

Delay Estimation and Data Detection in Long-Code DS-CDMA System

Pei Xiao, Erik Ström

Communication Systems Group, Dept. of Signals and Systems
Chalmers University of Technology, SE-412 96, Göteborg, Sweden

E-mail: pei.xiao@s2.chalmers.se, erik.strom@s2.chalmers.se

Abstract— In this paper, we aim at solving the problem of joint delay estimation and data detection of the orthogonal modulated signals in the asynchronous DS-CDMA system employing aperiodic long spreading codes over fading channels. The general system requirement of low error rate in data demodulation necessitates the reliable synchronization mechanisms. Synchronization of CDMA signals with long spreading codes is a more challenging task than that of CDMA signals with short spreading codes. In this work, two algorithms are introduced to perform acquisition and tracking of orthogonal modulated signals with long spreading codes, followed by data detection. The numerical results show that when applying to the asynchronous system with random propagation delays, the proposed algorithms approximate the performance that is attainable in the synchronized and chip-aligned system.

I. INTRODUCTION

The considered system in this paper is an asynchronous DS-CDMA system with orthogonal modulation. The transmitted chip sequence from a particular user is the concatenation of one of M possible Walsh codes (representing the transmitted symbol) and a long scrambling (spreading) code. The rationale for using orthogonal modulation is to avoid having to track the phase in a rapid fading environment. For example, in the IS-95 uplink, in addition to the 1/3 convolutional code, there is 64-ary orthogonal modulation, also acting as a block code, which is suitable for non-coherent demodulation [1].

Demodulation of M -ary orthogonal signalling formats in DS-CDMA system has been the subject of study in various papers. For instance, interference cancellation, i.e., removal of the multiple access interferences (MAI) by means of subtraction, and interference suppression, i.e., removal of MAI by means of orthogonal projection are studied in [2], [3] and references therein. They were shown to achieve significant performance gains compared to the conventional receiver.

Perfect synchronization and chip-synchronous were assumed in the data detection algorithms mentioned above. The impact of synchronization errors was studied in [3]. It was shown that errors in the delay estimates would drastically deteriorate the system performance.

Synchronization issues were dealt with, e.g., in [4], [5], [6], etc.. However, the synchronization algorithms proposed in those papers only apply to CDMA systems with periodic short spreading sequences and the modulation schemes other than orthogonal signalling. To make MAI more white noise like, thus better combat the interference and other manner of channel impairments [1], long spreading sequences are commonly used, for instance, in the existing IS-95 system as well as in the IMT-2000/UMTS standard-

ization. In general, a system with long spreading sequences is much more difficult to synchronize than the one with short spreading sequences, owing to the fact that the repetitive data structure available in the latter system is scrambled by the long spreading sequences employed in the former system.

Synchronization of the orthogonal modulated signals with long spreading sequences was explored in [7] and [8]. In [7], an adaptive algorithm was proposed to estimate the errors in synchronization. Based on the estimation, remedial actions were taken to alleviate the performance degradation caused by sampling the received signals at the incorrect timing. Simulation results showed considerable capacity gains when the proposed algorithms were applied on erroneously sampled signals. However, perfect acquisition and a constant synchronization error to all the users were assumed in this paper, which applies to the downlink scenario in which all the users transmit in a synchronous manner. In the uplink, different users transmit signals asynchronously, the propagation delays are therefore randomly distributed among different users. The problem of uplink synchronization was treated in [8], several acquisition algorithms were proposed to estimate the propagation delays of different users. Given enough pilot symbols, they achieve good acquisition performance in multiuser environment compared to the conventional delay estimator.

In this paper, we emphasize on how to utilize the estimated delays for detecting the transmitted data, as well as how to use the detected data to simplify the delay estimation process. In section II, the system model is presented. Different schemes suitable for estimating propagation delays of the M -ary orthogonal modulated signals are introduced in Section III. Data detection algorithms that can work jointly with delay estimators are described in Section IV. Synchronization schemes for a new user that joins the transmission at a later stage are discussed in Section V. Finally, the simulation results are shown in Section VI and conclusions drawn in Section VII.

II. SYSTEM MODEL

The passband received signal due to the k th user is denoted by $r_{\text{RF},k}(t)$. The k th user's j th symbol is denoted by $i_k(j) \in \{1, 2, \dots, M\}$, and the M orthogonal signal alternatives are defined by M Walsh sequences $\{w_1(n), w_2(n), \dots, w_M(n)\}$ of length N . The Walsh chips are randomized by a scrambling code $c_k(n) \in \{-1, +1\}$. Hence, every symbol which represents $\log_2 M$ binary bits is spread by N chips and each bit by $N/\log_2 M$ chips.

The baseband signal $s_k(t)$ is formed by pulse amplitude modulation with the unit-energy rectangular chip waveform $\psi(t)$, and the baseband signal is multiplied with a carrier with frequency ω_c and transmitted over the channel, which is represented by the complex channel gain $h_k(t)$ and assumed to be a slowly time-varying Rayleigh flat fading channel with delay τ_k .

The total received signal is the sum of the K users' signals plus additive white complex Gaussian noise $n(t)$ with power spectral density $N_0/2$. The complex envelope of the received signal is

$$r(t) = n(t) + \sum_{k=1}^K r_k(t), \quad r_{\text{RF},k}(t) = \sqrt{2} \text{Re}\{r_k(t)e^{j\omega_c t}\}$$

The received vector, $\mathbf{r}(j) \in \mathbb{C}^N$, due to transmission of the j th symbol can be formed as

$$\mathbf{r}(j) = \mathbf{A}(j, \boldsymbol{\tau})\mathbf{h}(j) + \mathbf{n}(j) \quad (1)$$

The elements of $\mathbf{r}(j)$ are samples from the chip-matched filter. The zero-mean complex Gaussian random vector $\mathbf{n}(j) \in \mathbb{C}^N$ has second moments $\mathbb{E}[\mathbf{n}(i)\mathbf{n}^T(j)] = \mathbf{0}$ and $\mathbb{E}[\mathbf{n}(i)\mathbf{n}^*(j)] = N_0\mathbf{I}_N\delta(i-j)$, where $\delta(i)$ is the Kronecker delta function. The vector $\mathbf{h}(j) \in \mathbb{C}^K$ is defined by the complex channel gains as $\mathbf{h}(j) = [h_1(jT) \ h_2(jT) \ \cdots \ h_K(jT)]^T$. The matrix $\mathbf{A}(j, \boldsymbol{\tau}) \in \mathbb{R}^{N \times K}$ can be expressed as

$$\mathbf{A}(j, \boldsymbol{\tau}) = [\mathbf{a}_1(j, \tau_1) \ \mathbf{a}_2(j, \tau_2) \ \cdots \ \mathbf{a}_K(j, \tau_K)] \quad (2)$$

where τ_k stands for the propagation delay of the k th user. The vector $\boldsymbol{\tau}$ is defined as $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_K]^T \in \mathbb{R}^K$. Let us denote $\tau_k = (p_k + \delta_k)T_c$ (T_c is one chip interval) such that $p_k \in \{0, 1, \dots, N-1\}$ stands for the chip delay, $\delta_k \in [0, 1)$ stands for the interchip delay, and denote $\mathbf{b}_k(j) \in \mathbb{R}^N$, the transmitted chip sequence due to the k th user as

$$\mathbf{b}_k(j) = \mathbf{C}_k(j)\mathbf{w}_{i_k(j)} \quad (3)$$

where $\mathbf{C}_k(j) \in \{-1, +1\}^{N \times N}$ is a diagonal matrix defined by the k th user's scrambling code, and $\mathbf{w}_m \in \{-1, +1\}^{N \times 1}$ is the m th column ($m = i_k(j)$) of the $N \times N$ Hadamard matrix. Then, each column of the matrix \mathbf{A} in (2) can be expressed as

$$\mathbf{a}_k(j, \tau_k) = (1 - \delta_k) [\text{ds}(\mathbf{b}_k(j), p_k) + \text{us}(\mathbf{b}_k(j-1), N - p_k)] + \delta_k [\text{ds}(\mathbf{b}_k(j), p_k + 1) + \text{us}(\mathbf{b}_k(j-1), N - p_k - 1)] \quad (4)$$

where $\text{us}(\cdot)$, $\text{ds}(\cdot)$ stand for the up-shift and down-shift operators respectively: $\text{us}([\mathbf{a}_1 \ \cdots \ \mathbf{a}_N]^T, q) = [\mathbf{a}_{N+1-q} \ \cdots \ \mathbf{a}_N \ 0 \ \cdots \ 0]^T$, $\text{ds}([\mathbf{a}_1 \ \cdots \ \mathbf{a}_N]^T, q) = [0 \ \cdots \ 0 \ \mathbf{a}_1 \ \cdots \ \mathbf{a}_{N-q}]^T$. For the periodic signal employed at the delay estimation stage, $\mathbf{b}_k(j)$ remains unchanged during the training period. Therefore, we can drop the symbol index j from $\mathbf{a}_k(j, \tau_k)$.

III. DELAY ESTIMATION

The task of the delay estimator in the receiver (Fig. 1) is to detect the propagation delays τ_k for $k = 1, 2, \dots, K$ given the received signal vector $\mathbf{r}(j)$. The decision on the

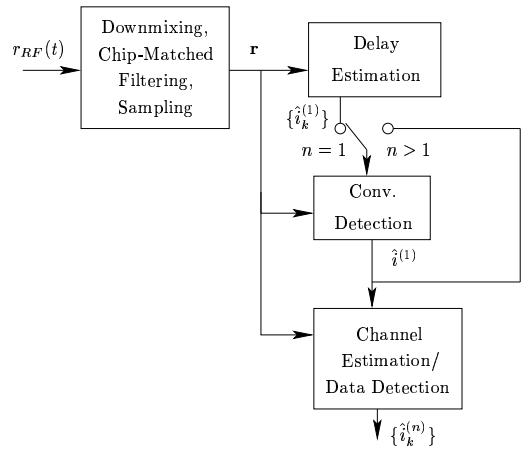


Fig. 1. Receiver Block Diagram

k th user's delay τ_k is found as the maximizer/minimizer of the merit/cost function $J_k(\tau_k)$

$$\hat{\tau}_k = \arg \max / \min_{\tau_k \in [0, T]} J_k(\tau_k)$$

We define the failure of acquisition (estimation outlier) to be the case when the estimated delay deviates from the true value by more than half chip interval, i.e., $|\hat{\tau}_k - \tau_k| > T_c/2$.

In the following, we shall introduce how the decision function $J_k(\tau_k)$ is formed for different delay estimation algorithms.

A. Conventional Sliding Correlator

The sliding correlator (SC) is the standard approach to propagation delay estimation. The merit function $J_{k,C}$ is formed by correlating the received signal with time delayed versions of the known code sequence $\mathbf{a}_k(\tau)$

$$\hat{\tau}_k^c = \arg \max_{\tau_k \in [0, T]} J_{k,C}(\tau_k) = \arg \max_{\tau_k \in [0, T]} \frac{|\mathbf{a}_k(\tau_k)^* \bar{\mathbf{r}}|^2}{\|\mathbf{a}_k(\tau_k)\|^2} \quad (5)$$

where $\bar{\mathbf{r}} = \frac{1}{L} \sum_{l=1}^L \mathbf{r}(l)$ is the sample mean of the received vector. The estimated delay is the candidate delay value that maximizes the correlation.

Like the standard receiver (i.e., the matched filter), the SC is only optimal in the AWGN single user channel or in strict orthogonal synchronous channel. It is highly unreliable in the presence of MAI. The subspace-based algorithm that is suitable for multiuser environment is presented below.

B. Subspace-based Delay Estimator

The covariance matrix for the vector \mathbf{r} is defined as

$$\mathbf{R} = \mathbb{E}[\mathbf{r}\mathbf{r}^*] = \mathbf{A} \mathbb{E}[\mathbf{h}\mathbf{h}^*] \mathbf{A}^* + N_0\mathbf{I}_N = \mathbf{A}\mathbf{P}\mathbf{A}^* + N_0\mathbf{I}_N \quad (6)$$

We assume that \mathbf{A} and \mathbf{P} in the equation (6) have full rank, and define the *signal subspace* as the column space of \mathbf{A} , i.e., $\text{range}(\mathbf{A})$, and the *noise subspace* as the orthogonal complement to $\text{range}(\mathbf{A})$. It can be shown that $\text{range}(\mathbf{A}\mathbf{P}\mathbf{A}^*) = \text{range}(\mathbf{A})$. Furthermore, since \mathbf{A} has rank K (the number of users), the signal subspace will have dimensionality K .

The subspace-based delay estimation involves an eigenvalue decomposition of the hermitian and positive definite matrix \mathbf{R} into *signal subspace*, i.e., $\text{range}(\mathbf{E}_s)$ and *noise subspace*, i.e., $\text{range}(\mathbf{E}_n)$

$$\mathbf{R} = \mathbf{E}_s \Lambda_s \mathbf{E}_s^* + N_0 \mathbf{E}_n \mathbf{E}_n^*$$

where $\mathbf{E}_s \in \mathbb{C}^{N \times K}$ and $\mathbf{E}_n \in \mathbb{C}^{N \times (N-K)}$ are such that $[\mathbf{E}_s \ \mathbf{E}_n]$ is unitary, and $\Lambda_s \in \mathbb{R}^{K \times K}$ is defined as

$$\Lambda_s = \text{diag}(\lambda_1 + N_0, \lambda_2 + N_0, \dots, \lambda_K + N_0)$$

where $\{\lambda_1, \dots, \lambda_K\}$ are the non-zero eigenvalues of $\mathbf{A} \mathbf{P} \mathbf{A}^*$.

Now we note that $\text{range}(\mathbf{A}) = \text{range}(\mathbf{E}_s)$ and consequently, the columns of $\mathbf{A}(\mathbf{a}_1(\tau_1), \mathbf{a}_2(\tau_2), \dots, \mathbf{a}_K(\tau_K))$ are orthogonal to the columns of \mathbf{E}_n . Given the knowledge of \mathbf{R} we can find τ_k as the solutions to $\|\mathbf{E}_n^* \mathbf{a}_k(\tau_k)\|^2 = 0$. In practice, the covariance matrix is unknown and usually estimated by the *sample covariance matrix* $\hat{\mathbf{R}}$, and a consistent estimate of \mathbf{E}_n is found in the eigenvalue decomposition of $\hat{\mathbf{R}}$

$$\hat{\mathbf{R}} = \frac{1}{L} \sum_{l=1}^L \mathbf{r}(l) \mathbf{r}^*(l) = \hat{\mathbf{E}}_s \hat{\Lambda}_s \hat{\mathbf{E}}_s^* + \hat{\mathbf{E}}_n \hat{\Lambda}_n \hat{\mathbf{E}}_n^*$$

where the columns of $\hat{\mathbf{E}}_n$ are the eigenvectors corresponding to the $N - K$ smallest eigenvalues of $\hat{\mathbf{R}}$. Note, however, that the columns of \mathbf{A} will now be only approximately orthogonal to the columns of $\hat{\mathbf{E}}_n$. The above idea leads to the subspace-based delay estimator formulated as follows

$$\hat{\tau}_k^s = \arg \min_{\tau_k \in [0, T]} J_{k,S}(\tau_k) = \arg \min_{\tau_k \in [0, T]} \frac{\|\hat{\mathbf{E}}_n^* \mathbf{a}_k(\tau_k)\|^2}{\|\mathbf{a}_k(\tau_k)\|^2} \quad (7)$$

The delay estimate $\hat{\tau}_k$ can be derived by exhaustive search at each time delay candidate and picking up the one which maximizes/minimizes the merit/cost function (5)/(7). Another approach is to find the solutions to $dJ_k(\tau_k)/d\tau_k = 0$ and search for a global maxima/minima [8].

This subspace-based delay estimator requires periodic training sequences for different users with good cross-correlation property. This can be fulfilled by assigning a distinct Walsh sequence to each user and spread it with periodic scrambling code to achieve low cross-correlation between the shifted versions of transmitted sequences.

Compared to the SC, the subspace-based estimator requires much longer training sequence in order to get an accurate estimate of the covariance matrix \mathbf{R} (see [8]). Due to the fast convergence of the SC, it can be applied in some specific situations, e.g., estimating the delay for a new user who joins the transmission at a later stage or in multiuser environment when the MAI can be effectively canceled. This is further discussed in Section V and VI.

IV. DATA DETECTION

After code synchronization, the receiver adjusts the sampling timing to align with the desired user k based on the estimated delay $\hat{\tau}_k$, when detecting the transmitted symbols from that user. Without loss of generality, we assume the first user to be the desired user. The new received vector

$\tilde{\mathbf{r}}(j)$ is formed as

$$\begin{aligned} \tilde{\mathbf{r}}(j) &= \tilde{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}}) \mathbf{h}(j) + \mathbf{n}(j) \\ \tilde{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}}) &= [\mathbf{a}_1(j, \tau_1 - \hat{\tau}_1) \quad \mathbf{a}_2(j, \tau_2 - \hat{\tau}_1) \quad \dots \quad \mathbf{a}_K(j, \tau_K - \hat{\tau}_1)] \\ &= [\mathbf{a}_1(j, \tilde{\tau}_1) \quad \mathbf{a}_2(j, \tilde{\tau}_2) \quad \dots \quad \mathbf{a}_K(j, \tilde{\tau}_K)] \\ \mathbf{a}_k(j, \tilde{\tau}_k) &= (1 - \tilde{\delta}_k) [\text{ds}(\mathbf{b}_k(j), \tilde{p}_k) + \text{us}(\mathbf{b}_k(j-1), N - \tilde{p}_k)] \\ &\quad + \tilde{\delta}_k [\text{ds}(\mathbf{b}_k(j), \tilde{p}_k + 1) + \text{us}(\mathbf{b}_k(j-1), N - \tilde{p}_k - 1)] \\ \mathbf{b}_k(j) &= \mathbf{C}_k(j) \mathbf{w}_{i_k(j)} \end{aligned} \quad (8)$$

where the vector $\tilde{\boldsymbol{\tau}}$ is defined as $\tilde{\boldsymbol{\tau}} = [\tilde{\tau}_1, \tilde{\tau}_2, \dots, \tilde{\tau}_K]$, and $\tilde{\tau}_k$ is defined as $\tilde{\tau}_k = \tau_k - \hat{\tau}_1 = (\tilde{p}_k + \tilde{\delta}_k) T_c$ such that \tilde{p}_k is an integer and $\tilde{\delta}_k \in [0, 1)$.

We denote the decision on the k th user's symbol at the n th iteration stage by $\hat{i}_k^{(n)}$, the multi-stage iterative detection can be formulated as

$$\hat{i}_k^{(n)} = \arg \max_{m \in \{1, 2, \dots, M\}} z_k^{(n)}(m)$$

The soft decision $z_k(m)$ for different receiver algorithms is derived below.

A. Conventional (Conv.) first stage

The conventional detection technique is to form the soft decision by correlating the received signal with the M possible transmitted waveforms $\mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \dots, \mathbf{a}_{k,m}, \dots, \mathbf{a}_{k,M}$. The soft decision can be expressed as $z_k(m) = |\mathbf{w}_m^* \mathbf{C}_k^* \tilde{\mathbf{r}}(j)| = |\mathbf{a}_{k,m}^* \tilde{\mathbf{r}}(j)|$ where we define $\mathbf{a}_{k,m} = \mathbf{C}_k \mathbf{w}_m$. This simple scheme is generally used in the beginning of the detection process, e.g., at the first iteration stage (see Fig. 1), when the estimate of the fading channel is lacking.

B. Parallel interference cancellation (PIC)

The PIC scheme detects all the users' signals at the same time and cancels the MAI in parallel at the next iteration stage. The soft decision of PIC can be formed as

$$z_k^{(n)}(m) = \text{Re}\{\hat{h}_k^* \mathbf{a}_{k,m}^* [\tilde{\mathbf{r}}(j) - \hat{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}}) \hat{\mathbf{h}}(j) + \hat{h}_k \hat{\mathbf{a}}_k(j, \tilde{\tau}_k)]\} \quad (9)$$

where $\hat{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}})$ and $\hat{\mathbf{a}}_k(j, \tilde{\tau}_k)$ are obtained by substituting $\hat{i}_k^{(n-1)}$ and delay estimate $\hat{\tau}_k$ into (8). The vector $\hat{\mathbf{h}}(j)$ is the estimate of the fading process $\mathbf{h}(j)$. An ML approach to channel estimation is $\hat{\mathbf{h}}(j) = \hat{\mathbf{A}}^\dagger(j, \tilde{\boldsymbol{\tau}}) \tilde{\mathbf{r}}(j)$, where $\hat{\mathbf{A}}^\dagger(j, \tilde{\boldsymbol{\tau}}) = [\hat{\mathbf{A}}^*(j, \tilde{\boldsymbol{\tau}}) \hat{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}})]^{-1} \hat{\mathbf{A}}^*(j, \tilde{\boldsymbol{\tau}})$ denotes the left pseudoinverse of $\hat{\mathbf{A}}(j, \tilde{\boldsymbol{\tau}})$ [2].

C. Iterative interference suppression (IIS)

IIS differs from PIC in that it suppresses the MAI via orthogonal projection rather than subtraction.

If we delete the column due to the k th user from the matrix $\tilde{\mathbf{A}}$ in (8) and form the matrix \mathbf{U} due to the MAI:

$$\mathbf{U} = [\mathbf{a}_1 \quad \dots \quad \mathbf{a}_{k-1} \quad \mathbf{a}_{k+1} \quad \dots \quad \mathbf{a}_K]^T$$

Then $\mathbf{P}_U^\perp = \mathbf{I} - \mathbf{U} \mathbf{U}^\dagger$ is the orthogonal projection matrix onto the orthogonal complement to the subspace spanned

by the columns of \mathbf{U} , where \mathbf{U}^\dagger denotes the left pseudoinverse of \mathbf{U} . The IIS receiver can be formulated as

$$z_k^{(n)}(m) = \text{Re}\{\hat{h}_k^* \mathbf{a}_{k,m}^* \mathbf{P}_{\hat{\mathbf{U}}}^\perp \tilde{\mathbf{r}}(j)\} - 0.5 \|\mathbf{P}_{\hat{\mathbf{U}}}^\perp \mathbf{a}_{k,m} \hat{h}_k\|^2 \quad (10)$$

where $\hat{\mathbf{U}}$ is the estimate of \mathbf{U} , i.e., the estimated MAI at the previous iteration stage.

Both PIC and IIS can be used for the subsequent data detection after the initial conventional stage (see Fig. 1). To improve the performance, we can shift the local reference $\mathbf{a}_{k,m}$ accordingly based on the delay estimate

$$\bar{\mathbf{a}}_{k,m} = (1 - \tilde{\delta}_k) \mathbf{a}_{k,m} + \tilde{\delta}_k \text{ds}(\mathbf{a}_{k,m}, 1)$$

Then $\mathbf{a}_{k,m}$ in equation (9) and (10) can be replaced by its shifted version.

V. SYNCHRONIZATION OF A NEW USER

Here we consider the situation that a new user joins the transmission after the previous K users have started. The data matrix $\mathbf{A}(j, \boldsymbol{\tau})$ becomes

$$\mathbf{A}(j, \boldsymbol{\tau}) = [\mathbf{a}_1(j, \tau_1) \quad \mathbf{a}_2(j, \tau_2) \quad \cdots \quad \mathbf{a}_K(j, \tau_K) \quad \mathbf{a}_{K+1}(j, \tau_{K+1})]$$

where $\tau_1, \tau_2, \dots, \tau_K$ were already estimated before the previous K users started their data transmission. Now, the task is to detect the delay τ_{K+1} for the new user. The following two scenarios lead to different solutions to the problem.

A. Scenario 1

We assume all the other users stop data transmission and start training together with the new user in order to help it get synchronized. This would be the ideal situation, probably not practical though. In this case, both aforementioned acquisition schemes can be applied. However, the subspace-based estimator is not able to utilize the previously estimated delays $\hat{\tau}_1, \hat{\tau}_2, \dots, \hat{\tau}_K$ and requires a long training sequence, which is waste of system resources and disturbance of ongoing transmission. It would be advantageous to apply the conventional SC under such a circumstance.

To synchronize the new user with the SC, we subtract/suppress other users' signal from the received signal \mathbf{r} using the known pilot symbols and previously detected delays of the other users. The channel becomes like a single user channel after the interference cancellation/suppression using PIC/IIS.

The simulation results are shown in Fig. 2. The SC achieves good acquisition performance for the new user with approximately 10 pilot symbols, when it is collaborated with PIC. As expected, the subspace-based estimator does not work well with a short training sequence.

B. Scenario 2

A more realistic scenario would be that all the previous K users ignore the existence of the new user and keep on transmitting data, the new user starts training on its own to figure out its propagation delay. In this case, the subspace-based algorithm wouldn't work since the data transmitted by the other users differ from symbol to symbol, the data

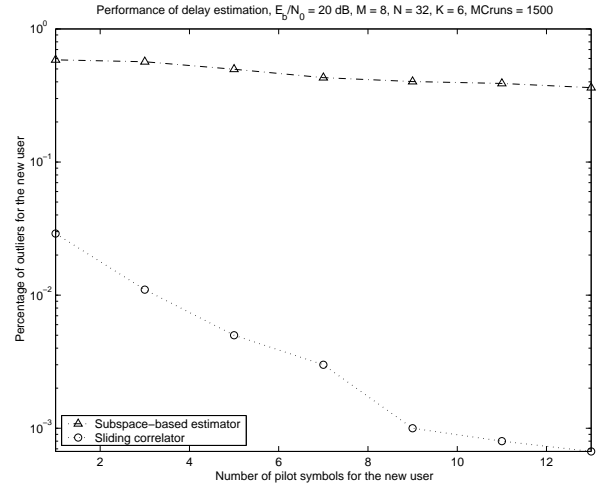


Fig. 2. Acquisition error probability for the new user as a function of the number of pilot symbols (scenario 1)

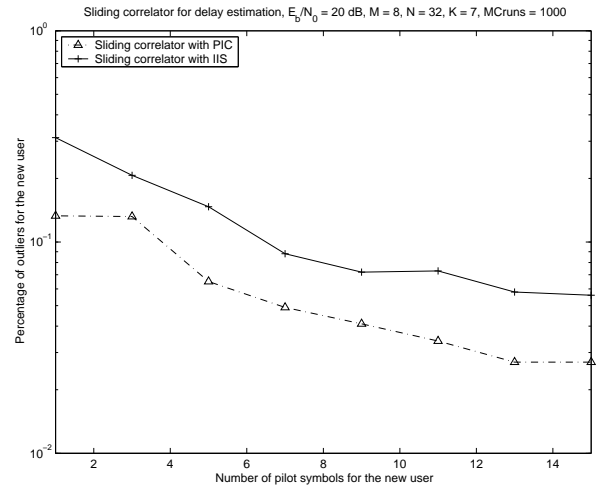


Fig. 3. Acquisition error probability for the new user as a function of the number of pilot symbols (scenario 2)

matrix \mathbf{A} in (6) keeps changing all the time. There is no way of tracking down the covariance matrix \mathbf{R} . A feasible solution is to perform PIC or IIS to remove the interferences using the data of other users detected at previous iteration. Then the new user applies the SC to estimate its delay. The result is depicted in Fig. 3. It is evident from the figure that the SC works better with PIC than IIS (the interference is more thoroughly removed, the delay of the new user is better estimated). This scheme has more practical value than the previous one because it does not require training from the other users.

This PIC-SC or IIS-SC scheme differs from the one introduced in scenario 1 in that the interference cancellation is done in decision directed mode, using the estimated data. While in scenario 1, the data transmitted by other users are known, as we assume all the users are in the training state. That is the reason why the performance of the PIC-SC scheme in scenario 1 (illustrated in Figure 2) is better than the one in scenario 2 (illustrated in Figure 3).

In consideration of the fact that the probability for different users to start up transmission at exactly the same

time is low, the scheme derived based on scenario 2 applies in general situation: after synchronizing the first active user, we can synchronize the second user who enters the transmission at a later time, and so on, until all the users are synchronized. In this way, we can get rid of the overhead induced by long training sequence which is required by the subspace-based and other delay estimators, and in the meantime, keep other users' ongoing transmission.

In addition to the acquisition, this PIC-SC algorithm is also proposed to be used for tracking¹ in the next section.

VI. NUMERICAL RESULTS

The simulations for joint delay estimation and data detection are based on the assumption of initial acquisition (which was treated in [8] and in the previous section of this paper) so that the receiver has the knowledge of the delay of each user and utilizes this information to demodulate transmitted data. However, due to the mobility of the mobile terminals, the delay of each user changes, the receiver is therefore out of synchronization over time. Experiments are conducted based on the two scenarios for tracking the changes of delay followed by the subsequent data demodulation process.

In the first experiment, we use the idea described in scenario 1. The tracking is done with the subspace-based algorithm, which needs a long periodic training sequence. After the training period is over, a long scrambling code is applied to each user for the whole packet data transmission at the transmitter. The receiver then starts iterative data detection process using the scrambling code of each user and the newly estimated delays.

In the second experiment, we use the scheme described in scenario 2. Suppose the receiver can detect the movement of the mobile user, e.g., by noticing the received power change from a particular user. In parallel with power control for this user, the receiver can perform PIC to cancel the interference from other users and use the SC to track down the change in the signal's propagation delay. For this algorithm to work, the receiver must know the delays of all the other users, at least with an accuracy of $0.1T_c$ in order to combat the MAI using PIC. This could be a reasonable assumption because the time needed to synchronize all the users using this algorithm is rapid enough not to let any user out of synchronization beyond $0.1T_c$ if tracking is performed to the mobile users in a fast round robin fashion. The major advantages of this algorithm are two-fold. First, the other users can continue their data transmission at the same time when the delay tracking is being carried out to one user. Secondly, much shorter training sequence is need for tracking (only 15 pilot symbols are used in the simulations, compared to 300 pilot symbols used for the subspace-based algorithm), thus the overhead imposed on the system is greatly reduced.

¹Tracking usually refers to the fine tuning of the code timing after the receiver finds the propagation delay of the received signal with the accuracy of half chip period, i.e., makes the code timing within the tracking range (acquisition), as defined in the classic literature on CDMA synchronization. Here we extend the concept so that it also means tracking down the changes in the delay of the mobile user (could be out of the tracking range) after the initial synchronization.

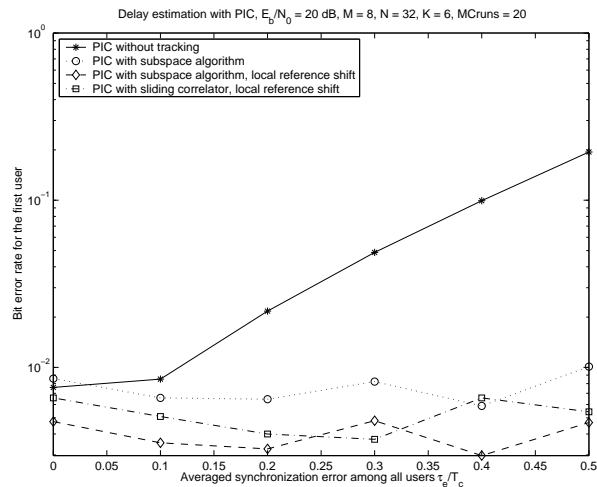


Fig. 4. Performance of joint delay estimation and data detection (PIC case)

The simulated system is a 6-user system with $N = 32$ chips per symbol and $M = 8$ signal alternatives. The signal to noise ratio is set to $10\log_{10} E_b/N_0 = 20dB$, the perfect power control is assumed. The channels are independent Rayleigh fading channels. The channel gain $h_k(t)$ is complex circular Gaussian process with auto-correlation function $E[h_k^*(t)h_k(t+\tau)] = J_0(2\pi f_D\tau)$ where f_D is the maximum Doppler frequency and $J_0(x)$ is the zeroth order Bessel function of the first kind. Simulation results are averaged over 20 Monte-Carlo runs with each MCrun representing a particular realization of the data sequences (2^{10} transmitted symbols), noise, and fading processes, as well as propagation delays.

The solid line in Fig. 4 shows how the receiver would react to the synchronization error (averaged error among all the users and assumed to have variance $0.01T_c^2$) when no attempt is made to track down the changes in propagation delays. Taking the first user as the user of interest, the bit error rate for demodulating the signal transmitted by the first user increases drastically after the synchronization error goes beyond $0.1T_c$.

The performance of the PIC-subspace algorithm (the first experiment) is illustrated with the dotted line and the dashed line. The dotted line with circle shows that the receiver works reliably and stably regardless of how much the system is out of synchronization when the multiuser detectors work in conjunction with subspace-based delay estimator. As indicated by the dashed line with diamond, the performance can be further improved by shifting the local reference according to the estimated synchronization errors.

The performance of the PIC-SC with local reference shift (the second experiment) lies somewhere in between as depicted by the dash-dot line. The PIC-subspace algorithm performs slightly better than the PIC-SC due to the reason that the former one achieves more accurate estimate of delays. However, the difference is minor because the synchronization error under $0.1T_c$ won't cause noticeable degradation in data demodulation, which is indicated by

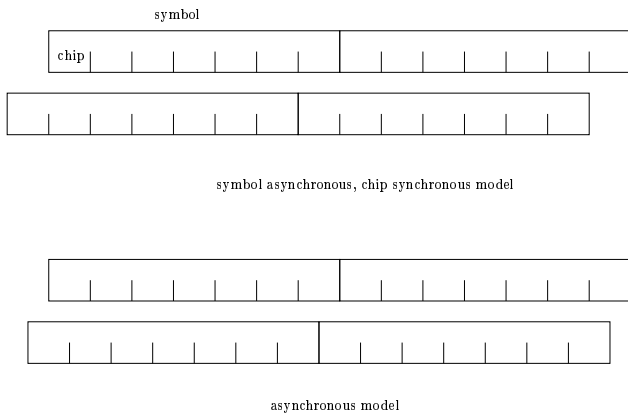


Fig. 5. Transmission models.

the solid line.

The BER obtained by both schemes falls well below 10^{-2} , the target physical layer error rate and approximates the results achievable by the system expressed by (4) when p_k is known and $\delta_k = 0$, i.e., a synchronized and chip-aligned asynchronous DS-CDMA system which is the case thoroughly studied in [2] and [3]. If we replace PIC with IIS, the result would be similar to the one shown in Fig. 4. The IIS plot is therefore omitted to conserve space.

VII. CONCLUSIONS

Two joint synchronization and data detection algorithms, namely, the PIC-subspace and PIC-SC are investigated in this paper for the orthogonal modulated signals in the asynchronous DS-CDMA system with long spreading codes. Proper synchronization is prerequisite to achieve low bit error rate in data detection. To this end, the receiver needs to have good knowledge about the delay of each user. Accurate delay estimates can be obtained either by the subspace-based algorithm or by the SC collaborated with interference cancellation. When coupled with data demodulation, the PIC-subspace algorithm exhibits a slight performance gain compared to the PIC-SC algorithm, at the cost of larger overhead and disruption of the other users' transmission. Therefore, the latter one could be a more feasible choice for the system in question, although the limitation is that it requires synchronization of the other users. If all the users are out of synchronization (e.g., sudden outage of the receiver) at the same time, the former scheme would be a more suitable solution. Otherwise, the PIC-SC algorithm is always preferred.

In addition to the proposal of the synchronization algorithms which are suitable for the long-code systems, a side product of this work is that we have proved the performance of asynchronous systems (see the bottom plot of Figure 5) is close to the performance attainable in the symbol asynchronous, chip aligned (semi-synchronous) systems (see the upper plot of Figure 5). This justifies the previous research on 'asynchronous CDMA systems' which are actually semi-synchronous in nature.

REFERENCES

- [1] S. Vembu, A. Viterbi. Two different philosophies in CDMA - a comparison. *Proc. VTC'1996*, vol. 2, pages 869-873, Jan. 1996.
- [2] E. Ström, S. Miller. Iterative demodulation of orthogonal signalling formats for DS-CDMA. *Proc. ICC'2000*, vol. 3, pages 1457-1461, 2000.
- [3] P. Xiao, E. Ström. Performance of iterative DS-CDMA M-ary demodulation in the presence of synchronization errors. *Proc. VTC'2001 spring*, pages 1703-1707, May, 2001.
- [4] S. Bensley, B. Aazhang. Subspace-based estimation of multipath channel parameters for CDMA communication systems. *Proc. GLOBECOM'1994*, pages 154-157, 1994.
- [5] E. Ström, S. Parkvall, S. Miller, B. Ottersten. Propagation delay estimation in asynchronous direct-sequence code-division multiple access systems. *IEEE Trans. Commun.*, vol. 44, pages 84-93, Jan. 1996.
- [6] U. Madhow. MMSE interference suppression for timing acquisition and demodulation in direct-sequence CDMA systems. *IEEE Trans. Commun.*, vol. 46, pages 1065-1075, August 1998.
- [7] P. Xiao, E. Ström. Synchronization algorithms for iteratively demodulated M-ary DS-CDMA systems. *Proc. GLOBECOM'2001*, pages 1371-1375, Nov. 2001.
- [8] P. Xiao, E. Ström. Pilot-aided acquisition algorithms for asynchronous DS-CDMA systems. *Selected for European Trans. Telecommun.*, vol. 14, first issue. 2003.