An Iterative Method for Code Timing Acquisition for DS-CDMA Systems

Madhur Chaturvedi, Gaurav Sinha and Ajit Kumar Chaturvedi
Department of Electrical Engineering, Indian Institute of Technology, Kanpur
E-mail: akc@iitk.ac.in

ABSTRACT

A recently proposed differential correlation based acquisition technique, the Differential Correlator-Matched Filter (DC-MF) algorithm, results in greater interference suppression by using differential correlators, but is computationally complex. We propose a modification to this algorithm, which reduces the computational complexity with little loss in performance. Simulations, for a multi-user system under varying load conditions have been done.

I. INTRODUCTION

A recently proposed algorithm [1] for code timing acquisition for DS-CDMA systems uses differential correlators for estimating the propagation delay. It is assumed that the user to be synchronized is sending a preamble of all 1's during the synchronization period. It is shown in [1] that the differential correlator matrix, $\mathbf{R}(\tau)$, contains the timing information of the desired user only and the contribution of the interfering users and background noise is suppressed. A delay estimator based on $\mathbf{R}(\tau)$ using a matched filter (MF) type approach is proposed in [1]. The problem of delay estimation reduces to finding the delay that corresponds to the peak of the delay estimator. This is called the Differential Correlator-Matched Filter (DC-MF) algorithm. It has a high computational complexity.

In this paper, we reduce the computational complexity of the DC-MF algorithm by modifying it into an iterative algorithm, the Iterative DC-MF (IDC-MF). This algorithm is based on the fact that the difference between the desired and the undesired peaks of the delay spectrum is large, so that the delay spectrum can be modeled as a concave function in the vicinity of the desired peak. Using a basic property of concave functions, the peak can be iteratively located.

Simulations for multi-user Gaussian noise channel show that for light to medium loaded systems, the proposed modification achieves a large reduction in the average number of computations, with a slight degradation in performance.

II. SYSTEM MODEL

The system model, [1], is assumed to be the base band downlink of a mobile communication system.

The base station transmits the signal

$$x(t) = \sum_{m=1}^{M}\sum_{k=1}^{K} b_{km} s_k(t - mT) \qquad (1)$$

where b_{km} is the k^{th} user's m^{th} symbol, $s_k(.)$ is the k^{th} user's signature sequence and K is the total number of active users.

The received signal has the form

$$r(t) = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L} b_{km} a_{lm} s_k(t - mT) + n(t) \qquad (2)$$

where L is the number of resolvable paths, T_c is the chip duration, and $\chi_l T_c$ is the delay of the l^{th} path, a_{lm} denotes the fading process of the l^{th} path for the m^{th} symbol duration and n(t) is the background Gaussian noise. The delay of each path is assumed to remain constant in the observation interval. The user to be synchronized sends a preamble of all 1's during the synchronization period.

III. THE DC-MF ALGORITHM

First the received data is sampled by chip matched filtering, and the equispaced samples are collected into N-vectors $\boldsymbol{r}_m$ where $N=T/T_c$. Then the correlation matrix $\boldsymbol{R}(\tau)$ of the sampled data is computed for a non-zero time lag τ. In [1] it is shown that $\boldsymbol{R}(\tau)$ contains the information of the desired user only and the contribution due to the interfering users and the background noise is suppressed.

The DC-MF algorithm, [1], can be understood using the following set of equations.

$$R(\tau) = \left(\sum_{m=1}^{M} r_m r_{m+\tau}^H\right) / M \qquad (3)$$

where M = preamble length

0-7803-7569-6/02/$17.00

Define

$$g_1(d) = \begin{bmatrix} s_1(N-d+1) \\ \vdots \\ s_1(N) \\ s_1(1) \\ \vdots \\ s_1(N-d) \end{bmatrix}^T \qquad (4)$$

where $s_1(\,)$ represents the spreading code of user 1, the desired user, and d is the integral part of the delay. Let δ be the fractional part, so that the delay $\chi = d + \delta$.
Then the replica of the spreading code of user 1 for delay χ is given by

$$c_1(\chi) = (1-\delta)g_1(d) + \delta g_1(d+1) \qquad (5)$$

The delay estimate is given by

$$\chi_{estimate} = \max_\chi \left|c_1(\chi)^T R(\tau) c_1(\chi)\right| / \|c_1(\chi)\|^2 \quad (6)$$

The above expression is evaluated for a large number of test values. This set of test values will be called the delay spectrum from now on. The test delay for which this value is maximized is the estimated delay.

IV. NATURE OF THE DELAY SPECTRUM

On studying the nature of the delay spectrum for a variety of load and noise conditions, **Fig 4.1** to **Fig 4.3**, it is seen that the difference in levels of the desired and the remaining peaks is large enough in most cases so that the delay spectrum can be approximated as a concave function, in the vicinity of the peak.

Also, it is seen that the delay spectrum values are very robust with respect to SNR, i.e. changing SNR by a reasonable amount does not change the nature and values of the delay spectrum much

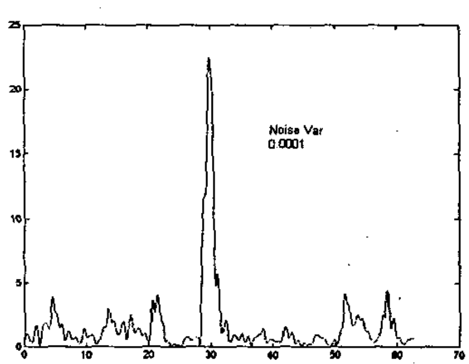

Fig 4.1: Delay Spectrum for single-user system with AWGN channel for SNR 40dB

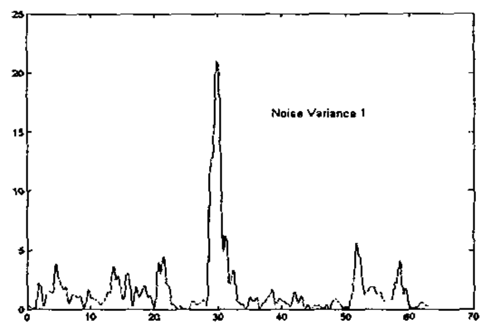

Fig 4.2: Delay Spectrum for single-user with AWGN

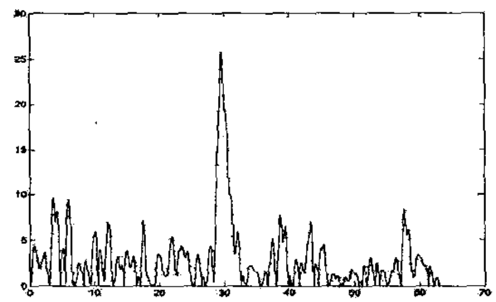

Fig 4.3: Delay Spectrum channel for SNR 0dB for multi-user system with AWGN channel for SNR 20dB, number of users 5

. Thus, the main source of randomness is the number of interfering users in the system.

V. MOTIVATING EXAMPLE

To locate the peak of any concave function in an interval in which we know that exactly one maxima exists, we can divide the interval into 3 sub-intervals and compare the function values at the 4 end-points of these sub-intervals. Referring to **Fig 5.1**, it is seen that depending on the ordering of these function values, one of the 3 sub-intervals can be rejected as not containing the peak. Also, since the function is known to be concave in the interval, the ordering of the functions values at the 4 end-points has cannot be anything other than the cases shown in **Fig 5.1**.In the next iteration, the same procedure can be applied to the reduced interval, thus coming closer to the actual peak.

VI. THE IDC-MF ALGORITHM

After studying the nature of the delay spectrum, as discussed in section 4, one can see that the method of section 5 can be applied to reduce the computational complexity, to get the IDC-MF.

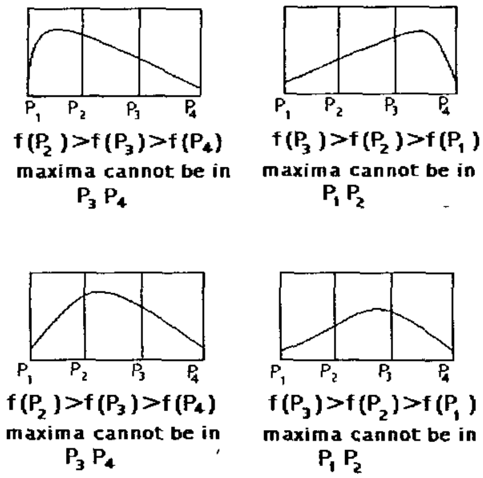

Fig 5.1 Possible cases for a function f(x) that is concave in the interval P_1P_4

Single-threshold testing algorithm

We do some offline simulations under a variety of system load conditions to get a threshold value, above which the highest peak is expected to exist.

When the acquisition procedure begins, using serial search, we get the first point on the delay spectrum such that the spectrum value at this point is greater than the threshold. For forming the window in which the peak is expected to exist, continue the serial search beyond the previously obtained point to locate a point in the vicinity such that the delay spectrum value at this point is below the threshold.

Use the iterative method as give in section 5 to locate the peak in the window within a given tolerance limit. Tolerance limit (in %) denotes the maximum permissible difference between the actual and the estimated delay.

n-threshold testing algorithm

There are n numbers of different thresholds. Apply the single-threshold testing algorithm for the first threshold. If the delay spectrum lies completely below the first threshold, apply the single-threshold testing algorithm for the second threshold and so on till either an acquisition occurs or all the n thresholds have been used one by one. This will give better performance than the individual single-threshold testing algorithms used.

VII. PERFORMANCE MEASURES

Apart from locking on to the correct delay, the IDC-MF algorithm may lock on to an incorrect value of delay, which is outside the specified tolerance limit. The algorithm may completely miss the highest peak, in case the highest peak is below the threshold. Hence performance measures discussed in [1] may not be adequate. We consider the following performance measures:

Probability of Correct Acquisition

P_a = (# of correct acquisitions)/ (# of trials)

An acquisition is said to happen when the estimated delay is within a specified tolerance limit of the actual delay.

Probability of False Acquisition

P_f = (# of false acquisitions)/ (# of trials)

A false acquisition is said to happen when the estimated delay is not within a specified tolerance limit of the actual delay.

Probability of Miss

P_m = (# of misses)/ (# of trials)

A 'miss' is said to happen when the algorithm fails to form the window based on the threshold value, i.e. the highest peak is below the threshold value.

<u>Note</u>: $P_a + P_f + P_a = 1$

Mean Acquisition Time

Mean Acquisition Time means the average time taken by the receiver to acquire the actual delay.

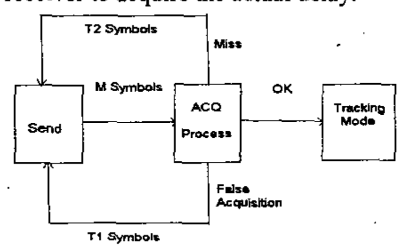

Fig 7.1: Code acquisition process

Using the block diagram given in **Figure 7.1** the expression for mean acquisition time comes out to be

$$T_{acq} = (M + T1\ P_f + T2\ P_m)/P_a$$

where M is the preamble length, T1 is the cost in symbols associated with a False Acquisition and T2 is the cost in symbols associated with a Miss.

$$RMSE = \sqrt{\sum_{i=1}^{S}(d_i - d)^2 \Big/ S}$$

Root Mean Square Error (RMSE)

This measures the accuracy of delay estimation according to

where d_i = delay estimate, d = actual delay and S = no. of realizations.

Calculating the Threshold

We considered and tested the IDC-MF algorithm for a number of intuitive thresholds under varying load conditions. The thresholds giving best performance were

1. Max (S_2) + 0.1
2. Average value of ($3S_1+S_2$)/4
3. Average value of ($2S_1+S_2$)/3

where S_1 denotes the value of the highest peak in the delay spectrum and S_2 denotes the value of the 2^{nd} highest peak.

To improve the performance, 'n' single-threshold testing algorithms can be combined to form the n-threshold testing algorithm. The results given in the following sections are for the 3-threshold testing algorithm formed by combining the above three thresholds.

VIII. SIMULATION RESULTS

Simulations were carried out for an AWGN channel under a variety of load conditions. Gold codes of length 63 were used for spreading. Two different situations were considered –one where the power of the interfering users was taken to be same as the desired user and the other where the interfering users had higher power.

Varying the number of interfering users

It can be seen from **Fig 8.1** to **Fig 8.3**, that as we vary the number of users, there is only a gradual decline in the performance measures. The root mean square error is smaller than in DC-MF. This is because once the window is located, the iterative method allows the calculation of the delay with much higher resolution. If we use a much higher resolution (which implies a very small step size) in DC-MF, its computational complexity will increase even further. In IDC-MF, a large step size in step 1 (locating the window) and a much higher resolution in step 2 (within the window) is simultaneously possible. Thus we reduce both the computations and root mean square error at the same time.

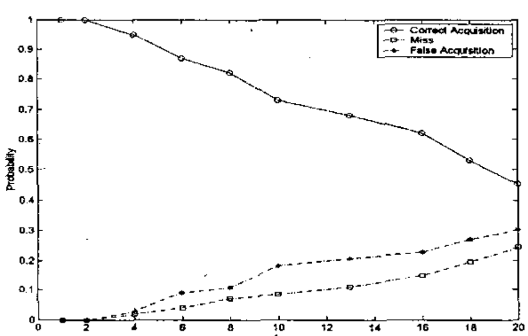

Fig 8.1:Probability of correct acquisition, false acquisition and miss for IDC-MF compound test formed by combining thresholds 1, 2 and 3, in that order, for tolerance limit 10%, SNR is 10 dB

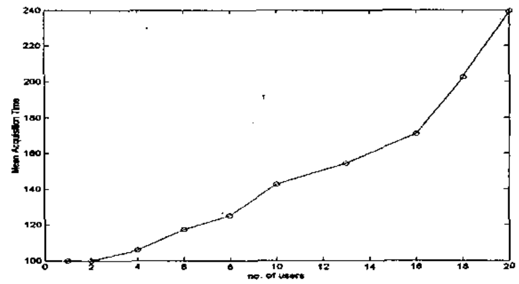

Fig 8.2: Mean acquisition time for IDC-MF compound test formed by combining thresholds 1, 2 and 3, in that order, tolerance limit 10%, SNR is 10 dB, preamble length is 100 symbols and cost in symbols associated with false acquisition and miss is 20 and 10 respectively

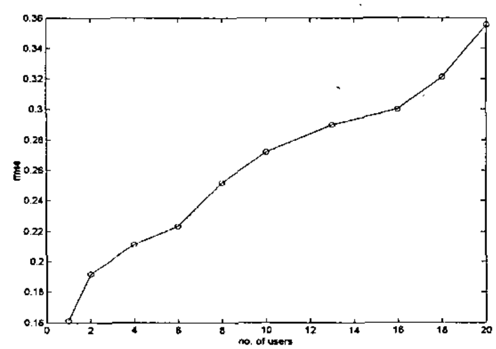

Fig 8.3: Root Mean Square Error for IDC-MF compound test formed by combining thresholds 1, 2 and 3, in that order, tolerance limit 10%, SNR is 10dB

Varying the power of interfering users

Figure 8.4 shows that the 3-threshold testing algorithm works well till the interfering users have around 18dB more power than the desired user when the number of users is 6.

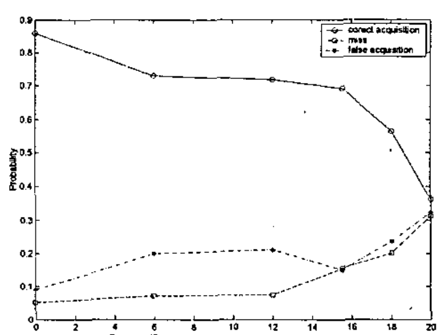

Fig 8.4: Probability of correct acquisition, false acquisition and miss for IDC-MF compound test formed by combining thresholds 1, 2 and 3, in that order, tolerance limit 10%, SNR is 10 dB and number of users is 6

IX. COMPUTATIONAL COMPLEXITY OF DC-MF AND IDC-MF ALGORITHMS

Define W = step size for calculating delay,

D = the delay, to be calculated,

DC-MF algorithm

The DC-MF algorithm can be broken up into the following two steps for the purpose of computational complexity analysis.

Step1: MN^2 multiplications and MN^2 additions are required to calculate the DC matrix.

Step 2: Each calculation of test value involves $N(1/W)(N+3)N$ multiplications and $N(1/W)(N+2)(N-1)$ additions. If N is large, we can simply write it as $N^3(1/W)$ multiplications and $N^3(1/W)$ additions.

Threshold testing IDC-MF algorithm

A combination of thresholds is used. The steps involved are:

Step 1: MN^2 multiplications and MN^2 additions are required to calculate the DC matrix.

Step 2: Denote the 2 thresholds by the subscripts 1, 2 and 3. Let X_n denote the number of calculations for the case of miss, Y_n denote the number of calculations for the case of acquisition for the n[th] threshold and P_{mn} denotes the probability of miss for the n[th] threshold testing. This gives

$(1 - P_{m,1})* Y_1 + P_{m,1}*(1 - P_{m,2})* Y_2 + P_{m,1}* P_{m,2}* (1 - P_{m,3})* Y_3 + P_{m,1}*X_1. + P_{m,1}* P_{m,2}* X_2 + P_{m,1}*P_{m,2}* P_{m,3}* X_3$

X_n involves $N*(1/W)* N^2$ multiplications and additions and X_n involves $D*(1/W)* N^2$ multiplications and additions. So in total, there are

$\{(1- P_{m1})+P_{m,1}(1 - P_{m,2}) +P_{m,1}P_{m2}(1-P_{m,3})\}D(1/W)N + \{P_{m,1} + P_{m,1}P_{m,2} + P_{m,1}P_{m,2}P_{m,3}\}(1/W)N^3$ multiplications and additions.

Step 3: For obtaining the delay using the iterative method in I iterations, there will be $4N^2I$ multiplications and $4N^2I$ additions.

Comparison of average computational complexities of DC-MF and IDC-MF

Fig 9.1 shows that the average computational complexity of IDC-MF is less than that of the DC-MF algorithm, although it is a function of the delay. Also, IDC-MF gives even better performance when the probability of miss is low, as can be seen in **Fig 9.2.** Thus the IDC-MF algorithm has much less average computationally complexity than the DC-MF algorithm for light to medium loaded system, for which the probability of miss will be low.

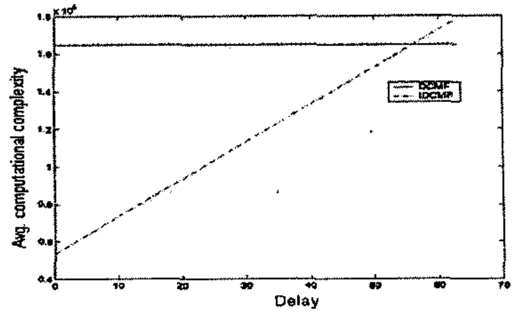

Fig 9.1: Average computational complexity of DCMF and IDCMF as the delay varies from 0 to 63, step size for DCMF and IDCMF=0.2, number of iterations in IDCMF window=4, number of users is 6

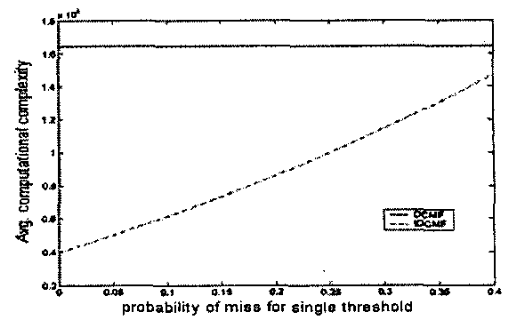

Fig 9.2 Average computational complexity of DCMF and IDCMF as the probability of miss varies, N=63, delay =32, step size for DCMF and IDCMF=0.2, number of iterations in IDCMF window =4, number of users =6

X. CONCLUSION AND COMMENTS

The DC-MF algorithm has been modified into an iterative method, the IDC-MF algorithm, using a very basic property of concave functions. This property can be used due to the generally favorable nature of the Delay Spectrum in the vicinity of the peak.

The computational complexity of this single-threshold testing IDC-MF algorithm is much less than that of the DC-MF algorithm, but at the cost of performance degradation. To improve the performance, we combined three thresholds that give the best performance to form the three-threshold testing IDC-MF algorithm. It is seen that the performance of the algorithm is good for light to medium loaded systems, with a significant decrease in the average computational complexity.

Future work would involve testing the IDC-MF algorithm for fading channels.

REFERENCES

[1] Tapani Ristaniemi and Jyrki Joutsensalo, "Code Timing Acquisition for DS-CDMA in fading channels by differential correlations", *IEEE Trans. Commun.*, vol.49, pp.899-910, May 2001.