Adaptive Delay Synchronisation and Reception in DS-CDMA Communications Systems

by

A. Manikas PhD, DIC, CEng and P. Wilkinson BSc

Digital Communications Research Section,
Department of Electrical and Electronic Engineering,
Imperial College of Science, Technology and Medicine,
London, SW7 2BT, UK.

Abstract

In this paper a new $H^\infty$ type delay synchronisation and tracking approach is presented. It is based on the partitioning of the PN-code matrix of the CDMA users in order to form a linear combiner. This linear combiner may be used to find the noise subspace in order to synchronise the receiver code signal with the transmitted signal. The problem is reformulated as a state-space model and an $H^\infty$ solution provides delay-tracking even in the presence of modelling errors such as over and underestimation of the number of signals present. The proposed novel formulation and approach provide a powerful near-far resistant solution for synchronisation and reception of DS-CDMA signals in a multiuser environment.

1 Introduction

Spread spectrum modulation schemes have been a topic of considerable interest over recent years as a means to provide increased user density and signal quality for cellular systems. However, a persistent issue in the implementation of such systems is the problem of parameter estimation. This is generally considered to be the limiting factor in achieving high performance with CDMA systems [1]. The parameters which are required for the successful demodulation of a DS-CDMA signal are: the frequency and phase of the desired carrier signal, the code sequence of the desired user and the timing of the code sequence. Of these, carrier frequency and phase tracking are problems which are common to any RF communications system and as such have been thoroughly investigated. The code sequence is assumed to be known by the desired receiver as a priori knowledge. However, the problem of acquiring and tracking the code sequence timing is currently an active research topic.

The predominant problem of conventional DS-CDMA systems is their performance degradation due to the near-far effect in a multi-user environment where the desired user signal is masked by interfering signals of higher power. This problem is not restricted to the demodulator, but also affects the performance of the delay time acquisition and tracking for code synchronisation at the receiver. This paper looks at the problem of time delay synchronisation for Direct Sequence Code Division Multiple Access (DS-CDMA) signals over scalar channels in the presence of multiple access interference (MAI).
Attempts to solve this problem may be seen in [2, 3, 4, 5, 6, 7] and the references therein. The most popular techniques for solving the near-far problem in time delay estimation are the subspace-based techniques, such as [4, 6, 7]. These techniques use some method to estimate the signal and noise subspaces - normally by eigendecomposing the covariance matrix of a received vector input signal. Then, assuming that the desired user code is known, the delay associated with the desired transmission may be obtained. The performance of such subspace-based techniques is in general insensitive to signal power, so they are naturally immune to the near-far problem.

There are two significant problems with the implementation of conventional subspace techniques to carry out signal reception in a DS-CDMA system. Firstly, they assume that the user delays are fixed throughout the observation interval. In any practical system the time delay associated with a given user may be expected to vary as the user moves around the cell. With vehicular mobiles travelling at up to 150kph, it is reasonable to expect significant signal delay changes for a particular time frame. Therefore, it is important that any synchronisation scheme is able to track moving delays; some method must be found to make the schemes adaptive.

Secondly, subspace techniques require certain assumptions about the statistics of noise signals and variations within the environment. They are very sensitive to modelling errors and tend to fail where any assumptions are violated. The cellular CDMA channel is quite complex and characterised by a number of different disturbances. As well as additive white Gaussian noise, there is likely to be significant multiple access interference, interference from other networks, shot noise, and countless other domestic and industrial sources whose effects cannot simply be considered in the thermal noise term. In addition, subspace methods require knowledge of the number of active users. Current detection methods such as AIC and MDL [8] tend to overestimate the number of users. All these abnormalities in the signal environment perturb the assumed signal model, potentially degrading the performance of the acquisition/tracking process.

In this paper, a new delay tracking algorithm based on adaptive $H^\infty$ estimation theory [9] and the so called "linear combiner" or "propagator" method [10] is proposed. The "linear combiner" method is a subspace method which eliminates the need to perform eigendecomposition of the received signal covariance matrix. This is achieved by determining a linear operator which may be used to obtain the noise subspace. $H^\infty$ estimation techniques not only provide the proposed algorithm with "adaptivity", but also reduce sensitivity to any modelling errors. The two techniques are combined via a reformulation of the linear combiner in the form of a state-space model. An adaptive $H^\infty$ estimator is then implemented to estimate the state matrix which may be used to track the noise subspace. A cost function similar to MuSIC [4] is used to estimate the propagation delay. The combination of these techniques provides a tracking solution that is robust to modelling errors in the signal environment as shown in the simulations.

2 Modelling and Problem Formulation

In an $M$-user asynchronous DS-CDMA system with BPSK modulation, each user sends an information signal, $m_i(t)$, of the form

$$m_i(t) = \sum_n a_i[n]c_i(t - nT_{cs}) \quad (1)$$

where $\{a_i[n], \forall n \in \mathbb{N}\}$ is the $i^{th}$ user's data sequence of $\pm 1$s and $c_i(t)$ is the unit amplitude rectangular pulse of width $T_{cs}$ shown in Figure (1); $T_{cs}$ is the channel symbol period. Note that in Equation (1), for a given $n$ (i.e. for the $n^{th}$ bit of the data sequence), the time $t$ satisfies

$$(n - 1)T_{cs} \leq t < nT_{cs}$$
The $i^{th}$ user data signal $m_i(t)$ is modulated by a periodic PN-code signal (of period $N_c$) modelled by

$$b_i(t) = \sum_k \alpha_i[k]c_2(t - kT_c) \quad (2)$$

where $\{\alpha_i[k], \forall k \in N, i = 1, 2, \ldots, M\}$ denotes the $i^{th}$ user’s PN-code sequence of $\pm 1$s at the $k$th chip interval, $T_c$ is the duration of one chip interval and $c_2$ is the unit amplitude rectangular pulse of duration $T_c$ shown in Figure (1). For the $k$th chip interval $t$ and $k$ are related by $kT_c \leq t < kT_c$.

![Figure 1: Rectangular Pulses $c_1(t)$ and $c_2(t)$.](image)

Given that the model is for a short code system where each data bit is modulated by one period of the code sequence it is obvious that the processing gain $N_c = T_c/T_c$. Assuming that for each user the clock of the data generator is synchronised with that of the pn-code generator, then the baseband DS-CDMA signal of the $i^{th}$ user becomes

$$s_i(t) = m_i(t)b_i(t) = \sum_k a_i[k][k/N_c]c_2(t - kT_c) \quad (3)$$

with $[\cdot]$ denoting the round down to integer operator. For Equation (3), for a given $k$, the time $t$ satisfies

$$k/N_c T_{cs} + (k \text{ mod } N_c)T_c \leq t < k/N_c T_{cs} + ((k \text{ mod } N_c) + 1)T_c$$

The total transmitted signal for the $i^{th}$ user may then be formed by multiplying $s_i(t)$ with a carrier $\sqrt{2P_i}\cos(2\pi f_c t + \zeta_i)$, where $P_i$ is the transmitted power, $\zeta_i$ is a random carrier phase uniformly distributed in $[0, 2\pi)$, and $f_c$ is the carrier frequency. The received signal due to the $i^{th}$ user may then be shown to be

$$x_i(t) = \Re\left(\sqrt{2P_i}s_i(t - \tau_i)e^{j2\pi f_c t + \psi_i}\right) \quad (4)$$

where $P_i = g_iP_c$ is the received signal power, $g_i \in [0, 1)$ and $\tau_i \in [0, T_{cs})$ are the propagation gain and delay, respectively, and $\psi_i = \zeta_i + \psi_i$ is the received phase of the signal due to the $i^{th}$ user so that $\psi_i$ is the phase shift introduced by the channel. The total received signal for all users in a noisy environment is

$$x(t) = \Re\left(\sum_{i=1}^M \sqrt{2P_i}s_i(t - \tau_i)e^{j2\pi f_c t + \psi_i}\right) + n(t) \quad (5)$$

where $M$ is the number of users and $n(t)$ is an additive noise waveform with unspecified statistics.

Discretising the delay associated with the $i^{th}$ user so that

$$\tau_i = (l_i + d_i)T_s \quad (6)$$

where $l_i$ and $d_i$ denote integer and fractional multiples of the sampling interval, $T_s = T_c/q$, where $q$ is the oversampling factor, then the sampled received signal for the $\ell$th sampling interval $x[\ell]$, is produced
where, for the $\ell$th sample interval, $t$ satisfies $(\ell - 1)T_s \leq t < \ell T_s$. In Equation (7) no assumptions are made about the statistical nature of the additive noise sequence $n[\ell]$.

In order to carry out delay estimation on this compound sampled signal the received samples must first be rearranged into frames of $qN_c$ samples using a tapped delay line.

Due to the asynchronism of the system the $n$th received data frame contains contributions from the $(n-1)$th transmitted data symbol as well as the $n$th transmitted data symbol of the $i$th user.

The tapped delay line produces a vector, $\mathbf{z}[n]$, which contains all the samples $\{x[(n-1)T_{cs} + 1], x[(n-1)T_{cs} + 2], \ldots, x[nT_{cs}]\}$ received over the $n$th symbol period $[(n-1)T_{cs}, nT_{cs})$. If $\mathcal{O}_M$ is an $M \times M$ matrix of zeros then

$$\mathbf{z}[n] = \begin{bmatrix} \mathbb{H}_i[n] & \mathbb{H}_p[n] \end{bmatrix} \begin{bmatrix} \mathcal{G} & \mathcal{O}_M \end{bmatrix} \begin{bmatrix} \mathcal{M}[n] \end{bmatrix} + \mathbf{n}[n]$$

where

$$\mathbb{H}_i[n] = [h_{i,1}[n], h_{i,2}[n], \ldots, h_{i,M}[n]] \quad \in \mathbb{R}^{qN_c \times M} \quad (9a)$$

$$\mathbb{H}_p[n] = [h_{p,1}[n], h_{p,2}[n], \ldots, h_{p,M}[n]] \quad \in \mathbb{R}^{qN_c \times M} \quad (9b)$$

$$\mathcal{G} = \operatorname{Re}\left\{\operatorname{diag}\left(\left[\sqrt{2P_1e^{j\phi_1}}, \sqrt{2P_2e^{j\phi_2}}, \ldots, \sqrt{2P_Me^{j\phi_M}}\right]\right)\right\} \quad \in \mathbb{R}^{M \times M} \quad (9c)$$

$$\mathcal{M}[n] = [a_1[n], a_2[n], \ldots, a_M[n]]^T \quad \in \mathbb{R}^{M} \quad (9d)$$

$$\mathbf{n}[n] = [n_1[n], n_2[n], \ldots, n_{qN_c}[n]]^T \quad \in \mathbb{R}^{qN_c} \quad (9e)$$

with the columns of the code signature matrices for the current and previous bits ($\mathbb{H}_i[n]$ and $\mathbb{H}_p[n]$), respectively) given by

$$h_{i,i}[n] = \operatorname{diag}\left(\left[\begin{array}{c} qN_c \\ 0, 0, \ldots, 0, 1, 1, \ldots, 1 \end{array}\right]\right) \quad \mathcal{O}_i[n]$$

$$h_{p,i}[n] = \operatorname{diag}\left(\left[\begin{array}{c} qN_c \\ 1, 1, \ldots, 1, 0, 0, \ldots, 0 \end{array}\right]\right) \quad \mathcal{O}_i[n]$$

where the oversampled code vector $\mathcal{O}_i[n] \in \mathbb{R}^{qN_c}$ is given by

$$\mathcal{O}_i[n] = \left[\alpha_i\left(\frac{(n-1)qN_c + q - (\ell_i + d_i)}{q}\right), \alpha_i\left(\frac{(n-1)qN_c + q - (\ell_i + d_i) + 1}{q}\right), \ldots, \alpha_i\left(\frac{qN_c + q - (\ell_i + d_i) - 1}{q}\right)\right]$$

For Equation 9c-e, $\mathcal{G}$ is referred to as the gain matrix, $\mathcal{M}[n]$ as the message vector and $\mathbf{n}[n]$ as the noise vector.
3 Proposed Adaptive H∞Delay-Tracking

In the previous section we saw that the $i$th user contributes two columns to the matrix $\mathbb{H}[n]$ where

$$\mathbb{H}[n] = \begin{bmatrix} \mathbb{H}_c[n] & \mathbb{H}_p[n] \end{bmatrix}$$

These are shifted versions of the code sequence for the $i$th user corresponding to the current and previous data symbol periods. This section demonstrates the application of a suitable subspace-based technique to this matrix. The linear combiner, or propagator was first proposed for antenna arrays in [10], and here the concept is extended for DS-CDMA delay estimation. The matrix $\mathbb{H}[n]$ may be partitioned into two submatrices, $\mathbb{H}_1$ and $\mathbb{H}_2$ as follows

$$\mathbb{H}[n] = \begin{bmatrix} 2M \\ \mathbb{H}_1[n] \\ \mathbb{H}_2[n] \end{bmatrix} \begin{bmatrix} 2M \\ qN_c - 2M \end{bmatrix}$$

where $\mathbb{H}_1[n]$ is full rank (it has linearly independent column vectors). Then we may show that it is possible to find a unique linear operator $\mathbb{L}[n]$ such that

$$\mathbb{L}^H[n] \mathbb{H}_1[n] = \mathbb{H}_2[n]$$

where $\mathbb{L}_m$ is a $2M \times (qN_c - 2M)$ matrix. We may then define the following matrix

$$\mathbb{Q}[n] = \begin{bmatrix} \mathbb{L}[n] \\ -I_{qN_c - 2M} \end{bmatrix} \in \mathbb{R}^{pN_c \times (qN_c - 2M)}$$

where $I_p$ denotes the $p \times p$ identity matrix. It can be shown that

$$\mathbb{Q}^H[n] \mathbb{H}[n] = \mathbb{Q}_{(qN_c - 2M) \times 2M}$$

where $\mathbb{Q}_{p \times q}$ denotes the $p \times q$ zero matrix. This means that the subspace spanned by the columns of $\mathbb{Q}[n]$ is orthogonal to the subspace spanned by the columns of $\mathbb{H}[n]$. That is to say that if we can estimate the linear combiner $\mathbb{L}[n]$, we may find a matrix $\mathbb{Q}[n]$ which describes the noise subspace of our system. This may then be used in a signal subspace type cost function to carry out delay estimation.

Note that the assumption that the matrix $\mathbb{H}_1[n]$ has linearly independent column vectors may not be true in the case of this delay estimator. However, we can always find $2M$ rows that are linearly independent using the method described below.

In [12], a method is given for determining the rank of a matrix in the presence of roundoff error, using QR factorization with column pivoting ($A\Pi = QR$). Given a priori knowledge of the rank of the matrix ($\text{rank}(\mathbb{H}) = 2M$), this method may be adapted to carry out a partial $A\Pi = QR$ factorization on the rows of $\mathbb{H}$, using only the first $2M$ steps to find the rows which best form an orthogonal basis for $\mathbb{H}$. This method was found to be insensitive to high noise levels, working consistently at signal-to-noise ratios below 10dB, as well as in the presence of moving sources.

After row determination has been carried out and the $\mathbb{H}[n]$ matrix partitioned accordingly, the received signal-vector $\mathbf{x}[n]$ can also be partitioned into two vectors $\mathbf{x}_1[n]$ and $\mathbf{x}_2[n]$ as follows

$$\begin{bmatrix} \mathbf{x}_1[n] \\ \mathbf{x}_2[n] \end{bmatrix} = \begin{bmatrix} \mathbb{H}_1[n] \\ \mathbb{H}_2[n] \end{bmatrix} \mathbf{a} + \begin{bmatrix} p_1[n] \\ p_2[n] \end{bmatrix}$$

Using Equation (13), the relationship between $\mathbf{x}_1[n]$ and $\mathbf{x}_2[n]$ can be expressed as
A. MANIKAS AND P. WILKINSON

\[ x_2[n] = L^H[n]x_1[n] + (p_2[n] - L^H[n]p_1[n]) \]  

(17)

It may then be shown that the complete system may be described by the following state-space model

\[
\begin{align*}
Q[n] &= \begin{bmatrix} -\frac{L[n]}{\gamma} & -\frac{L[n]}{\gamma} \\ -\frac{L[n]}{\gamma} & -\frac{L[n]}{\gamma} \end{bmatrix} \\
\varepsilon_1^H[n] &= \varepsilon_1^H[n]L[n] + \varepsilon[n] \\
\tilde{L}[n] + 1 &= \tilde{L}[n] + \mathbb{B} (\Delta L[n])
\end{align*}
\]  

(18)

where the linear combiner, \( L[n] \), now becomes the state matrix for the \( n \)th symbol, \( \mathbb{B} \) is a user-defined matrix which describes the upper bound on the change in the matrix \( L[n] \) from the \( n \)th to the \( (n + 1) \)th symbol and \( \Delta L[n] \) denotes the unknown variation in the state-matrix \( L[n] \) that will occur between the \( n \)th and the \( (n + 1) \)th symbol (the process noise). The unknown measurement noise vector \( \varepsilon[n] \) is defined as follows

\[ \varepsilon[n] = (p_2[n] - L^H[n]p_1[n]) \]

Having transformed the linear system model into a state space model we have a potential method for tracking the noise subspace as it changes in time using the state matrix and the orthogonal operator \( \mathbb{Q}[n] \). It is necessary to find a filter which may be applied to estimate the matrix \( \tilde{L}[n] \) given only the input vectors \( \{x_1[p], x_2[p], \forall p = 1, 2, \cdots, n\} \).

The filter chosen to perform the estimation in this paper belongs to the class of \( H^\infty \) filters. As already mentioned, the mobile environment is complex and contains multiple poorly characterised noise sources. The \( H^\infty \) estimation methods can be seen as a powerful and robust solution to handle parameter variations, modelling uncertainties and noise effects with only limited statistical information. By choosing the \textit{a priori} \( H^\infty \) estimator for the state-space model defined in Equation (18) the following result can be derived

\[ \tilde{L}[n + 1] = \tilde{L}[n] + \mathbb{P}[n]x_1[n] \left( 1 + \varepsilon_1^H[n]\mathbb{P}[n]x_1[n] \right)^{-1} (\varepsilon_1^H[n] - \varepsilon_1^H[n]L[n]) \]  

(19)

Where \( \tilde{L}[n] \) denotes the estimate of the state matrix \( L[n] \) at the \( n \)th data symbol. The initial estimate, \( \tilde{L}[0] \) may be obtained from the following least squares criterion

\[ \tilde{L}[0] = (X_1X_1^H)^{-1}X_2 \]

where \( X_1 = [x_1[1], \cdots, x_1[n]] \), \( X_2 = [x_2[1], \cdots, x_2[n]] \) and \( n_s \) denotes the number of samples used to form the initial estimate of the state matrix. In Equation (19) the matrix \( \mathbb{P}[n] \) is defined as follows

\[ \mathbb{P}[n] = \tilde{\mathbb{P}}[n] - \gamma^{-2}x_1[n]x_1^H[n] \]  

(20)

where \( \tilde{\mathbb{P}}[n] \) satisfies the recursive equation

\[ \tilde{\mathbb{P}}[n + 1] = \left[ \tilde{\mathbb{P}}[n] + (1 - \gamma^{-2})x_1[n]x_1^H[n] \right]^{-1} + \mathbb{B} \]  

(21)

initialized with \( \tilde{\mathbb{P}}[0] = \mu \mathbb{I} \) and \( \mathbb{B} = \beta \mathbb{I}_M \). The parameters \( \gamma \), \( \mu \) and \( \beta \) are user-defined and may be optimized for any particular environment.

A stable solution of the estimator exists as long as

\[ \gamma^2 \leq \sup_n (\beta + 1/x_1^H[n]x_1[n]) \]  

(22)
At each iteration of the recursive function described in Equation (19), the orthogonal transformation
\[ \mathbf{Q}[n] = \left[ \mathbf{I}_T[n], -\mathbf{I}_{N_c - 2M} \right]^T \]
is determined. We can then use any signal subspace method (e.g. [3]) to
obtain the delay of the desired signal. This delay estimate may then be supplied to a subspace type receiver
to complete the demodulation for the desired user.

4 Simulations

In order to demonstrate the performance of this delay tracking technique, the filter described above will be
compared with two others, a recursive least mean squares (LMS) filter and a recursive least squares filter
(RLS). These well known filters may be applied to the state space model described in Equation 18, to
form an estimate for the matrix \( \mathbf{Q}[n] \) and hence to carry out delay estimation. As a basis for comparison
with alternative estimation methods a simple exact eigendecomposition (ED) algorithm is used [11], as well
as the standard sliding correlator.

In the subsequent simulations, all users are assigned Gold sequences of length \( N_c = 15 \), generated by the
polynomials \( g_1(x) = x^4 + x + 1 \) and \( g_2(x) = x^4 + x^3 + 1 \). The signal is scaled so that the received
power of the desired user is 1. The chip period \( T_c \) is taken equal to 0.8\( \mu \)sec (as in IS-95) and the rest of the
signal parameters such as carrier phase and frequency are assumed to be perfectly tracked. The data bits
are binary random variables with \( \pm 1 \) having equal probability. Each run of a simulation is carried out
over 300 iterations (symbol periods).

The parameters used for each delay estimator are summarised in Table 1. Note that it may be possible to
improve on the performance of any of the algorithms for a particular scenario by changing the receiver
parameters, but in order to demonstrate the overall performance the parameters were chosen to produce
the best possible performance across a range of environments.

<table>
<thead>
<tr>
<th>Receiver</th>
<th>Delay Estimation Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H^\infty )</td>
<td>( \gamma_H = 1.4, \mu_H = 0.9, \beta_H = 5 \times 10^{-4} )</td>
</tr>
<tr>
<td>ED</td>
<td>( \lambda_E = 0.95 )</td>
</tr>
<tr>
<td>LMS</td>
<td>( \mu_L = 0.01 )</td>
</tr>
<tr>
<td>RLS</td>
<td>( \delta_R = 0.01, \lambda_R = 0.9 )</td>
</tr>
</tbody>
</table>

Table 1: Delay Estimation Parameters

The first scenario considers the performance of the algorithms with 8 users (1 desired, 7 interferences),
they are under perfect power control conditions (i.e. \( P_i = 1, i = 1, \cdots, M \)) and with an input signal-to-
noise ratio (\( \text{SNR}_\text{in} \)) of 10dB. The noise is additive white Gaussian, uncorrelated from sample to sample.
An exemplary set of results for such an environment are shown in Figure 2 for both the proposed
algorithm and the RLS based algorithm. In order to compare the algorithms in the less hospitable non-
Gaussian noise environment, Figure 3 shows an equivalent set of results for the case of uniform noise,
again with an \( \text{SNR}_\text{in} \) of 10dB.

To find the mean squared delay tracking error for each algorithm, the simulation was repeated 100 times in
each noise scenario with the same delays, but different realisations of the noise waveform. The mean squared delay tracking error is shown in Table 2 for each of the four subspace based algorithms. No result is shown for the sliding correlator since in this environment the high level of MAI causes delay
acquisition and tracking to fail.
In the presence of Gaussian noise, the ED algorithm performs the best, followed by the $H^\infty$ based algorithm. The RLS algorithm provides some level of tracking, while the LMS performs very poorly due to the large number of users present. Clearly the $H^\infty$ method is the most effective in the uniform noise environment; the RLS and ED algorithms both provide an acceptable level of tracking, while the LMS algorithm once again performs badly.

Figure 2: Example of Tracking Performance and Adaptation of Proposed Algorithm in Power Controlled Environment. $M = 8$, $q = 4$, Gaussian Noise, $SNR_{\text{in}} = 10$dB

Figure 3: Example of Tracking Performance and Adaptation of Proposed Algorithm in Power Controlled Environment. $M = 8$, $q = 4$, Uniform Noise, $SNR_{\text{in}} = 10$dB
Table 2: MSE Delay Tracking Results in units of $T_c$ for $M = 8, q = 4$, $\text{SNR}_{in} = 10\text{dB}$

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$H^\infty$</th>
<th>ED</th>
<th>RLS</th>
<th>LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td>0.556</td>
<td>0.434</td>
<td>0.893</td>
<td>2.863</td>
</tr>
<tr>
<td>Uniform Noise</td>
<td>0.422</td>
<td>0.518</td>
<td>0.570</td>
<td>2.242</td>
</tr>
</tbody>
</table>

In order to demonstrate the usefulness of the LMS and RLS algorithms, the same simulation was repeated with a smaller number of users. In this environment these two algorithms may be shown to work rather better. The results are shown in Table (3). Indeed for this small number of users, under perfect power control, the performance of the RLS algorithm surpasses that of both the $H^\infty$ and ED algorithms, and even the LMS algorithm performs reasonably well. The $H^\infty$ based algorithm slightly outperforms the ED algorithm in both Gaussian and uniform noise.

Table 3: MSE Delay Tracking Results in units of $T_c$ for $M = 4, q = 4$, $\text{SNR}_{in} = 10\text{dB}$

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$H^\infty$</th>
<th>ED</th>
<th>RLS</th>
<th>LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td>0.305</td>
<td>0.345</td>
<td>0.195</td>
<td>0.340</td>
</tr>
<tr>
<td>Uniform Noise</td>
<td>0.341</td>
<td>0.393</td>
<td>0.225</td>
<td>0.394</td>
</tr>
</tbody>
</table>

Another interesting point is that for this small number of users under perfect power control, all the estimators perform better in Gaussian noise than in uniform noise. This is a result of the fact that where a small number of users are present, the signal environment may be considered to be truly Gaussian and uncorrelated. However, where there are a large number of users present, the relatively short code length leads to a signal environment that is quite correlated. This means that even in the Gaussian noise case, the environment contains unknown correlations and disturbances. This mixed environment degrades the performance of all the algorithms where there a larger number of users present, but the propagator based algorithms are better able to cope with it than the ED algorithm.

It should be noted that in considering uniform noise we are not carrying out an exhaustive treatment of the non-Gaussian noise case - it simply serves to demonstrate that in the case where the noise environment is non-Gaussian, the $H^\infty$ filter tends to produce better results than the other estimators. As noted above, the majority of channels will contain significant non-Gaussian components, and hence the proposed method is an attractive one for dealing with such cases.

Figure (4) shows how the mean delay tracking error varies with the number of active users for the sliding correlator, the ED algorithm and the $H^\infty$ algorithm. All the users are perfectly power controlled and the noise is additive white Gaussian. Clearly the sliding correlator performs the best for a single user, but due to cross-correlations between the user codes its performance degrades severely as the number of users increases until eventually it totally fails to track the delay.

For a small number of active users (less than 9) the performance of the proposed algorithm is virtually identical to that of the ED algorithm. Although this means that there is no performance gain from implementing the proposed algorithm in this scenario, there are two points that are worth noting. Firstly, this is the best case for the ED algorithm. Since it is based on Eigendecomposition it performs best in an AWGN (additive white Gaussian noise) environment. In contrast, the $H^\infty$ algorithm may be expected to perform well across a whole range of noise environments because it makes no assumptions concerning the nature of any disturbances. Secondly, since it requires an Eigenvector update at each iteration, the ED algorithm requires a lot of processing, particularly for large $N_e$. Tracking the subspace using the Linear
Combiner involves matrix inversions at every step, but is still not so computationally expensive as the ED
algorithm. Therefore, it may be worth considering the proposed estimator even in this scenario.

The next case to consider is the environment, where the desired user signal is masked at the
receiver by higher powered interferences. The first scenario under consideration contains 4 users, with the
desired user received at a power level 10dB below that of the interfering users. The noise power is 10dB
below that of the desired user. The results for this scenario are shown in Table 4. Clearly the
performance of all the algorithms has worsened to some degree. In fact, the assertion that the performance
of subspace algorithms is independent of signal power is only true for the estimation of static delays. In the
independent case where the delay changes over the observation interval, we can see various degrees of robustness to the
near far effect.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$H^\infty$</th>
<th>ED</th>
<th>RLS</th>
<th>LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Noise</td>
<td>1.504</td>
<td>1.450</td>
<td>3.887</td>
<td>4.902</td>
</tr>
<tr>
<td>Uniform Noise</td>
<td>1.560</td>
<td>1.481</td>
<td>3.836</td>
<td>5.836</td>
</tr>
</tbody>
</table>

Table 4: MSE Delay Tracking Results. Near Far Ratio of 10dB, $M = 4$, $q = 4$, Gaussian Noise,
$\text{SNR}_{\text{in}} = 10\text{dB}$.

Further data on the performance of the various algorithms in the presence of the near far effect is presented
in Figure (5). This figure shows the variation of the delay tracking MSE for a selection of near far ratios.
Once again the noise is Gaussian and received at a power 10dB below that of the desired user. The
interference users are received at higher power than the desired user, according to the specified Near Far
ratio.
As noted in Table (3), the RLS algorithm actually produces the best performance for 4 perfectly power controlled users, and even the LMS algorithm produces acceptable performance. However, Figure (5) shows that in both cases this performance rapidly worsens as the near far ratio increases, so that the $H^\infty$ and ED algorithms may be seen to have the best performance at medium to high near far ratios, providing roughly equivalent results.

Figure 5: MSE Delay Tracking Results in Near Far Conditions, $M = 4, q = 4$, Gaussian Noise, $\text{SNR}_\text{in} = 10\text{dB}$.

5 Conclusion

The problem of delay estimation for DS-CDMA systems is an important one which must be examined exhaustively if the future mobile phone networks are to be able to meet subscriber demand for signal quality and channel density. In this paper a novel approach has been proposed to deal with the problem of delay estimation in dynamic channel environments. This involved the manipulation of the user PN-code matrix in order to define a matrix operator known as the linear combiner. This linear combiner was then reformulated as the state matrix in a state space model. The preferred solution to this state space model was chosen to be the a priori $H^\infty$ solution. This permits the delay estimator to operate successfully even in the presence of poorly characterised disturbances, such as noise for which only limited statistical information is available.

Through the use of simulation studies the approach has been shown to successfully track the delay of a single desired user over a range of signal environments, in the presence of multiple access interference and noise (both Gaussian and non-Gaussian). It has also been shown to be robust to the near far effect. The $H^\infty$ formulation has been compared with the RLS and LMS filters, applied to the same state space model and also with the conventional sliding correlator and a standard subspace method.
The proposed approach has been shown to provide a powerful near-far resistant solution to the problem of delay tracking in a multi-user environment, robust to any modelling errors.

6 References