymbol{x}_J^T[n] \end{bmatrix}^T \end{cases},$$

$$\text{with } \begin{cases} H_j &= \begin{bmatrix} \boldsymbol{h}_j[n_j + K_j] & \cdots & \boldsymbol{h}_j[n_j] \end{bmatrix} \\ \boldsymbol{x}_j[n] &= \begin{bmatrix} x_j[n - n_j - K_j] & \cdots & x_j[n - n_j] \end{bmatrix}^T \end{cases}.$$

The parameters K_j and n_j respectively represent the order and the delay of the 'composite' vector channel $\boldsymbol{h}_j[n]$. Note that K_j and n_j depend on L_j, m_j and also on the offset p. Suppose we now group l successive received vectors, starting from time step n,

$$Y_n^{(l)} = \begin{bmatrix} \boldsymbol{y}[n] & \boldsymbol{y}[n+1] & \cdots & \boldsymbol{y}[n+l-1] \end{bmatrix}$$

and define $V_n^{(l)}$ and $X_n^{(l)}$ in the same way as $Y_n^{(l)}$, then we can write

$$Y_n^{(l)} = HX_n^{(l)} + V_n^{(l)}. \qquad (3)$$

We also introduce the following notation

$$X_n^{(l)} = \begin{bmatrix} X_{1,n}^{(l)} \\ \vdots \\ X_{J,n}^{(l)} \end{bmatrix}, \text{ with } X_{j,n}^{(l)} = \begin{bmatrix} \boldsymbol{x}_{j,n-n_j-K_j}^{(l)} \\ \vdots \\ \boldsymbol{x}_{j,n-n_j}^{(l)} \end{bmatrix}, \qquad (4)$$

where $\boldsymbol{x}_{j,n}^{(l)} = \begin{bmatrix} x_j[n] & x_j[n+1] & \cdots & x_j[n+l-1] \end{bmatrix}$.

When we have $M > 1$ receiver antennas, model (3) can easily be extended. The orders L_j and K_j and delays m_j and n_j are then related to the union of all antennas.

3 BLIND EQUALIZATION

Assume that P is even and $L \leq P/2$, where L is the maximal order L_j. Based on the knowledge of m_j (accuracy $\pm P/2$), this allows us, for a certain user j, to determine the delay n_j and to select the offset $p \in \{0, P/2\}$ such that the order $K_j = 1$. Using the obtained values for n_j and p, we can finally isolate that user j based on the knowledge of $c_j[n]$, which is pointed out in the next sections. Note that the selected value of the offset p determines the matrix H and therefore the 'composite' channel order of each user. The total number of users for which this order is 1 is denoted as I. For all other $J - I$ users this order is 2.

3.1 Block Processing

Let us now focus on $\boldsymbol{x}_{j,kN}^{(wN)}$, where we take a window length $w > 1$. Following (1), $\boldsymbol{x}_{j,kN}^{(wN)}$ can be written as a function of the long code $c_j[n]$ of user j

$$\boldsymbol{x}_{j,kN}^{(wN)} = \begin{bmatrix} s_j[k]\tilde{\boldsymbol{c}}_j[k] \\ \vdots \\ s_j[k+w-1]\tilde{\boldsymbol{c}}_j[k+w-1] \end{bmatrix}^T. \qquad (5)$$

Since $K_j = 1$, it follows from (3) and (4) that $\boldsymbol{x}_{j,kN}^{(wN)}$ is contained in $Y_{kN+n_j}^{(wN)}$ and $Y_{kN+n_j+1}^{(wN)}$. The problem addressed here is to compute the data symbols $\{s_j[k], \ldots, s_j[k+w-1]\}$ from $Y_{kN+n_j}^{(wN)}$ and $Y_{kN+n_j+1}^{(wN)}$ based on the knowledge of $c_j[n]$. To solve this problem we make the following assumptions

A1) $\mathcal{H}$ has full column rank $3J - I$.

A2) $X_{kN+n_j}^{(wN)}$ has full row rank $3J - I$.

These require that $MP \geq 3J - I$ and $wN \geq 3J - I$.

A subspace based approach for $p = 0$ with an elimination of some of the output samples is given in [1] but can easily be extended for this framework. However, in this paper we focus on a linear equalizer based approach. For the sake of clarity we initially ignore the additive noise. Under Assumptions A1) and A2), the vector equalizers $\boldsymbol{f}_{j,k}^{(0)}$ and $\boldsymbol{f}_{j,k}^{(1)}$ of order 0, that satisfy

$$\boldsymbol{f}_{j,k}^{(0)H} Y_{kN+n_j}^{(wN)} = \boldsymbol{f}_{j,k}^{(1)H} Y_{kN+n_j+1}^{(wN)} = \boldsymbol{x}_{j,kN}^{(wN)}, \qquad (6)$$

are zero-forcing and every vector equalizer has $MP - 3J + I$ degrees of freedom. The known long code information that is in $\boldsymbol{x}_{j,kN}^{(wN)}$ (see (5)) then admits us to write (6) as

$$\left[\begin{array}{cc|c} \boldsymbol{f}_{j,k}^{(0)H} & \boldsymbol{f}_{j,k}^{(1)H} & \boldsymbol{s}_{j,k}^{(w)} \end{array}\right] \left[\begin{array}{cc} Y_{kN+n_j}^{(wN)} & O \\ O & Y_{kN+n_j+1}^{(wN)} \\ \hline -C_{j,k} & -C_{j,k} \end{array}\right]$$

$$= \left[\begin{array}{c|c} \boldsymbol{f}_{j,k}^{H} & \boldsymbol{s}_{j,k}^{(w)} \end{array}\right] \left[\begin{array}{c} \boldsymbol{Y}_k \\ \hline -\boldsymbol{C}_{j,k} \end{array}\right] = \boldsymbol{0}, \qquad (7)$$

with $C_{j,k} = \text{diag}\left\{\tilde{\boldsymbol{c}}_j^T[k], \tilde{\boldsymbol{c}}_j^T[k+1], \ldots, \tilde{\boldsymbol{c}}_j^T[k+w-1]\right\}$, where $\text{diag}\{X_1, X_2, \ldots, X_k\}$ represents a block diagonal matrix with $X_1, X_2, \ldots, X_k$ as her block diagonal elements.

The next theorem gives a sufficient condition for a unique solution for $\boldsymbol{s}_{j,k}^{(w)}$ of (7) (up to a possible complex factor). The proof is omitted due to space limitations.

Theorem 1. *If at least one of the matrices $X_{kN+n_j}^{(wN)}$ and $X_{kN+n_j+1}^{(wN)}$, extended with an extra row $\boldsymbol{x}'^{(wN)}_{j,kN}$, has full row rank $3J - I + 1$, irrespective of $\boldsymbol{s}'^{(w)}_{j,k}$ (independent of $\boldsymbol{s}_{j,k}^{(w)}$), then (7) has a unique solution $\boldsymbol{s}_{j,k}^{(w)}$ for the data symbols (up to a possible complex factor).*

Following this theorem we need to take $wN > 3J - I$, but because the theorem is satisfied if and only if $\left[\begin{array}{c|c} \boldsymbol{Y}_k^T & -\boldsymbol{C}_{j,k}^T \end{array}\right]$ has rank $2(3J - I) + w - 1$, we more specifically need to take $2(wN - 3J + I) \geq w - 1$. Note that Theorem 1 and the corresponding condition for the window length are the same as the ones for the subspace based approach. Note also that (7) does not have to be overdetermined in order to find a unique solution for the data symbols.

Let us now again suppose that the additive noise $V_n^{(wN)}$ is present in (3). Then we consider the following minimization problem

$$\min_{\boldsymbol{f}_{j,k}, \boldsymbol{s}_{j,k}^{(w)}} \left\{ \left\| \left[\begin{array}{c|c} \boldsymbol{f}_{j,k}^{H} & \boldsymbol{s}_{j,k}^{(w)} \end{array}\right] \left[\begin{array}{c} \boldsymbol{Y}_k \\ \hline -\boldsymbol{C}_{j,k} \end{array}\right] \right\|^2 \right\}, \qquad (8)$$

where some constraint is necessary to avoid the trivial solution.

3.2 Adaptive Processing

We now derive an adaptive processing problem at time step k, by combining $k+1$ block processing problems of the form (8) (let the value for k in (8) vary form 0 to k). This leads to the following minimization problem

$$\min_{\boldsymbol{f}_{j,0:k}, \boldsymbol{s}_{j,0}^{(k+w)}} \left\{ \left\| \underbrace{\left[\begin{array}{c|c} \boldsymbol{f}_{j,0}^{H} \cdots \boldsymbol{f}_{j,k}^{H} & \boldsymbol{s}_{j,0}^{(k+w)} \end{array}\right]}_{\left[\begin{array}{c|c} \boldsymbol{f}_{j,0:k}^{H} & \boldsymbol{s}_{j,0}^{(k+w)} \end{array}\right]} \left[\begin{array}{c} \mathcal{Y}_k \\ \hline -\mathcal{C}_{j,k} \end{array}\right] \right\|^2 \right\},$$

with $\mathcal{Y}_k = \text{diag}\{\boldsymbol{Y}_0, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_k\}$ and

$$\mathcal{C}_{j,k} = \begin{bmatrix} \boxed{\boldsymbol{C}_{j,0}} & \boldsymbol{0} & & \boldsymbol{0} \\ \boldsymbol{0} & \boxed{\boldsymbol{C}_{j,1}} & \ddots & \vdots \\ \vdots & \vdots & & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & & \boxed{\boldsymbol{C}_{j,k}} \end{bmatrix}.$$

Again, some constraint is necessary to avoid the trivial solution.

In this paper we make no constraints on the equalizer taps and only consider the following linear constraint on the data symbols

$$\boldsymbol{s}_{j,0}^{(k+w)} \left[\boldsymbol{e}_1 \quad \cdots \quad \boldsymbol{e}_{k-l}\right] = \hat{\boldsymbol{s}}_0^{(j)(k-l)} \ (l > -w),$$

where $\boldsymbol{e}_i$ is the i^{th} unit vector and $\hat{\boldsymbol{s}}_{j,0}^{(k-l)}$ is assumed known. The problem can then be formulated as a least squares (LS) problem

$$\underbrace{\left[\begin{array}{c|c} \mathcal{Y}_k^H & -\mathcal{C}_{j,k}^H(:, k-l+1 : k+w) \end{array}\right]}_{\left[\begin{array}{c|c} \mathcal{Y}_k^H & -\bar{\mathcal{C}}_{j,k}^H \end{array}\right]} \left[\begin{array}{c} \boldsymbol{f}_{j,0:k} \\ \hline \boldsymbol{s}_{j,k-l}^{(l+w)H} \end{array}\right] \overset{LS}{=} \underbrace{-\mathcal{C}_{j,k}^H(:, 1 : k-l)\hat{\boldsymbol{s}}_{j,0}^{(k-l)H}}_{\boldsymbol{d}_{j,k}}, \qquad (9)$$

which can be solved recursively by an RLS algorithm. Because we are only interested in the solution for $\boldsymbol{s}_{j,k-l}^{(l+w)}$ this RLS algorithm can be implemented with fixed matrix dimensions $2MP + l + w$ (for the subspace based algorithm this is only $l + w$). For every time step k, the algorithm projects the first element of the solution for $\boldsymbol{s}_{j,k-l}^{(l+w)}$ onto the finite alphabet Ω (hard-decision feedback). Next, we focus on the RLS algorithm.

RLS Algorithm

First rewrite (9) as

$$\left[\begin{array}{c|c} \mathcal{Y}_k^H & -\text{fc}\{\bar{\mathcal{C}}_{j,k}^H\} \end{array}\right] \left[\begin{array}{c} \boldsymbol{f}_{j,0:k} \\ \hline \text{fc}\{\boldsymbol{s}_{j,k-l}^{(l+w)}\}^H \end{array}\right] \overset{LS}{=} \boldsymbol{d}_{j,k}, \qquad (10)$$

where $\text{fc}\{\cdot\}$ represents a flipping operation of the columns. The reason for this modified representation will become clear further on. To solve (10) we make use of the QRD. The QRD of $\left[\begin{array}{c|c|c} \mathcal{Y}_k^H & -\text{fc}\{\bar{\mathcal{C}}_{j,k}^H\} & \boldsymbol{d}_{j,k} \end{array}\right]$, which is assumed to have full column rank, gives us

$$\left[\begin{array}{c|c|c} Q_k^{(1)} & Q_k^{(2)} & \boldsymbol{q}_k \end{array}\right] \left[\begin{array}{c|c|c} R_k^{(1,1)} & R_k^{(1,2)} & \boldsymbol{z}_k^{(1)} \\ \hline O & R_k^{(2,2)} & \boldsymbol{z}_k^{(2)} \\ \hline \multicolumn{2}{c|}{\boldsymbol{0}} & x_k \end{array}\right].$$

The solution for $\boldsymbol{s}_{k-l}^{(l+w)}$ of (10) then satisfies

$$R_k^{(2,2)} \text{fc}\{\boldsymbol{s}_{j,k-l}^{(l+w)}\}^H = \boldsymbol{z}_k^{(2)},$$

$$\left[\begin{array}{c|c|c} \bar{R}^{(1,1)}_{k+1} & \bar{R}^{(1,2)}_{k+1} & \bar{\boldsymbol{z}}^{(1)}_{k+1} \\ \hline O & R^{(2,2)}_{k+1} & \boldsymbol{z}^{(2)}_{k+1} \\ \hline \multicolumn{2}{c|}{O} & \star \end{array}\right] \leftarrow Q_g^H \left[\begin{array}{c|c|c|c} \multicolumn{4}{c}{O} \\ \hline O & \mathbf{0} & R_k^{(2,2)}(:,1:l+w-1) & \boldsymbol{z}_k^{(2)} - R_k^{(2,2)}(:,l+w)\hat{s}_j^*[k-l] \\ \hline \boldsymbol{Y}^H_{k+1} & \multicolumn{2}{c|}{-\mathrm{fc}\{\boldsymbol{C}^H_{j,k+1}(:,-l+1:w)\}} & -\boldsymbol{C}^H_{j,k+1}(:,1:-l)\hat{\boldsymbol{s}}^{(-l)H}_{j,k+1} \end{array}\right] \tag{11}$$

which can be solved through backsubstitution.

Assume now that $R_k^{(2,2)}$ and $\boldsymbol{z}_k^{(2)}$ are known at time step k. The aim is then to find an efficient updating rule to update these. Note that this updating can be performed without the knowledge of $R_k^{(1,1)}$, $R_k^{(1,2)}$ and $\boldsymbol{z}_k^{(1)}$, which allows us to work with fixed matrix dimensions $2MP + w + l$. In a first step we calculate an estimate for the first element of $\boldsymbol{s}_{j,k-l}^{(l+w)}$. Because we use the representation of (10) this estimate can easily be derived by only executing the first step of the backsubstitution scheme. Once the decision on the data symbol is made we are ready to update $R_k^{(2,2)}$ and $\boldsymbol{z}_k^{(2)}$ in a second step. The updating formula for $-w < l < 0$ is summarized in (11) (the formula for $l \geq 0$ looks similar). The matrices $\bar{R}_{k+1}^{(1,1)}$, $\bar{R}_{k+1}^{(1,2)}$ and $\bar{\boldsymbol{z}}_{k+1}^{(1)}$ are given by the right lower $2MP \times 2MP$ submatrix of $R_{k+1}^{(1,1)}$, the lower $2MP$ rows of $R_{k+1}^{(1,2)}$ and the lower $2MP$ rows of $\boldsymbol{z}_{k+1}^{(1)}$, respectively. The matrix Q_g^H is a product of Givens rotations and $\star$ is a column vector of don't care entries.

4 SIMULATION RESULTS

In this section we perform some simulations, comparing the proposed algorithm with the subspace based algorithm ([1] extended for the framework of this paper). We consider DBPSK modulation and antipodal (± 1) spreading sequences. We assume that $v[k]$ is additive white zero-mean complex circular Gaussian noise with variance σ_v^2. For the j^{th} user, the signal-to-noise ratio (SNR) and the near-far ratio (NFR) are defined as $\text{SNR} = E_j/(M\sigma_v^2)$, where E_j is the expected received energy per chip, averaged over the total burst, and $\text{NFR} = E/E_j$, where $E_i = E$ for $i \neq j$.

Now consider an 8-user asynchronous DS-CDMA system ($J = 8$) with $d = 10$, $N = 4$ and $P = 12$. The short and long code sequences are randomly generated. The shaping pulse we use is a raised-cosine with a roll-off factor of 0.5 and two sidelobes. We take $M = 2$, which is sufficient, irrespective of I. The multipath channels are designed such that, for every user, the delay spread related to the union of all antennas is not greater than $6T_c$ (so we approximately have $L = 6$). We consider Rayleigh fading paths with a Doppler frequency of 75 Hz (speed of 90 km/h for a carrier frequency of 900 MHz). Applying a symbol rate of $1/T = 25$ kHz, transmitting bursts of 100 data symbols, this fading excludes the use of block processing. We further assume the same variance for all the paths corresponding to the same user. For our simulations we take $w = 7$ and SNR = 10 dB. For each simulation 500 Monte-Carlo trials are conducted. As we can see from Figure 1, when the NFR increases the performance of the proposed algorithm overshoots the one of the subspace based algorithm. As l increases this switch occurs at a lower NFR.

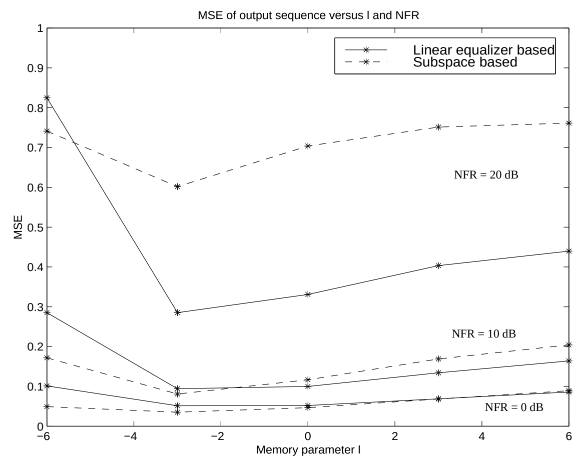

Figure 1: MSE of the output sequence of one of the users as a function of the memory parameter l and the NFR.

5 CONCLUSIONS

We developed an RLS algorithm for linear equalizer based blind direct symbol estimation, which can handle time-varying channels. Compared with the subspace based algorithm, the algorithm has increased dimensions but does not make use of the computationally demanding SVD. Furthermore, simulations showed that the performance is more robust against MUI.

REFERENCES

[1] G. Leus and M. Moonen. An Adaptive Blind Receiver for Asynchronous DS-CDMA Based on Recursive SVD and RLS. In *Proc. ProRISC/IEEE Workshop on CSSP*, pages 615–620, Mierlo, The Netherlands, November 1997.

[2] H. Liu and G. Xu. Blind Equalization for CDMA Systems with Aperiodic Spreading Sequences. In *Proc. ICASSP'96*, Atlanta, GA, May 1996.

[3] H. Liu and M. D. Zoltowski. Blind Equalization in Antenna Array CDMA Systems. *IEEE Trans. Signal Processing*, 45(1):161–172, January 1997.