

# Synchronization Algorithms for Iterative Demodulated M-ary DS-CDMA Systems

Pei Xiao, Erik Ström

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

**Abstract** — In this paper, we developed several algorithms to combat the impact of synchronization errors on demodulating M-ary orthogonal signaling formats in asynchronous DS-CDMA systems. The system under study resembles the uplink of an IS-95 system. The channel is assumed to be a time-varying flat Rayleigh-fading channel. Investigation shows that synchronization errors severely deteriorate the performance of multi-user detectors. We proposed an adaptive algorithm to estimate the errors in synchronization. Based on this information, remedial actions are taken to alleviate the performance degradation caused by sampling the received signals at the incorrect timing. Simulation results show considerable capacity gains when the proposed algorithms are performed to erroneously sampled signals.

## I. Introduction

The considered system in this paper is a DS-CDMA system with orthogonal modulation. The system resembles the uplink (reverse link) of an IS-95 system in that the narrowband bit stream is spread by one of  $M$  possible Walsh codes, which are not used for separating users from each other, but for M-level modulation. Then the modulated data is scrambled with a long PN-code. Different scrambling codes are used to separate users.

Demodulation of M-ary orthogonal signalling formats in DS-CDMA system has been the subject of study in various literatures. 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 [4], [5] and references therein. They were shown to achieve significant performance gains compared to the conventional receiver. Time-varying Rayleigh fading channel is assumed in those papers, which entails the necessary channel estimation for effective interference cancellation and suppression.

However, errors in delay estimates would drastically degrade the system performance and near-far robustness, which has been shown in [2] and [3], for the system with BPSK and M-ary modulation respectively. This paper aims at robustifying the receiver algorithms to combat synchronization errors. To this end, three algorithms, namely, perfect interpolation, original received vector estimate, local reference adjustment are proposed and their performance compared with computer simulations.

In section II, the system model is presented. Data detection and channel estimation are briefly introduced in Section III. Section IV describes the synchronization algorithms. Different algorithms are compared in Section V based on the numerical results and conclusions are drawn in Section VI.

## II. System Model

The passband received signal due to the  $k$ th user is denoted by  $r_{\text{RF},k}(t)$  and is formed as shown in the block diagram in Fig. 1. 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  $\log_2 M$  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  $\omega_c$  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  $\tau_k$  and additive white Gaussian noise with power spectral density  $N_0/2$ .

The total received signal is the sum of the  $K$  users' signals plus additive white complex Gaussian noise  $n(t)$ . 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 signal vector,  $\mathbf{r}(j) \in \mathbb{C}^N$ , due to transmission of the  $j$ th symbol can be formed as

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

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  $\mathbf{E}[\mathbf{n}(i)\mathbf{n}^T(j)] = \mathbf{0}$  and  $\mathbf{E}[\mathbf{n}(i)\mathbf{n}^*(j)] = N_0\mathbf{I}_N\delta(i-j)$ . The vector  $\mathbf{h}(j) \in \mathbb{C}^K$  is defined by the complex channel gains as  $\mathbf{h}(j) = [h_1(jT) \ h_2(jT) \ \dots \ h_K(jT)]^T$ . The matrix  $\mathbf{A}(j) \in \mathbb{R}^{N \times K}$  is defined as

$$\begin{aligned} \mathbf{A}(j) &= [\mathbf{a}_1(j) \ \mathbf{a}_2(j) \ \dots \ \mathbf{a}_K(j)] \\ \mathbf{a}_k(j) &= \mathbf{C}_k(j)\mathbf{w}_{i_k(j)} \end{aligned} \quad (2)$$

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  $\mathbf{w}_m$  is the  $m$ th column of the  $N \times N$  Hadamard matrix.

Equation (1) is defined under the assumption of perfect synchronization. In case there is an error in the delay estimation at the receiver end, i.e.,  $\hat{\tau}_k = \tau_k + \tau_e$ , where  $\tau_e \in [0, T_c]$ ,  $T_c$  is chip interval (acquisition is assumed and the residual error is in the range  $[0, T_c]$ , and equal for each user), the received vector

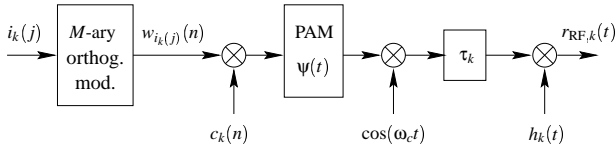


Figure 1. Block diagram of signal path for the  $k$ th user

should be reformed as:

$$\tilde{\mathbf{r}}(j) = (\tau_e/T_c) [\text{us}(\mathbf{r}(j), 1) + \text{ds}(\mathbf{r}(j+1), N-1)] \\ + (1 - \tau_e/T_c) \mathbf{r}(j)$$

where  $\text{us}(\cdot)$ ,  $\text{ds}(\cdot)$  stand for the up-shift and down-shift operators respectively:  $\text{us}([a_1 \cdots a_N]^T, q) = [a_{q+1} \cdots a_N \ 0 \cdots 0]^T$ ,  $\text{ds}([a_1 \cdots a_N]^T, q) = [0 \cdots 0 \ a_1 \cdots a_{N-q}]^T$ .

For notation simplicity, all the above equations are derived for synchronous model in which  $\tau_k = 0$  for  $k = 1, 2, \dots, K$ . In asynchronous case, the vector  $\mathbf{a}_k(j)$  in (2) should be shifted accordingly based on the delay  $\tau_k$  of each user.

### III. Data Detection and Channel Estimation

The task of the receiver is to detect the symbols from all users given the received signal vector  $\mathbf{r}(j)$ , i.e., detect  $i_k(j)$  for  $k = 1, 2, \dots, K$ .

The decision on the  $k$ th user's symbol at the  $l$ th iteration stage is denoted by  $\hat{i}_k^{(l)}$  (multistage detection is applied here) and is found as

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

In the following, we shall briefly introduce how the soft decision  $z_k(m)$  is obtained in different receiver algorithms.

#### A Conventional matched filter (Conv.)

The conventional detection technique is to form the soft decision by correlating the received signal with the  $M$  possible transmitted waveforms  $\mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \dots, \mathbf{a}_{k,m}$ . The soft decision can be formulated as  $z_k(m) = |\mathbf{w}_m^* \mathbf{C}_k^* \mathbf{r}(j)| = |\mathbf{a}_{k,m}^* \mathbf{r}(j)|$  where we defined  $\mathbf{a}_{k,m} = \mathbf{C}_k \mathbf{w}_m$ . This simple scheme is particularly useful in the beginning of the detection process, e.g., at the first iteration stage, when the estimates of the fading channel are lacking, we must therefore carry out the detection in a non-coherent manner.

#### B Parallel interference cancellation (PIC)

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

$$z_k^{(l)}(m) = \text{Re}\{\hat{h}_k^* \mathbf{a}_{k,m}^* [\mathbf{r}(j) - \hat{\mathbf{A}}(j) \hat{\mathbf{h}}(j) + \hat{h}_k \hat{\mathbf{a}}_k(j)]\} \quad (3)$$

where  $\hat{\mathbf{A}}(j)$  and  $\hat{\mathbf{a}}_k(j)$  are obtained by substituting  $\hat{i}_k^{(l-1)}$  into (2), and  $\hat{\mathbf{h}}(j)$  is the estimate of the fading vector  $\mathbf{h}(j)$ .

#### 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  $\mathbf{A}$  and form the matrix  $\mathbf{U}$ :

$$\mathbf{U} = [\mathbf{a}_1(j) \ \mathbf{a}_2(j) \ \cdots \ \mathbf{a}_{k-1}(j) \ \mathbf{a}_{k+1}(j) \ \cdots \ \mathbf{a}_K(j)]^T$$

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

$$z_k^{(l)}(m) = \text{Re}\{\hat{h}_k^* \mathbf{a}_{k,m}^* \mathbf{P}_U^\perp \mathbf{r}(j)\} - 0.5 \|\mathbf{P}_U^\perp \mathbf{a}_{k,m} \hat{h}_k\|^2 \quad (4)$$

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

#### D Channel estimation

Multuser detectors require channel estimation to provide substantial capacity gains. Coherent detection always performs better than is noncoherent counterpart provided that the complex channel gains are accurately estimated [3].

Two data-aided algorithms were presented in [4]. The channel estimation in the  $l$ th iteration stage uses the data estimates  $\hat{\mathbf{A}}(j)$  from the previous stage, i.e.,  $\hat{i}_k^{(l-1)}$ . Given the estimate of the transmitted data  $\hat{\mathbf{A}}(j)$ , the channel can be estimated as

$$\hat{\mathbf{h}}(j) = \hat{\mathbf{A}}^\dagger(j) \mathbf{r}(j) \quad (5)$$

The matrix  $\hat{\mathbf{A}}^\dagger(j)$  denotes the left pseudoinverse of  $\hat{\mathbf{A}}(j)$ .

Another alternative is the linear minimum mean square estimate of  $\mathbf{h}$ :

$$\hat{\mathbf{h}}(j) = \hat{\mathbf{p}}^* \hat{\mathbf{R}}^{-1} \mathbf{r}(j) \quad (6)$$

where  $\mathbf{R} = \text{E}[\mathbf{r}(j)\mathbf{r}^*(j)]$  is the correlation matrix for  $\mathbf{r}$ , and  $\mathbf{p} = \text{E}[\mathbf{r}(j)\mathbf{h}^*(j)]$  is the crosscorrelation matrix between  $\mathbf{r}(j)$  and  $\mathbf{h}(j)$ . The matrices  $\mathbf{R}$  and  $\mathbf{p}$  are uniquely determined by the matrix  $\mathbf{A}(j)$  and the signal to noise ratio which is assumed to be known, so they can be estimated based on the data estimate  $\hat{\mathbf{A}}(j)$ .

### IV. Synchronization Algorithms

We denote  $\varepsilon = \tau_e/T_c, \varepsilon \in [0, 1)$  as the normalized synchronization error (perfect acquisition is implied), which is assumed to be equal for all users. Let  $\tilde{\mathbf{r}}$  and  $\mathbf{r}$  stand for the received vector of the entire packet in presence of synchronization error and in absence of synchronization error respectively, then<sup>1</sup>

$$\tilde{\mathbf{r}} = (1 - \varepsilon)\mathbf{r} + \varepsilon \text{us}(\mathbf{r}, 1)$$

The  $i$ th chip of the  $j$ th symbol in the received signal can be denoted as:

$$\tilde{r}(jN + i) = (1 - \varepsilon)r(jN + i) + \varepsilon r(jN + i + 1)$$

<sup>1</sup>This equation holds for noiseless case. If the noise  $\tilde{\mathbf{n}}$  is taken into account in the received signal vector  $\tilde{\mathbf{r}}$ , then  $\tilde{\mathbf{n}}$  has the same statistics as the noise vector  $\mathbf{n}$  in equation (1).

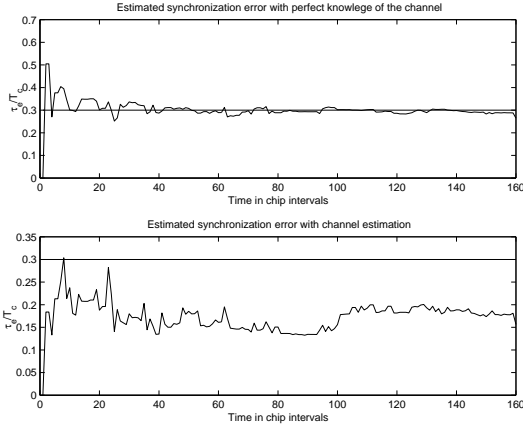


Figure 2. Adaptive algorithm for synchronization error estimate

We define the optimum estimate of  $\epsilon$  as

$$\hat{\epsilon}_{opt} = \arg \min_{\epsilon} \mathbf{E} [\|\tilde{\mathbf{r}} - (1 - \epsilon)\hat{\mathbf{r}} - \epsilon \text{us}(\hat{\mathbf{r}}, 1)\|^2]$$

where  $\hat{\mathbf{r}}$  is defined in (8).

A method that would find an approximation to  $\hat{\epsilon}_{opt}$  with relatively low complexity is the following stochastic gradient descent algorithm:

$$\hat{\epsilon}(m) = \hat{\epsilon}(m-1) - \mu [\tilde{r}(m) - \hat{r}(m) + \hat{\epsilon}(m-1)(\hat{r}(m) - \hat{r}(m+1))] \times (\hat{r}(m) - \hat{r}(m+1)) \quad (7)$$

where  $\mu$  is the adaptation step size (here  $\mu$  is set to  $1/m$ ,  $m = jN + i$ , the principle is to progressively shrink it as the algorithm converges). The original received signal  $\mathbf{r}$  in the absence of synchronization error is unknown, it can be estimated as:

$$\hat{\mathbf{h}}^{tr} = \mathbf{p}^* \mathbf{R}^{-1} \tilde{\mathbf{r}}, \quad \hat{\mathbf{r}}^{tr} = \mathbf{A}^{tr} \hat{\mathbf{h}}^{tr} \quad (8)$$

The matrices  $\mathbf{A}^{tr}$ ,  $\hat{\mathbf{h}}^{tr}$  and the vector  $\hat{\mathbf{r}}^{tr}$  represent the pilot symbols, the channel estimate and the original received vector estimate corresponding to pilots respectively.

Fig. 2 shows the experiment results of this adaptive algorithm using 5 pilot symbols, i.e., 160 chips (the spreading factor  $N = 32$ ). First, we tested the genie-aided case, assuming the fading channel is known to the receiver. The algorithm converges nicely to the real value of synchronization error 0.3 after 40 chips (less than 2 symbols) adaptation. On the other hand, if we use the channel coefficients estimated by equation (8), the results deviate from the real value regardless of how many symbols we use for the training.

Apparently, the accuracy of the synchronization error estimation depends on the accuracy of the channel estimation. We learned from multistage channel estimation and data detection algorithms ([4], [5]) that the channel is better estimated when the transmitted data are more accurately detected, the performance is improved by repeating the process in an iterative manner. This idea leads to the following algorithms proposed below.

### A Initial iteration stage

The initial stage is common for all the proposed algorithms. Given the received vector  $\tilde{\mathbf{r}}$  and the training sequence  $\mathbf{A}^{tr}$ , we can estimate the channel coefficients  $\hat{\mathbf{h}}^{tr}$  corresponding to pilots. Then the received signal  $\hat{\mathbf{r}}^{tr}$  in the absence of synchronization error can be estimated using equation (8). The adaptive algorithm (7) is performed to estimate the synchronization error  $\hat{\epsilon}$ . The receiver detects the data  $\hat{\mathbf{A}}$  after adjusting the sampling timing based on the estimated synchronization error.

Different schemes differ in the subsequent iterations, and are described below.

### B Scheme 1: Perfect interpolation

Assume the receiver can perform perfect interpolation and fully recover the baseband signal, the receiver re-samples the recovered signal every time when a new estimate of synchronization error is obtained at each iteration. The algorithm proceeds as follows:

Step 1: Using the synchronization error estimated at the previous iteration  $\hat{\epsilon}^{(l-1)}$ , the receiver re-samples the baseband signal. The received vector would appear differently to the receiver and is denoted by  $\tilde{\mathbf{r}}^{(l)}$ .

Step 2: With  $\tilde{\mathbf{r}}^{(l)}$  and the data  $\hat{\mathbf{A}}^{(l-1)}$  detected earlier, we estimate the channel  $\hat{\mathbf{h}}^{(l)}$  using the equation (5) or (6).

Step 3: Estimate the received vector  $\hat{\mathbf{r}}^{tr(l)}$  corresponding to the training sequence  $\mathbf{A}^{tr}$  using the equation (8).

Step 4: Perform the adaptive algorithm (7) to estimate the synchronization error  $\hat{\epsilon}^{(l)}$ .

Step 5: Perform multiuser detection algorithms, e.g., PIC or IIS, to detect the transmitted data  $\hat{\mathbf{A}}^{(l)}$  using  $\hat{\mathbf{h}}^{(l)}$ ,  $\tilde{\mathbf{r}}^{(l)}$ ,  $\hat{\mathbf{A}}^{(l-1)}$ .

Step 6: Repeat step 1 through 5 using the newly estimated synchronization error  $\hat{\epsilon}^{(l)}$  and data  $\hat{\mathbf{A}}^{(l)}$ .

### C Scheme 2: Original received vector estimate

Let us temporarily ignore the noise contribution, which allows us to write  $\mathbf{r} = \mathbf{A}\mathbf{h}$  and  $\tilde{\mathbf{r}} = (1 - \epsilon)\mathbf{r} + \epsilon \text{us}(\mathbf{r}, 1)$ . Evidently, all the useful information is contained in the vector  $\mathbf{A}\mathbf{h}$ . It is the original received vector free from synchronization error and noise corruption. Data detection, channel estimation and synchronization can be accomplished if we have accurate knowledge about  $\mathbf{A}\mathbf{h}$ . An estimate of  $\mathbf{A}\mathbf{h}$  can be formulated as

$$\hat{\mathbf{A}\mathbf{h}}^{(l)} = \left[ \tilde{\mathbf{r}} - \hat{\epsilon}^{(l)} \text{us}(\hat{\mathbf{A}\mathbf{h}}^{(l-1)}, 1) \right] / (1 - \hat{\epsilon}^{(l)}) \quad (9)$$

We initialize  $\hat{\mathbf{A}\mathbf{h}}^{(1)} = \tilde{\mathbf{r}}$ , where  $\tilde{\mathbf{r}}$  now is the noisy received vector and enter into the following iteration loop:

Step 1: Estimate the channel  $\hat{\mathbf{h}}^{(l)}$  using  $\hat{\mathbf{A}}^{(l-1)}$  and  $\hat{\mathbf{A}\mathbf{h}}^{(l-1)}$  using the equation (5). The vector  $\mathbf{r}$  in (5) is replaced by  $\hat{\mathbf{A}\mathbf{h}}^{(l-1)}$ .

Step 2: Using the channel estimate  $\hat{\mathbf{h}}^{(l)}$  to calculate the received vector  $\hat{\mathbf{r}}^{tr(l)}$  corresponding to the training sequence  $\mathbf{A}^{tr}$ .

Step 3: Perform adaptive algorithm to estimate the synchronization error  $\hat{\epsilon}^{(l)}$ .

Step 4: Compute  $\hat{\mathbf{A}\mathbf{h}}^{(l)}$  with  $\hat{\epsilon}^{(l)}$  using the equation (9).

Step 5: Replace the received vector  $\tilde{\mathbf{r}}$  with  $\hat{\mathbf{A}}\hat{\mathbf{h}}^{(l)}$  and use it together with  $\hat{\mathbf{A}}^{(l-1)}$ ,  $\hat{\mathbf{h}}^{(l)}$  to detect the data  $\hat{\mathbf{A}}^{(l)}$ . Multiuser detection algorithms like the PIC or IIS can be used here.

Step 6: Repeat step 1 through 5 using the new estimated  $\hat{\mathbf{h}}^{(l)}$  and  $\hat{\mathbf{A}}^{(l)}$ .

#### D Scheme 3: Local reference adjustment

This algorithm is identical to algorithm 2 except the step 5 is modified as follows:

Step 5.1: Shift the local reference  $\mathbf{a}_{k,m}$  accordingly based on the synchronization error estimate  $\hat{\epsilon}^{(l)}$ :

$$\bar{\mathbf{a}}_{k,m} = (1 - \hat{\epsilon}^{(l)})\mathbf{a}_{k,m} + \hat{\epsilon}^{(l)}\text{us}(\mathbf{a}_{k,m}, 1)$$

Step 5.2: Replace  $\mathbf{a}_{k,m}$  with  $\bar{\mathbf{a}}_{k,m}$  in equations (3) and (4), then detect data  $\hat{\mathbf{A}}^{(l)}$  with  $\tilde{\mathbf{r}}$ ,  $\hat{\mathbf{h}}^{(l)}$ ,  $\hat{\mathbf{A}}^{(l-1)}$ , using the PIC or IIS scheme.

### V. Numerical Results

In the simulations, each user transmits one of  $M = 8$  Walsh codes spread to a total length of  $N = 32$  chips. The scrambling codes  $c_k(n)$  are random. The signal to noise ratio is set to  $10\log_{10}E_b/N_0 = 25\text{dB}$ . With the common initial stage,  $L = 10$  iterations are performed on all the algorithms. A total of 20 Monte-Carlo runs were done for each simulation with 1024 symbols transmitted for each Monte-Carlo run. Also perfect power control is assumed in the sense that the average power is equal for all users.

Table 1, 2, 3, 4 show the estimated synchronization errors obtained by adaptive algorithm (7) at the last iteration of each algorithm. Fig. 3, 5, 4, and 6 present the results of different synchronization schemes as a function of the normalized synchronization error. Based on the experiments, we have made the following observations:

- Multiuser detectors without synchronization robustification like PIC and IIS behave similarly in presence of synchronization errors. The logarithm of the bit error rate (BER) increases linearly as  $\tau_e/T_c$  increases, i.e., the performance gain achievable by multiuser detectors over conventional receiver becomes small when the system is out of synchronization. They converge to conventional receiver when the synchronization error approaches half of the chip duration. On the contrary, conventional receiver is less sensitive to synchronization error.

- The adaptive algorithm proposed in (7) achieves the accuracy of  $\pm 0.04$  code chip interval for synchronization error estimate. The channel estimation is a prerequisite for synchronization, and different schemes estimate the channel differently. However, the results of synchronization error estimate obtained by all the three schemes are quite close to each other.

- Scheme 1 and 2 achieve better performance compared to the scheme 3. They can almost fully remedy the impact of the synchronization errors on multiuser detectors no matter how big the synchronization error is.

- Scheme 3 work better with IIS rather than PIC. The higher complexity of IIS trades for better performance on combatting the synchronization errors.

Table 1: Estimated synchronization errors by different schemes in PIC 6-user case:

Real value	0	0.1	0.2	0.3	0.4	0.5
Scheme 1	-0.005	0.063	0.176	0.291	0.372	0.490
Scheme 2	0.002	0.071	0.178	0.289	0.378	0.490
Scheme 3	0.002	0.069	0.173	0.284	0.373	0.485

Table 2: Estimated synchronization errors by different schemes in IIS 6-user case:

Real value	0	0.1	0.2	0.3	0.4	0.5
Scheme 1	0.010	0.074	0.184	0.272	0.392	0.502
Scheme 2	0.018	0.078	0.187	0.278	0.392	0.497
Scheme 3	0.016	0.078	0.187	0.275	0.387	0.490

Table 3: Estimated synchronization errors by different schemes in PIC 12-user case:

Real value	0	0.1	0.2	0.3	0.4	0.5
Scheme 1	-0.041	0.083	0.184	0.303	0.377	0.510
Scheme 2	-0.002	0.084	0.187	0.302	0.387	0.507
Scheme 3	-0.005	0.083	0.184	0.300	0.380	0.500

Table 4: Estimated synchronization errors by different schemes in IIS 12-user case:

Real value	0	0.1	0.2	0.3	0.4	0.5
Scheme 1	-0.019	0.073	0.225	0.321	0.414	0.496
Scheme 2	0.002	0.072	0.210	0.310	0.402	0.501
Scheme 3	0.004	0.073	0.205	0.306	0.403	0.495

### VI. Summary and Conclusions

In this paper, several synchronization algorithms are proposed and their performance are studied and compared with computer simulations. They considerably robustify the receiver's performance in the presence of the synchronization errors. Especially, the perfect interpolation and the original received vector estimate algorithms approach the BER obtained in the case of perfect synchronization regardless of how much the system is out of synchronization. However, considering the fact

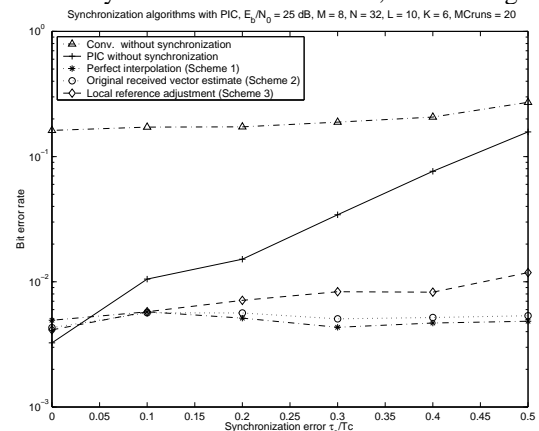


Figure 3. Comparison of different synchronization schemes (PIC 6-user case)

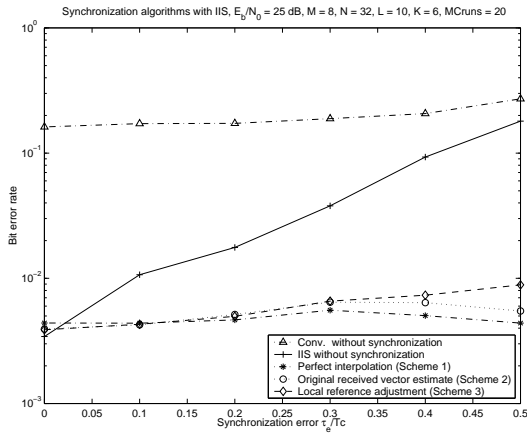


Figure 4. Comparison of different synchronization schemes (IIS 6-user case)

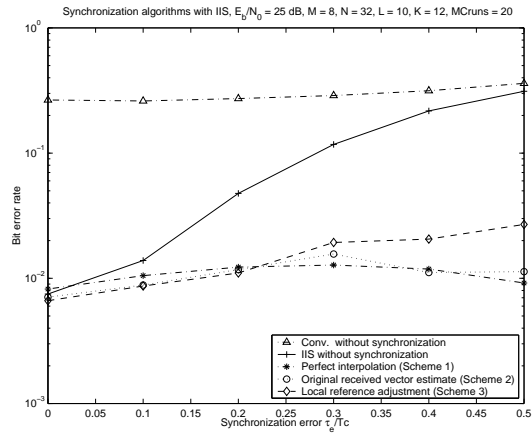


Figure 6. Comparison of different synchronization schemes (IIS 12-user case)

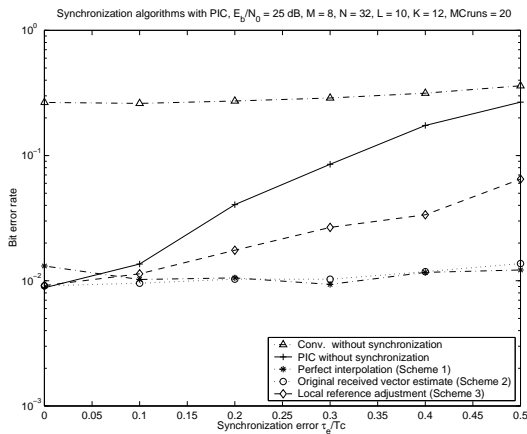


Figure 5. Comparison of different synchronization schemes (PIC 12-user case)

that perfect interpolation is impossible with rectangular chip pulses as well as its marginal gain over the second scheme, the original received vector estimate is much preferred way of combating synchronization errors. Nevertheless, compared with the rectangular chip pulses, the more spectrally efficient pulse shapes, e.g., Raised Cosine waveforms would be of more practical use for modulation. This along with other issues like random distribution of synchronization errors among different users, initial acquisition schemes, etc., are the topics for future research by the authors.

#### References

- [1] S. Parkvall, E. Ström, B. Ottersten. The impact of Timing Errors on the Performance of Linear DS-CDMA Receivers *IEEE Journal on Selected Areas in Communications*, pp. 1660–1168, vol. 14, October 1996.
- [2] P. Orten, T. Ottosson. Robustness of DS-CDMA multiuser detectors *Proc. IEEE Communication Theory Mini-Conference*, pages 144–148, Nov. 1997.

- [3] P. Xiao, E. Ström. Performance of Iterative DS-CDMA M-ary Demodulation in the Presence of Synchronization Errors. *Proc. VTC'2001 spring*, paper 310, May, 2001
- [4] E.G. Ström, S.L. Miller. Iterative demodulation of orthogonal signalling formats for DS-CDMA. *Proc. ICC2000*, pages 1457–1461, vol.3, 2000.
- [5] E.G. Ström, S.L. Miller. Iterative Demodulation of Orthogonal Signalling Formats in Asynchronous DS-CDMA Systems. *To appear in Proc. ISSSE, 2001*
- [6] S. Verdú. *Multuser Detection*. Cambridge Univ. Press, 1998.