Spreading Code Optimization and Adaptation in CDMA Via Discrete Stochastic Approximation

Vikram Krishnamurthy, *Senior Member, IEEE*, Xiaodong Wang, *Senior Member, IEEE*, and George Yin, *Fellow, IEEE*

***Abstract*—The aim of this paper is to develop discrete stochastic approximation algorithms that adaptively optimize the spreading codes of users in a code-division multiple-access (CDMA) system employing linear minimum mean-square error (MMSE) receivers. The proposed algorithms are able to adapt to slowly time-varying channel conditions. One of the most important properties of the algorithms is their self-learning capability—they spend most of the computational effort at the global optimizer of the objective function. Tracking analysis of the adaptive algorithms is presented together with mean-square convergence. An adaptive-step-size algorithm is also presented for optimally adjusting the step size based on the observations. Numerical examples, illustrating the performance of the algorithms in multipath fading channels, show substantial improvement over heuristic algorithms.**

***Index Terms*—Discrete stochastic optimization, linear minimum mean-square error (MMSE) multiuser detector, spreading code optimization, tracking.**

I. Introduction

CONTINUOUS-valued stochastic approximation algorithms (e.g., adaptive filtering algorithms such as least mean squares (LMS)) have been studied in the signal processing and communications literature [4], [24]. In comparison, little has been done in stochastic optimization when the underlying parameter set takes discrete values. Yet, in several stochastic optimization problems such as the spreading code optimization considered in this paper, the underlying parameter (spreading code) takes only discrete values. This paper develops discrete stochastic approximation algorithms inspired by the recently proposed techniques in [1]–[3] appearing in the operations research literature. We also develop and analyze a tracking version of the algorithm for time-varying channels. The adaptive discrete stochastic optimization algorithms and tracking analysis presented in this paper are also of independent interest with applications in other areas.

Manuscript received October 21, 2002; revised February 1, 2004. The work of V. Krishnamurthy was supported in part by an NSERC Strategic Research Grant and by the British Columbia Advanced Systems Institute (BCASI). The work of X. Wang was supported in part by the National Science Foundation under Grant DMS-0225692 and in part by the Office of Naval Research under Grant N00014-03-1-0039. The work of G. Yin was supported in part by the National Science Foundation under Grant DMS-0304928 and in part by the Wayne State University Research Enhancement Program.

V. Krishnamurthy is with the Department of Electrical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: vikramk @ece.ubc.ca).

X. Wang is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

G. Yin is with the Department of Mathematics, Wayne State University, Detroit, MI 48202 USA (e-mail: gyin@math.wayne.edu).

Communicated by G. Caire, Associate Editor for Communications.

Digital Object Identifier 10.1109/TIT.2004.833338

For a code-division multiple-access (CDMA) system to efficiently serve heterogeneous traffic (e.g., data, video) in a dynamic environment, it is often necessary that the design should involve adaptive optimization at the receiver and transmitter. A significant amount of literature exists on adaptive multiuser detection for CDMA. These works consider optimization at the receiver and assume that the transmitter parameters (rate, power, spreading codes, error-correction codes, spreading gain) are fixed [43]. Several recent papers investigate transmission optimization in the context of rate control, power control, and beamforming, e.g., [20], [33], [40]. The aim of this paper is to consider adaptive spreading code optimization at the transmitter using discrete stochastic optimization algorithms assuming that linear minimum mean-square error (MMSE) multiuser detectors are employed at the receiver.

In [41], [42], optimal real-valued spreading sequences for synchronous CDMA over additive white Gaussian noise (AWGN) channels are analyzed. In [27], good spreading sequences are identified with conventional matched-filter receivers and equal received power for all users. In [36], the problem of code sequence design is addressed in an information-theoretic setting for which the sum of the rates of all users is maximized. Spreading code design based on the total squared correlation criterion has been addressed in [18], [19], [17]. In [35], the spreading code optimization is formulated as a sophisticated continuous optimization problem from the viewpoint of interference avoidance and a greedy algorithm is used. That work is generalized to vector channels in [31]. Note that in realistic multipath fading channels, spreading codes that have been *a priori* designed to have good correlation properties can lead to composite signature waveform sets (due to convolution of the channel with the spreading code) that have poor correlation properties. Good composite signature waveform sets can only be constructed with information about the channel. In [9], [32], [34], the problem of adapting the spreading codes for CDMA in multipath environments is addressed. However, the formulations there are still continuous optimization problems. In this paper, we formulate the spreading code optimization in fading multipath channels with linear MMSE multiuser detector as a discrete stochastic optimization problem, and develop novel discrete stochastic approximation algorithms.

The contributions and organization of the rest of this paper are summarized as follows. Section II describes the CDMA system under consideration and formulates the spreading code optimization problem as a discrete stochastic optimization problem. In Section III, two decreasing step-size discrete stochastic approximation algorithms for spreading code optimization are presented together with a brief survey of discrete stochastic optimization algorithms and justifications of using such algorithms. The algorithms are motivated by the random search algorithms proposed in [2] and [11].

We show that the spreading code optimization problem satisfies the conditions given in [2] for convergence of the algorithm to a globally optimal spreading code. Section IV presents *adaptive* discrete stochastic approximation algorithms for spreading code optimization in time-varying environments. Here we assume an ideal error-free feedback channel. Two adaptive algorithms are presented. The first comprises of the discrete stochastic optimization algorithm in tandem with a *fixed* step-size adaptive filtering algorithm to track the occupation probabilities of the spreading codes. A mean-square analysis of the algorithm is presented together with an upper bound on the error probability of the algorithm choosing the wrong spreading code. In addition, a computationally efficient asynchronous version of this algorithm is given. The second scheme consists of a discrete stochastic approximation algorithm in tandem with an *adaptive*-step-size filtering algorithm. The adaptive-step-size algorithm itself consists of two cross-coupled adaptive filtering algorithms—one for updating the occupation probabilities of the spreading codes, the other to optimize the step size. A weak convergence analysis of the algorithm is given which shows that the step size converges weakly to the optimal step size. Section V shows that the techniques developed in this paper can be extended straightforwardly to fading multipath channels and systems employing multiple antennas. Numerical examples are given in Section VI. These examples illustrate the performance of the discrete stochastic approximations algorithm in multipath fading channels. The performance of a computationally efficient modification of the adaptive algorithm is also illustrated. This modification consists of replacing the adaptive filtering algorithm with a sign error algorithm that operates in tandem with the discrete stochastic approximation algorithm. Also, the performance of the discrete stochastic approximation algorithms are compared with an algorithm proposed in [34] that solves a continuous optimization problem and then quantizes the solution to the nearest discrete solution. Section VI concludes the paper. Some mathematical proofs are given in two appendices.

II. System Description and Problem Formulation

A. Notation

Throughout the paper, we use the following notation.

- $\iota = 1, 2, \dots$ denotes discrete time (at the bit-time scale resolution).
- $n = 1, 2, \dots$ denotes frame time on a coarser (slower) time scale than ι. Frame time n denotes bit time interval $[M\iota, M\iota+1, \dots, (M+1)\iota-1]$ where the positive integer M denotes the frame size.
- $k = 1, 2 \dots, K$ denotes user number.
- $\mathcal{E} \triangleq \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_{2^P}\}$ is the collection of 2^P unit vectors, where $\boldsymbol{e}_j$ denotes the 2^P-dimensional unit vector with 1 in the jth component and zeros elsewhere.
- i, j, l are generic integer valued indices.

Consider the following basic real-valued discrete-time synchronous CDMA channel model:

$$\boldsymbol{r}[\iota] = \sum_{k=1}^{K} A_k b_k[\iota] \boldsymbol{s}_k + \boldsymbol{n}[\iota], \qquad \iota = 1, 2, \dots \tag{1}$$

where $A_k > 0$, $b_k[\iota] \in \{+1, -1\}$, and $\boldsymbol{s}_k$ are the received amplitude, data bit, and unit-energy spreading code of the kth user, respectively; and $\boldsymbol{n}[\iota] \sim \mathcal{N}(\boldsymbol{0}, \eta \boldsymbol{I}_P)$ is finite variance AWGN. (In this paper, we denote the $n \times n$ identity matrix by $\boldsymbol{I}_n$.) In a direct-sequence spread-spectrum system with spreading gain P, the spreading code of the kth user is of the form

$$\boldsymbol{s}_k = \frac{1}{\sqrt{P}} \left[s_{1,k} s_{2,k} \dots s_{P,k} \right]^T, \qquad s_{j,k} \in \{+1, -1\}, \; j = 1, \dots, P. \tag{2}$$

Denote

$$\boldsymbol{r}_1[\iota] \triangleq \boldsymbol{r}[\iota] - A_1 b_1[\iota] \boldsymbol{s}_1$$
$$\boldsymbol{C}_1 \triangleq \mathbf{E}\{\boldsymbol{r}_1[\iota] \boldsymbol{r}_1^T[\iota]\} = \sum_{k=2}^{K} A_k^2 \boldsymbol{s}_k \boldsymbol{s}_k^T + \eta \boldsymbol{I}_P. \tag{3}$$

The linear MMSE detector for a given user, say User 1, is given by [26]

$$\boldsymbol{w}_1 = \boldsymbol{C}_1^{-1} \boldsymbol{s}_1. \tag{4}$$

The signal-to-interference-plus-noise ratio (SINR) at the output of $\boldsymbol{w}_1$ is given by [26]

$$\text{SINR}_1 \triangleq \frac{\mathbf{E}\left\{\mathbf{E}\left\{\boldsymbol{w}_1^T \boldsymbol{r}[\iota] \mid b_1[\iota]\right\}\right\}^2}{\mathbf{E}\left\{\text{Var}\left\{\boldsymbol{w}_1^T \boldsymbol{r}[\iota] \mid b_1[\iota]\right\}\right\}} = A_1^2 \left(\boldsymbol{s}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{s}_1\right). \tag{5}$$

We consider the single-user spreading code optimization problem. (The multiuser case is addressed in Section III-E.) Suppose that the spreading codes for all users except that of User 1, $\boldsymbol{s}_2, \dots, \boldsymbol{s}_K$, are fixed. Assume that the transmitted bits are assembled into frames each comprising of M bits. (For example, in wideband CDMA (WCDMA), the frame length M can range from 150 to 9600 bits; and in CDMA 2000, the available frame sizes range from 192 to 768 bits.) Let $n = 1, 2 \dots$ denote frame time—where frame time n represents the time interval $M\iota, M\iota + 1, \dots, (M + 1)\iota - 1$. User 1 observes a sequence of independent and identically distributed (i.i.d.) random variables $\{X[n, \boldsymbol{s}_1]\}$, such that at each frame time n, $X[n, \boldsymbol{s}_1]$ is an unbiased estimator of the cost $\varphi(\boldsymbol{s}_1)$ when the spreading code chosen is $\boldsymbol{s}_1$. Denote the set of all 2^P possible spreading codes as

$$\mathcal{S} \triangleq \{+1, -1\}^P = \left\{\boldsymbol{s}(1), \boldsymbol{s}(2), \dots, \boldsymbol{s}(2^P)\right\}. \tag{6}$$

Our discrete stochastic optimization problem is as follows. Compute

$$\max_{\boldsymbol{s}_1 \in \mathcal{S}} \varphi(\boldsymbol{s}_1) = \max_{\boldsymbol{s}_1 \in \mathcal{S}} \mathbf{E}\left\{X[n, \boldsymbol{s}_1]\right\}. \tag{7}$$

Let $\mathcal{M}_1$ be the set of global maximizers and use $\boldsymbol{s}_1^* \in \mathcal{M}_1$ to denote any of the elements of $\mathcal{M}_1$.

Example—Maximizing SINR of User 1: If we choose $\varphi(\boldsymbol{s}_1) = \boldsymbol{s}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{s}_1$, then we maximize the output SINR for User 1. In this case, $X[n, \boldsymbol{s}_1]$ represents a noisy estimate of

User 1's SINR. There are a number of ways of obtaining such an estimate.

i) Observation-based estimate: Obtain a sample estimate $\hat{\boldsymbol{C}}_1[n]$ of $\boldsymbol{C}_1$ based on the nth frame comprising M observations, i.e.,

$$\hat{\boldsymbol{C}}_1[n] = \frac{1}{M} \sum_{j=nM}^{(n+1)M-1} \boldsymbol{r}_1[j]\boldsymbol{r}_1^T[j]. \tag{8}$$

Compute the SINR estimate as

$$X[n, \boldsymbol{s}_1] = \boldsymbol{s}_1^T \hat{\boldsymbol{C}}_1[n]^{-1} \boldsymbol{s}_1. \tag{9}$$

Note that $X[n, \boldsymbol{s}_1]$ is an asymptotically unbiased estimate of $\varphi(\boldsymbol{s}_1)$. (cf. Proposition 1.)

ii) Bit-error based estimate: Another way of obtaining $X[n, \boldsymbol{s}_1]$ is based on the measurement of bit-error rate. Since for linear MMSE receivers, the bit-error rate is well approximated by the formula $P_1 = Q(\sqrt{\mathrm{SINR}_1})$ [30]. We can measure the bit-error rate $P_1[n]$ based on a segment of M received signals at frame time n, and then let $X[n, \boldsymbol{s}_1] = \left(Q^{-1}(P_1[n])\right)^2$. In general, $\mathbf{E}\left\{X[n, \boldsymbol{s}_1]\right\}$ cannot be evaluated analytically since its distribution is difficult to compute, which motivates the need to devise recursive stochastic approximation algorithms.

III. Discrete Stochastic Approximation Algorithms for Spreading Code Optimization

There are several different classes of methods that can be used to solve the discrete stochastic optimization problem (7); see [3], [39] for a recent survey. When the feasible set $\mathcal{S}$ is small (usually 2 to 20 elements), ranking and selection methods as well as multiple comparison methods [14] can be used to locate the optimal solution. However, for large $\mathcal{S}$, the computational complexity of these methods becomes prohibitive.

Problem (7) can also be viewed as a multiarmed bandit problem that is a special type of infinite-horizon Markov decision process with an "indexable" optimal policy [16]. However, as mentioned in [3], multiarmed bandit solutions and learning automata procedures often tend to be conservative in exploring the solution space. Moreover, in the tracking case, where the optimal spreading code slowly evolves with time, a bandit formulation would require explicit knowledge of the dynamics of the change of the optimum spreading code which is virtually unknown.

Recently, a number of discrete stochastic approximation algorithms have been proposed. Several of these algorithms [1]–[3], [13], [44] including simulated annealing type procedures [12] and stochastic ruler [44] fall into the category of random search. In this section, we present two algorithms. i) The first one (see Section III-B) is an *aggressive* random search procedure that is based on the algorithms in [1], [2]. The basic idea is to generate a homogeneous Markov chain taking values in $\mathcal{S}$ which spends more time at the global optimum than at any other element of $\mathcal{S}$. ii) The second one (see Section III-D) is a *conservative* random search motivated by the recent paper [11].

In Section IV, we will show that these algorithms can be modified for tracking time-varying optima. Finally, it is worthwhile mentioning that there are other classes of simulation-based discrete stochastic optimization algorithms such as nested partition methods [37] which combine partitioning, random sampling and backtracking to create a Markov chain that converges to the global optimum, which will be examined in our future work.

A. Rationale for Discrete Stochastic Approximation

If $\boldsymbol{s}_1 \in \mathcal{S}$ were a continuous-valued variable (e.g., $\mathcal{S}$ being a compact subset of the real numbers) and $X[n, \boldsymbol{s}_1]$ were differentiable with respect to $\boldsymbol{s}_1$, (7) would be a continuous-valued stochastic optimization problem that can be solved via the following stochastic approximation (adaptive filtering) algorithm:

$$\hat{\boldsymbol{s}}_1^{(n+1)} = \hat{\boldsymbol{s}}_1^{(n)} + \mu^{(n)} \nabla_{\boldsymbol{s}_1} X[n, \boldsymbol{s}_1] \big|_{\boldsymbol{s}_1 = \hat{\boldsymbol{s}}_1^{(n)}}. \tag{10}$$

Here $\nabla_{\boldsymbol{s}_1}$ denotes the gradient with respect to $\boldsymbol{s}_1$. For decreasing step size $\mu^{(n)} = 1/n$, it can be proved that $\lim_{n\to\infty} \hat{\boldsymbol{s}}_1^{(n)} = \boldsymbol{s}_1^*$ almost surely (a.s.) under suitable conditions. If the statistics of the system slowly vary with time, then a constant step size $\mu^{(n)} = \mu$ in the above algorithm can be used to track the time-varying optimal $\boldsymbol{s}_1^*$.

However, for the optimization problem (7), $\mathcal{S}$ is a finite discrete set and the objective function in (7) is only defined on the discrete set $\mathcal{S}$. Thus, the notion of gradient cannot be used and approximation using the gradient information to identify a "good" direction is impossible. Since $\mathbf{E}\left\{X[n, \boldsymbol{s}_1]\right\}$ cannot be evaluated analytically, a brute-force method [28, Ch. 5.3] of solving the discrete stochastic optimization problem involves an exhaustive enumeration and proceeds as follows. For each possible $\boldsymbol{s}_1 \in \mathcal{S}$ and for large N, compute the empirical average

$$\hat{X}_N(\boldsymbol{s}_1) \triangleq \frac{1}{N} \sum_{n=1}^{N} X[n, \boldsymbol{s}_1]$$

via simulation. Then pick $\hat{\boldsymbol{s}}_1 = \arg\max_{\boldsymbol{s}_1 \in \mathcal{S}} \hat{X}_N(\boldsymbol{s}_1)$.

Since for any fixed $\boldsymbol{s}_1 \in \mathcal{S}$, $\{X[n, \boldsymbol{s}_1]\}$ is an i.i.d. sequence of random variables, by virtue of the strong law of large numbers, $\hat{X}_N(\boldsymbol{s}_1) \to \mathbf{E}\{X[n, \boldsymbol{s}_1]\}$ a.s. as $N \to \infty$. This and the finiteness of $\mathcal{S}$ imply that as $N \to \infty$

$$\begin{aligned} \arg\max_{\boldsymbol{s}_1 \in \mathcal{S}} \hat{X}_N(\boldsymbol{s}_1) &\to \arg\max_{\boldsymbol{s}_1 \in \mathcal{S}} \mathbf{E}\left\{X[n, \boldsymbol{s}_1]\right\} \\ &= \arg\max_{\boldsymbol{s}_1 \in \mathcal{S}} \varphi(\boldsymbol{s}_1) \ \text{ a.s.} \end{aligned} \tag{11}$$

The above brute-force simulation method can, in principle, solve the discrete stochastic optimization problem (7) for large N and the estimate is *consistent*, i.e., (11) holds. However, the method is highly inefficient since $\hat{X}_N(\boldsymbol{s}_1)$ needs to be evaluated for all 2^P spreading codes in $\mathcal{S}$.

The idea of discrete stochastic approximation is to design an algorithm that is both *consistent* and *attracted* to the maximum. That is, the algorithm should spend more time obtaining observations $X[n, \boldsymbol{s}_1]$ in areas of the spreading code space $\mathcal{S}$ near the maximizer $\boldsymbol{s}_1^*$, and less in other areas. Thus, in discrete stochastic approximation the aim is to devise an *efficient* [28, Ch. 5.3] adaptive search (sampling) plan that allows one to find the maximizer with as few samples as possible by not making unnecessary observations at nonpromising values of $\boldsymbol{s}_1$.

B. Aggressive Discrete Stochastic Approximation Algorithm

The following discrete stochastic approximation algorithm implemented at the receiver resembles an adaptive filtering (LMS) algorithm—it generates a sequence of spreading code estimates where each spreading code estimate is obtained from the old one by moving in a good direction in the sense that it converges to an optimizer of the objective function. The algorithm aggressively searches through $\mathcal{S}$ at each time instant looking for better spreading codes. As will be shown in Section III-E, the algorithm is consistent and attracted to the global maximum $\boldsymbol{s}_1^*$.

Notation: In Algorithm 1, the following notation is used: $\{\boldsymbol{s}_1^{(n)}\} \in \mathcal{S}$ is a random signature sequence generated by the algorithm that can be thought as the state of the algorithm at frame time n. It is convenient to map the state sequence $\{\boldsymbol{s}_1^{(n)}\}$ to the sequence $\{\boldsymbol{Y}[n]\} \in \mathcal{E}$ of unit vectors where $\mathcal{E}$ is defined at the beginning of Section II

$$\boldsymbol{Y}[n] \triangleq \boldsymbol{e}_j \text{ if } \boldsymbol{s}_1^{(n)} = \boldsymbol{s}(j), \qquad j = 1, \ldots, 2^P. \tag{12}$$

That is,

$$\boldsymbol{Y}[n] \triangleq \sum_{j}^{2^P} \boldsymbol{e}_j 1_{\{\boldsymbol{s}_1^{(n)} = \boldsymbol{s}(j)\}}$$

where 1_A is an indicator function.

$$\boldsymbol{\pi}[n] = \left[\pi[n,1] \cdots \pi[n, 2^P]\right]^T$$

denotes the 2^P-dimensional empirical state occupation probability measure up to time n of the state $\{\boldsymbol{s}_1^{(n)}\}$ or, equivalently, $\{\boldsymbol{Y}[n]\}$, i.e.,

$$\boldsymbol{\pi}[n] \triangleq \frac{1}{n}\left[W^{(n)}[\boldsymbol{s}(1)], \ldots, W^{(n)}[\boldsymbol{s}(2^P)]\right]^T \tag{13}$$

where $W^{(n)}[\boldsymbol{s}]$ for each $\boldsymbol{s} \in \mathcal{S}$ is a counter that measures the number of times the state sequence $\{\boldsymbol{s}_1^{(l)},\ 0 \le l \le n\}$ has visited the spreading code $\boldsymbol{s}$.

Finally, $\hat{\boldsymbol{s}}_1^{(n)}$ is the estimate generated by the algorithm at frame time n of the optimal signature sequence $\boldsymbol{s}_1^* \in \mathcal{M}_1$. It is the main output of the algorithm and is fed back from the receiver to the transmitter on a frame time scale.

Algorithm 1 (Single-user spreading code adaptation—Aggressive Search)

*Step 0—**Initialization:*** At frame time $n = 0$, select initial state $\boldsymbol{s}_1^{(0)} \in \mathcal{S}$. Set $\pi[0, \boldsymbol{s}_1^{(0)}] = 1$, $\pi[n, \boldsymbol{s}] = 0$ for all $\boldsymbol{s} \in \mathcal{S}, \boldsymbol{s} \ne \boldsymbol{s}_1^{(0)}$. Set $\hat{\boldsymbol{s}}_1^{(0)} = \boldsymbol{s}_1^{(0)}$.

*Step 1—**Sampling and evaluation:*** Given the state $\boldsymbol{s}_1^{(n)}$, compute $X[n, \boldsymbol{s}_1^{(n)}]$ based on $M/2$ observations in the nth frame (see example below). Generate a candidate state $\tilde{\boldsymbol{s}}_1^{(n)}$ from $\mathcal{S} - \{\boldsymbol{s}_1^{(n)}\}$ according to a uniformly distributed random variable. Compute $X[n, \tilde{\boldsymbol{s}}_1^{(n)}]$ using the remaining $M/2$ observation in the nth frame.

*Step 2—**Acceptance:*** If $X[n, \tilde{\boldsymbol{s}}_1^{(n)}] > X[n, \boldsymbol{s}_1^{(n)}]$, then set $\boldsymbol{s}_1^{(n+1)} = \tilde{\boldsymbol{s}}_1^{(n)}$; otherwise set $\boldsymbol{s}_1^{(n+1)} = \boldsymbol{s}_1^{(n)}$.

*Step 3—**Adaptive filter for updating state occupation probabilities:*** Update state occupation probabilities

$$\boldsymbol{\pi}[n+1] = \boldsymbol{\pi}[n] + \mu[n+1]\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]\right) \tag{14}$$

with the decreasing step size $\mu[n] = 1/n$, where $\boldsymbol{Y}[n]$ is defined in (12).

*Steo 4—**Update estimate of optimal spreading code**:* If

$$\pi[n+1, \boldsymbol{s}_1^{(n+1)}] > \pi[n+1, \hat{\boldsymbol{s}}_1^{(n)}]$$

then set $\hat{\boldsymbol{s}}_1^{(n+1)} = \boldsymbol{s}_1^{(n+1)}$; otherwise, set $\hat{\boldsymbol{s}}_1^{(n+1)} = \hat{\boldsymbol{s}}_1^{(n)}$ and send this new signature sequence to transmitter. Set $n \leftarrow n+1$ and go to Step 1.

Example—Maximizing SINR of User 1: If the objective function is $\varphi(\boldsymbol{s}_1) = \boldsymbol{s}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{s}_1$ (SINR of User 1), and SINR estimates of the form $X[n, \boldsymbol{s}_1] = \boldsymbol{s}_1^T \hat{\boldsymbol{C}}_1[n]^{-1} \boldsymbol{s}_1$ (see (9)) are used, then the sampling in Step 1 can be implemented as follows: For frame n, using the first $M/2$ observation samples

$$\boldsymbol{r}_1[\iota] = \boldsymbol{r}[\iota] - A_1 b_1[\iota] \boldsymbol{s}_1^{(n)}, \qquad i = Mn+1, \ldots, Mn + (M/2)$$

compute the sample covariance $\hat{\boldsymbol{C}}_1^0[n]$ via (8). Compute $X[n, \boldsymbol{s}_1^{(n)}]$ using (9). Using the next $M/2$ observation samples

$$\boldsymbol{r}_1[\iota] = \boldsymbol{r}[\iota] - A_1 b_1[\iota] \boldsymbol{s}_1^{(n)}, i = Mn + M/2 + 1, \ldots, M(n+1)$$

in frame n, compute the sample covariance $\hat{\boldsymbol{C}}_1^1[n]$ via (8). Sample $\tilde{\boldsymbol{s}}_1^{(n)}$ uniformly from $\mathcal{S} - \{\boldsymbol{s}_1^{(n)}\}$. Compute

$$X[n, \tilde{\boldsymbol{s}}_1^{(n)}] = \left(\tilde{\boldsymbol{s}}_1^{(n)}\right)^T \hat{\boldsymbol{C}}_1^1[n]^{-1} \tilde{\boldsymbol{s}}_1^{(n)}.$$

Obviously, from the preceding algorithm, $X[n, \boldsymbol{s}_1^{(n)}]$ and $X[n, \tilde{\boldsymbol{s}}_1^{(n)}]$ are independent samples.

Note that the state $\{\boldsymbol{s}_1^{(n)}\}$ (or equivalently, $\{\boldsymbol{Y}[n]\}$) of Algorithm 1 does not converge at all. Instead, it *aggressively* explores the spreading code space $\mathcal{S}$. The main idea behind Algorithm 1 is that the state of the algorithm $\{\boldsymbol{s}_1^{(n)}\}$ is a homogeneous Markov chain that is designed to spend more time at the global maximizer $\mathcal{M}_1$ than any other state.

The maximization in Step 4 yields

$$\hat{\boldsymbol{s}}_1^{(n)} = \arg\max_{\boldsymbol{s} \in \mathcal{S}} \pi[n, \boldsymbol{s}] = \arg\max_{\boldsymbol{s} \in \mathcal{S}} W^{(n)}[\boldsymbol{s}].$$

Hence, the estimate of the optimal spreading code $\hat{\boldsymbol{s}}_1^{(n)}$ is merely the particular state in $\mathcal{S}$ at which the Markov chain $\{\boldsymbol{s}_1^{(n)}\}$ has spent most time. We will show that $\hat{\boldsymbol{s}}_1^{(n)} \to \boldsymbol{s}_1^*$ a.s., meaning that the algorithm is both attracted to the maximum (i.e., spends more time in $\boldsymbol{s} \in \mathcal{M}_1$ compared to any other $\boldsymbol{s}$) and is consistent.

C. Implementation Aspects and Variations of Algorithm 1

Algorithm 1 is depicted in Fig. 1. Steps 1 to 4 are computed locally at the receiver. The output of Step 4, namely, the estimate of the optimal signature sequence $\hat{\boldsymbol{s}}_1^{(n)}$ (or more efficiently its index) at frame time n, is fed back to the transmitter at a much slower time scale than the bit time scale.

One of the key advantages of using the SINR cost function above is that the SINR estimate $X[n, \boldsymbol{s}]$ of any candidate spreading code $\boldsymbol{s}$ can be evaluated directly at the receiver, without requiring the receiver to send this candidate $\boldsymbol{s}$ to the

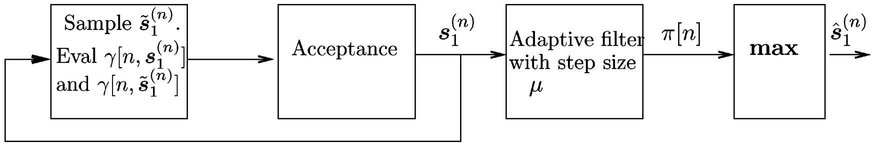


Fig. 1. Schematic diagram of Algorithm 1 implementation at the receiver.

transmitter. Thus, it is not necessary to send the state $\boldsymbol{s}_1^{(n)}$ of the algorithm to the transmitter. Since the algorithm is attracted to the globally optimal signature sequence, the sequence of estimates $\hat{\boldsymbol{s}}_1^{(n)}$ sent by the receiver to the transmitter are consistently closer to the globally optimal signature sequence than any other signature sequence.

Memory and Computational Complexity: It is clearly not necessary to store the sequences $\boldsymbol{s}_1^{(n)}$, $\hat{\boldsymbol{s}}_1^{(n)}$, and $\boldsymbol{\pi}[n]$ for all n. They can be overwritten at each time. The main memory overhead required for the algorithm is storing the local variable $\boldsymbol{\pi}[n]$ which requires $\mathcal{O}(2^P)$ memory.

The total computational cost is the cost per iteration times the number of iterations. Regarding the number of iterations, due to the attraction property of the algorithm, it spends more time at the globally optimal signature sequence than any other candidate, see discussion following Theorem 1 in Section III-E. The computational cost per iteration is as follows. Sampling $\tilde{\boldsymbol{s}}_1^{(n)}$ uniformly from $\mathcal{S} - \{\boldsymbol{s}_1^{(n)}\}$ (Step 1) requires minimal cost since sampling uniformly from $\{1, \ldots, 2^P - 1\}$ is implemented as $\lfloor (2^P - 1)U \rfloor + 1$ where $U \sim U[0,1]$. Regarding Step 3, as in [2], it can be written in terms of the occupation time $W^{(n)}$ vector as $W^{(n+1)} = W^{(n)} + \boldsymbol{Y}[n+1]$ rather than the empirical occupation probabilities $\boldsymbol{\pi}[n]$. This update only requires one integer addition per iteration. Note that (14) is merely a recursive way of computing $W^{(n)}/n$. Our formulation in terms of $\boldsymbol{\pi}[n]$ and the interpretation of (14) as an adaptive filtering algorithm will be used in Section IV to devise and analyze a tracking version of the algorithm, see Section IV-A for the complexity of the tracking algorithm and an asynchronous implementation.

The main computational cost at each iteration involves evaluating $X[n, \boldsymbol{s}_1^{(n)}]$, $X[n, \tilde{\boldsymbol{s}}_1^{(n)}]$ which requires $\mathcal{O}(P^2)$ computations (since $\hat{\boldsymbol{C}}_1[n]$ is Toeplitz). Note that this calculation is needed for calculating the linear MMSE receiver anyway, hence it does not require additional computational overhead for the spreading sequence adaptation.

D. Conservative Discrete Stochastic Approximation Algorithm

Here, motivated by the recent work [11], a conservative random search algorithm is presented. Unlike Algorithm 1, where the state of the algorithm jumps around as an irreducible Markov chain, the state of Algorithm 2 converges a.s. to the globally optimal spreading code. That is, the evolution of the state of the algorithm becomes more and more *conservative* as time evolves. The advantage of Algorithm 2 that follows is that its convergence analysis holds for any frame size $M > 0$. This is unlike Algorithm 1, where we require large frame size M. However, in adaptive tracking, it is not as attractive as the tracking version of Algorithm 1, see Section IV.

Algorithm 2 (Single-user spreading code adaptation—Conservative Search)

*Step 0—**Initialization:*** At frame time $n = 0$, initialize the 2^P-dimensional vectors $H[0]$, $L[0]$ to zero and $\bar{K}[0] = \mathbf{1}$ (vector of ones). Select initial state $\boldsymbol{s}_1^{(0)} \in \mathcal{S}$.

*Step 1—**Sampling and evaluation:*** Given the state $\boldsymbol{s}_1^{(n)}$, generate, as in Step 1 of Algorithm 1, $\tilde{\boldsymbol{s}}_1^{(n)}$, $X[n, \boldsymbol{s}_1^{(n)}]$, and $X[n, \tilde{\boldsymbol{s}}_1^{(n)}]$. Update the accumulated cost, occupation times, and average cost as

$$\begin{aligned} L[n+1, \tilde{\boldsymbol{s}}_1^{(n)}] &= L[n, \tilde{\boldsymbol{s}}_1^{(n)}] + X[n, \tilde{\boldsymbol{s}}_1^{(n)}],\ L[n+1, \boldsymbol{s}_1^{(n)}] \\ &= L[n, \boldsymbol{s}_1^{(n)}] + X[n, \boldsymbol{s}_1^{(n)}] \\ \bar{K}[n+1, \tilde{\boldsymbol{s}}_1^{(n)}] &= \bar{K}[n, \tilde{\boldsymbol{s}}_1^{(n)}] + 1,\ \bar{K}[n+1, \boldsymbol{s}_1^{(n)}] \\ &= \bar{K}[n, \boldsymbol{s}_1^{(n)}] + 1 \\ H[n+1, \tilde{\boldsymbol{s}}_1^{(n)}] &= L[n+1, \tilde{\boldsymbol{s}}_1^{(n)}]/\bar{K}[n+1, \tilde{\boldsymbol{s}}_1^{(n)}],\ H[n+1, \boldsymbol{s}_1^{(n)}] \\ &= L[n+1, \boldsymbol{s}_1^{(n)}]/\bar{K}[n+1, \boldsymbol{s}_1^{(n)}]. \end{aligned} \tag{15}$$

*Step 2—**Acceptance:*** If

$$H[n+1, \tilde{\boldsymbol{s}}_1^{(n)}] > H[n+1, \boldsymbol{s}_1^{(n)}]$$

set $\boldsymbol{s}_1^{(n+1)} = \tilde{\boldsymbol{s}}_1^{(n)}$; otherwise set $\boldsymbol{s}_1^{(n+1)} = \boldsymbol{s}_1^{(n)}$.

*Step 3—**Update Estimate of Spreading Code:*** $\hat{\boldsymbol{s}}_1^{(n)} = \boldsymbol{s}_1^{(n)}$. Set $n \leftarrow n+1$ and go to Step 1.

Instead of using (15), Step 1 of Algorithm 2 can also be re-expressed as an adaptive filter as follows. Update the average cost vector estimate $H[n] = (H[n, s(1)], \ldots, H[n, s(2^P)])$ and occupation time matrix $K[\cdot]$ ($(2^P \times 2^P)$ diagonal matrix) as

$$\begin{aligned} K[n+1] &= K[n] + \mu[n+1]\big(\mathrm{diag}(\boldsymbol{Z}[n] + \boldsymbol{Y}[n]) - K[n]\big) \\ H[n+1] &= H[n] + \mu[n+1]K^{-1}[n+1]\Big(X[n, \tilde{\boldsymbol{s}}_1^{(n)}]\boldsymbol{Z}[n] \\ &\quad + X[n, \boldsymbol{s}_1^{(n)}]\boldsymbol{Y}[n] - \mathrm{diag}(\boldsymbol{Z}[n] + \boldsymbol{Y}[n])H[n]\Big). \end{aligned} \tag{16}$$

Here, $\mu[n] = 1/n$, $\boldsymbol{Z}[n] \triangleq \boldsymbol{e}_i$ if $\tilde{\boldsymbol{s}}_1^{(n)} = \boldsymbol{s}(i)$, $i = 1, \ldots, 2^P$, $\boldsymbol{Y}[n]$ is defined in (12), $\mathrm{diag}(\boldsymbol{Z}[n])$ is the diagonal matrix with (i,i) element $\boldsymbol{Z}^i[n]$ (where $\boldsymbol{Z}^i[n]$ is the ith component of $\boldsymbol{Z}[n]$). The $(2^P \times 2^P)$-dimensional diagonal matrix $K[n]$ is initialized as $K[0] = I$ (or any arbitrary positive element diagonal matrix). It is straightforwardly shown (e.g., standard derivation of recursive least squares with forgetting factor) that this is equivalent to Step 1 of Algorithm 2. In terms of the schematic setup in Fig. 1, whereas Algorithm 1 implements an adaptive filter after the sampling and acceptance, Algorithm 2 (16) implements and adaptive filter before the sampling and acceptance step. This makes Algorithm 2 more conservative since the sampling and acceptance is done based on the averaged behavior of the adaptive filter. The computational complexity and memory overheads of Algorithms 1 and 2 are similar.

It is well known in reinforcement learning algorithms [6] that there is a tradeoff between *exploration* (seeking new candidate solutions) and *exploitation* (staying with the best estimate so as to minimize cost). The passive algorithm above does more exploitation and less exploration than the aggressive algorithm.

E. Convergence of Algorithms 1 and 2

In order to prove the consistency and attraction of the discrete stochastic approximation Algorithm 1, we summarize the following result from [1], [2]. The following conditions are required for global convergence and attraction of the discrete stochastic approximation Algorithm 1.

For $\boldsymbol{s}_1^* \in \mathcal{M}_1$ and $\boldsymbol{s}_1, \bar{\boldsymbol{s}}_1 \in \mathcal{S} - \mathcal{M}_1$ and *independent* samples $X[n, \boldsymbol{s}_1]$, $X[n, \bar{\boldsymbol{s}}_1]$, $X[n, \boldsymbol{s}_1^*]$

$$\text{(C1)} \quad P(X[n, \boldsymbol{s}_1^*] > X[n, \boldsymbol{s}_1]) > P(X[n, \boldsymbol{s}_1] > X[n, \boldsymbol{s}_1^*]) \tag{17}$$

$$\text{(C2)} \quad P(X[n, \boldsymbol{s}_1^*] > X[n, \bar{\boldsymbol{s}}_1]) > P(X[n, \boldsymbol{s}_1] > X[n, \bar{\boldsymbol{s}}_1]). \tag{18}$$

Theorem 1: [2, Theorem 2.1] Under Conditions (C1) and (C2), the sequence $\{\boldsymbol{s}_1^{(n)}\}$ generated by Algorithm 1 is a homogeneous irreducible and aperiodic Markov chain with state space $\mathcal{S}$. Furthermore, for sufficiently large n, this Markov chain $\{\boldsymbol{s}_1^{(n)}\}$ spends more time in $\mathcal{M}_1$ than any other states, i.e., $\pi[n, \boldsymbol{s}_1] > \pi[n, \boldsymbol{s}]$ for $\boldsymbol{s}_1 \in \mathcal{M}_1$, $\boldsymbol{s} \in \mathcal{S} - \mathcal{M}_1$. Moreover, $\hat{\boldsymbol{s}_1}^{(n)}$ is attracted to and converges a.s. to an element in $\mathcal{M}_1$.

The proof of the theorem is given in [2]. The proof of convergence of the local search algorithm in Section III-C to a local maximum is given in [1]. The following interpretation is useful in understanding why Theorem 1 works. The sequence $\{\boldsymbol{s}_1^{(n)}\}$ is a homogeneous Markov chain on the state space $\mathcal{S}$. This follows directly from Steps 1 and 2 of Algorithm 1. Let $\boldsymbol{A}$ be the corresponding $2^P \times 2^P$ transition probability matrix. It is well known that if $\{\boldsymbol{s}_1^{(n)}\}$ is aperiodic irreducible, then the strong law of large numbers for Markov chains states that $\boldsymbol{\pi}[n] \to \boldsymbol{\pi}$ a.s., where $\boldsymbol{\pi}$ is the Perron–Frobenius eigenvector (invariant distribution) of the Markov chain $\{\boldsymbol{s}_1^{(n)}\}$, i.e., $\boldsymbol{\pi}$ is a probability vector satisfying

$$\boldsymbol{A}^T \boldsymbol{\pi} = \boldsymbol{\pi}, \qquad \sum_i \pi^i = 1$$

where π^i denotes the ith component of $\boldsymbol{\pi}$. Assumptions (C1) and (C2) shape the transition probability matrix $\boldsymbol{A}$ and hence invariant distribution $\boldsymbol{\pi}$ as follows: (C1) imposes the condition that $\boldsymbol{A}(i,j) < \boldsymbol{A}(j,i)$ for $i \in \mathcal{M}_1$, $j \in \mathcal{S} - \mathcal{M}_1$, i.e., it is more probable to jump from a state outside $\mathcal{M}_1$ to a state in $\mathcal{M}_1$ than the reverse. Assumption (C2) says that $\boldsymbol{A}(i,j) < \boldsymbol{A}(l,j)$ for $i \in \mathcal{M}_1$, $j, l \in \mathcal{S} - \mathcal{M}_1$, i.e, it is less probable to jump out of the global optimum i to another state j compared to any other state l. Intuitively, one would expect that such a transition probability matrix would generate a Markov chain that is attracted to the set $\mathcal{M}_1$ (spends more time in $\mathcal{M}_1$ than other states), i.e., $\pi[i] > \pi[j]$, $i \in \mathcal{M}_1$, $j \notin \mathcal{M}_1$. This is what Theorem 1 says and this is proved in [2] using algebraic arguments.

Verification of (C1) and (C2) for Algorithm 1: Examples of distributions that satisfy (C1) and (C2) are given in [2]. For example, if for fixed i, $X[i, \boldsymbol{s}_1] - \mathbf{E}\{X[i, \boldsymbol{s}_1]\}$, $\boldsymbol{s}_1 \in \mathcal{S}$ are i.i.d. symmetric random variables, then (C1) and (C2) hold, see [2]. However, in our case, the distributions are not identical. We will verify (C1) and (C2) for large frame size $M \to \infty$ using a Gaussian approximation by virtue of the central limit theorem. This is justified from a practical point of view since, as mentioned in Section II, in WCDMA and CDMA 2000, typically the frame size M is several hundreds. Actually, our numerical experiments show that convergence and attraction of Algorithm 1 hold even for moderate batch size $M = 50$.

We start with the following proposition regarding the asymptotic normality of $X[n, \boldsymbol{s}]$.

Proposition 1: For any $\boldsymbol{s}_1 \in \mathcal{S}$ and large frame size M, $X[n, \boldsymbol{s}_1] = \boldsymbol{s}_1^T \hat{\boldsymbol{C}}_1^{-1}[n] \boldsymbol{s}_1$ satisfies

$$\sqrt{M}\Big(X[n, \boldsymbol{s}_1] - \varphi(\boldsymbol{s}_1)\Big) \sim \mathcal{N}(0, v^2(\boldsymbol{s}_1)) \tag{19}$$

where

$$v^2(\boldsymbol{s}_1) = 2\left[\varphi^2(\boldsymbol{s}_1) - \sum_{k=2}^{K} A_k^4 \left(\boldsymbol{w}_1^T \boldsymbol{s}_k\right)^4\right] \approx 2\varphi^2(\boldsymbol{s}_1). \tag{20}$$

Here, $\boldsymbol{w}_1 \triangleq \boldsymbol{C}_1^{-1} \boldsymbol{s}_1$ is the linear MMSE detector for User 1, and the approximation (20) follows from the fact that $\boldsymbol{w}_1^T \boldsymbol{s}_k \approx 0$ for $k = 2, \ldots, K$.

Proof: The differential of $\varphi(\boldsymbol{s}_1)$ with respect to $\boldsymbol{C}_1$ is given by

$$\Delta\varphi(\boldsymbol{s}_1) = -\boldsymbol{s}_1^T \boldsymbol{C}_1^{-1} \Delta\boldsymbol{C}_1 \boldsymbol{C}_1^{-1} \boldsymbol{s}_1 = -\boldsymbol{w}_1^T \Delta\boldsymbol{C}_1 \boldsymbol{w}_1 \tag{21}$$

where $\Delta\boldsymbol{C}_1 \triangleq \hat{\boldsymbol{C}}_1[n] - \boldsymbol{C}_1$. Hence,

$$v^2 = \mathbf{E}\left\{(\Delta\varphi(\boldsymbol{s}_1))^2\right\} = \boldsymbol{w}_1^T \mathbf{E}\left\{\Delta\boldsymbol{C}_1 \boldsymbol{w}_1 \boldsymbol{w}_1^T \Delta\boldsymbol{C}_1\right\} \boldsymbol{w}_1. \tag{22}$$

Now, we make use of the following result proved in [15]: $\sqrt{M}\Delta\boldsymbol{C}_1$ converges in distribution to a matrix valued Gaussian random variable with mean $\mathbf{0}$ and a $(P^2 \times P^2)$ covariance matrix whose elements are specified by

$$\begin{aligned}\lim_{M\to\infty} M \cdot \mathrm{Cov}\Big\{[\Delta\boldsymbol{C}_1]_{i,j}, [\Delta\boldsymbol{C}_1]_{m,n}\Big\} &= [\boldsymbol{C}_1]_{i,m}[\boldsymbol{C}_1]_{j,n} + [\boldsymbol{C}_1]_{i,n}[\boldsymbol{C}_1]_{j,m} \\ &\quad - 2\sum_{\alpha=2}^{K} A_\alpha^4 [\boldsymbol{s}_\alpha]_i [\boldsymbol{s}_\alpha]_j [\boldsymbol{s}_\alpha]_m [\boldsymbol{s}_\alpha]_n. \end{aligned} \tag{23}$$

Using this result together with (22), we have for large M

$$\begin{aligned} M \cdot \mathbf{E}\left\{\Delta\boldsymbol{C}_1 \boldsymbol{w}_1 \boldsymbol{w}_1^T \Delta\boldsymbol{C}_1\right\}_{i,j} &= M \cdot \mathbf{E}\left\{\sum_{m=1}^{P}\sum_{n=1}^{P} [\Delta\boldsymbol{C}_1]_{i,m} [\boldsymbol{w}_1]_m [\Delta\boldsymbol{C}_1]_{j,n} [\boldsymbol{w}_1]_n\right\} \\ &= \sum_{m=1}^{P}\sum_{n=1}^{P} \Bigg([\boldsymbol{C}_1]_{i,j}[\boldsymbol{C}_1]_{m,n} + [\boldsymbol{C}_1]_{i,n}[\boldsymbol{C}_1]_{m,j} \\ &\qquad - 2\sum_{\alpha=1}^{K} A_\alpha^4 [\boldsymbol{s}_\alpha]_i [\boldsymbol{s}_\alpha]_j [\boldsymbol{s}_\alpha]_l [\boldsymbol{s}_\alpha]_m \Bigg) [\boldsymbol{w}_1]_m [\boldsymbol{w}_1]_n \\ &= [\boldsymbol{C}_1]_{i,j} \underbrace{\left(\sum_{m=1}^{P}\sum_{n=1}^{P} [\boldsymbol{C}_1]_{m,n} [\boldsymbol{w}_1]_m [\boldsymbol{w}_1]_n\right)}_{\boldsymbol{w}_1^T \boldsymbol{C}_1 \boldsymbol{w}_1} \end{aligned}$$

$$+\underbrace{\left(\sum_{m=1}^{P}[\boldsymbol{C}_1]_{m,j}[\boldsymbol{w}_j]_k\right)}_{[\boldsymbol{C}_1\boldsymbol{w}_1]_j}\underbrace{\left(\sum_{n=1}^{P}[\boldsymbol{C}_1]_{i,n}[\boldsymbol{w}_1]_i\right)}_{[\boldsymbol{C}_1\boldsymbol{w}_1]_i}$$
$$-2\sum_{\alpha=2}^{K}[\boldsymbol{s}_\alpha]_i[\boldsymbol{s}_\alpha]_j A_\alpha^4\underbrace{\left(\sum_{m=1}^{P}[\boldsymbol{s}_\alpha]_m[\boldsymbol{w}_1]_m\right)}_{\boldsymbol{s}_\alpha^T\boldsymbol{w}_1}^2. \quad (24)$$

Writing (24) in a matrix form, we have

$$M\cdot\mathbf{E}\left\{\Delta\boldsymbol{C}_1\boldsymbol{w}_1\boldsymbol{w}_1^T\Delta\boldsymbol{C}_1\right\} = \boldsymbol{C}_1\left(\boldsymbol{w}_1^T\boldsymbol{C}_1\boldsymbol{w}_1\right)+\boldsymbol{C}_1\boldsymbol{w}_1\boldsymbol{w}_1^T\boldsymbol{C}_1-2\boldsymbol{S}_1\boldsymbol{D}_1\boldsymbol{S}_1^T \quad (25)$$

with

$$\boldsymbol{S}_1 \triangleq [\boldsymbol{s}_2\,\boldsymbol{s}_3\,\ldots\,\boldsymbol{s}_K] \quad (26)$$
$$\boldsymbol{D}_1 \triangleq \mathrm{diag}\left\{A_2^4(\boldsymbol{s}_2^T\boldsymbol{w}_1)^2,\ldots,A_K^4(\boldsymbol{s}_K^T\boldsymbol{w}_1)^2\right\}. \quad (27)$$

Substituting (25)–(27) into (22), we obtain the proposition. □

Based on the above proposition, in the rest of this paper we will assume that for some large finite size frame M

$$X[n,\boldsymbol{s}_1]\sim\mathcal{N}(\varphi(\boldsymbol{s}_1),\upsilon^2(\boldsymbol{s}_1)/M).$$

This is justified from a practical point of view since typically the frame size M is several hundreds (as mentioned in Section II). Then the discrete stochastic optimization problem (7) can be reformulated as follows. There are 2^P possible spreading codes $\boldsymbol{s}_1\in\mathcal{S}$. On picking out any one of these spreading codes $\boldsymbol{s}_1$, one observes an independent sample $X[n,\boldsymbol{s}_1]$ drawn from the normal distribution $\mathcal{N}(\varphi(\boldsymbol{s}_1),\upsilon^2(\boldsymbol{s}_1)/M)$. The aim is to devise an algorithm which computes the maximizer of $\varphi(\boldsymbol{s}_1)$ defined in (7) with minimum effort.

Let us verify the Conditions (C1) and (C2) for our spreading code optimization problem. According to Algorithm 1, since $X[n,\boldsymbol{s}_1^*]$ and $X[n,\boldsymbol{s}_1]$ are statistically independent samples, we have

$$X[n,\boldsymbol{s}_1^*]-X[n,\boldsymbol{s}_1]\sim\mathcal{N}\left(\varphi(\boldsymbol{s}_1^*)-\varphi(\boldsymbol{s}_1),\frac{(\upsilon^2(\boldsymbol{s}_1^*)+\upsilon^2(\boldsymbol{s}_1))}{M}\right).$$

Hence Condition (C1) is equivalent to

$$\Phi\left(\frac{\sqrt{M}(\varphi(\boldsymbol{s}_1^*)-\varphi(\boldsymbol{s}_1))}{\sqrt{\upsilon^2(\boldsymbol{s}_1^*)+\upsilon^2(\boldsymbol{s}_1)}}\right)>\Phi\left(\frac{\sqrt{M}(\varphi(\boldsymbol{s}_1)-\varphi(\boldsymbol{s}_1^*))}{\sqrt{\upsilon^2(\boldsymbol{s}_1^*)+\upsilon^2(\boldsymbol{s}_1)}}\right)$$

where $\Phi(\cdot)$ denotes the standard Gaussian distribution function. Since $\varphi(\boldsymbol{s}_1^*)>\varphi(\boldsymbol{s}_1)$ (because $\boldsymbol{s}_1^*$ is the global maximum) and $\Phi(\cdot)$ is monotonically increasing, the above equation holds.

Consider (C2). Similar to the preceding argument, (C2) is equivalent to

$$\Phi\left(\frac{\sqrt{M}(\varphi(\boldsymbol{s}_1^*)-\varphi(\boldsymbol{s}_1))}{\sqrt{\upsilon^2(\boldsymbol{s}_1^*)+\upsilon^2(\boldsymbol{s}_1)}}\right)>\Phi\left(\frac{\sqrt{M}(\varphi(\boldsymbol{s}_1)-\varphi(\bar{\boldsymbol{s}}_1))}{\sqrt{\upsilon^2(\boldsymbol{s}_1)+\upsilon^2(\bar{\boldsymbol{s}}_1)}}\right).$$

Then using (20) since $\Phi(\cdot)$ is monotonically increasing, this is equivalent to showing that

$$F(x,z)\triangleq\sqrt{M}(x-z)/\sqrt{2x^2+2z^2}$$

is monotonically increasing in x for $x\in(0,1)$ and any fixed $z\in(0,1)$ and fixed positive integer M. It is an elementary exercise in calculus to verify this.

Convergence of Algorithm 2: It is readily verified that $E|X[n,\boldsymbol{s}]|^k<\infty$ for $k\geq 1$ so it is uniformly integrable. This and the Kolmogorov's strong law of large numbers imply

$$\lim_{M\to\infty}\mathbf{E}\{X[n,\boldsymbol{s}]\}=\varphi(\boldsymbol{s}).$$

Thus, by a similar proof to [11, Theorem 4.1] we have the following.

Theorem 2: There exists a finite integer n_0 a.s. such that the estimates $\hat{\boldsymbol{s}}_1^{(n)}$ generated by Step 3 of Algorithm 2 for any finite frame size $M>0$ satisfy $\hat{\boldsymbol{s}}_1^{(n)}\in\mathcal{M}_1$ for all $n\geq n_0$, where $\mathcal{M}_1$ is the set of global maximizers.

F. Spreading Code Optimization for Multiple Users

In lieu of (7), consider the problem of optimizing the spreading codes of $\tilde{K}$ users ($\tilde{K}\leq K$). Compute

$$\max_{\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}\in\mathcal{S}^{\tilde{K}}}\varphi(\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}) = \max_{\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}\in\mathcal{S}^{\tilde{K}}}\mathbf{E}\left\{X[n,\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}]\right\} \quad (28)$$

where $X[n,\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}]$ is an unbiased estimator of the cost $\varphi(\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}})$ when the spreading codes chosen by the $\tilde{K}$ users are $\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}$, respectively.

Example—Maximizing Sum of SINRs: One possible objective is to maximize the sum of the SINRs of these $\tilde{K}$ users, i.e.,

$$\varphi(\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}})=\sum_{k=1}^{\tilde{K}}\mathrm{SINR}_k$$

where $\mathrm{SINR}_k=\boldsymbol{s}_k^T\boldsymbol{C}_k^{-1}\boldsymbol{s}_k$ denotes the SINR of user k, and

$$\boldsymbol{C}_k=\sum_{j=1,j\neq k}^{\tilde{K}}\left(A_j^2\boldsymbol{s}_j\boldsymbol{s}_j^T+\eta\boldsymbol{I}_P\right).$$

One can then use the estimator

$$X[n,\boldsymbol{s}_1,\boldsymbol{s}_2,\ldots,\boldsymbol{s}_{\tilde{K}}]=\sum_{k=1}^{\tilde{K}}\boldsymbol{s}_k^T\left[\hat{\boldsymbol{C}}[n]+\sum_{j=1,j\neq k}^{\tilde{K}}A_j^2\boldsymbol{s}_j\boldsymbol{s}_j^T\right]^{-1}\boldsymbol{s}_k \quad (29)$$

where $\hat{\boldsymbol{C}}[n]$ denotes the sample correlation of the total interference signals, i.e.,

$$\hat{\boldsymbol{C}}[n]=\frac{1}{M}\sum_{j=nM}^{(n+1)M-1}\boldsymbol{i}[j]\,\boldsymbol{i}^T[j],\quad \boldsymbol{i}[\iota]\triangleq\boldsymbol{r}[\iota]-\sum_{k=1}^{\tilde{K}}A_kb_k[\iota]\boldsymbol{s}_k.$$

A simple modification of Algorithm 1 or Algorithm 2 can be used. In Step 1, $\{\tilde{\boldsymbol{s}}_1^{(n)},\ldots,\tilde{\boldsymbol{s}}_{\tilde{K}}^{(n)}\}$ is sampled uniformly from $\mathcal{S}^{\tilde{K}}-\{\boldsymbol{s}_1^{(n)},\ldots,\boldsymbol{s}_{\tilde{K}}^{(n)}\}$. Steps 2 and 3 are similar to Algorithm 1 or Algorithm 2 with straightforward modifications (details are omitted). Just as for the single-user case, a local search can be used, see Section III-C. Another alternative is the coordinate ascent algorithm [5]. Fixing $(\boldsymbol{s}_2^{(n)},\ldots,\boldsymbol{s}_{\tilde{K}}^{(n)})$, let $\boldsymbol{s}_1^{(n+1)}$ denote the optimum of (29) with respect to $\boldsymbol{s}_1$. Then fix $(\boldsymbol{s}_1^{(n+1)},\boldsymbol{s}_3^{(n)},\ldots,\boldsymbol{s}_{\tilde{K}}^{(n)})$ and optimize with respect to $\boldsymbol{s}_2$ to obtain $\boldsymbol{s}_2^{(n+1)}$, and so on.

Similar to Proposition 1, it follows that as frame size $M\to\infty$

$$\sqrt{M}\left(X[n,\boldsymbol{s}_1^{(n)},\ldots,\boldsymbol{s}_{\tilde{K}}^{(n)}]-\sum_{k=1}^{\tilde{K}}\mathrm{SINR}_k\right)\sim\mathcal{N}\left(0,\,\upsilon^2\right)$$

where

$$\upsilon^2=2\sum_k(\mathrm{SINR}_k)^2. \quad (30)$$

Then verifications of Conditions (C1) and (C2) are similar to the single-code optimization. For example, verifying (C2) is equivalent to showing that the function

$$F(x_1,\dots,x_{\tilde{K}},z_1,\dots,z_{\tilde{K}}) \triangleq \frac{\sum_{k=1}^{\tilde{K}}(x_k - z_k)}{\sqrt{\frac{2}{M}\sum_{k=1}^{\tilde{K}}(x_k^2 + z_k^2)}}$$

is monotonically increasing in every component $x_k, 0 \le x_k \le 1$ for fixed $z_1,\dots,z_{\tilde{K}}$ where $0 \le z_k \le 1$. This is easily shown.

Example—Maximizing Sum Capacity:

$$\varphi(\boldsymbol{s}_1,\boldsymbol{s}_2,\dots,\boldsymbol{s}_{\tilde{K}}) = \sum_{k=1}^{\tilde{K}} \log(1+\text{SINR}_k)$$

with estimator

$$X[n,\boldsymbol{s}_1,\boldsymbol{s}_2,\dots,\boldsymbol{s}_{\tilde{K}}] = \sum_{k=1}^{\tilde{K}} \log\left[1+\boldsymbol{s}_k^T\left(\hat{\boldsymbol{C}}[n]+\sum_{j=1,j\neq k}^{\tilde{K}} A_j^2\boldsymbol{s}_j\boldsymbol{s}_j^T\right)^{-1}\boldsymbol{s}_k\right].$$

Then applying δ-method of asymptotic normality [25, Theorem 1.8.12] to (30) yields as frame size $M\to\infty$

$$\sqrt{M}\left(X[n,\boldsymbol{s}_1^{(n)},\dots,\boldsymbol{s}_{\tilde{K}}^{(n)}])-\varphi(\boldsymbol{s}_1,\boldsymbol{s}_2,\dots,\boldsymbol{s}_{\tilde{K}})\right) \sim \mathcal{N}\left(0,\sum_{k=1}^{\bar{K}}\frac{\text{SINR}_k^2}{(1+\text{SINR}_k)^2}\right).$$

Conditions (C1) and (C2) are easily verified. For example, (C2) is equivalent to showing that

$$\left[\sum_{k=1}^{\bar{K}}[\log(1+x_k)-\log(1+z_k)]\right]\Big/\sqrt{\frac{2}{M}\sum_{k=1}^{\bar{K}}\left(\frac{x_k^2}{1+x_k^2}+\frac{z_k^2}{1+z_k^2}\right)}$$

is monotone increasing in every component x_k, $0 \le x_k \le 1$, for fixed $z_1,\dots,z_{\tilde{K}}$ where $0 \le z_k \le 1$.

IV. Adaptive Discrete Stochastic Approximation Algorithms

Algorithms 1 and 2 deal with the case when the optimal spreading code is time invariant. Here, we consider the case where due to slow fading or user arrivals or departures, the optimal spreading code $\boldsymbol{s}_1^* \in \mathcal{S}$ is time varying. Denote the time-varying optimal spreading code by $\boldsymbol{s}_1^*[n]$. Such nonstationary environments are at the very heart for applications of adaptive stochastic approximation algorithms to track time-varying parameters.

A. Constant Step-Size Discrete Stochastic Approximation Algorithm

We propose the following constant step-size discrete stochastic approximation algorithm for tracking the time-varying parameter.

Algorithm 3 (Constant step-size spreading code adaptation)

Steps 1 and 2: Identical to Algorithm 1.

*Step 3—**constant step-size:*** Replace (14) in Step 3 of Algorithm 1 with a fixed step-size algorithm, i.e.,

$$\boldsymbol{\pi}[n+1] = \boldsymbol{\pi}[n] + \mu\left(\boldsymbol{Y}[n+1]-\boldsymbol{\pi}[n]\right) \tag{31}$$

where the step-size μ is a small positive constant.

Step 4: Identical to Algorithm 1.

Remark: As long as the step size satisfies $0<\mu<1$, $\boldsymbol{\pi}[n]$ is guaranteed to be a probability vector. To see this, note that $\boldsymbol{1}^T(\boldsymbol{Y}[n+1]-\boldsymbol{\pi}[n])=0$ implying that $\boldsymbol{1}^T\boldsymbol{\pi}[n+1]=\boldsymbol{1}^T\boldsymbol{\pi}[n]=1$. Also rewriting (31) as $(1-\mu)\boldsymbol{\pi}[n]+\mu\boldsymbol{Y}[n+1]$ implies that all elements of $\boldsymbol{\pi}[n+1]$ are nonnegative.

The constant step size μ essentially introduces an exponential forgetting of the past occupation probabilities and permits us to track slowly time-varying $\boldsymbol{s}_1^*[n]$.

A similar constant step size version of Algorithm 2 can be formulated. However, note that while the sampling and evaluation steps of Algorithm 1 and its adaptive version Algorithm 3 are identical (since the adaptive filtering is done after the sampling and evaluation steps), this is not so for an adaptive version of Algorithm 2. Indeed, one would expect the adaptive version of Algorithm 2 to track time-varying parameters slower since it does not aggressively explore the state space. Moreover, the analysis of the adaptive version of Algorithm 2 is much more difficult and beyond the scope of the current paper. The rest of this section is devoted to mean-square analysis of Algorithm 3.

Complexity and Asynchronous Implementation: The complexity of (31) is $\mathcal{O}(2^P)$ additions and multiplications at each time instant. In practical implementation, the following asynchronous version of Algorithm 3 can be used. Replace $\boldsymbol{\pi}[n]$ in Algorithm 3 with $\hat{\boldsymbol{\pi}}[n]$ which is updated as (32) at the bottom of the page. Thus, at each time instant n, only one component of $\hat{\boldsymbol{\pi}}[n+1]$ is updated which requires one addition and multiplication. Note that the update times of the jth component of $\hat{\boldsymbol{\pi}}[n]$ occur at the (random) time instants when the Markov chain $\boldsymbol{Y}[n]$ visits the state $\boldsymbol{e}_j$—thus, the algorithm can be viewed as an asynchronous implementation of Algorithm 3; see [23], [24], and the references therein. Note that the components of $\hat{\boldsymbol{\pi}}[n]$ are nonnegative but do not sum to one. Let $\bar{\boldsymbol{\pi}}[n]$ be the normalized version of $\hat{\boldsymbol{\pi}}[n]$, i.e., the elements of $\bar{\boldsymbol{\pi}}[n]$ add to one. Then obviously the estimate of the optimal spreading code $\hat{\boldsymbol{s}}_1^{(n)} = \arg\max_j \hat{\boldsymbol{\pi}}[n,j] = \arg\max_j \bar{\boldsymbol{\pi}}[n,j]$.

Tracking Analysis: In adaptive filtering (e.g., LMS), a typical method for analyzing the tracking performance of an adaptive algorithm is to postulate a *hypermodel* for the variation in the true parameter. Since the optimal spreading code $\boldsymbol{s}_1^*[n]$ belongs to a finite set $\mathcal{S}$ and its evolution can be correlated in time (e.g., due to the correlation of fading), we choose to describe $\boldsymbol{s}_1^*[n]\in\mathcal{S}$ as a slow Markov chain on $\mathcal{S}$ for our mean square convergence analysis. The Markov chain hypermodel below is one of the most general models available for a finite-state model. Note that the hypermodel assumption is only used for our subsequent analysis, it does not enter the actual algorithm imple-

$$\hat{\boldsymbol{\pi}}[n+1] = \hat{\boldsymbol{\pi}}[n] + \mu\text{diag}(1_{\{\boldsymbol{Y}[n+1]=\boldsymbol{e}_1\}},\dots,1_{\{\boldsymbol{Y}[n+1]=\boldsymbol{e}_{2^P}\}})\left(\boldsymbol{Y}[n+1]-\hat{\boldsymbol{\pi}}[n]\right), \qquad \hat{\boldsymbol{\pi}}[0]=\boldsymbol{0}. \tag{32}$$

mentation. Let $\theta[n] \in \mathcal{E}$ be such that $\theta[n] = \boldsymbol{e}_i$ if $\boldsymbol{s}_1^*[n] = \boldsymbol{s}(i)$. We make the following assumptions.

(M1) *Hypermodel*: Assume that $\theta[n]$ (or, equivalently, the optimal spreading code $\boldsymbol{s}_1^*[n]$) is a "slow" Markov chain on $\mathcal{E}$ with transition probability matrix $\boldsymbol{P}^\varepsilon = (p_{ij}^\varepsilon)$ given by

$$\boldsymbol{P}^\varepsilon = \boldsymbol{I} + \varepsilon \boldsymbol{Q} \tag{33}$$

where $\varepsilon > 0$ is a small parameter, $\boldsymbol{I}$ denotes the $(2^P \times 2^P)$ identity matrix, and $\boldsymbol{Q} = (q_{ij})$ is a $(2^P \times 2^P)$ matrix with $q_{ij} \geq 0$ for $i \neq j$ and $\sum_{j=1} q_{ij} = 0$ for each $i = 1, \ldots, 2^P$. Assume for simplicity that the initial distribution $P(\theta[0] = \boldsymbol{e}_i) = p_{0,i}$ is independent of ε for each $i = 1, \ldots, 2^P$, where $p_{0,i} \geq 0$ and $\sum_{i=1}^{2^P} p_{0,i} = 1$. Denote

$$\boldsymbol{p}_0 \triangleq [p_{0,1}, \ldots, p_{0,2^P}] \in \mathbb{R}^{1 \times 2^P}.$$

The small parameter ε specifies how slowly the hypermodel evolves with time. It belongs to nearly completely decomposable matrix models. Such a notion has been applied in queueing networks for organizing resources in a hierarchical fashion in computer systems for aggregating memory levels, and in economics for reduction of complexity of large-scale systems; see [10] and [38]. Recently, such time-scale separation has gained renewed interest owing to its ability for reduction of complexity; see [47] and also the continuous-time counter part in [29], [46], and the references therein. Note that for sufficiently small $\varepsilon > 0$, $\boldsymbol{P}^\varepsilon$ is a valid transition probability matrix (i.e., each element is nonnegative and each row sums to 1) and, furthermore, the corresponding Markov chain is irreducible. It is clear from Theorem 1 that for any fixed optimal spreading code, $\boldsymbol{s}_1^*$ (or, equivalently, θ), Algorithm 1 generates an irreducible aperiodic Markov chain. This straightforwardly translates to the following corollary for Algorithm 3.

Corollary 1: Given $\{\theta[n]\}$, the state $\{\boldsymbol{Y}[n]\}$ (or, equivalently, $\{\boldsymbol{s}_1^{(n)}\}$) of Algorithm 3 is a Markov chain with $2^P \times 2^P$ irreducible aperiodic transition probability matrix $\boldsymbol{A}_\theta = (a_{ij}(\theta))$, where

$$\begin{aligned} a_{ij}(\theta) &= P(\boldsymbol{Y}[n+1] = \boldsymbol{e}_j \mid \boldsymbol{Y}[n] = \boldsymbol{e}_i, \theta[n+1] = \theta) \\ &= P(\boldsymbol{Y}[1] = \boldsymbol{e}_j \mid \boldsymbol{Y}[0] = \boldsymbol{e}_i, \theta[1] = \theta). \end{aligned}$$

Remark: Let $\boldsymbol{\pi}_\theta$ denote the invariant distribution of $\boldsymbol{A}_\theta$, i.e., $\boldsymbol{\pi}_\theta = \boldsymbol{A}_\theta^T \boldsymbol{\pi}_\theta$, $\mathbf{1}^T \boldsymbol{\pi}_\theta = 1$. The quantity ε being small implies that the Markov chain $\theta[n]$ and, thus, the true optimum $\boldsymbol{s}_1^*[n]$ has slow dynamics, i.e., it jumps infrequently. Note that μ is the step size of the adaptive algorithm for estimating $\boldsymbol{\pi}[n]$. Typically, for an adaptive algorithm to successfully track a time-varying optimum, the rate of change in the true optimum (i.e., ε) should be smaller than the tracking speed of the tracking algorithm (i.e., μ).

A Markov chain with transition matrix (33) is known to belong to the class of singularly perturbed Markov chains. It is a Markov chain with two time scales. The small parameter ε serves the purpose of separating the fast and the slow transition rates. In view of (33), it is clear that the Markov chain $\theta[n]$ depends on ε and should be more appropriately written as $\theta^\varepsilon[n]$. The idea behind this hyper-model is that the chain will spend most of its time at a constant value. However, due to the presence of the generator $\boldsymbol{Q}$, from time to time, the chain jumps into some other location, and the parameter is time varying.

To proceed, we consider the probability distribution vector of $\theta^\varepsilon[n]$. A moment of reflection shows that we can write the $2^P \times 2^P$ identity matrix in (33) as $\boldsymbol{I} = \operatorname{diag}\{1, \ldots, 1\}$, which can be viewed as a transition matrix that consists of 2^P-irreducible transition matrices, each of which is simply the number 1. It is evident that this matrix 1 is irreducible and aperiodic, and a result, its stationary distribution is also 1. Denote the 2^P-dimensional probability vector by

$$\boldsymbol{p}^\varepsilon(n) = \left[P(\theta^\varepsilon[n] = \boldsymbol{e}_i), \qquad i = 1, \ldots, 2^P\right] \in \mathbb{R}^{1 \times 2^P}$$

with $\boldsymbol{p}^\varepsilon(0) = \boldsymbol{p}_0$. By applying Theorems 3.5 and 4.3 in [47], the following assertion holds.

Lemma 1: For some $0 < \lambda < 1$, $\boldsymbol{p}^\varepsilon(n) = \overline{\boldsymbol{p}}(\varepsilon n) + \mathcal{O}(\varepsilon + \lambda^n)$. where $\overline{\boldsymbol{p}}(t)$ is a solution of the initial value problem

$$\frac{d\overline{\boldsymbol{p}}(t)}{dt} = \overline{\boldsymbol{p}}(t)\boldsymbol{Q}, \;\; \overline{\boldsymbol{p}}(0) = \boldsymbol{p}_0. \tag{34}$$

Moreover, consider the n-step transition matrices resulting from (33), denoted by $(\boldsymbol{P}^\varepsilon)^n$. Then $(\boldsymbol{P}^\varepsilon)^n = \overline{\boldsymbol{P}}(\varepsilon n) + \mathcal{O}(\varepsilon + \lambda^n)$, where $\overline{\boldsymbol{P}}(t)$ is a solution of the differential equation

$$\frac{d\overline{\boldsymbol{P}}(t)}{dt} = \overline{\boldsymbol{P}}(t)Q, \;\; \overline{\boldsymbol{P}}(0) = \boldsymbol{I}. \tag{35}$$

The following theorem gives a mean-squares bound on the tracking error of the occupation probability estimate $\boldsymbol{\pi}[n]$ generated by Algorithm 3, i.e., a discrete stochastic approximation algorithm in tandem with a fixed step-size adaptive filtering algorithm. The proof of the result is given in the Appendix . A remark on the mean-square error analysis of the asynchronous implementation (32) is also in the Appendix after the proof.

Theorem 3: Under the Conditions (C1), (C2), and (M1) for sufficiently large n, the mean-square error of the estimate $\boldsymbol{\pi}[n]$ generated by the tracking Algorithm 3 satisfies

$$\mathbf{E}\|\boldsymbol{\pi}[n] - \boldsymbol{\pi}_{\theta[n]}\|^2 = \mathcal{O}\left(\mu + \frac{\varepsilon}{\mu}\right) \text{ for sufficiently large } n. \tag{36}$$

Note that $\boldsymbol{\pi}_{\theta[n]}$ is the invariant distribution corresponding to $\theta = \theta[n]$. The proof of the above theorem is given in Appendix A. Looking at the order of magnitude estimate $\mathcal{O}(\mu + \varepsilon/\mu)$, to balance the two terms μ and ε/μ, we need to choose $\mu = \mathcal{O}(\sqrt{\varepsilon})$. That is, the rate of change of the true parameter should be slower than the adaptation of the tracking algorithm for the algorithm to successfully track the time-varying parameter. To some extent, this is a tractability condition.

Due to the discrete nature of our problem, it makes sense to give bounds on the probability of error of the estimates $\hat{\boldsymbol{s}}_1^{(n)}$ rather than the mean-square error. Our main result (Corollary 2) shows that for large n, the probability of error in choosing the optimal spreading code is $\mathcal{O}(\mu^{1-2\alpha})$ where $0 < \alpha < 1/2$ providing $\mu = \mathcal{O}(\sqrt{\varepsilon})$. The proof appears in Appendix A.

Corollary 2: Under the conditions of Theorem 3, if $\mu = \mathcal{O}(\sqrt{\varepsilon})$, then for sufficiently large n, the error probability of the estimate $\hat{\boldsymbol{s}}_1^{(n)}$ generated by the discrete stochastic approximation tracking algorithm satisfies

$$P\left(\hat{\boldsymbol{s}}_1^{(n)} \neq \boldsymbol{s}_1^*[n]\right) \leq C\mu^{1-2\alpha}, \qquad 0 < \alpha < 1/2 \tag{37}$$

where C is a positive constant independent of μ and ε.

The main use of the above corollary is as a consistency check. As μ and $\varepsilon \to 0$, the probability of error of the tracking algorithm goes to zero. An identical result holds for the asynchronous algorithm (32).

Remark: We also refer the reader to our recent work [45] where a more sophisticated ordinary differential equation (ODE) weak convergence analysis of Algorithm 3 is given for the case $\mu = \mathcal{O}(\epsilon)$. Somewhat remarkably this leads to a switched Markov ODE. Also explicit error probability estimates are obtained in [45] from the diffusion limit.

B. Adaptive Step-Size Discrete Stochastic Approximation Algorithm

Section IV-A presented a discrete stochastic approximation algorithm in tandem with a constant step-size adaptive filtering algorithm for tracking the occupation probabilities $\boldsymbol{\pi}[n]$. Here we propose a discrete stochastic approximation algorithm where the occupation probabilities are tracked by an *adaptive* step-size algorithm.

The choice of step size μ is of critical importance in the tracking algorithm. Ideally, one would want μ to be large when the current estimate is far away from the optimal spreading code and μ to be small when the current estimate is close to the optimal sequence. However, selecting *a priori* the best step size is not straightforward since it depends on the dynamics of the tracking algorithm and the time-varying nature of the optimal spreading code $\boldsymbol{s}_1^*[n]$ which is essentially unknown. One enticing alternative is to replace the fixed μ in (31) by a time-varying step-size sequence $\{\mu[n]\}$, and adaptively adjust $\mu[n]$ as the dynamics evolve. The origin of the adaptive step-size approach can be traced back to [4]. Such ideas were further exploited in [8] with examples from sonar signal processing and in [21] where such algorithms were used for adaptive blind multiuser detection. The proof of the results and technical developments can be found in [22] (see also [24]).

In what follows, we set up the framework and present the asymptotic results. Define $\boldsymbol{e}^\mu[n] \triangleq \boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]$. Note that $\boldsymbol{e}^\mu[n]$ depends on μ since $\boldsymbol{\pi}[n]$ should really have been written as $\boldsymbol{\pi}^\mu[n]$. Denote the mean-square derivative of $\boldsymbol{\pi}^\mu[n]$ by $\boldsymbol{D}^\mu[n]$. By mean-square derivative, we mean

$$\lim_{\delta \to 0} E \left| \frac{\boldsymbol{\pi}^{\mu+\delta}[n] - \boldsymbol{\pi}^\mu[n]}{\delta} - \boldsymbol{D}^\mu[n] \right|^2 = 0.$$

Formally differentiating (31) with respect to μ yields

$$\boldsymbol{D}^\mu[n+1] = \boldsymbol{D}^\mu[n] - \mu \boldsymbol{D}^\mu[n] + (\boldsymbol{Y}[n+1] - \boldsymbol{\pi}^\mu[n]).$$

Following the ideas suggested in [4], [8], [22], we propose an adaptive step-size discrete stochastic approximation algorithm as follows.

Algorithm 4 (Adaptive step-sizes spreading code adaptation)

Steps 1 and 2: Identical to Algorithm 1.

*Step 3—**adaptive step-size:*** Replace (14) in Step 3 of Algorithm 1 with

$$\begin{aligned} \boldsymbol{e}^\mu[n] &= \boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n] \\ \boldsymbol{\pi}[n+1] &= \boldsymbol{\pi}[n] + \mu[n] \boldsymbol{e}^\mu[n] \\ \mu[n+1] &= \left\{ \mu[n] + \rho \boldsymbol{e}^\mu[n]^T \boldsymbol{D}[n] \right\}_{\mu_-}^{\mu_+} \\ \boldsymbol{D}[n+1] &= (1 - \mu[n]) \boldsymbol{D}[n] + \boldsymbol{e}^\mu[n], \quad \boldsymbol{D}[0] = \mathbf{0}. \end{aligned} \tag{38}$$

Step 4: Identical to Algorithm 1.

In the preceding algorithm, $\rho > 0$ denotes the "learning rate." $\{X\}_{\mu_-}^{\mu_+}$ denotes the projection of X onto the interval $[\mu_-, \mu_+]$. That is, if $X > \mu_+$ (respectively, $X < \mu_-$), its value will be reset to μ_+ (respectively, μ_-). The third equation in (38) can be written as

$$\mu[n+1] = \mu[n] + \rho \boldsymbol{e}^\mu[n]^T \boldsymbol{D}[n] + \rho Z[n]$$

where $Z[n]$ is known as a reflection term. That is,

$$\rho Z[n] = \mu[n+1] - (\mu[n] + \rho \boldsymbol{e}^\mu[n]^T \boldsymbol{D}[n])$$

is the term with minimal absolute value needed to bring $\mu[n] + \rho \boldsymbol{e}^\mu[n]^T \boldsymbol{D}[n]$ back to the constraint interval $I_\mu = [\mu_-, \mu_+]$ if it is not in I_μ.

Note that the above algorithm comprises of three parts. A randomized search algorithm which feeds its candidate solution $\boldsymbol{s}_1^{(n)}$ to an adaptive step-size-LMS algorithm. The adaptive step-size-LMS algorithm itself comprises of two cross-coupled LMS algorithms—one LMS algorithm estimates the occupancy probabilities $\boldsymbol{\pi}[n]$, the other adapts the step size $\mu[n]$. If the learning rate $\rho = 0$, then the adaptive-step-size algorithm reduces to the constant step-size algorithm. Finally, if the optimal spreading code $\boldsymbol{s}_1^*[n]$ was time invariant, then $\mu[n] \to 0$ as expected. As one would expect, numerical studies in [22] (see also [21]) have shown that the sensitivity of the adaptive step-size algorithm to the choice of learning rate ρ is much smaller than the sensitivity of a fixed step-size stochastic approximation algorithm to the choice of step size μ.

Remark: Note that our framework is blind in that it does not explicitly assume a model for the time variations of the true spreading code $\boldsymbol{s}_1^*[n]$ (and hence $\boldsymbol{\pi}[n]$) as they are essentially unknown. Instead, we bury the time variations in $\boldsymbol{Y}[n]$, see [22], [21] for further motivation.

To study the convergence of the adaptive step size $\{\mu[n]\}$, we use the ODE approach and weak convergence methods detailed in [24]. Define continuous-time interpolations

$$\mu^\rho(t) = \begin{cases} \mu[0], & \text{for } t \le 0 \\ \mu[n], & \text{for } 0 < t \in [n\rho, n\rho + \rho) \end{cases}$$

and

$$Z^\rho(t) = \begin{cases} 0, & \text{for } t < 0 \\ \rho \sum_{j=0}^{t/\rho - 1} Z[j], & \text{for } t \ge 0. \end{cases}$$

The following theorem shows that the estimated step size $\mu[n]$ of the adaptive step-size algorithm will spend nearly all of the time in an arbitrarily small neighborhood of the local minima of the cost function $\mathbf{E}\left\{\|\boldsymbol{e}^\mu[n]\|^2\right\}$. Recall that weak convergence is a generalization of convergence in distribution [24]. Suppose that X_n and X are vector-valued random variables. We say that X_n converges weakly to X iff for any bounded and continuous function $f(\cdot)$, $\mathbf{E}f(X_n) \to \mathbf{E}f(X)$. $\{X_n\}$ is said to be *tight* iff for each $\rho > 0$, there is a compact set K_ρ such that $P(X_n \in K_\rho) \geq 1-\rho$ for all n—roughly speaking, tightness is equivalent to boundedness in probability. These definitions have extensions for random variables living in a more general metric space. In weak convergence analysis of stochastic approximation algorithms, the most convenient function space to consider is $D(-\infty,\infty)$, which is the space of functions that are right continuous with left limits ("cadlag" functions). In the theorem to follow, $D^2(-\infty,\infty)$ refers to the function space $D(-\infty,\infty)\times D(-\infty,\infty)$.

Theorem 4: Assume that Conditions (C1), (C2), and (M1) are satisfied. Then $\{\mu^\rho(\cdot), Z^\rho(\cdot)\}$ is tight in $D^2(-\infty,\infty)$, and any weakly convergent subsequence has the limit $(\mu(\cdot), Z(\cdot))$ which is a solution of

$$\frac{d\mu}{dt} = g(\mu(t)) + z(t), \;\; \mu(0)=\mu[0]$$

where

$$g(\mu) \triangleq \mathbf{E}\left\{\boldsymbol{e}^\mu[n]^T \boldsymbol{D}^\mu[n]\right\} = -\frac{1}{2}\frac{\partial \mathbf{E}\left\{\boldsymbol{e}^\mu[n]^T\boldsymbol{e}^\mu[n]\right\}}{\partial \mu}.$$

Proof: Due to the boundedness of $\{\boldsymbol{Y}[n]\}$ and $\{\boldsymbol{\pi}[n]\}$, it is easily proved that $(\mu^\rho(\cdot), Z^\rho(\cdot))$ is tight in $D^2(-\infty,\infty)$. By Prohorov's theorem, we can extract convergent subsequences. Do so and still index the subsequence by ρ for simplicity. Denote the limit process by $(\mu(\cdot), Z(\cdot))$. We proceed to characterize the limit process.

By Corollary 1, $\{\boldsymbol{Y}[n]\}$ is irreducible and aperiodic, thus, by [7, pp. 167–168], it is ϕ-mixing with exponential mixing rate. Then this mixing property implies that for each m

$$\sum_{j=m}^{\infty} \left|\mathbf{E}_m\left\{\boldsymbol{Y}[j]\right\} - \boldsymbol{\pi}[j]\right| = \mathcal{O}(1)$$

where $\mathbf{E}_m$ denotes the conditional expectation on the past data up to time m. In addition, the process $\{\boldsymbol{Y}[n]\}$ is ergodic. The stationarity then implies that for any m, as $n\to\infty$

$$\frac{1}{n}\sum_{j=m}^{n+m-1} \mathbf{E}_m\left\{\boldsymbol{e}^\mu[j]^T\boldsymbol{Y}[j]\right\} \to g(\mu) \text{ in probability}$$

for each $\mu \in I_\mu = [\mu_-, \mu_+]$. Now all the conditions in Theorem 5.1 of [22] are verified. By virtue of that theorem, the desired result follows. □

The above theorem deals with large but still bounded t. A related result concerns the case when $\rho\to 0$, $n\to\infty$, and $\rho n\to\infty$; see [24].

Theorem 5: Assume the conditions in Theorem 4. Let $\{q_\rho\}$ denote a sequence of integers such that $q_\rho\to\infty$ as $\rho\to 0$ and $\rho q_\rho \to \infty$. Suppose that there is a unique local minimum μ^* of $\mathbf{E}\{||\boldsymbol{e}^\mu[n]||^2\}$ such that $\mu^* \in (\mu_-, \mu_+)$. Then $(\mu^\rho(\rho q_\rho + \cdot))$ converges weakly to μ^*.

C. Extension to Multiantenna Multipath Fading Channels

In the preceding sections, we have designed and analyzed discrete stochastic approximation algorithms for single-user and multiuser spreading code optimization for the simple synchronous CDMA system (1). In fact, the more complicated fading multipath CDMA channels with single or multiple receive antennas can be described by a model in the same form as (1) [43], that is, we have the following general signal model:

$$\boldsymbol{r}[\iota] = \tilde{\boldsymbol{S}}\boldsymbol{b}[\iota] + \boldsymbol{n}[\iota] \tag{39}$$

where $\tilde{\boldsymbol{S}}$ is a matrix comprising of the effective signature sequences (i.e., original signatures convolved with the channels). Furthermore, a linear MMSE decision on the ιth symbol of User 1 is made based on the output of the linear MMSE detector $\boldsymbol{w}_1^H\boldsymbol{r}[\iota]$, where $\boldsymbol{w}_1$ has the form

$$\boldsymbol{w}_1 = \boldsymbol{C}_1^{-1}\tilde{\boldsymbol{s}}_1 \tag{40}$$

where, as before

$$\boldsymbol{C}_1 \triangleq E\left\{\boldsymbol{r}_1[\iota]\boldsymbol{r}_1[\iota]^H\right\}, \qquad \text{with } \boldsymbol{r}_1[\iota] \triangleq \boldsymbol{r}[\iota] - b_1[\iota]\tilde{\boldsymbol{s}}_1.$$

Moreover, the composite signature waveform $\tilde{\boldsymbol{s}}_1$ of the desired user is determined by the original signature waveform of this user $\boldsymbol{s}_1$, and its channel response and antenna array response.

The discrete stochastic approximation Algorithms 1–4 can be used to optimize the spreading sequence in this case as well. The only modification that needs to be made is that after getting a candidate spreading code $\boldsymbol{s}_1$, we then compute the composite signature waveform by convolving $\boldsymbol{s}_1$ with its channel response. The composite signature waveform is then used in computing $X[n, \boldsymbol{s}_1^{(n)}]$, $X[n, \tilde{\boldsymbol{s}}_1^{(n)}]$ in Algorithm 1.

The convergence of the discrete stochastic approximation algorithms in fading multipath and multiantenna channels can be analyzed in the same way as that for the simple synchronous channel case, thanks to the following central limit theorem, which is the counterpart of Proposition 1 for complex-valued signals. The proof is given in Appendix B.

Proposition 2: Let $X_1(\tilde{\boldsymbol{s}}_1) = \tilde{\boldsymbol{s}}_1^H\boldsymbol{C}_1^{-1}\tilde{\boldsymbol{s}}_1$ and $\hat{\gamma}_1(\tilde{\boldsymbol{s}}_1) = \tilde{\boldsymbol{s}}_1^H\hat{\boldsymbol{C}}_1^{-1}\tilde{\boldsymbol{s}}_1$ with

$$\hat{\boldsymbol{C}}_1 = \frac{1}{M}\sum_{\iota=1}^{M}\boldsymbol{r}_1[\iota]\boldsymbol{r}_1[\iota]^H.$$

Then for large M, we have

$$\sqrt{M}\Big(\hat{\gamma}_1(\tilde{\boldsymbol{s}}_1) - X_1(\tilde{\boldsymbol{s}}_1)\Big) \sim \mathcal{N}\left(0, X_1^2(\tilde{\boldsymbol{s}}_1)\right). \tag{41}$$

V. Simulation Examples

In this section, we provide several simulation examples to demonstrate the performance of various discrete stochastic approximation algorithms developed in this paper. Both single-user and multiuser spreading code adaptations are considered.

A. Single-User Spreading Code Adaptation

Example 1: Static Synchronous CDMA Channel: We first consider a real-valued synchronous CDMA system in AWGN. The processing gain is $P=10$. The total number of users in the system is $K=8$. The spreading code of User 1 is to be optimized. The other users' spreading codes are randomly generated and kept fixed. The signal-to-noise ratio for each user is 10 dB. The window size for forming the sample covariance matrices $\hat{\boldsymbol{C}}_1^0$ and $\hat{\boldsymbol{C}}_1^1$ (see Algorithm 1) is chosen to be $M=50$. The performance of Algorithm 1 is illustrated in Fig. 2. In Fig. 2(a), the SINR of the chosen code as a function of iteration number n in one simulation run is plotted. In the same graph, we also plot the maximum, median, and minimum SINR among all 2^P possible spreading sequence for User 1. Moreover, the performance of a code optimization method proposed in [34], which quantizes the minimum eigenvector of the estimated covariance matrix of the received signal is also shown in the same figure. (Note that here

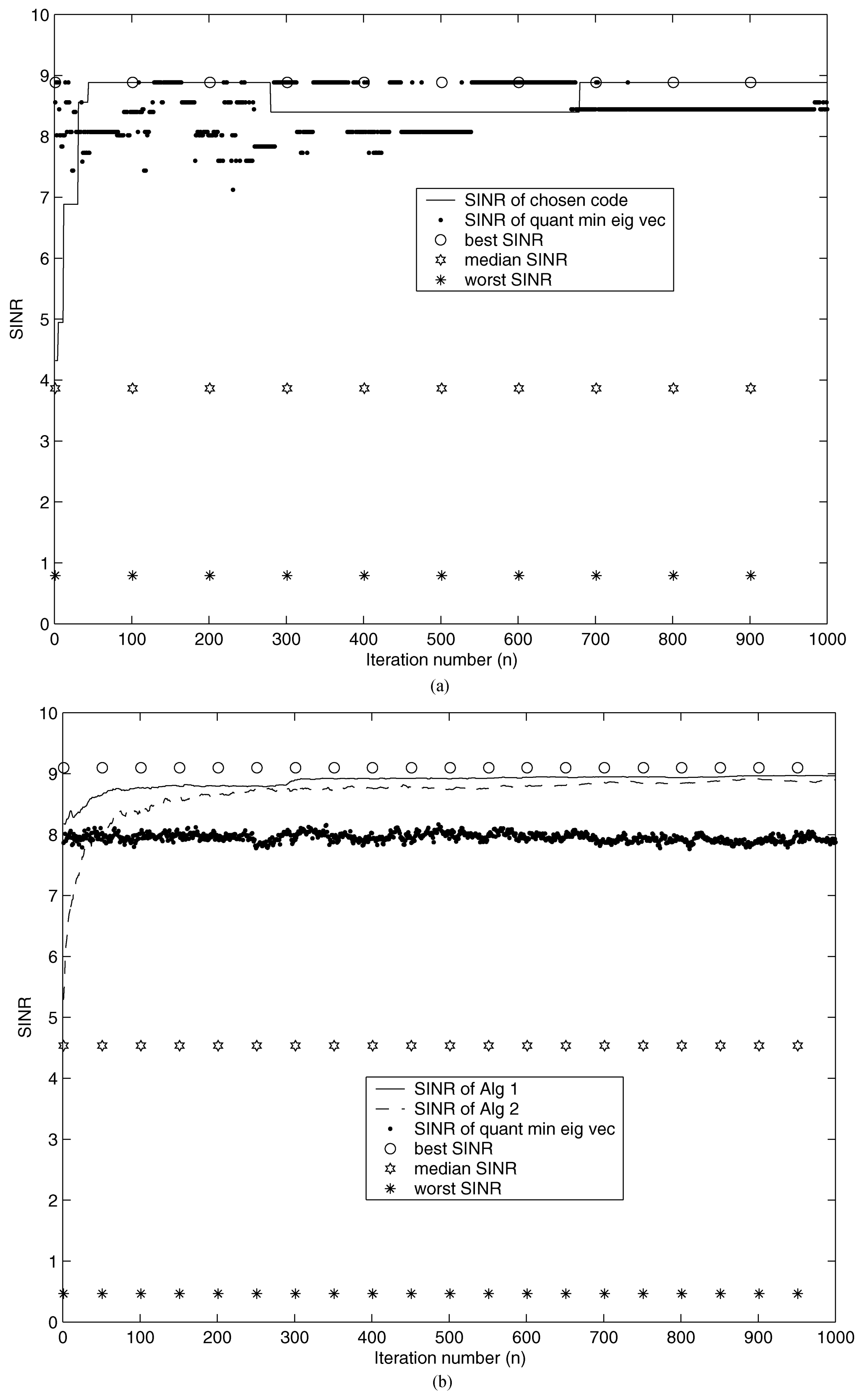


Fig. 2. Single-user code adaptation in synchronous CDMA system: SINR of the chosen code versus iteration number n. (a) One simulation run. (b) Average performance over 100 simulation runs.

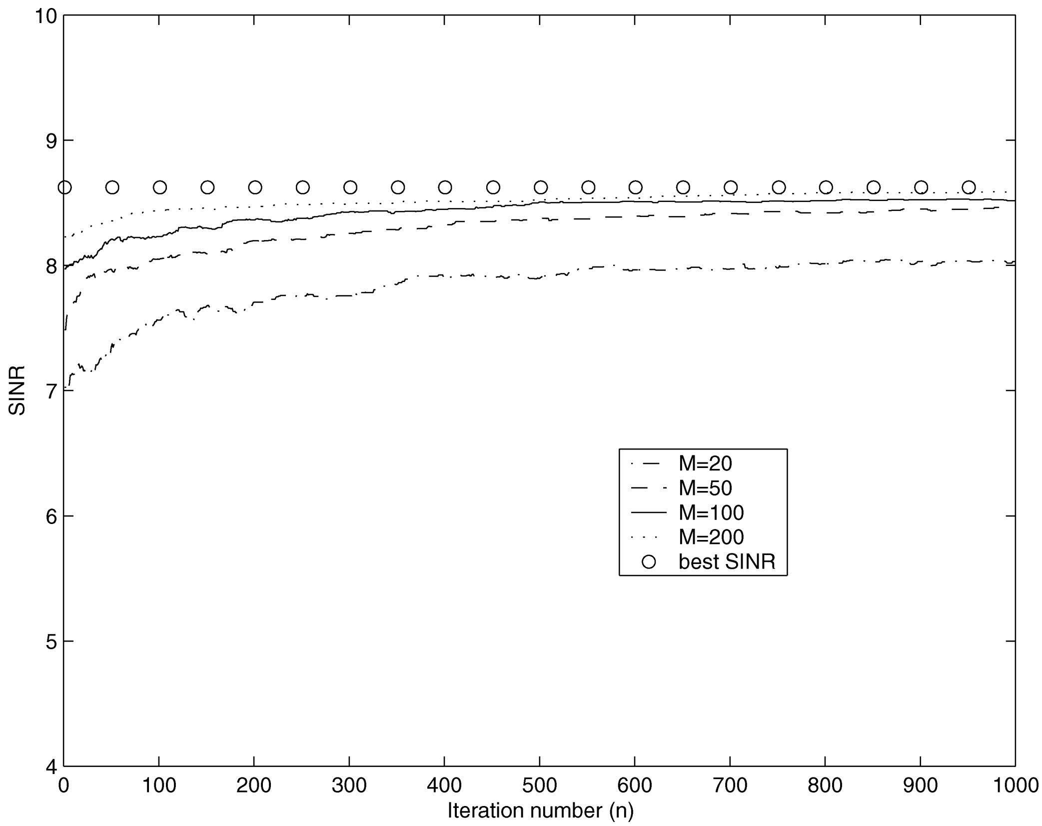


Fig. 3. Single-user code adaptation in synchronous CDMA system: the effect of the window size M.

the estimated covariance matrix is obtained accumulatively as the algorithm iterates, that is, at iteration n, the covariance matrix is based on the past Mn signal samples; whereas for the stochastic approximation algorithm, the matrix $\hat{\boldsymbol{C}}_1^0$ is based on only M received signal samples.) In Fig. 2(b), the average SINR performance of both schemes over 100 simulation runs is plotted. Also shown in the same figure is the performance of Algorithm 2. It is seen that the discrete stochastic approximation algorithm locks on the optimal code very quickly; it significantly outperforms the solution obtained by quantizing the continuous maximizer. Moreover, compared with the aggressive algorithm (i.e., Algorithm 1), the conservative algorithm (i.e., Algorithm 2) has a slower convergence rate and slightly inferior steady-state performance. In Fig. 3, we show the average SINR performance under different window sizes $M = 20, 50, 200$. Although the performance improves monotonically with the window size M, beyond certain value, e.g., $M = 50$, the performance gain from increasing M is diminishing.

Example 2: Static Multipath CDMA Channel: Next, we illustrate the performance of Algorithm 1 in a static multipath environment. The simulated system is the same as that in the previous example, except now each user's signal is subject to multipath distortion. The multipath channel of each user has three taps, each delayed by integer number of chip intervals. The complex path gain of each path is randomly generated and kept fixed throughout the simulation. The path gains of each user are normalized such that the received composite signature waveform has unit energy. The signal-to-noise ratio for each user is still 10 dB. The code adaptation performance is shown in Fig. 4 for both a single simulation run and the average of 100 runs. It is seen that similar to the synchronous case, the algorithm locks on the optimal code very quickly.

Example 3: Constant Step-Size Algorithm in Fading Multipath CDMA Channel: We demonstrate the tracking performance of the adaptive discrete stochastic approximation Algorithm 3 and its asynchronous version (32) in multipath fading channels. The simulated system is the same as that in the previous example, except now the channels are subject to multipath fading. We assume each channel tap remains the same over a period of $\tau = Mq$ symbol intervals, and follows a first-order autoregressive model on a time scale of τ

$$h_t = \alpha h_{t-1} + \beta\zeta \tag{42}$$

where $\zeta \sim \mathcal{N}_c(0,1)$, and α and β are related through $\beta = (1-\alpha^2)^{-\frac{1}{2}}$. In the simulations, we set $q = 100$, $\alpha = 0.9$, and the constant step size $\mu = 0.002$. The tracking performance of Algorithm 3 is illustrated in Fig. 5. We also ran the asynchronous version of Algorithm 3 given by (32)—the performance was virtually identical. The maximum, median, and minimum SINR as a function of time are also shown. Note that since the channels are time-varying now, these quantities need to computed for each channel realizations. It is seen that our code adaptation algorithm can closely track the optimal code under the time-varying channel conditions.

We next illustrate the performance of a variant of Algorithm 3, where (31) in Step 3 is replaced by a sign error LMS algorithm

$$\boldsymbol{\pi}[n+1] = \boldsymbol{\pi}[n] + \mu\,\mathrm{sign}\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]\right). \tag{43}$$

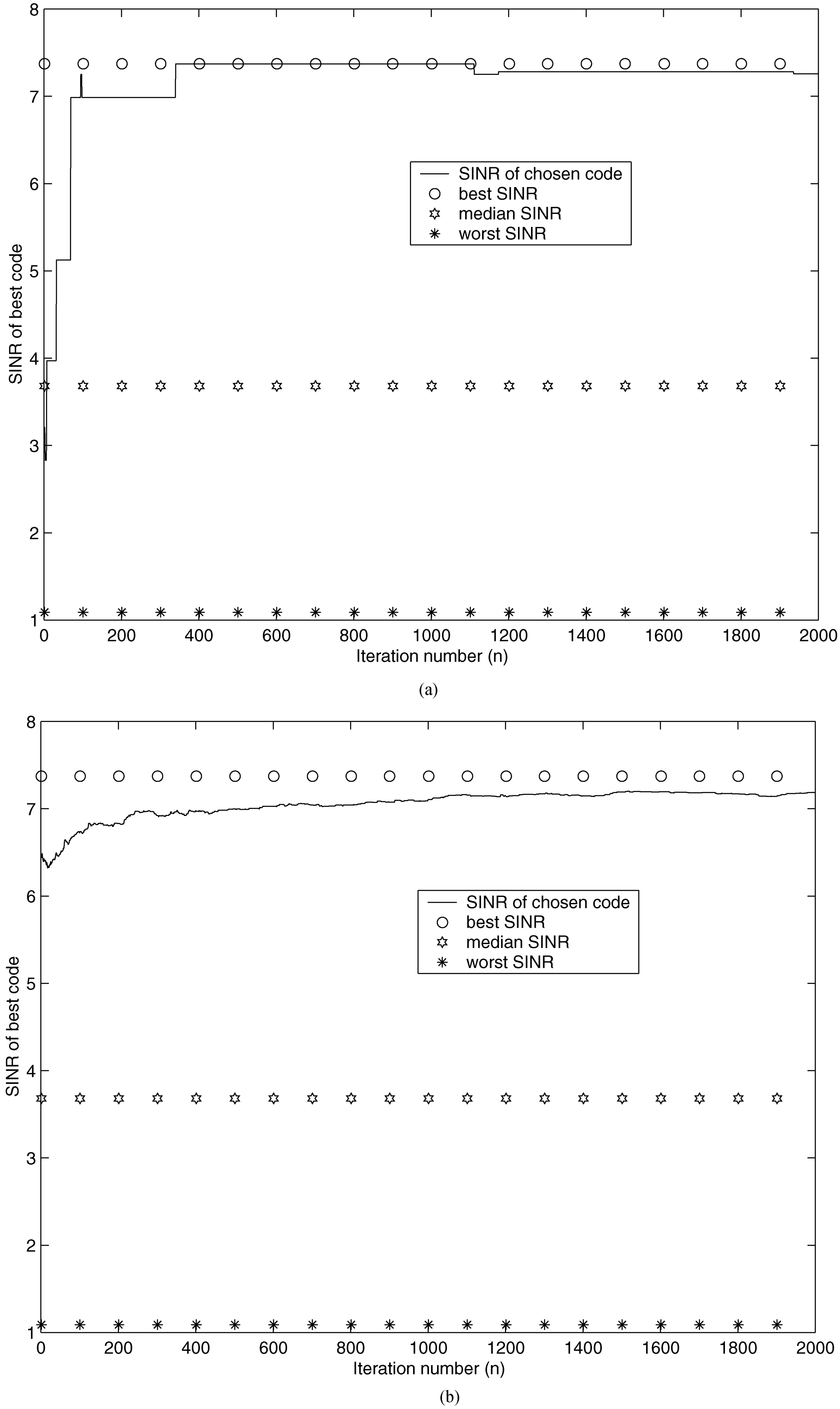


Fig. 4. Single-user code adaptation in static multipath CDMA system: SINR of the chosen code versus iteration number n. (a) One simulation run. (b) Average performance over 100 simulation runs.

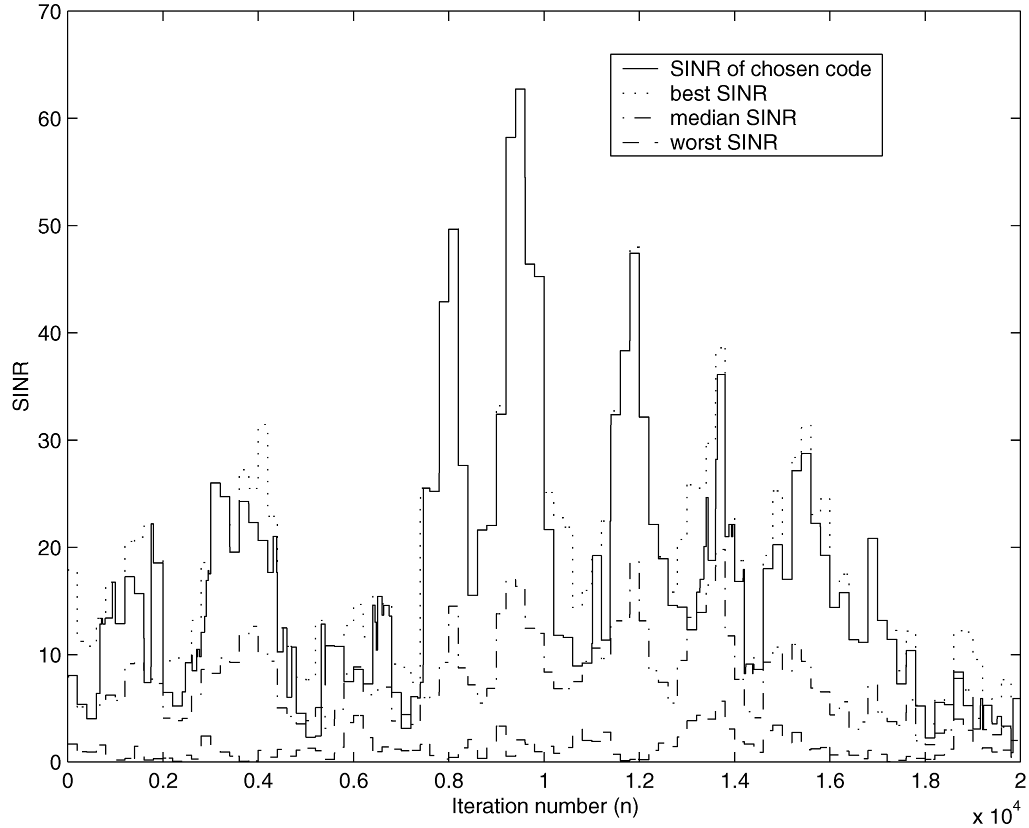


Fig. 5. Single-user code adaptation in multipath fading CDMA system with constant step size: SINR of the chosen code versus iteration number n.

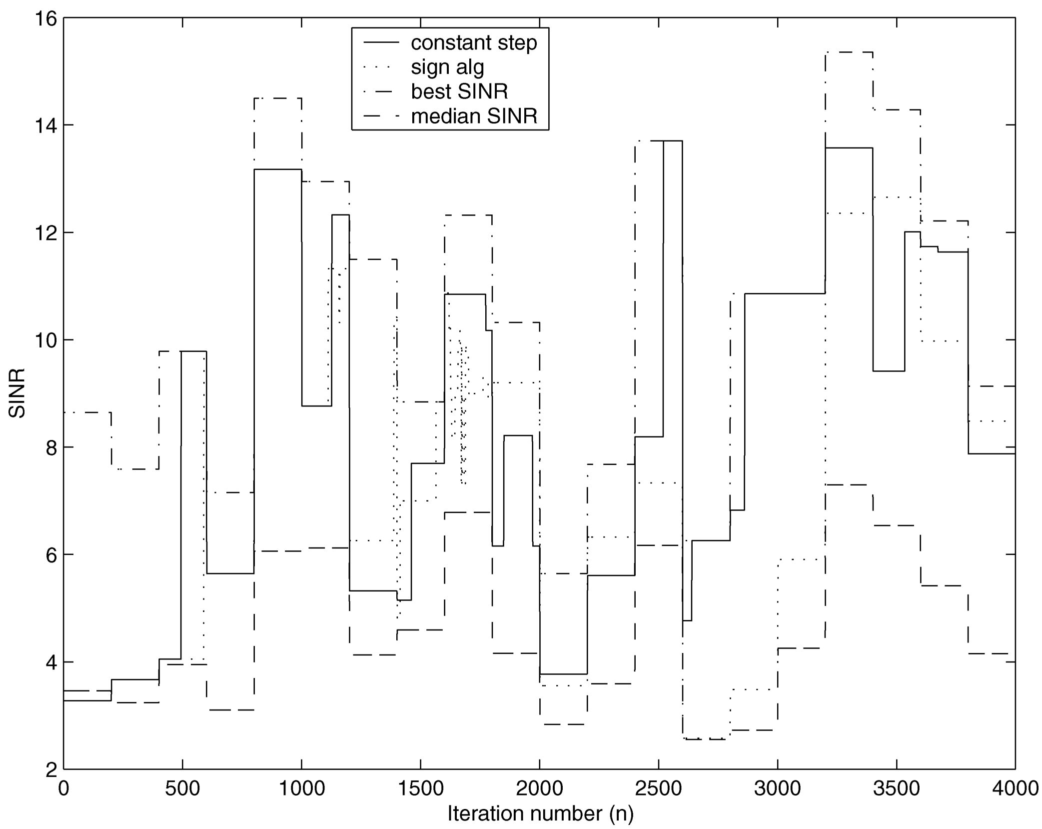


Fig. 6. Single-user code adaptation in multipath fading CDMA system with constant step size and the sign algorithm.

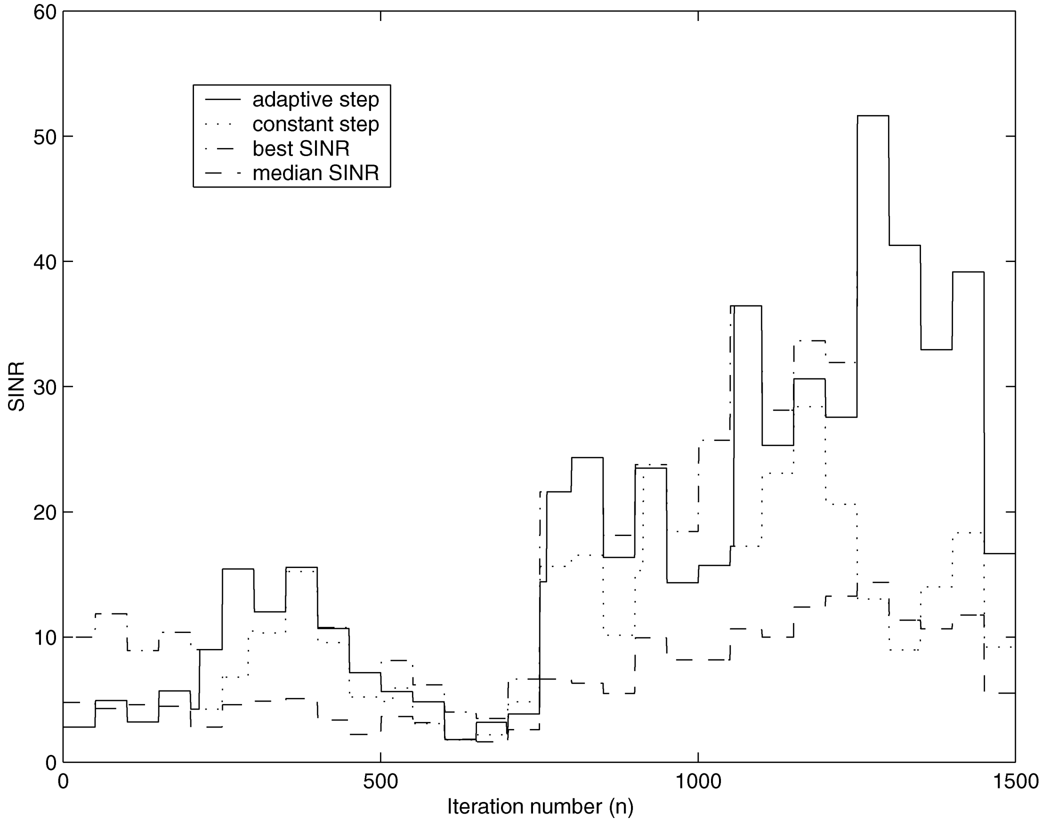


Fig. 7. Single-user code adaptation in multipath fading CDMA system with adaptive step size.

Such a sign algorithm can reduce the computational complexity if μ is chosen as $2^{-\ell}$ for some integer ℓ since the multiplications $\mu\text{sign}\,(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n])$ can be computed using bit shifts. The performance of the sign algorithm in tandem with the discrete stochastic optimization algorithm is shown in Fig. 6. It is seen that although there is a slight performance degradation compared with Algorithm 3, the sign algorithm can still efficiently track the channel variation and produce the optimal spreading codes.

Example 4: Adaptive Step-Size Algorithm in Multipath CDMA Channel: The performance of the discrete stochastic approximation Algorithm 4 in the same multipath fading CDMA system as in the previous example is shown in Fig. 7. The upper and lower bounds for the step size μ are set as $\mu_- = 0$ and $\mu_+ = 0.005$, respectively. It is seen from Fig. 7 that the algorithm with an adaptive step size converges much faster than the one with a constant step size.

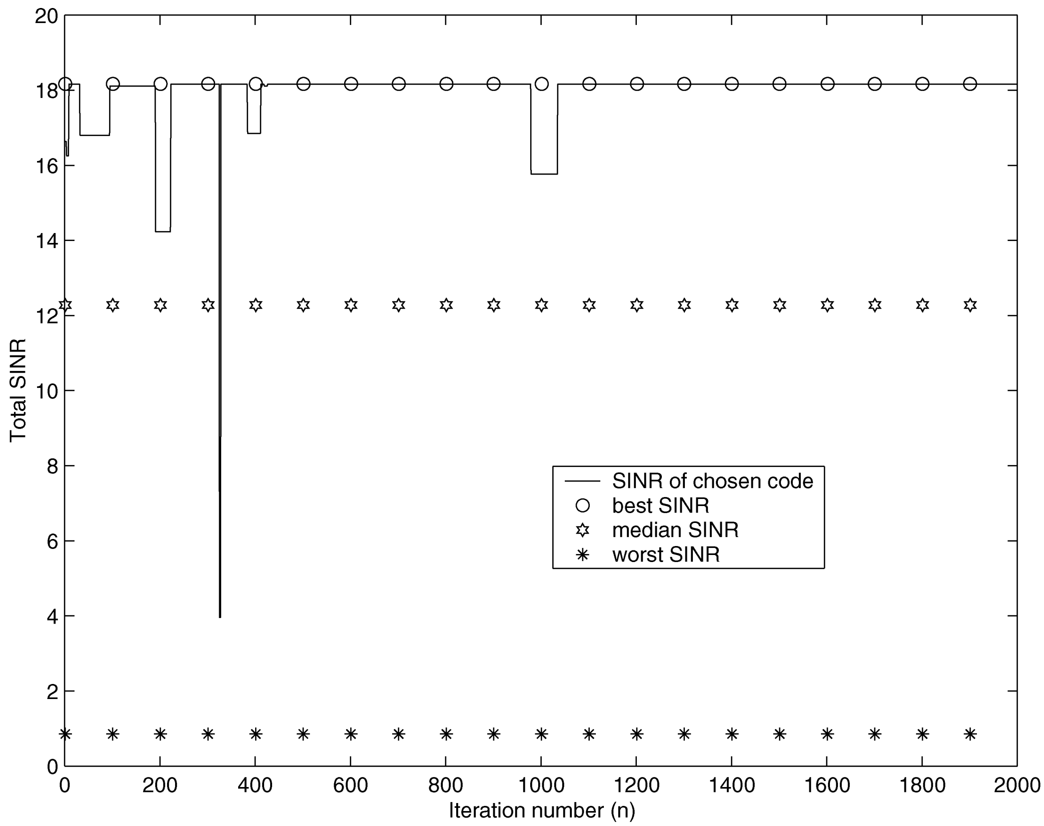
B. Two-User Spreading Code Adaptation

Examples 5, 6, and 7: Two-User Spreading Code Adaptation: We consider the performance discrete stochastic optimization algorithms for multiuser spreading code adaptation. We will primarily focus on adapting two users' codes, since it becomes computationally prohibitive to calculate the optimal codes when adapting more than two users. That is, in order to optimize $\tilde{K}$ users' codes, we need to calculate $2^{P\tilde{K}}$ possible SINR values corresponding to all possible combination of code selections. The simulated systems are the same as the ones for the single-user adaptation case discussed above. The processing gain is $P = 10$. There are $K = 8$ users in the channel, and the first two users' codes are to be optimized. The performance in static channels is shown in Figs. 8 and 9, for synchronous case and multipath case. It is seen that like in the single-user adaptation case, the convergence speed is quite fast. The tracking performance in fading multipath channels is shown in Fig. 10. Again the algorithm can effectively track the dynamic nature of the channels.

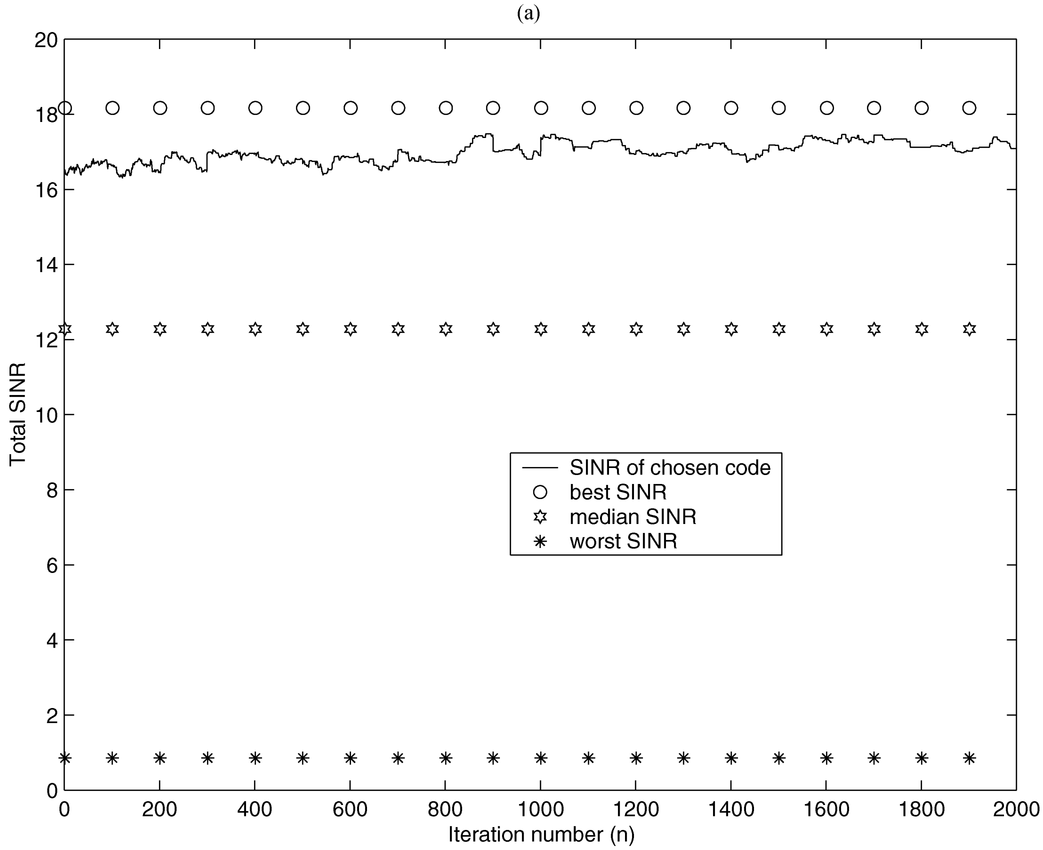
VI. Concluding Remarks

This paper presented discrete stochastic approximation algorithms for spreading code optimization and adaptation in CDMA systems. First, aggresive as well as conservative decreasing-step-size algorithms (Algorithms 1 and 2) were presented. Then two adaptive discrete stochastic approximation algorithms (Algorithms 3 and 4) were presented for tracking the optimal spreading code in a time-varying environment. A mean-square error tracking analysis of Algorithm 3 is given along with a weak convergence analysis of Algorithm 4. Numerical examples show that the algorithm performs well in multipath fading channels.

The algorithms and analysis presented in this paper are of independent interest in other applications that involve discrete stochastic optimization. In future work, we will consider joint power control and spreading code optimization. In such cases, the objective function is typically a utility function. We will also derive a weak convergence tracking analysis for the algorithms proposed. The advantage of such a weak convergence analysis is that an asymptotic Gaussian distribution can be obtained for

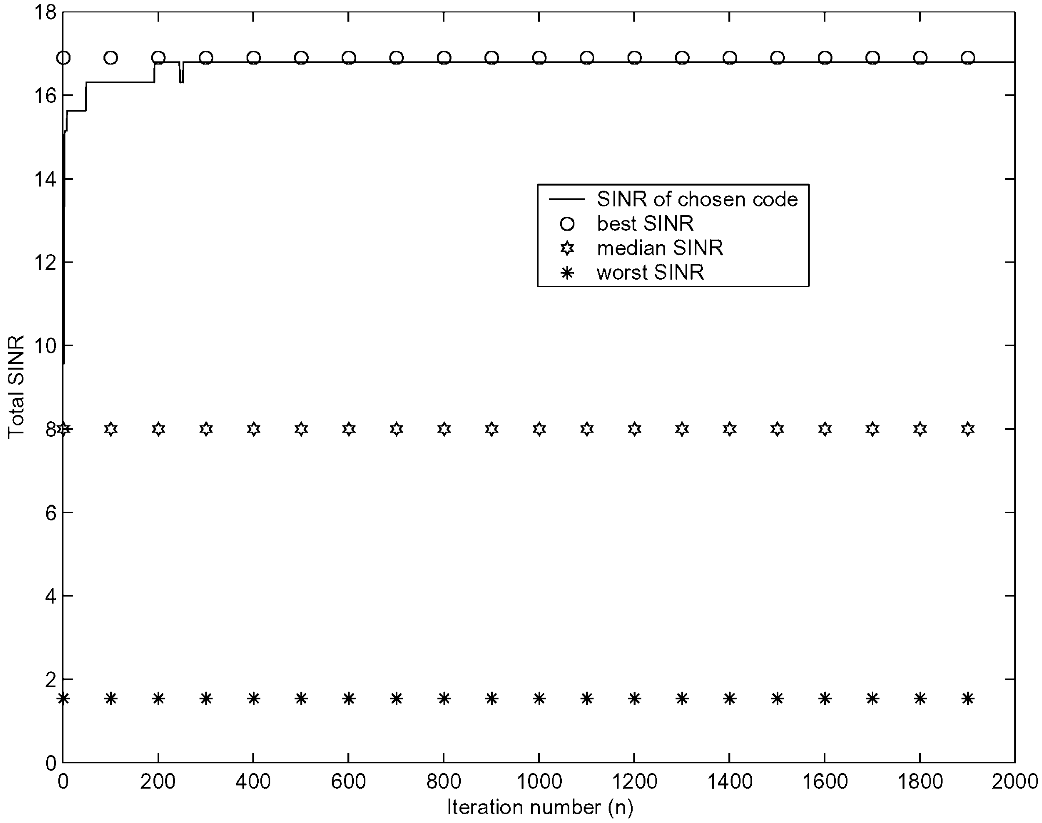


(a)

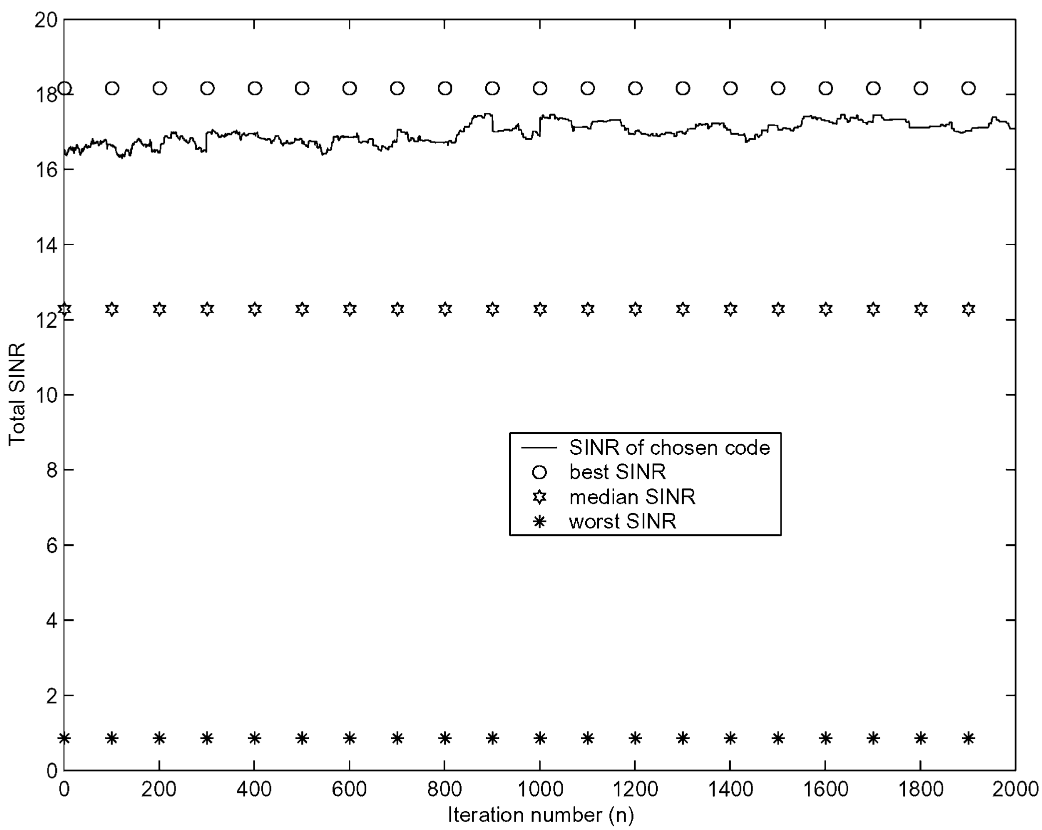


(b)

Fig. 8. Two-user code adaptation in static synchronous CDMA system: SINR of the chosen code versus iteration number n. (a) One simulation run. (b) Average performance over 100 simulation runs.



(a)



(b)

Fig. 9. Two-user code adaptation in static multipath CDMA system: SINR of the chosen code versus iteration number n. (a) One simulation run. (b) Average performance over 100 simulation runs.

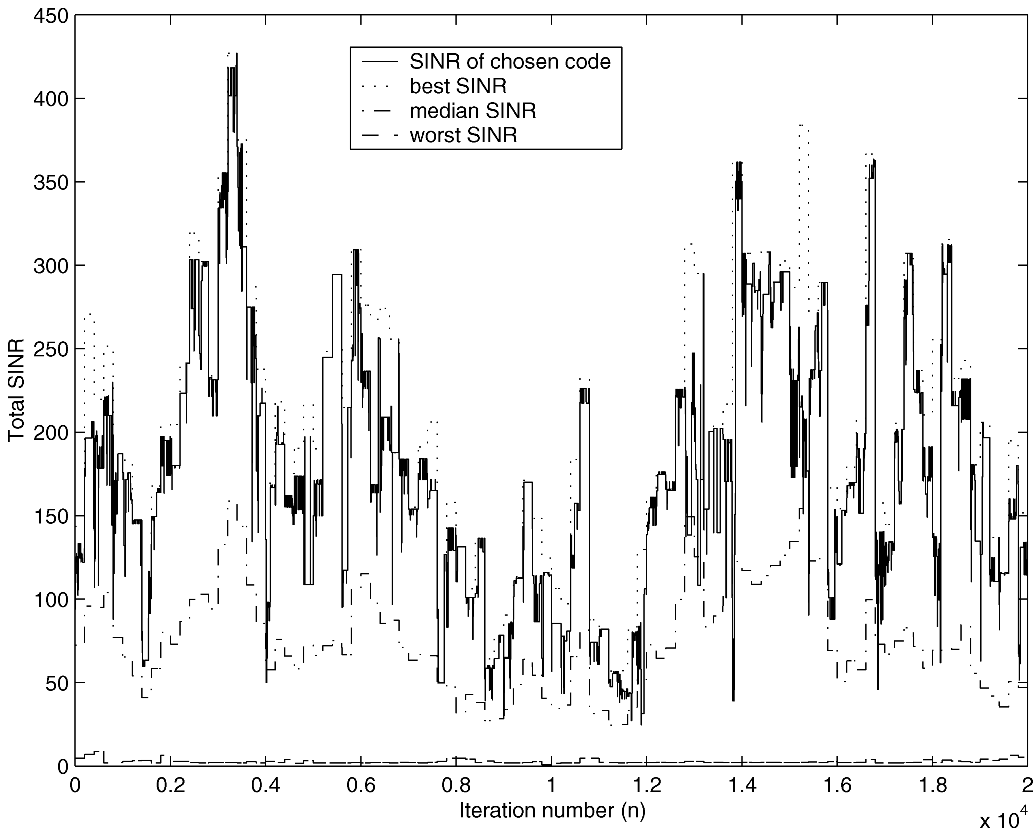


Fig. 10. Two-user code adaptation in multipath fading CDMA SINR of the chosen code versus iteration number n.

the tracking error resulting in explicit expressions for the error probability.

APPENDIX A
PROOF OF THEOREM 3 AND COROLLARY 2

Throughout this appendix, $||\cdot||$ will denote the Euclidean norm for vectors and the induced 2-norm for matrices. We will need the following two lemmas.

Lemma 2: Let $\mathcal{F}_n$ be the σ-algebra generated by

$$\{\boldsymbol{Y}[0],\ldots,\boldsymbol{Y}[n],\theta[0],\ldots,\theta[n+1]\}$$

and $\mathbf{E}_n$ the conditional expectation with respect to (w.r.t.) $\mathcal{F}_n$. Under hypermodel condition (M1)

$$\begin{aligned}||\mathbf{E}_n(\theta[n]-\theta[n+1])|| &= \mathcal{O}(\varepsilon)\\ ||\mathbf{E}_n(\boldsymbol{A}_{\theta[n]}-\boldsymbol{A}_{\theta[n+1]})|| &= \mathcal{O}(\varepsilon)\\ ||\mathbf{E}_n(\boldsymbol{\pi}_{\theta[n]}-\boldsymbol{\pi}_{\theta[n+1]})|| &= \mathcal{O}(\varepsilon). \end{aligned} \tag{44}$$

Proof: The proof mainly uses the Markov property. Note that $p^\varepsilon = \delta_{ij}+\varepsilon q_{ij}$. Consequently

$$\begin{aligned}||\mathbf{E}_n(\theta[n]-\theta[n+1])|| &= ||\mathbf{E}\{(\theta[n]-\theta[n+1]) \mid \theta[n]\}||\\ &= ||\sum_{i=1}^{2^P}\mathbf{E}\{(\boldsymbol{e}_i-\theta[n+1]) \mid \theta[n]=\boldsymbol{e}_i\}I(\theta[n]=\boldsymbol{e}_i)||\\ &= ||\sum_{i=1}^{2^P}\{\boldsymbol{e}_i-\sum_{j=1}^{2^P}\boldsymbol{e}_j p_{ij}^\varepsilon\}I(\theta[n]=\boldsymbol{e}_i)||\\ &\le \varepsilon\sum_{i=1}^{2^P}||\sum_{j=1}^{2^P}\boldsymbol{e}_j q_{ij}||.\end{aligned}$$

Thus, $||\mathbf{E}_n(\theta[n]-\theta[n+1])|| = \mathcal{O}(\varepsilon)$. Similar argument yields that

$$\begin{aligned}||\mathbf{E}_n(\boldsymbol{\pi}_{\theta[n]}-\boldsymbol{\pi}_{\theta[n+1]})|| &= ||\mathbf{E}\{(\boldsymbol{\pi}_{\theta[n]}-\boldsymbol{\pi}_{\theta[n+1]}) \mid \theta[n]\}||\\ &= ||\sum_{i=1}^{2^P}\mathbf{E}\{(\boldsymbol{\pi}_{\boldsymbol{e}_i}-\boldsymbol{\pi}_{\theta[n+1]}) \mid \theta[n]=\boldsymbol{e}_i\}I(\theta[n]=\boldsymbol{e}_i)||\\ &= ||\sum_{i=1}^{2^P}\{\boldsymbol{\pi}_{\boldsymbol{e}_i}-\sum_{j=1}^{2^P}\boldsymbol{\pi}_{\boldsymbol{e}_j} p_{ij}^\varepsilon\}I(\theta[n]=\boldsymbol{e}_i)||\\ &\le \varepsilon\sum_{i=1}^{2^P}||\sum_{j=1}^{2^P}\boldsymbol{\pi}_{\boldsymbol{e}_j} q_{ij}||.\end{aligned}$$

So $||\mathbf{E}_n(\boldsymbol{\pi}_{\theta[n]}-\boldsymbol{\pi}_{\theta[n+1]})|| = \mathcal{O}(\varepsilon)$. Exactly the same argument yields the second inequality in (44). □

Lemma 3: Suppose p_i and q_i for $i=1,\ldots,2^P$ are nonnegative real numbers. Then $|\max_i p_i - \max_j q_j| \le \max_l |p_l - q_l|$.

Proof: For any fixed m,

$$\max_i p_i - \max_j q_j \le \max_l p_l - q_m$$

which implies

$$\max_i p_i - \max_j q_j \le \max_l (p_l - q_l) \le \max_l |p_l - q_l|.$$

A similar reasoning yields

$$\max_j q_j - \max_i p_i \le \max_l |p_l - q_l|. \qquad \square$$

Proof of Theorem 3

Since $\{\boldsymbol{Y}[n]\}$ has Markovian dynamics, we follow a similar approach to [4, pp. 246–247] where mean-square convergence analysis of general stochastic approximation algorithms is presented. However, the analysis in [4] does not deal with hypermodel dynamics, and does not consider the finite-state Markov chain case. In the finite-state Markov chain case we consider here, the impenetrable technicalities of verifying bounded moments and stability which make stochastic approximation proofs inaccessible to engineers do not arise.

Define the tracking error $\tilde{\boldsymbol{\pi}}[n+1] \triangleq \boldsymbol{\pi}[n+1] - \boldsymbol{\pi}_{\theta[n+1]}$. From (31), this evolves according to

$$\tilde{\boldsymbol{\pi}}[n+1] = \tilde{\boldsymbol{\pi}}[n] + \mu\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]\right) + \boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}. \quad (45)$$

We decompose the term $\mu\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]\right)$ in (45) as follows:

$$\mu\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}[n]\right) = \mu\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}_{\theta[n+1]}\right) + \mu(\boldsymbol{\pi}_{\theta[n+1]} - \boldsymbol{\pi}_{\theta[n]}) + \mu(\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}[n]). \quad (46)$$

The simplest proofs of stochastic approximation algorithms deal with martingale processes (which includes i.i.d. signals as a special case). However, because in our case $\boldsymbol{Y}[n]$ has Markovian dynamics, we will need to express it in terms of martingales. The so called "Poisson's equation" approach [4, pp. 216–217] allows us to give a martingale decomposition of $\boldsymbol{Y}[n+1] - \boldsymbol{\pi}_{\theta[n+1]}$ in terms of the term F_n (see (49)) as will be shown later. Since the Markov chain $\{\boldsymbol{Y}[n]\}$ parameterized by $\theta[n]$ is geometrically ergodic (i.e., irreducible and aperiodic since we are dealing with a finite-state space), there exists a process $\{\boldsymbol{\nu}_{\theta[n]}\}$ which is a solution to the following Poisson's equation [4]:

$$\boldsymbol{Y}[n+1] - \boldsymbol{\pi}_{\theta[n+1]} = \boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1]) - \boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1]). \quad (47)$$

The solution $\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1])$ to Poisson's (47) is explicitly given by

$$\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1]) = \sum_{i=0}^{\infty}\left[\boldsymbol{A}^i_{\theta[n+1]} - \mathbf{1}\boldsymbol{\pi}^T_{\theta[n+1]}\right]\boldsymbol{Y}[n+1]. \quad (48)$$

Consider now the term $\mu(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}_{\theta[n+1]})$ in (46). Using Poisson's equation yields

$$\begin{aligned}
&\mu\left(\boldsymbol{Y}[n+1] - \boldsymbol{\pi}_{\theta[n+1]}\right)\\
&\quad = \mu\left(\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1]) - \boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1])\right)\\
&\quad = \mu\left(\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1]) - \boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n])\right)\\
&\qquad + \mu\left(\boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n]) - \boldsymbol{A}^T_{\theta[n]}\boldsymbol{\nu}_{\theta[n]}(\boldsymbol{Y}[n])\right)\\
&\qquad + \mu\left(\boldsymbol{A}^T_{\theta[n]}\boldsymbol{\nu}_{\theta[n]}(\boldsymbol{Y}[n]) - \boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n+1])\right)\\
&\quad = F_{n+1} + G_{n+1} + H_{n+1}.
\end{aligned} \quad (49)$$

Then from (45) and the (49), the tracking error $\tilde{\boldsymbol{\pi}}[n+1]$ evolves according to

$$\tilde{\boldsymbol{\pi}}[n+1] = (1-\mu)\tilde{\boldsymbol{\pi}}[n] + F_{n+1} + G_{n+1} + H_{n+1} + L_{n+1} \quad (50)$$

where

$$L_{n+1} = (1-\mu)(\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}). \quad (51)$$

Squaring (50) yields

$$\begin{aligned}
&\|\tilde{\boldsymbol{\pi}}[n+1]\|^2\\
&\quad = (1-\mu)^2\|\tilde{\boldsymbol{\pi}}[n]\|^2 + \|F_{n+1} + G_{n+1} + H_{n+1} + L_{n+1}\|^2\\
&\qquad + 2(1-\mu)\tilde{\boldsymbol{\pi}}^T[n](F_{n+1} + G_{n+1} + H_{n+1} + L_{n+1})\\
&\quad \le (1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\|^2 + 4(\|F_{n+1}\|^2 + \|G_{n+1}\|^2\\
&\qquad + \|H_{n+1}\|^2 + \|L_{n+1}\|^2)\\
&\qquad + 2(1-\mu)\tilde{\boldsymbol{\pi}}^T[n](F_{n+1} + G_{n+1} + H_{n+1} + L_{n+1}).
\end{aligned}$$

Let $\mathcal{F}_n$ denote the sigma algebra generated by

$$\{\boldsymbol{Y}[0], \ldots, \boldsymbol{Y}[n], \theta[0], \ldots, \theta[n+1]\}$$

and define $\mathbf{E}_n$ as conditional expectation w.r.t. $\mathcal{F}_n$. Taking conditional expectations $\mathbf{E}_n$, using norms on both sides of the above inequality, and noting that $\tilde{\boldsymbol{\pi}}[n]$ is $\mathcal{F}_n$-measurable, we have (52) at the bottom of the page. Since F_n is an $\mathcal{F}_n$-martingale increment process, $\mathbf{E}_n\{F_{n+1}\} = 0$. Also by definition (49)

$$\begin{aligned}
&\mathbf{E}_n\{G_{n+1}\}\\
&= \mathbf{E}_n\left\{\mu\boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n]) - \mu\boldsymbol{A}^T_{\theta[n]}\boldsymbol{\nu}_{\theta[n]}(\boldsymbol{Y}[n])\right\}.
\end{aligned} \quad (53)$$

But from (48), we have

$$\boldsymbol{A}^T_{\theta[n+1]}\boldsymbol{\nu}_{\theta[n+1]}(\boldsymbol{Y}[n]) = \sum_{i=1}^{\infty}\left[\boldsymbol{A}^i_{\theta[n+1]} - \mathbf{1}\boldsymbol{\pi}^T_{\theta[n+1]}\right]\boldsymbol{Y}[n].$$

Hence, (53) yields

$$\mathbf{E}_n\{G_{n+1}\} = \mu\sum_{i=1}^{\infty}\left(\boldsymbol{A}^i_{\theta[n+1]} - \boldsymbol{A}^i_{\theta[n]}\right)^T\boldsymbol{Y}[n].$$

Finally, using Lemma 2 we have

$$\|\mathbf{E}_n\{G_{n+1}\}\| = \mathcal{O}(\mu\varepsilon).$$

Similarly, using Lemma 2 we have

$$\begin{aligned}
\|\mathbf{E}_n\{L_{n+1}\}\| &= (1-\mu)\|\boldsymbol{\pi}_{\theta[n]} - \mathbf{E}_n\boldsymbol{\pi}_{\theta[n+1]}\| = \mathcal{O}(\varepsilon)\\
\|\mathbf{E}_n\{H_{n+1}\}\| &= \mathcal{O}(\mu\varepsilon).
\end{aligned}$$

In addition

$$\begin{aligned}
&\mathbf{E}_n\|\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}\|^2\\
&\quad = \boldsymbol{\pi}^T_{\theta[n]}\mathbf{E}_n(\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}) - \mathbf{E}_n\boldsymbol{\pi}^T_{\theta[n+1]}(\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}).
\end{aligned}$$

$$\begin{aligned}
\mathbf{E}_n\left\{\|\tilde{\boldsymbol{\pi}}[n+1]\|^2\right\} \le{}& (1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\|^2 + 2(1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\mathbf{E}_n\{F_{n+1}\}\|\\
&+ 2(1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\|(\|\mathbf{E}_n\{G_{n+1}\} + \mathbf{E}_n\{H_{n+1}\} + \mathbf{E}_n\{L_{n+1}\}\|)\\
&+ 4\mathbf{E}_n\left\{\|F_{n+1}\|^2 + \|G_{n+1}\|^2\|H_{n+1}\|^2 + \|L_{n+1}\|^2\right\}.
\end{aligned} \quad (52)$$

The first term on the right-hand side is $\mathcal{O}(\varepsilon)$. As for the second term, we have

$$\begin{aligned}&\mathbf{E}_n \boldsymbol{\pi}^T_{\theta[n+1]}(\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]})\\ &= \sum_{i=1}^{2^P} \mathbf{E}\{\boldsymbol{\pi}^T_{\theta[n+1]}(\boldsymbol{\pi}_{\boldsymbol{e}_i} - \boldsymbol{\pi}_{\theta[n+1]}) \mid \theta[n] = \boldsymbol{e}_i\} I(\theta[n] = \boldsymbol{e}_i)\\ &= \sum_{i=1}^{2^P}\sum_{j=1}^{2^P} \boldsymbol{\pi}^T_{e_j}(\boldsymbol{\pi}_{\boldsymbol{e}_i} - \boldsymbol{\pi}_{\boldsymbol{e}_j}) p^{\varepsilon}_{ij} I(\theta[n] = \boldsymbol{e}_i) = \mathcal{O}(\varepsilon).\end{aligned}$$

Thus, we have

$$\mathbf{E}_n \|L_{n+1}\|^2 \le \mathbf{E}_n \|\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}_{\theta[n+1]}\|^2 = \mathcal{O}(\varepsilon).$$

It is also easily shown using similar arguments that

$$\begin{aligned}\mathbf{E}_n\{\|F_{n+1}\|^2\} &= \mathcal{O}(\mu^2)\\ \mathbf{E}_n\{\|G_{n+1}\|^2\} &= \mathcal{O}(\mu^2)\\ \mathbf{E}_n\{\|H_{n+1}\|^2\} &= \mathcal{O}(\mu^2).\end{aligned}$$

Thus, from (52) it follows that

$$\begin{aligned}\mathbf{E}_n\left\{\|\tilde{\boldsymbol{\pi}}[n+1]\|^2\right\} &\le (1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\|^2\\ &\quad + 2(1-\mu)\|\tilde{\boldsymbol{\pi}}[n]\| \Big[\mathcal{O}(\mu\varepsilon) + \mathcal{O}(\varepsilon)\Big] + \mathcal{O}(\mu^2 + \varepsilon).\end{aligned}$$

Taking expectations on both sides and using the smoothing property of conditional expectations we have

$$\begin{aligned}\mathbf{E}\left\{\|\tilde{\boldsymbol{\pi}}[n+1]\|^2\right\} &\le (1-\mu)\mathbf{E}\left\{\|\tilde{\boldsymbol{\pi}}[n]\|^2\right\}\\ &\quad + \mathcal{O}(\mu\varepsilon + \varepsilon)\mathbf{E}\left\{\|\tilde{\boldsymbol{\pi}}[n]\|\right\} + \mathcal{O}(\mu^2 + \varepsilon). \quad (54)\end{aligned}$$

However, since $\boldsymbol{\pi}_{\theta[n]}$ and $\boldsymbol{\pi}[n]$ are probability vectors

$$\|\tilde{\boldsymbol{\pi}}[n]\| = \|\boldsymbol{\pi}_{\theta[n]} - \boldsymbol{\pi}[n]\| = \mathcal{O}(1) \text{ a.s.}$$

which, in turn, implies that $\mathbf{E}\|\tilde{\boldsymbol{\pi}}[n]\| = \mathcal{O}(1)$. Thus, we have

$$\mathbf{E}\left\{\|\tilde{\boldsymbol{\pi}}[n+1]\|^2\right\} \le (1-\mu)\mathbf{E}\left\{\|\tilde{\boldsymbol{\pi}}[n]\|^2\right\} + \mathcal{O}(\mu^2 + \varepsilon). \quad (55)$$

Iterating on this inequality yields

$$\mathbf{E}\|\tilde{\boldsymbol{\pi}}[n+1]\|^2 \le (1-\mu)^n \mathbf{E}\|\tilde{\boldsymbol{\pi}}[0]\|^2 + \sum_{j=0}^{n}(1-\mu)^{n-j}\mathcal{O}(\mu^2+\varepsilon).$$

By taking n sufficiently large we can make $(1-\mu)^n \le \mathcal{O}(\mu)$. Then the above inequality yields that for sufficiently large n, $\mathbf{E}\|\tilde{\boldsymbol{\pi}}[n+1]\|^2 \le \mathcal{O}(\mu + \varepsilon/\mu)$. □

Mean Square Convergence of Asynchronous Implementation (32): A similar analysis can be carried out to the normalized version asynchronous algorithm (32). To illustrate, denote the ith components of $\bar{\boldsymbol{\pi}}[n]$ and $\boldsymbol{Y}[n]$ by $\bar{\boldsymbol{\pi}}^i[n]$ and $\boldsymbol{Y}^i[n]$, respectively. Replacing $\hat{\boldsymbol{\pi}}[n]$ replaced by $\bar{\boldsymbol{\pi}}[n]$ in (32), the corresponding scheme can be written in a component form as

$$\bar{\boldsymbol{\pi}}^i[n+1] = \bar{\boldsymbol{\pi}}^i[n] + \mu 1_{\{\boldsymbol{Y}[n+1]=\boldsymbol{e}_i\}}(\boldsymbol{Y}^i[n+1] - \bar{\boldsymbol{\pi}}^i[n]).$$

To further simplify, let us fix a component i and denote $\pi_n = \bar{\boldsymbol{\pi}}^i[n]$ and $y_n = \boldsymbol{Y}^i[n+1]$. Define $\tau_0 = 0$, $\tau_1 = \min\{\ell : \boldsymbol{Y}[\ell] = \boldsymbol{e}_i\}$, and $\tau_{k+1} = \min\{\ell > \tau_k : \boldsymbol{Y}[\ell] = \boldsymbol{e}_i\}$. This is an increasing sequence of stopping times. In terms of this sequence, the recursion can be written as

$$\pi_{\tau_{k+1}} = \pi_{\tau_k} + \mu(y_{\tau_k} - \pi_{\tau_k}). \quad (56)$$

Note that for any positive integer n, there is a k such that $\tau_k \le n < \tau_{k+1}$. Thus, we can work with (56) and proceed as in the proof of previous theorem with the use of the stopping times (strong Markov property). We can show

$$E|\bar{\boldsymbol{\pi}}^i[n]|^2 = \mathcal{O}(\mu + \varepsilon/\mu), \text{ for each } i = 1, 2 \ldots, 2^P$$

hence, $E\|\bar{\boldsymbol{\pi}}[n]\|^2 = \mathcal{O}(\mu + \varepsilon/\mu)$ as desired.

Proof of Corollary 2

The estimate of the maximum generated by the discrete stochastic approximation algorithm at time n is

$$\hat{\boldsymbol{s}}^{(n)} = \arg\max_j \pi[n, j].$$

The actual maximum at time n is $\boldsymbol{s}^*_{\theta[n]} = \arg\max_i \boldsymbol{\pi}_{\theta[n]}(i)$. Define the error event E as (where $I(\cdot)$ denotes the indicator function)

$$E \triangleq \left\{I\left(\arg\max_i \boldsymbol{\pi}_{\theta[n]}(i) \ne \arg\max_j \pi[n,j]\right)\right\}.$$

Then clearly the complement event

$$\bar{E} = \left\{I\left(\arg\max_i \boldsymbol{\pi}_{\theta[n]}(i) = \arg\max_j \pi[n,j]\right)\right\}$$

satisfies the equation at the bottom of the page, where

$$L \le \min_{i,j} |\boldsymbol{\pi}_{\theta[n]}(i) - \pi[n,j]| \quad (57)$$

is a positive constant.

Then the probability of no error $P(\bar{E})$ is

$$\begin{aligned}P(\bar{E}) &= P(\arg\max_i \boldsymbol{\pi}_{\theta[n]}(i) = \arg\max_j \pi[n,j])\\ &> P(|\max_i \boldsymbol{\pi}_{\theta[n]}(i) - \max_j \pi[n,j]| \le L)\end{aligned}$$

for any sufficiently small positive number L. Using the preceding equation and Lemma 3 we have

$$\begin{aligned}P(E) &\le P(|\max_i \boldsymbol{\pi}_{\theta[n]}(i) - \max_j \pi[n,j]| > L)\\ &\le P(\max_i |\boldsymbol{\pi}_{\theta[n]}(i) - \pi[n,i]| > L). \quad (58)\end{aligned}$$

Choosing $\mu = \mathcal{O}(\sqrt{\varepsilon})$ and applying Chebyshev's inequality to (36) yields for any i

$$P(|\boldsymbol{\pi}_{\theta[n]}(i) - \pi[n,i]| > L) \le \frac{1}{L^2}\mathcal{O}(\mu).$$

Thus, (58) yields

$$P(\max_i |\boldsymbol{\pi}_{\theta[n]}(i) - \pi[n,i]| > L) \le \frac{1}{L^2}\mathcal{O}(\mu). \quad (59)$$

It only remains to pick a sufficiently small L. Choose $L = \mu^\alpha$ where $0 < \alpha < \frac{1}{2}$ is arbitrary. It is clear that for sufficiently small μ, L satisfies (57). Then (59) yields $P(E) \le \mu^{1-2\alpha}$. □

$$\begin{aligned}\bar{E} &\supseteq \left\{I\left(|\max_i \boldsymbol{\pi}_{\theta[n]}(i) - \max_j \pi[n,j]| \le \min_{i,j} |\boldsymbol{\pi}_{\theta[n]}(i) - \pi[n,j]|\right)\right\}\\ &\supseteq \left\{I\left(|\max_i \boldsymbol{\pi}_{\theta[n]}(i) - \max_j \pi[n,j]| \le L\right)\right\}\end{aligned}$$

$$\upsilon^2 = \mathbf{E}\left\{(\Delta X_1)^2\right\} = \boldsymbol{w}_1^H \mathbf{E}\left\{\Delta \boldsymbol{C}_1 \boldsymbol{w}_1 \boldsymbol{w}_1^H \Delta \boldsymbol{C}_1^H\right\} \boldsymbol{w}_1 = \frac{1}{M}\left[\left(\boldsymbol{w}_1^H \boldsymbol{C}_1 \boldsymbol{w}_1\right)\boldsymbol{C}_1 + \mu\left(\tilde{\boldsymbol{S}}_1 \tilde{\boldsymbol{S}}_1^T\right)\boldsymbol{w}_1^* \boldsymbol{w}_1^T \left(\tilde{\boldsymbol{S}}_1 \tilde{\boldsymbol{S}}_1^T\right)^* + \nu \tilde{\boldsymbol{S}}_1 \boldsymbol{D}_1 \tilde{\boldsymbol{S}}_1^H\right] \quad (65)$$

$$\text{with} \quad \boldsymbol{D} \triangleq diag\left\{|\boldsymbol{s}_2^H \boldsymbol{w}_1|^2, \ldots, |\boldsymbol{s}_K^H \boldsymbol{w}_1|^2\right\}. \quad (66)$$

APPENDIX B
PROOF OF PROPOSITION 2

The following lemma can be proved following along the same lines of argument as that of the proof of Lemma 1 in [15].

Lemma 4: Define $\Delta \boldsymbol{C}_1 \triangleq \hat{\boldsymbol{C}}_1 - \boldsymbol{C}_1$, where $\boldsymbol{C}_1$ and $\hat{\boldsymbol{C}}_1$ are give by

$$\boldsymbol{C}_1 = \mathbf{E}\left\{\boldsymbol{r}_1[i]\boldsymbol{r}[i]_1^H\right\} = \tilde{\boldsymbol{S}}_1 \tilde{\boldsymbol{S}}_1^H + \sigma^2 \boldsymbol{I} \quad (60)$$

$$\hat{\boldsymbol{C}}_1 = \frac{1}{M}\sum_{i=0}^{M-1} \boldsymbol{r}_1[i]\boldsymbol{r}_1[i]^H. \quad (61)$$

Then $\sqrt{M}\Delta \boldsymbol{C}_1$ converges in distribution to a complex matrix valued Gaussian random variable with mean $\mathbf{0}$ and $(P^2 \times P^2)$ Hermitian covariance matrix whose elements are specified by

$$M \cdot \mathrm{Cov}\left\{[\Delta \boldsymbol{C}_1]_{i,j}, [\Delta \boldsymbol{C}_1]_{m,n}^*\right\} = [\boldsymbol{C}_1]_{i,m}[\boldsymbol{C}_1^*]_{j,n} + \mu\left[\tilde{\boldsymbol{S}}_1 \tilde{\boldsymbol{S}}_1^T\right]_{i,n}\left[\tilde{\boldsymbol{S}}_1 \tilde{\boldsymbol{S}}_1^T\right]_{j,m}^* + \nu \sum_{\alpha=2}^{K} [\tilde{\boldsymbol{s}}_\alpha]_i [\tilde{\boldsymbol{s}}_\alpha^*]_j [\tilde{\boldsymbol{s}}_\alpha^*]_m [\tilde{\boldsymbol{s}}_\alpha]_n \quad (62)$$

where

$$\mu = \left|\mathbf{E}\left\{b^2\right\}\right|, \quad (63)$$

$$\nu = \mathbf{E}\left\{|b|^4\right\} - 2\mathbf{E}\left\{|b|^2\right\}^2 - \left|\mathbf{E}\left\{b^2\right\}\right|^2. \quad (64)$$

After straightforward algebra, we then have (65) and (66) at the top of the page. Proposition 2 then follows from the fact that $\tilde{\boldsymbol{S}}_1^H \boldsymbol{w}_1 \approx \mathbf{0}$. □

REFERENCES

[1] S. Andradottir, "A method for discrete stochastic optimization," *Manag. Sci.*, vol. 41, no. 12, pp. 1946–1961, 1995.

[2] ——, "A global search method for discrete stochastic optimization," *SIAM J. Optimiz.*, vol. 6, no. 2, pp. 513–530, May 1996.

[3] ——, "Accelerating the convergence of random search methods for discrete stochastic optimization," *ACM Trans. Modeling and Comp. Simulation*, vol. 9, no. 4, pp. 349–380, Oct. 1999.

[4] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, ser. Applications of Mathematics. New York: Springer-Verlag, 1990, vol. 22.

[5] D. Bertsekas, *Nonlinear Programming*. Belmont, MA.: Athena Scientific, 2000.

[6] D. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA.: Athena Scientific, 1996.

[7] P. Billingsley, *Convergence of Probability Measures*. New York: Wiley, 1968.

[8] J. M. Brossier, "Egalization adaptive et estimation de phase: Application aux communications sous-marines," Ph.D. dissertation, Institut National Polytechnique de Grenoble, Grenpble, France, 1992.

[9] J. I. Concha and S. Ulukus, "Optimization of CDMA signature sequences in multipath channels," in *Proc. 2001 IEEE Vehicular Technology Conf.*, 2001, pp. 1978–1982.

[10] P. J. Courtois, *Decomposability, Queueing and Computer Systems Applications*. New York: Academic, 1977.

[11] T. H. de Mello, "Variable-sample methods for stochastic optimization," *ACM Trans. Modeling and Comp. Simulation*, vol. 13, no. 2, Apr. 2003.

[12] S. B. Gelfand and S. K. Mitter, "Simulated annealing with noisy or imprecise energy measurements," *J. Opt. Theory Appl.*, vol. 62, pp. 49–62, 1989.

[13] W. Gong, Y. Ho, and W. Zhai, "Stochastic comparison algorithm for discrete optimization with estimation," *SIAM J. Optimiz.*, vol. 10, no. 2, pp. 384–404, 1999.

[14] Y. Hochberg and A. C. Tamhane, *Multiple Comparison Procedures*. New York: Wiley, 1987.

[15] A. Høst-Madsen and X. Wang, "Performance of blind and group-blind multiuser detection," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1849–1872, July 2002.

[16] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*. New York: Wiley, 1989.

[17] G. N. Karystinos and D. A. Pados, "Minimum total squared correlation design of DS-CDMA binary signature sets," in *Proc. 2001 IEEE GLOBECOM*, 2001, pp. 801–805.

[18] ——, "New bounds on the total squared correlation and optimum design of DS-CDMA binary signature sets," *IEEE Trans. Commun.*, vol. 51, pp. 48–51, Jan. 2003.

[19] ——, "Binary CDMA signature sets with concurrently minimum total sqaured correlation and maximum squared correlation," in *Proc. 2003 IEEE Int. Conf. Communications*, 2003, pp. 2500–2503.

[20] D. Kim, "Rate-regulated power control for supporting flexible transmission in future CDMA mobile networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 968–977, May 1999.

[21] V. Krishnamurthy, G. Yin, and S. Singh, "Adaptive step size algorithms for blind interference suppression in DS/CDMA systems," *IEEE Trans. Signal Processing*, vol. 49, pp. 190–201, Jan. 2001.

[22] H. J. Kushner and J. Yang, "Analysis of adaptive step-size SA algorithms for parameter tracking," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1403–1410, Aug. 1995.

[23] H. J. Kushner and G. Yin, "Stochastic approximation algorithms for parallel and distributed processing," *Stochastics*, vol. 22, pp. 219–250, 1987.

[24] ——, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

[25] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York: Springer-Verlag, 1998.

[26] U. Madhow and M. L. Honig, "MMSE interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Trans. Commun.*, vol. 42, pp. 3178–3188, Dec. 1994.

[27] J. L. Massey and T. Mittelholzer, "Welch's bound and sequence sets for code-division multiple access systems," in *Sequences II, Methods in Communication, Security and Computer Science*, R. Capocelli, A. De Santis, and U. Vaccaro, Eds. New York: Springer-Verlag, 1993.

[28] G. Pflug, *Optimization of Stochastic Models: The Interface between Simulation and Optimization*. Norwell, MA: Kluwer, 1996.

[29] R. G. Phillips and P. V. Kokotovic, "A singular perturbation approach to modeling and control of Markov chains," *IEEE Trans. Automat. Contr.*, vol. AC-26, pp. 1087–1094, Oct. 1981.

[30] H. V. Poor and S. Verdú, "Probability of error in MMSE multiuser detection," *IEEE Trans. Inform. Theory*, vol. 43, pp. 858–871, May 1997.

[31] D. C. Popescu, O. Popescu, and C. Rose, "Interference avoidence for multiaccess vector channels," in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, 2002, p. 499.

[32] G. S. Rajappan and M. L. Honig, "Signature sequence adaptation for DS-CDMA with multipath," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 384–395, Feb. 2002.

[33] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna array," *IEEE Trans. Commun.*, vol. 46, pp. 1313–1324, Oct 1998.

[34] D. Reynolds and X. Wang, "Adaptive transmitter optimization for blind and group-blind multiuser detection," *IEEE Trans. Signal Processing*, vol. 51, pp. 825–838, Mar. 2003.

[35] A. Rose, "CDMA codeword optimization: Interference avoidance and convergence via class warfare," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2368–2382, Sept. 1999.

[36] M. Rupf, F. Tarkoy, and J. L. Massey, "User-separating demodulation for code-division multiple-access systems," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 785–795, June 1994.

[37] L. Shi and S. Olafsson, "An integrated framework for deterministic and stochastic optimization," in *Proc. 1997 Winter Simulation Conf.*, Phoenix, AZ, 1997, pp. 358–365.

[38] H. A. Simon and A. Ando, "Aggregation of variables in dynamic systems," *Econometrica*, vol. 29, pp. 111–138, 1961.

[39] J. R. Swisher, S. H. Jacobson, P. D. Hyden, and L. W. Schruben, "A survey of simulation optimization techniques and procedures," in *Proc. 2000 Winter Simulation Conf.*, Orlando, FL, 2000.

[40] S. Ulukus and R. D. Yates, "Adaptive power control and MMSE interference suppression," *Wireless Networks*, vol. 4, pp. 489–496, 1998.

[41] ——, "Iterative construction of optimum signature sequence sets in synchronous CDMA systems," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1989–1998, July 2001.

[42] P. Viswanath, V. Anantharam, and D. N. C. Tse, "Optimal sequences, power control and user capacity of synchronous CDMA systems with linear MMSE multiuser receivers," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1968–1983, Sept. 1999.

[43] X. Wang and H. V. Poor, *Wireless Communication Systems: Advanced Techniques for Signal Reception* . Upper Saddle River, NJ: Prentice-Hall, 2003.

[44] D. Yan and H. Mukai, "Stochastic discrete optimization," *SIAM J. Contr. and Optimiz.*, vol. 30, no. 3, pp. 594–612, 1992.

[45] G. Yin, V. Krishnamurthy, and C. Ion, "Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization," SIAM J. Opt., 2004, to be published.

[46] G. Yin and Q. Zhang, *Continuous-Time Markov Chains and Applications: A Singular Perturbation Approach*. New York: Springer-Verlag, 1998.

[47] ——, "Singularly perturbed discrete-time Markov chains," *SIAM J. Appl. Math.*, vol. 61, pp. 833–854, 2000.