

# BLIND TIME DELAY ESTIMATION IN ASYNCHRONOUS CDMA VIA SUBSPACE INTERSECTION AND ESPRIT

Nicolas Petrochilos and Alle-Jan van der Veen

Department of Electrical Engineering, Delft University of Technology  
Mekelweg 4, 2628 CD Delft, The Netherlands

**Abstract**— The implementation of a RAKE receiver for a quasi-synchronous DS-CDMA (direct sequence code division multiple access) system requires the estimation of the dominant path delays for each user. Presented here is a blind method to estimate these parameters. The algorithm takes advantage of various signal space invariances in the frequency domain to isolate the subspace of interest for each user, then uses ESPRIT on these subspaces to estimate the delays. The method processes each user independently of the others, is near-far resistant, and allows several delays per user. Simulations indicate a fair accuracy.

## 1. INTRODUCTION

Third generation wireless communications networks such as UMTS will be based on wideband Direct Sequence Code Division Multiple Access (DS-CDMA) schemes. A specific problem of such systems in the uplink is that the transmitted user signals are received over multipath fading channels with long delay spreads, while the power ratio between any two users can be dramatically large (the near-far problem). This necessitates advanced equalization and detection structures at the base-station receiver. Practical receivers are based on RAKE structures: a bank of matched filters which correlate the received signal with the desired user code at several delays, corresponding to the dominant path delays of the user channel. In this case, but also in more advanced multi-user receivers, accurate estimation of these delays is essential. A second consideration is that in wideband CDMA, the amount of training chips needed for channel estimation is proportional to both the maximal delay spread and the number of users, and can be quite substantial. It is therefore interesting to look at algorithms that estimate path delays without using training chips, i.e. blindly. In this paper we propose such an algorithm.

In the literature, recent works that specifically address the blind delay estimation problem are [1, 2, 3]. In these papers, only a single delay per user is assumed (this situation applies to asynchronous narrowband CDMA). In [1], the Cramer-Rao Bound (CRB) on the estimation variance is derived for the multi-user case with one path delay per user. The CRB indicates that the error variance of the delay estimate of one user is independent from the parameters of the other users, which suggests that near-far resistant algorithms are possible. While a Maximum-Likelihood estimator has been derived, it has a high complexity because all delays are estimated jointly. For this reason it is desirable to have separate estimators for each user, as in the aforementioned articles. Our contribution differs from the precedent in two aspects: we consider more than one delay per user and we propose an algebraic, noniterative solution acting on a block (frame) of data. The algorithmic principle is similar to SI-JADE in [4].

## 2. DATA MODEL FORMULATION

We consider a  $K$ -user CDMA system with periodic codes of length  $L_c$ . In a frame, user  $k$  emits  $N$  symbols  $s_k(n)$  taken from some communication alphabet. His spreading code is denoted by a sequence  $[c_k(m)]_{m=1}^{L_c}$ , and his pulse shape is  $p_k(t)$ . Let the sampling period be  $T_s$ , normalized to  $T_s = 1$ , so that the chip period is  $T_c = 1/L_c$ . The code waveform by which the symbols are modulated thus becomes

$$g_k(t) = \sum_{m=1}^{L_c} c_k(m) p_k(t - mT_c),$$

which spans slightly more than a symbol period, and the emitted signal of user  $k$  is

$$y_k(t) = \sum_{n=1}^N s_k(n) g_k(t - nT_s).$$

We consider a quasi-synchronous system, in which differences in synchronization are small and can be incorporated as delays in the channel model.

We consider in this contribution a discrete multipath channel disturbed by additive white complex Gaussian noise. The maximal path delay is assumed to be less than a symbol period, or such that the convolution of the channel with the user code waveform spans at most two symbol periods. Each user  $k$  has  $L_k$  paths, where the  $\ell$ -th path is parametrized by a delay  $\tau_k^\ell$  and a complex gain  $\alpha_k^\ell$ . The received signal is thus

$$\begin{aligned} x(t) &= \sum_{k=1}^K \sum_{\ell=1}^{L_k} \alpha_k^\ell y_k(t - \tau_k^\ell) + n(t) \\ &= \sum_{k=1}^K \sum_{n=1}^N s_k(n) h_k(t - nT_s) + n(t), \end{aligned}$$

where the channel seen by the  $k$ -th user symbols is

$$h_k(t) = \sum_{\ell=1}^{L_k} \alpha_k^\ell g_k(t - \tau_k^\ell).$$

The assumption on the maximal delay entails that  $h_k(t)$  has support only on the interval  $t \in [0, 2T_s]$ .

The received signal is subsequently sampled at a rate of  $P$  times the chip rate (typically  $P = 2$ ), and the samples are stacked

in  $PL_c$ -dimensional vectors spanning a symbol period,

$$\mathbf{x}(i) = \begin{bmatrix} x(i) \\ x(i + \frac{1}{PL_c}) \\ \vdots \\ x(i + 1 - \frac{1}{PL_c}) \end{bmatrix}, \quad i = 1, \dots, N.$$

Using the channel model and the implication that at most two symbols per user play a role in  $\mathbf{x}(i)$ , we can derive that it has a model of the form

$$\mathbf{x}(i) = \sum_{k=1}^K [\mathbf{h}_k^{(1)} \ \mathbf{h}_k^{(2)}] \begin{bmatrix} s_k(i) \\ s_k(i-1) \end{bmatrix} + \mathbf{n}(i),$$

where  $\mathbf{n}(i)$  is the noise vector defined similarly as  $\mathbf{x}(i)$ , and

$$\mathbf{h}_k = \begin{bmatrix} \mathbf{h}_k^{(1)} \\ \mathbf{h}_k^{(2)} \end{bmatrix} = \begin{bmatrix} h_k(0) \\ h_k(\frac{1}{PL_c}) \\ \vdots \\ h_k(\frac{2PL_c-1}{PL_c}) \end{bmatrix} = \sum_{\ell=1}^{L_k} \alpha_k^\ell \begin{bmatrix} g_k(-\tau_k^\ell) \\ g_k(\frac{1}{PL_c} - \tau_k^\ell) \\ \vdots \\ g_k(\frac{2PL_c-1}{PL_c} - \tau_k^\ell) \end{bmatrix} \quad (1)$$

We collect  $N$  samples of the data vector into a data matrix

$$\mathbf{X} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{x}(1) & \mathbf{x}(2) & \dots & \mathbf{x}(N-1) \\ \mathbf{x}(2) & \mathbf{x}(3) & \dots & \mathbf{x}(N) \end{bmatrix}.$$

$\mathbf{X}$  is a  $2PL_c \times (N-1)$  matrix with model

$$\mathbf{X} = \sum_{k=1}^K \mathbf{H}_k \mathbf{S}_k + \mathbf{N} = \mathbf{H} \mathbf{S} + \mathbf{N}, \quad (2)$$

where  $\mathbf{H} = [\mathbf{H}_1 \dots \mathbf{H}_K]$  is a  $2PL_c \times 3K$  matrix, with

$$\mathbf{H}_k \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{0} & \mathbf{h}_k^{(1)} & \mathbf{h}_k^{(2)} \\ \mathbf{h}_k^{(1)} & \mathbf{h}_k^{(2)} & \mathbf{0} \end{bmatrix} \quad (3)$$

and

$$\mathbf{S}_k = \begin{bmatrix} s_k(2) & s_k(3) & \dots & s_k(N) \\ s_k(1) & s_k(2) & \dots & s_k(N-1) \\ s_k(0) & s_k(1) & \dots & s_k(N-2) \end{bmatrix}.$$

We will require that the factorization  $\mathbf{H} \mathbf{S}$  is a low-rank factorization. For this it is necessary that  $2PL_c \geq 3K$ , and  $3K \leq N-1$ . We also need to assume that the noise bandwidth is larger than the oversampling rate, so that the noise covariance matrix is diagonal with equal diagonal elements.

### 3. DELAY ESTIMATION

Knowing  $\mathbf{X}$  and the  $g_k(t)$ 's, our goal is to estimate the time delays for each user blindly, i.e. without knowledge of the symbols  $\mathbf{S}$ . We first present the basic principle of the algorithm, and then derive the details. The idea behind the algorithm is similar to that of the SI-JADE algorithm [4].

#### 3.1. Principle

The algorithm is based on the property that a convolution in time is equivalent to a multiplication in frequency. For finite sequences and the Discrete Fourier Transform (DFT), this property is strictly speaking not valid, but approximately so if we perform sufficient oversampling (this model mismatch causes the estimator to be slightly biased). Denote by  $\mathcal{F}$  the Discrete Fourier Transform matrix, and apply  $\mathcal{F}$  to the column  $\mathbf{h}_k$  with model as in (1),

$$\begin{aligned} [\mathcal{F} \mathbf{h}_k](f) &\approx G_k(f) \cdot \sum_{\ell=1}^{L_k} \alpha_k^\ell \exp\left(j\pi f \tau_k^\ell\right) \\ \text{with } f &\in \left\{-\frac{1}{2}, \dots, 0, \frac{1}{2PL_c}, \dots, \frac{1}{2} - \frac{1}{2PL_c}\right\}. \end{aligned} \quad (4)$$

Hence, the spectrum is equal to the frequency code pulse  $G_k(f)$  times a sum of complex exponentials. Denote  $\theta_{k,\ell} = \exp\left(\frac{j\pi \tau_k^\ell}{2PL_c}\right)$ ,  $\boldsymbol{\alpha}_k = [\alpha_k^1, \dots, \alpha_k^{L_k}]^T$ , construct  $\boldsymbol{\Theta}_k^{(+)}$  as

$$\boldsymbol{\Theta}_k^{(+)} = \begin{bmatrix} 1 & \dots & 1 \\ \theta_{k,1} & \dots & \theta_{k,L_k} \\ \theta_{k,1}^2 & \dots & \theta_{k,L_k}^2 \\ \vdots & \ddots & \vdots \\ \theta_{k,1}^{2PL_c-1} & \dots & \theta_{k,L_k}^{2PL_c-1} \end{bmatrix},$$

and let  $\mathbf{G}_k$  be a diagonal matrix whose diagonal entries are  $G_k(f)$ . Then we can write (4) as

$$\mathcal{F} \mathbf{h}_k = \mathcal{F} \begin{bmatrix} \mathbf{h}_k^{(1)} \\ \mathbf{h}_k^{(2)} \end{bmatrix} \approx \mathbf{G}_k \boldsymbol{\Theta}_k^{(+)} \boldsymbol{\alpha}_k.$$

Similarly, let  $\mathbf{J}$  be a diagonal matrix with entries  $(\mathbf{J})_{i,i} = -1^{1+i}$  and construct the matrix  $\boldsymbol{\Theta}_k^{(-)} = \mathbf{J} \boldsymbol{\Theta}_k^{(+)}$ , then

$$\mathcal{F} \begin{bmatrix} \mathbf{h}_k^{(2)} \\ \mathbf{h}_k^{(1)} \end{bmatrix} \approx \mathbf{G}_k \boldsymbol{\Theta}_k^{(-)} \boldsymbol{\alpha}_k.$$

Let us now consider  $\mathbf{H}_k$  in (3). Note that

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{h}_k^{(1)} & \mathbf{h}_k^{(2)} & -\mathbf{h}_k^{(2)} \\ \mathbf{h}_k^{(2)} & \mathbf{h}_k^{(1)} & \mathbf{h}_k^{(1)} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 1/2 & 0 & -1/2 \end{bmatrix}.$$

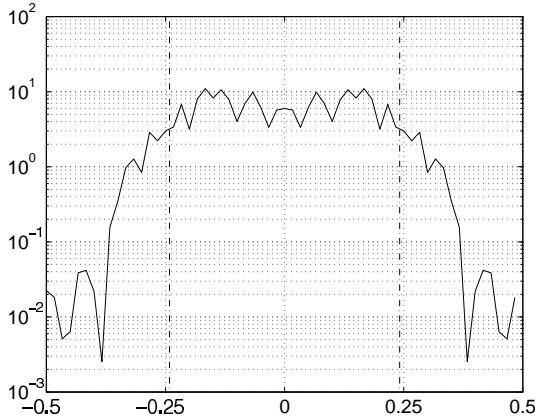
Denoting the latter matrix by  $\mathbf{O}$ , we obtain that

$$\mathcal{F} \mathbf{H}_k = \left[ \begin{array}{c|c} \mathbf{G}_k \boldsymbol{\Theta}_k \begin{bmatrix} \boldsymbol{\alpha}_k & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\alpha}_k \end{bmatrix} & \mathcal{F} \begin{bmatrix} -\mathbf{h}_k^{(2)} \\ \mathbf{h}_k^{(1)} \end{bmatrix} \end{array} \right] \mathbf{O}, \quad (5)$$

where

$$\boldsymbol{\Theta}_k = \begin{bmatrix} \boldsymbol{\Theta}_k^{(+)} & \boldsymbol{\Theta}_k^{(-)} \end{bmatrix}.$$

Observe that  $\boldsymbol{\Theta}_k$  satisfies a shift-invariance property: all subsets of  $n$  consecutive rows have the same column span. Our approach will thus be to estimate the column span of  $\mathcal{F} \mathbf{H}_k$ , and look for vectors that have this invariance structure, discarding all other vectors that do not have this property. The parameters of interest can then be obtained by an application of the ESPRIT algorithm.



$G_k(f)$ , and the interval selected by the truncating matrix.

### 3.2. Noiseless case

We first describe the algorithm in the noisefree case. Assume that  $\mathbf{X}$  has a low-rank factorization, and consider the SVD of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{H}\mathbf{S} = \mathbf{U}_x \mathbf{\Sigma}_x \mathbf{V}_x,$$

where  $\mathbf{U}_x : 2PL_c \times 3K$  is an isometric matrix,  $\mathbf{\Sigma}_x : 3K \times 3K$  is a diagonal matrix containing the singular values, and  $\mathbf{V}_x : 3K \times (N-1)$  is a co-isometric matrix. Assuming that  $\mathbf{S}$  has full rank, we know that  $\text{span}\{\mathbf{U}_x\} = \text{span}\{\mathbf{X}\} = \text{span}\{\mathbf{H}\}$ , so that there is an invertible matrix that maps  $\mathbf{U}_x$  to  $\mathbf{H}$ . We now apply  $\mathcal{F}$  to  $\mathbf{U}_x$ :

$$\mathcal{F}\mathbf{U}_x = [\mathbf{G}_1\Theta_1 \mid \dots \mid \mathbf{G}_K\Theta_K \mid *] \mathbf{P},$$

where  $\mathbf{P}$  is a tall  $(2\sum_{k=1}^K L_k + K) \times 3K$  matrix of full column rank, and  $*$  represents the last columns of equation (5) for each  $k$ . Note that the column span of  $\Theta_k$  is invariant under left multiplication by  $\mathbf{J}$ , and the same holds for  $\mathbf{G}_k\Theta_k$ , since  $\mathbf{G}_k$  is diagonal. On the other hand, the last columns denoted as  $*$  do not have this invariance property. Thus, we can remove these columns by intersecting the subspaces spanned by the columns of  $\mathcal{F}\mathbf{U}_x$  and  $\mathbf{J}\mathcal{F}\mathbf{U}_x$ :

$$\begin{aligned} \mathbf{U} &\stackrel{\text{def}}{=} \text{span}\{\mathcal{F}\mathbf{U}_x\} \cap \text{span}\{\mathbf{J} \cdot \mathcal{F}\mathbf{U}_x\} \\ &= [\mathbf{G}_1\Theta_1 \mid \dots \mid \mathbf{G}_K\Theta_K] \mathbf{P}', \end{aligned}$$

where  $\mathbf{P}'$  is a tall  $2\sum_{k=1}^K L_k \times 2K$  matrix of full column rank.

The next objective is to retrieve the column span corresponding to the user of interest. We would like to premultiply  $\mathbf{U}$  by the inverse of  $\mathbf{G}_k$ , so that we obtain columns of the desired user that are shift-invariant (the columns of  $\Theta_k$ ), whereas all other columns are not. A complication is that  $\mathbf{G}_k$  contains small entries (see figure 1), due to the oversampling of the bandlimited pulse shape  $g_k(t)$ , and we do not wish to invert these entries. However, we can define a rectangular selection matrix  $\mathbf{J}_c$  that selects a consecutive subset of rows of  $\mathbf{G}_k$  containing the large entries. Typically,  $\mathbf{J}_c$  has size  $2L_c \times 2PL_c$ . (We assume that the user codes are such that they do not introduce spectral nulls in the selected interval.)

Suppose that user 1 is the user of interest. Define  $\mathbf{D}_1 = (\mathbf{J}_c \mathbf{G}_1 \mathbf{J}_c^H)^{-1} \mathbf{J}_c$ . We premultiply  $\mathbf{U}$  by  $\mathbf{D}_1$ :

$$\mathbf{D}_1 \mathbf{U} = [\mathbf{J}_c \Theta_1 \mid \mathbf{D}_1 \mathbf{G}_2 \cdot \Theta_2 \mid \dots \mid \mathbf{D}_1 \mathbf{G}_K \cdot \Theta_K] \mathbf{P}'.$$

User	Delay 1	Delay 2
1	3.6285	10.3155
2	1.5345	3.1875
3	1.4895	5.8155
4	5.1285	11.8155
5	4.6380	7.5015

**Table 1.** Delays used for each user (in chips).

Our goal will be to extract from  $\mathbf{D}_1 \mathbf{U}$  a matrix with the same column span as  $\mathbf{J}_c \Theta_1$ . Since this component has the desired shift-invariance structure, we can apply ESPRIT to estimate the delays. There are however two complications: (i) the columns of  $\mathbf{J}_c \Theta_1$  are not individually present in the column span of  $\mathbf{D}_1 \mathbf{U}$ , but only a linear combination of them (due to the compression by  $\mathbf{G}_1$ ); (ii) the column span of  $\mathbf{D}_1 \mathbf{U}$  also contains contributions from all other users, although they are not shift-invariant.

We take care of the first problem by smoothing  $\mathbf{U}$  in the usual way, by collecting  $L_1$  shifts in a larger matrix. Thus let  $\mathbf{J}_i$  be a selection matrix that keeps rows  $i$  to  $2L_c - L_1 + i$  of a matrix, and construct

$$\mathbf{U}_e = [\mathbf{J}_1 \mathbf{D}_1 \mathbf{U}, \dots, \mathbf{J}_{L_1} \mathbf{D}_1 \mathbf{U}].$$

Let  $\Theta'_1$  consist of the first  $2L_c - L_1 + 1$  rows of  $\Theta_1$ . Due to the shift-invariance of  $\Theta_1$ , we know that

$$\text{span}\{\mathbf{U}_e\} \supset \text{span}\{(\mathbf{J}_1 \mathbf{J}_c) \Theta_1\} = \text{span}\{\Theta'_1\}.$$

To filter out the other columns in the span of  $\mathbf{U}_e$ , we again invoke the shift-invariance of the desired subspace. By intersecting the column spans of two or more shifts of  $\mathbf{U}_e$ , only the shift-invariant subspace will remain. Thus compute

$$\mathbf{U}_f = \text{span}\{\mathbf{U}_e^{(1)}\} \cap \text{span}\{\mathbf{U}_e^{(2)}\} \cap \dots \cap \text{span}\{\mathbf{U}_e^{(n)}\}, \quad (6)$$

where the notation  $^{(k)}$  indicates the  $k$ -th shift on  $\mathbf{U}_e$ . Note that the smoothing step also introduced shifts of the undesired vectors corresponding to the other users. However, these are not shift invariant over more than  $L_1$  shifts, hence if  $n > L_1$ , we know that

$$\mathbf{U}_f = \Theta''_1 \mathbf{P}'',$$

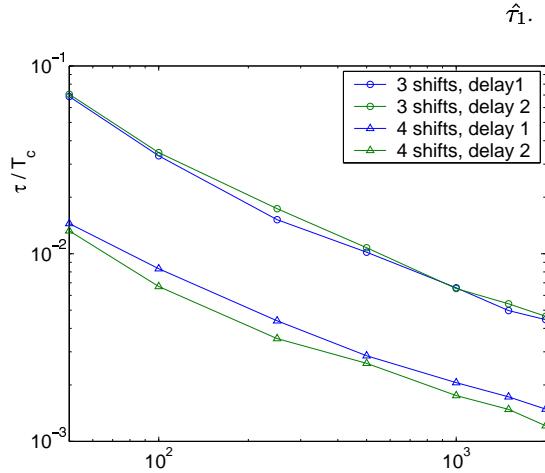
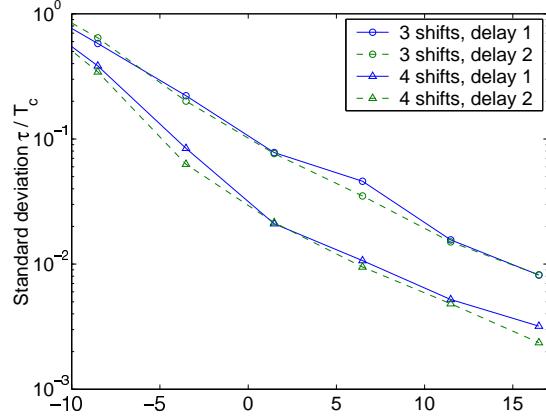
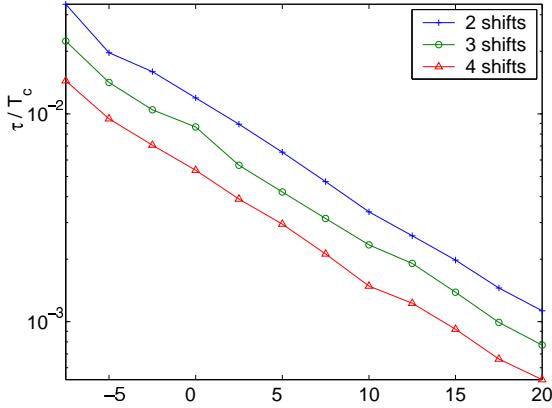
where  $\mathbf{P}''$  is an invertible matrix and  $\Theta''_1$  consists of the first  $2L_c - L_1 - n + 2$  rows of  $\Theta_1$ . The final step is to apply ESPRIT [5] to  $\mathbf{U}_f$  in order to get  $\{\pm\theta_{1,\ell}, \ell \in [1, \dots, L_1]\}$ , and to estimate the delays  $\tau_1^\ell$  from these.

### 4. SIMULATIONS

We test the algorithm on simulated data involving  $K = 5$  users with random codes of length  $L_c = 15$ . Each user has either 1 or 2 propagation paths, with delays as specified in table 1 (in simulations where only one delay is used, it is the first one in the table). We used for the pulse shape a raised-cosine pulse with a roll-off factor 0.52 truncated at a total length of  $6T_c$ . The signal was oversampled  $P = 2$  times. The results are averaged over 1000 independent runs, with the worst 5% removed to suppress the outliers.

The received power  $\sigma_k^2$  of user  $k$  is defined as  $\|\mathbf{h}_k\|^2$ . In the simulations, SNR denotes the ratio between the power of user 1 and the noise variance  $\sigma^2 = \frac{1}{P} E\|\mathbf{n}(i)\|^2$ , i.e.  $\sigma_1^2/\sigma^2$ . The Near-Far Ratio (NFR) is defined as the ratio  $\sigma_2^2/\sigma_1^2$ , and we have given all other users have the same power as user 2.

In the first simulation, we consider 5 users, each with one delay, and a frame of  $N = 200$  symbols. The matrices  $\mathbf{U}_e$  are then



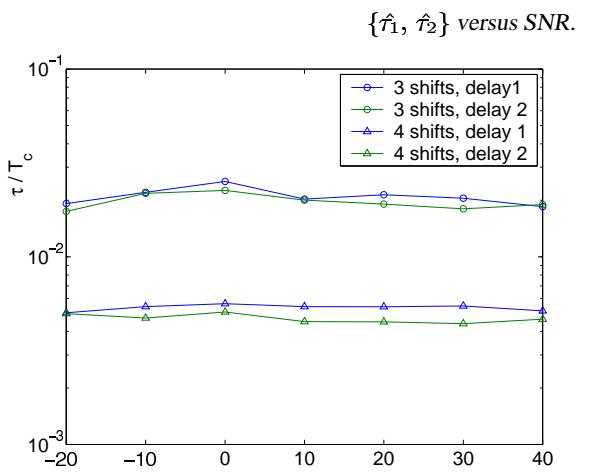
$\{\hat{\tau}_1, \hat{\tau}_2\}$  versus the number of samples.

$30 \times 10$ , and with  $L_1 = 1$ , we can use  $n = 2, 3, 4$  shifts in (6) for the estimation of  $\mathbf{U}_f$ . Figure 2 shows the standard deviation of  $\hat{\tau}_1$  in chips for the first user as a function of SNR. (The bias of the estimates was smaller than their standard deviation.) All curves have the expected log-linear behavior. Apparently, taking more shifts in (6) improves the delay estimate.

In a second simulation, we consider two paths per user, which results in a matrix  $\mathbf{U}_e$  of dimension  $30 \times 20$ . Since  $L_1 = 2$  we can only take  $n = 3, 4$  shifts. Figure 4 presents the standard deviation for the estimator as a function of the SNR, for  $N = 200$  and a near-far ratio of 0 dB. Both delays are equally well estimated, and the version with 4 shifts has better performance than the 3 shift version. Figure 3 shows the performance as function of  $N$ , with the SNR fixed at 15 dB and an NFR of 0 dB, with similar conclusions. Finally, figure 5 presents the standard deviation as a function of the NFR, while the SNR is 10 dB and  $N = 500$ . It is seen that the proposed algorithm is near-far resistant.

## 5. CONCLUDING REMARKS

A path delay estimator for DS-CDMA has been proposed, based on subspace intersection and ESPRIT. Simulations indicate that the proposed algorithm is Near-Far resistant and offers good performance.



$\{\hat{\tau}_1, \hat{\tau}_2\}$  versus the Near Far Ratio.

## 6. REFERENCES

- [1] E.G. Strom, S. Parkvall, S.L. Miller, and B. Ottersten, "Propagation delay estimation in asynchronous direct-sequence code-division multiple access systems," *IEEE Transactions on Communications*, vol. 44, no. 1, pp. 84–93, Jan. 1996.
- [2] A. Jakobsson, A. Lee Swindlehurst, D. Aszely, and C. Tidestav, "A blind frequency domain method for DS-CDMA synchronization using antenna arrays," in *Proc. of the 32nd Asilomar Conf.*, Vail, Colorado, Oct 1998, vol. 2, pp. 1848–1852.
- [3] T. Ostman, S. Parkvall, and B. Ottersten, "An improved MUSIC algorithm for estimation of time delays in asynchronous DS-CDMA systems," *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1628–1631, Nov. 1999.
- [4] A.J. van der Veen, M.C. Vanderveen, and A. Paulraj, "Joint angle and delay estimation using shift-invariance techniques," *IEEE Tr. Signal Processing*, vol. 46, no. 2, pp. 405–418, Feb. 1998.
- [5] R. Roy and T. Kailath, "ESPRIT estimation of signal parameters via rotational invariance techniques," *IEEE Trans. on acoustics, speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, July 1989.