Tracking Fading Multipath Channel Parameters, in CDMA Systems, Using a Subspace-based Method - An Implementation Perspective

Chaitali Sengupta Joseph R. Cavallaro Behnaam Aazhang

Electrical and Computer Engineering Department,

Rice University, MS 366, Houston, TX 77005-1892.

Ph: (713)-527-4020 Fax: (713)-524-5237 Email: {chaitali,cavallar,aaz}@rice.edu

ABSTRACT

In this paper, we evaluate several implementation issues in the application of subspace based methods to tracking channel parameters in Code Division Multiple Access (CDMA) communication systems, in the presence of multipath fading. We focus on the behavior of Singular Value Decomposition (SVD) based schemes while tracking the time variations in the signal subspace, due to fading. We also evaluate the application of several techniques to reduce the complexity of the computationally expensive SVD procedure, to the channel estimation problem.

1. INTRODUCTION

In a *Code Division Multiple Access (CDMA)* communication system, a communication channel with a given bandwidth is accessed by all the users simultaneously. The different mobile users are distinguished at the base station receiver by the unique *spreading code* assigned to the users to modulate their signals. Hence, the CDMA signal transmitted by any given user consists of that user's data which modulates the unique spreading code assigned to that user, which in turn modulates a carrier (the frequency of which is the same for all the users), using any well-known modulation scheme such as *binary phase shift keying (BPSK)*. The receiver receives a linear superposition of the signals transmitted by all the users, attenuated by arbitrary factors and delayed by an arbitrary amount. The goal of channel parameter estimation is to determine these unknown and time varying attenuation factors and delays by processing the received signal, to facilitate recovery of the data transmitted by each user.

Channel estimation is one of the major problems in radio communications particularly when the mobile system is subject to multipath fading, that is, the transmission channel consists of more than one distinct propagation path for each user's signal. Moreover, when the CDMA technique is used to allow multiple users access to a single channel, the system is susceptible to the *near-far* effect. The near-far problem arises when the signals from the different users arrive at the receiver with widely varying power levels. The near-far problem has been shown to severely degrade the performance of standard single user techniques (e.g., matched filters, correlators, etc.) in conventional CDMA systems.

In this paper we deal with a subspace-based algorithm

that has been developed in [1], and has been shown to be near-far resistant and effective in the presence of multiple propagation paths. This algorithm exploits the signal subspace spanned by certain observation vectors, in order to estimate the unknown channel parameters. The tool typically used to estimate the signal and noise subspaces of the received signal vectors is the Singular Value Decomposition (SVD) of an observation matrix formed from received data vectors. Our studies show that the number of data vectors in the observation matrix is an important factor that influences the ability of any subspace estimator to capture the time-variations in the signal subspace. We present preliminary results from our experiments to determine the best size of the observation matrix required to track the signal or noise subspace in the presence of multipath fading, multiple access interference, and background white noise.

The SVD is computationally expensive. Several techniques (such as, SVD updating [6, 5, 8], approximate orthogonal rotations [3]) have been proposed in the literature to reduce the complexity of the SVD when tracking a slowly changing subspace. We explore the use of such techniques in tracking the signal subspace in the presence of multipath fading. We also evaluate a new technique to reduce the SVD workload, by not using all the signal samples available.

2. CDMA CHANNEL ESTIMATION AS A SUBSPACE-BASED ALGORITHM

When the CDMA channel estimation problem are formulated as a subspace-based algorithm, observations by the receiver is modeled as follows :

$$y_t = A c_t + \eta_t, \quad (1)$$

where y_t is the observation vector, the columns of matrix A are known functions of the users' spreading code and the unknown channel parameters (τ and α), c_t contains information about the different users' data, and η_t is an additive white gaussian noise vector. The unknown parameters, τ , are the relative time delays of each path of each user, and α are the complex valued attenuation factors of each path of each user.

Although each observation vector corresponds to one *symbol (or bit)* interval, since the system is asynchronous, each vector will contain at least components from two adjacent bits. Also, each bit period corresponds to N *chips*, where N is the size of the spreading codes of the users. For details of how the continuous

time received signal is processed using a chip matched filter, to result in the above discrete model, the reader is referred to the previously mentioned paper [1]. The channel parameters, are assumed to vary slowly and consequently the system response is approximately constant over several observations. Given this linear model, it is possible through a process of averaging to estimate the unknown parameters.

Given observation vectors y_t , for all t , the first step is the estimation of the noise subspace. The subspace spanned by the columns of A is called the *signal subspace* and the orthogonal subspace is termed *noise subspace*. This basis may be obtained by computing an eigenvalue decomposition (EVD) of the correlation matrix, R , of the observation vectors:

$$R = \mathcal{E}[y_t y_t'], \quad (2)$$

where $\mathcal{E}[\cdot]$ denotes the expected value and $'$ denotes the conjugate transpose operator. In practice, the correlation matrix is estimated as a time average of $y_t y_t'$ over several observations. Since this average may be performed in a variety of ways, several formulations of the problem result and in effect result in different algorithms, which in turn may be implemented in a variety of different ways. One of these formulations is the *sliding window formulation*, where the last w successive observations are used at each time step t , to estimate the correlation matrix as:

$$\hat{R}_t = \frac{1}{w} \sum_{i=t-w+1}^t y_i y_i' = \frac{1}{w} Y_t' Y_t. \quad (3)$$

The matrix Y_t is an observation matrix, which is defined as follows for the sliding window formulation:

$$Y_t = \begin{pmatrix} y_{t-w+1}' \\ y_{t-w+2}' \\ \vdots \\ y_t' \end{pmatrix}. \quad (4)$$

In practice, the EVD of the estimated correlation matrix, $\hat{R}_t$ is computed using the SVD of the observation matrix Y_t . This obviates the need for the matrix multiply operation required to compute the correlation matrix and helps achieve higher numerical accuracy.

The SVD of Y_t is defined as:

$$Y_t = U \Sigma V', \quad (5)$$

where, U and V are unitary matrices and Σ is a diagonal matrix of singular values. The basis for the noise subspace, V_n , is determined as the last $N - r$ columns of the matrix V , where r is the rank of matrix A . In this paper, we will assume that r is known; however a variety of techniques have been proposed for estimating the model order (e.g. [9]). It may be noted that the rank of matrix A is determined by the number of users K , and the number of propagation paths for each user, L .

Once an orthogonal basis for the noise subspace, V_n , has been determined, the unknown time delays and attenuation factors are estimated by solving non-linear optimization problems which may again be posed in numerous ways. One of these formulations has been proposed

in [1], based on the well-known *Multiple Signal Classification (MUSIC)* [7] algorithm. In this formulation, the different users' time delays and attenuation factors are estimated independently of each other. The formulation essentially estimates the channel impulse response which is related to the users' time delays and attenuation factors as :

$$h_k(t) = \sum_{p=1}^L \alpha_{k,p} \delta(t - \tau_{k,p}), \quad (6)$$

where $h_k(t)$ is the channel impulse response for user k , $\alpha_{k,p}$ is the attenuation factor and $\tau_{k,p}$ is the time delay of the p th path of user k . The channel impulse response for each user is estimated as that value chosen from a set of all feasible impulse responses, (as determined by some *a priori* channel model), which minimizes the ℓ_2 -norm of the projection of the user's signal vectors into the estimated noise subspace.

The subspace based technique evaluated in this paper allows a multidimensional parameter estimation problem to be decomposed into a series of one-dimensional problems making the solution less computationally intensive. The algorithm has also been shown to be near-far resistant and no preamble or decision feedback is necessary [1]. Moreover, in this paper we show that it also performs well in time varying situations.

The major limitation of this method is that the number of users are limited by the size of the spreading code ($K \leq N/J$, where J is the number of signal components per user in the input vector). However this problem can be alleviated by sampling the received signal faster than the chip rate, although at the cost of increased complexity. Also, as K increases, the dimension of the noise subspace decreases, thus increasing the error in the parameter estimates. This problem can be handled by maximizing the ℓ_2 -norm of the projection of the user's signal vectors into the estimated signal subspace, for values of K for which the dimension of the noise subspace is too small to get a good estimate of the channel parameters.

3. TRACKING A TIME VARYING SUBSPACE

Following the ideas presented in [6], we analyze the tracking behavior of any subspace estimator in a time varying situation. Let, the matrix $V n_t^{exact}$ denote the actual matrix that defines the noise subspace at time t , and the corresponding noise subspace be called $\mathcal{R}(V n_t^{exact})$. However, at time t , any subspace tracking mechanism gives an estimate of $V n_t^{exact}$, denoted by $V n_t$, and the corresponding noise subspace is called $\mathcal{R}(V n_t)$.

The function $dist(\mathcal{R}(V_1), \mathcal{R}(V_2))$ is a measure of the distance between the two subspaces specified as its arguments. The distance between two subspaces is defined as [2] :

$$dist(\mathcal{R}(V_1), \mathcal{R}(V_2)) = \|P_1 - P_2\|_2, \quad (7)$$

where P_i , ($i = 1, 2$), is the orthogonal projection onto the subspace $\mathcal{R}(V_i)$. Here, the columns of V_i are an orthonormal basis for the subspace $\mathcal{R}(V_i)$. Hence, it follows that $P_i = V_i V_i'$.

The following quantities - tracking error (TE), and time variation (TV) are defined using the notion of distance between noise subspaces. The tracking error, TE_t , at time step t is defined as :

$$TE_t = \text{dist}(\mathcal{R}(Vn_t^{\text{exact}}), \mathcal{R}(Vn_t)). \quad (8)$$

The time variation ($TV_{[t-w] \rightarrow [t]}$) in the received vectors from time step $t-w$ to time step t , is defined as :

$$TV_{[t-w] \rightarrow [t]} = \text{dist}(\mathcal{R}(Vn_{t-w}^{\text{exact}}), \mathcal{R}(Vn_t^{\text{exact}})). \quad (9)$$

At time step $t-w$, we have a tracking error of TE_{t-w} . If there was no change in the observation matrix over the next w steps, the tracking error would get reduced by some reduction factor, RF_w , after w time steps. However, the observation matrix does change due to actual change of input data. So, we have the following equation:

$$TE_t \leq \frac{TE_{t-w}}{RF_w} + TV_{[t-w] \rightarrow [t]}. \quad (10)$$

The best window size, w , for any subspace tracking mechanism is dictated by the two terms on the right hand side of equation 10. The first term will be reduced by the use of a larger window size because in the presence of noise, RF_w will be larger for larger w . On the other hand, the second term will be reduced by the use of a smaller window size, so that the variation of the incoming data over the window is small. Thus the optimum window size is determined by a tradeoff between these two terms. This analysis will help us explain the results of our experiments in the rest of the paper.

4. DETERMINATION OF THE SIZE OF A SLIDING WINDOW

The experiments described in this section will help to throw light upon the factors influencing the size of the sliding window in the presence of multipath fading. All the experiments described below use Gold codes of length 31 as spreading codes. The values used for multiple access interference (MAI), signal-to-noise ratio (SNR) of the additive background noise, number of users (K), number of paths for each user (L), and the fading parameters are listed against the results of each experiment.

4.1. Channel estimation in the absence of any time variation

When the signal and noise subspace does not vary with time (which would happen if the delays and attenuation factors of each path of each user does not change over time), the optimum size of the sliding window required to get a 'good' estimate of the channel parameters, depends on the SNR and the number of users in the system. A smaller SNR would require a larger window size as more observation vectors would be necessary to 'average out' the noise. This observation has been illustrated in Figure 1 which shows the variation in the root mean square error (RMSE) in delay estimate of first path of the weakest user with varying window size, for two values of SNR. The dependence of the window size w only on the SNR when the subspace does not vary with time

is explained by equation 10. In this case, the second term in the right hand side of equation 10 is zero, hence only the first term affects the tracking error.

Figure 1. RMSE in delay estimate of first path of the weakest user vs. window size, $L = 2$, fixed delays and attenuation factors for each path, $MAI = 20\text{db}$.

4.2. Channel estimation in the presence of multipath fading

In the presence of fading, the attenuation factors vary with time, causing variations in the subspace. In this case, the best window size is dictated both by the SNR and the localized time variation in the subspace being estimated. If the window size is too small, the noise cannot be 'averaged out' successfully, whereas if the window size is too large, the subspace estimation process cannot respond to the variation in time as it is biased by old data points. The dependence of the window size w on the SNR as well as the time variation of the subspace is again explained by equation 10. In this case, both terms in the right hand side of equation 10, influence the choice of the best window size, but in opposite ways.

In the experiments in this section, time variation is introduced through Rayleigh fading simulated according to the Jakes model [4]. The experiments have been performed with a Doppler frequency of $1.04 \times 10^{-3}/T_b$, where T_b is the bit period. This corresponds to a carrier frequency of 900 MHz, bit rate of 40kbps, and a relative velocity between transmitter and receiver of 50km/h. In these experiments, the attenuation factor of each path of each user varies with time due to fading. However, the time delay of each path of each user is fixed. All statistics related to tracking are collected for the first path of the weakest user, over 1500 bits.

The effect of time variation in a channel parameter on the required window size is shown in Figure 2, which shows the variation in the estimate of the attenuation factor of the first path of the weakest user with window size. The error curve attains a minima at a particular window size - which depends upon the fading characteristics, number of users in the system and the SNR.

However, since the delays of each path of each user is

Figure 2. Error in attenuation factor estimate of first path of weakest user vs. window size, $K = 3$, $L = 2$, Rayleigh fading with Doppler frequency of $1.04 \times 10^{-3}/T_b$, time varying attenuation factor and fixed delay of each path of each user, $SNR = 15db$, $MAI = 20db$

fixed, the error in the delay estimate decreases monotonically with increasing window size (Figure 3). A comparison of Figure 3 and Figure 1 shows that a larger window size is required to attain the same performance, in the presence of fading, for the same SNR. Here, successful tracking is defined as $|\tau_k - \hat{\tau}_k| \leq T_c/2$, where τ_k is the actual delay, $\hat{\tau}_k$ is the estimate of the delay and T_c is the chip period. Figure 3 shows that the choice of the best window size is influenced both by SNR and the number of users K in the system. The error in the delay estimate from a sliding correlator, (used in conventional CDMA systems and computationally the least expensive), has been provided for reference.

However, for the sliding correlator, all users were assumed to have equal power (as would be the case with strict power control). This is because, the sliding correlator, unlike the subspace based estimator is not near-far resistant and requires power control. The fading parameters used were the same.

These experiments show how a parameter which remains fixed over time, as well as, a time varying parameter, effects the best window size. Also, the best window size for estimating two different parameters may be different depending on how fast each parameter is varying. Hence there is a tradeoff involved in the choice of the window size, based on how the inaccuracy in the estimation of a parameter effects the bit error rate of the receiver using these estimated parameters. If a particular parameter is more important to the receiver for achieving lower bit error rates, the choice of the window size would lean towards the best window size required for the estimation of that parameter.

Figure 3. Probability of tracking first path of weakest user vs. window size. $L = 2$, Rayleigh fading with Doppler frequency of $1.04 \times 10^{-3}/T_b$, time varying attenuation factor and fixed delay of each path of each user, $MAI = 20db$. The sliding correlator has been evaluated at $SNR = 15db$ and $K = 4$.

5. TECHNIQUES TO REDUCE THE COMPUTATIONAL COST OF THE SUBSPACE ESTIMATION PROCESS

5.1. Use of approximate SVD updating techniques

Several SVD updating techniques have been proposed in the literature which reduces the computation that needs to be performed at each time step. The SVD updating problem is defined as calculating the SVD of the matrix Y_t from the SVD of the matrix Y_{t-1} at minimum computational cost. Typically approximate SVD updating algorithms can be implemented in time which is at least an order of complexity less than a full SVD. These techniques are also extremely amenable to implementations on parallel architectures, either on application specific hardware [6] or as a software solution on an array of off-the-shelf processors [8].

These techniques suggest that typically one or less than one sweep of the orthogonalization procedure embedded within the SVD algorithms be performed at each update, instead of the $O(\log(n))$ sweeps required for full convergence of the SVD algorithms. It has been shown, via simulations, for other subspace based applications such as frequency estimation using MUSIC, that a 'good' enough estimate of the signal subspace is obtained by performing only one sweep per update, that is, one sweep per time step. In this section, we present results from our experiments to evaluate the performance of a Hestenes method based SVD updating scheme [8], while tracking channel parameters in the presence of fading. In the updating scheme, only one sweep of the orthogonalization procedure is performed per update.

Table 1 lists the RMSE in delay estimate when full convergence is achieved at each update (5 sweeps) and the error when only 1 sweep is used, along with the window size used. The table shows that, when only 1 sweep

is used at each update, we get almost as good performance as when full convergence is achieved at each update, especially at larger window sizes. This technique, reduces the computational load to a great extent at a very small cost in terms of reduced accuracy.

w	Full convergence	1 sweep	1 sweep and App. rotation
150	2.9511	3.0292	3.1437
250	1.8658	1.8692	-
350	1.7045	1.7170	1.7246
750	0.0297	0.0297	0.0297

Table 1. RMSE in delay estimate of first path of weakest user, $K = 3$, $L = 2$, Rayleigh fading with Doppler frequency of $1.04 \times 10^{-3}/T_b$, $SNR = 10db$, $MAI = 20db$, w denotes window size

5.2. Use of approximate orthogonal rotations in the SVD

In addition, to the SVD updating schemes mentioned in the previous section, several techniques have been proposed to reduce the computation required in the orthogonalization procedure embedded within the SVD algorithms. This is done by approximating the exact Jacobi rotations by less accurate but faster approximate orthogonal rotations [3]. It has been shown that in practice, these approximate rotations provide a 'good' enough estimate of the signal subspace. Table 1 shows the loss in accuracy of the delay estimate by using one of the approximations mentioned in [3]. The approximation can be described as follows.

Say, we are trying to calculate the SVD of matrix Y , using Hestenes method which applies Jacobi rotations to orthogonalize the columns of matrix Y . In Hestenes method for SVD, starting with any given data matrix Y , an orthogonal matrix V is built such that YV has orthogonal columns. To construct V , we take $Y^{(0)} = Y$ and iterate :

$$Y^{(i-1)} = Y^{(i)}Q^{(i)}, \quad (11)$$

(where i represents a *sweep* and $Q^{(i)}$ is orthogonal), until some $Y^{(i)}$ has orthogonal columns. $Q^{(i)}$ is chosen to be a product of $n(n-1)/2$ Jacobi rotations :

$$Q^{(i)} = \prod_1^{n(n-1)/2} Q_j^{(i)}, \quad (12)$$

where n is the number of columns of matrix Y . Every possible pair (r, s) , $1 \leq r < s \leq n$, is associated with one of the rotations $Q_j^{(i)}$. Let us denote the rotation associated with columns (r, s) by Q_{rs} .

The exact Jacobi rotation Q_{rs} is constructed by replacing 4 elements of an identity matrix : (q_{ij} is the $(i, j)^{th}$ element of matrix Q_{rs}):

$$\begin{bmatrix} q_{pp} & q_{pq} \\ q_{qp} & q_{qq} \end{bmatrix} = \begin{bmatrix} -s & c \\ c & s \end{bmatrix}, \quad c = \cos\phi, \quad s = \sin\phi, \quad (13)$$

$$t_{ex} = \tan\phi = \frac{\text{sign}(\tau)}{|\tau| + \sqrt{1 + \tau^2}}, \quad (14)$$

$$\tau = \frac{\|Y(:, r)\|^2 - \|Y(:, s)\|^2}{Y(:, r)^T Y(:, s)}, \quad (15)$$

$$c = \frac{1}{\sqrt{1 + t_{ex}^2}}, \quad s = t_{ex}c. \quad (16)$$

where, $Y(:, r)$ denotes the r^{th} column of matrix Y .

In the approximate rotation used in our simulations, $t = \tan\phi$ is calculated as follows :

$$t_{ap} = \tan\phi = \frac{1}{2\tau}, \quad (17)$$

instead of as equation 14. This approximate t_{ap} is less costly to compute but less accurate, especially for smaller values of $|\tau|$. However, Table 1 shows that this approximation can be used (with only 1 SVD sweep) with very little loss of accuracy, for our application.

5.3. Subspace estimate without using all available data vectors

In this section, we form the subspace estimate, without using the data vectors corresponding to every bit observed. Only a fraction of the data vectors available are used. If the channel is slowly varying with time, this should give as good an estimate as the subspace estimate using all the data vectors, while more time would be available between updates. In this way, we gain more time to perform each update.

Figure 4 shows the results in the presence of Rayleigh fading with a Doppler frequency of $1.04 \times 10^{-3}/T_b$. In this figure, the error is plotted against number of bits observed. Please recall from Section 2. that every data vector corresponds to one bit interval. The parameter *load* defines the size of the data matrix or the number of data vectors used to form the observation matrix. For example, when number of bits observed is 200 - if *load* = 1, size of the data matrix is also 200; if *load* = 1/2, size of the data matrix is 100; and when *load* = 1/5, size of the data matrix is 40. In other words, *load* = 1/2 implies that the data vector corresponding to every other bit has been used to form the data matrix. Similarly, *load* = 1/5, implies that every fifth data vector has been used to form the data matrix.

Figure 4 shows that when *load* = 1/2, the same performance is obtained as when all data vectors are used (*load* = 1), but after observing a larger number of bits. However the number of bits that need to be observed to get the same performance with *load* = 1/2, is less than twice that are required when all the data vectors are used. For example, to get a probability of tracking of 0.9, either we need to observe 65 bits and use *load* = 1 (every bit observed) or we need to observe 115 bits and use *load* = 1/2 (every other bit observed). However, when *load* = 1, we need to update every bit interval, but with *load* = 1/2, we need to update every other bit interval. Also, the size of the data matrix with *load* = 1/2 is half of that with *load* = 1. Using this technique, we can gain more time in which to perform each update, but at the cost of collecting data over more bit intervals or at the cost of some loss in performance.

Figure 4. (a) Probability of tracking first path of weakest user vs. number of bits observed, $SNR = 15\text{db}$, $K = 3$, $L = 2$, $MAI = 20\text{db}$, Doppler frequency is $1.04 \times 10^{-3}/T_b$

6. CONCLUSION AND FUTURE WORK

In this paper, we have evaluated implementation issues related to computational complexity of subspace tracking, techniques and approximations to reduce the complexity, and performance of the approximations while tracking channel parameters in CDMA systems, in the presence of multipath fading. The future emphasis will be on evaluating the same issues when the delays of each user varies (either slowly due to linear movement of the mobile from the base station, or abruptly due to a sudden change in one of the distinct propagation paths) and when the different paths of the different users fade with different Doppler frequencies. We also need to evaluate other computationally viable subspace tracking schemes (e.g. [5]) for this problem. Ultimately, this work is aimed at combining the developments in multiuser CDMA communications, computationally inexpensive subspace tracking schemes, and parallel architectures involving DSP and general purpose processors, for a successful real time implementation of a subspace based channel parameter estimation scheme.

7. ACKNOWLEDGMENTS

This work was supported in part by Nokia Corporation, by the Texas Advanced Technology Program under grant #003604-049, and by NSF under grant NCR 9506681.

REFERENCES

- [1] S. E. Bensley and B. Aazhang. Subspace-based channel estimation for code division multiple access communication systems. *IEEE Trans. Commun.*, 44(8):1009–1020, August 1996.
- [2] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 2nd edition, 1989.
- [3] J. Gotze. On the parallel implementation of Jacobi and Kogbetliantz algorithms. *SIAM J. Sci. Comput.*, 15(6):1331–1348, 1994.
- [4] W. C. Jakes. Chapter 1: Multipath interference. *Microwave mobile communications*. ed. W. C. Jakes, IEEE Press, pages 11–77, 1974, 1993.
- [5] A. Kavcic and B. Yang. A new efficient subspace tracking algorithm based on SVD. *ICASSP*, pages IV485–IV488, 1994.
- [6] M. Moonen, P. Van Dooren, and J. Vandewalle. A systolic array for SVD updating. *SIAM J. Matrix Anal. Appl.*, 14:353–371, 1993.
- [7] R. O. Schmidt. *A signal subspace approach to multiple emitter location and spectral estimation*. PhD thesis, Stanford Univ., Stanford, CA, 1981.
- [8] C. Sengupta, J. R. Cavallaro, and B. Aazhang. Solving the SVD updating problem for subspace tracking on a fixed sized linear array of processors. *ICASSP*, 1997.
- [9] M. Wax and T. Kailath. Detection of signals by information theoretic criteria. *IEEE Trans. on Acoustics, Speech and Signal processing*, ASSP-33:387–392, April 1985.