

Adaptive OFDM Synchronization Algorithms Based on Discrete Stochastic Approximation

Vikram Krishnamurthy*, Chandranath R. N. Athaudage, Dawei Huang

Abstract

This paper presents discrete stochastic approximation algorithms (DSA) for time synchronization in OFDM systems. It is shown that the discrete stochastic approximation algorithms can be effectively used to achieve a significant reduction in computational complexity compared to brute force maximum-likelihood (ML) methods for OFDM synchronization. The most important property of the proposed algorithms is their *recursive self-learning capability* - most of the computational effort is spent at the global or a local optimizer of the objective function. The convergence of the algorithms is analysed. An adaptive version of the discrete stochastic approximation algorithm is also presented for tracking time varying time delays and frequency offsets in time selective fading channels. Detailed numerical examples illustrate the performance gains of these DSA based synchronization algorithms.

Index Terms

OFDM, time synchronization, frequency synchronization, discrete stochastic approximation, computational efficiency, fading channels.

Some of the results in this paper will appear in the International Communications Conference ICC2004, Paris.

V. Krishnamurthy is with the Department of Electrical and Computer Engineering, University of British Columbia V6T 1Z4, Vancouver, Canada. e-mail:vikramk@ece.ubc.ca . tel: +1 604 822 2653 – **corresponding author**

C. R. N. Athaudage is with the ARC Special Research Center for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, University of Melbourne, Victoria 3010, Australia. e-mail:chandra@ee.mu.oz.au. tel +61 3 8344 3815

Bell Laboratories, Beijing China

I. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) is a promising technique for high-bit-rate wireless communications [4], [9], [16], [18]. Multipath immunity, bandwidth efficiency and resistance to narrow-band interference and impulse noise are the key advantages of OFDM. It has been employed for digital audio/video broadcasting (ZV95) and wireless LAN (IEEE 802.11a and Hiperlan/2) standards [16]. However, a major drawback of OFDM is its relatively high sensitivity to time and frequency synchronization errors compared to a single carrier system [4], [9], [18]. The time synchronization error refers to the incorrect timing of OFDM symbols at the receiver introducing subcarrier phase rotation and possible inter-symbol-interference (ISI) and inter-carrier-interference (ICI). Frequency synchronization error is caused by the misalignment in subcarrier frequencies due to fluctuations in receiver RF oscillators or the channel's Doppler frequency. This frequency offset can destroy the subcarrier orthogonality of the OFDM signal introducing inter-carrier-interference (ICI). Both ISI and ICI cause degradation of the bit-error-rate (BER) performance of the OFDM systems.

The synchronization problem considered here is an instance of a stochastic optimization problem of the form:

$$\text{Compute } \max_{\theta \in \Theta} \mathbf{E}\{|G_m(\theta)|\} \quad (1)$$

where $\theta \in \Theta$ is an unknown parameter, and $\{G_m\}$ is a sequence of iid random variables parameterized by θ . For continuous parameter stochastic optimization problems, i.e., when the parameter set Θ is some compact subset of the Euclidean space, continuous stochastic gradient algorithms [14], [21] such as the Least Mean Squares (LMS) and Recursive Least Squares (RLS) have been studied *ad nauseam* in the signal processing literature. They have the form

$$\hat{\theta}_{m+1} = \hat{\theta}_m + \mu \frac{\partial}{\partial \hat{\theta}_m} |G_m(\hat{\theta}_m)|$$

where μ denotes a small step size (matrix valued step size in the RLS case).

However, in the OFDM synchronization problem addressed in this paper, the underlying parameter set of possible time delays (symbol timings) Θ is discrete-valued. Thus maximizing the expected value of an objective function (e.g. likelihood function) cannot be achieved by stochastic gradient algorithms such as LMS since the concept of a gradient does not exist for such a *discrete valued optimization problem*. An obvious brute force approach to solve such a discrete stochastic optimization problem is to approximate $E\{|G_m(\theta)|\}$ for each $\theta \in \Theta$ as $|\hat{G}_M(\theta)| = \sum_{m=1}^M |G_m(\theta)|/M$. For large M by the strong law of large numbers, this approximation becomes accurate. Then simply pick the $\theta \in \Theta$ that maximizes $|\hat{G}_M(\theta)|$.

This brute force procedure is computationally inefficient since evaluating $|\hat{G}_M(\theta)|$ for a particular value of θ says nothing about $|\hat{G}_M(\theta)|$ for the optimal value of θ .

What one needs to devise is an adaptive resource allocation scheme which dynamically decides at each time instant m , which sample $\theta \in \Theta$ to pick to evaluate $|\hat{G}_m(\theta)|$ so as to converge to the optimum with minimum effort. This paper presents novel low complexity time and frequency synchronization algorithms for OFDM based on the recently proposed discrete-stochastic approximation algorithms [1]-[2] appearing in the operations research literature. Like the LMS algorithm, the discrete stochastic approximation algorithm recursively generates a sequence of parameter estimates $\{\hat{\theta}_m\}$ where each new parameter estimate is obtained from the old one by moving in a good direction. However, instead of a instantaneous derivative to guide the evolution of the algorithm, a randomized sampling and acceptance step is used. The most important property of the proposed algorithms is their *recursive self-learning capability* - most of the computational effort is spent at the global or a local optimizer of the objective function. Since the algorithm is recursive it can be used on-line to adaptively synchronize time varying time delays (symbol timings) and frequency offsets caused by time selective fading channels. The self learning capability is equivalent to statistical efficiency of the algorithm and is important in discrete stochastic optimization algorithms since an obvious brute force approach would involve exhaustive trial of all possible parameter values (as outlined above).

There are several different classes of methods that can be used to solve discrete stochastic optimization problems of the form (1), see [2], [23] for a recent survey. When the feasible set Θ is small (usually 2 to 20 elements), ranking and selection methods, and multiple comparison methods can be used to locate the optimal solution. However for large Θ the computational complexity of these methods becomes prohibitive. It is for this reason we have chosen the discrete stochastic approximation algorithms in [1], [2] as our basis for designing synchronization algorithms. We also refer the reader to [22] for a unified introduction to stochastic search and optimization.

The synchronization algorithm presented in this paper is cyclic prefix based. Cyclic prefix (CP) based techniques [4], [10], [18], and reference symbol (training sequences) based techniques [15], [20], have been proposed for OFDM synchronization in the literature. Among these, CP based techniques are considered more advantageous because the use of reference symbols lowers the achievable data rate (bandwidth inefficient). In this paper we propose low-complexity synchronization techniques useful for frequency selective (multipath) and slowly time-varying channels. Such fading channels are generally encountered when there is zero or very little relative movement between the transmitter and the receiver, i.e. zero Doppler frequency. Wireless LANs and xDSL are examples for time-flat fading situation. Indoor

environments where the receiver mobility is naturally limited also provide slowly time-varying channel conditions. In such channels, it is reasonable to assume that the time and frequency offsets are constant over a large number of OFDM symbols and that they change slowly after a large number of OFDM symbols.

The organization and main results of this paper are as follows:

In Sec.II, the synchronization problem is formulated for an OFDM system. Also a brute force maximum likelihood estimation algorithm (which involves exhaustive enumeration) presented in [10] is briefly outlined.

In Sec.III the discrete stochastic approximation algorithm is presented for solving the synchronization problem. It is shown that the algorithm can be interpreted as a decreasing step size stochastic approximation algorithm in tandem with a adaptive search/sampling scheme.

In Sec.IV convergence and efficiency of the proposed algorithm is shown. In order to show this, we verify the stochastic ordering conditions of [1] for the OFDM model.

In Sec.V, a novel adaptive version of the discrete stochastic approximation synchronization algorithm is given for OFDM systems in time selective fading channels. This algorithm can track time varying delays and frequency offsets. A mean square analysis of the tracking properties of the algorithm based on our recent work [11] is briefly outlined.

Finally, Sec.VI presents numerical examples that illustrate the performance of the synchronization algorithms in frequency selective and time selective fading channels. In particular, the performance of the adaptive synchronization algorithm for tracking time varying time delays (symbol timings) and frequency offset is illustrated for time selective fading channels.

II. TIME AND FREQUENCY SYNCHRONIZATION

A. Synchronization Task

Let N denote the total number of subcarriers. The transmitted OFDM signal sequence $\{s(n)\}$ is obtained by performing the IFFT operation over consecutive blocks of data of length N where the data comprises of QAM or PSK symbols. The front of the resulting N block IFFT sample is augmented with a N_{cp} block cyclic-prefix (CP) to form a **transmitted OFDM symbol** comprising of $N + N_{cp}$ samples. Note that the N data samples are assumed iid whereas the N_{cp} length cyclic prefix sequence is a copy of the last N_{cp} data samples of the OFDM symbol.

In an OFDM system, the complex discrete time signal $r(n)$ at the receiver is modelled as [4]-[18]

$$r(n) = s(n - T)e^{j2\pi n\Omega/N} + g(n), \quad n = 0, 1, \dots \quad (2)$$

Here $T \in \Theta$ denotes the integer valued channel delay (timing information) where

$$\Theta = \{0, 1, \dots, N_{cp} + N - 1\}$$

denotes the set of possible timing delays. The carrier frequency offset Ω (normalized using the subcarrier spacing Δf) lies in the closed interval $[0, 0.5]$. $\{g(n)\}$ denotes a complex valued additive white Gaussian noise (AWGN) process statistically independent of the process $\{s(n)\}$. The real and imaginary components of $g(n)$ are also assumed statistically independent.

For a multipath (time-dispersive) channel $s(n)$ in (2) is replaced by the multipath signal

$$p(n) = \sum_{i=1}^I a_i(n)s(n - \tau_i) \quad (3)$$

In the frequency dispersive case, the coefficients $a_i(n)$ are time invariant.

Denote the sequence of **received OFDM symbols** as $\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_m, \dots$. Here $m = 0, 1, 2, \dots$, denotes symbol time. The $N + N_{cp}$ sample received OFDM symbol \mathbf{S}_m is

$$\mathbf{S}_m = [r(m(N_{cp} + N)), r(m(N_{cp} + N) + 1), \dots, r((m + 1)(N_{cp} + N) - 1)]$$

Note that in each received OFDM symbol, both the N_{cp} cyclic prefix samples as well as the N data samples are corrupted by noise.

The synchronization task at the receiver involves the estimation of two parameters: the integer valued delay $T \in \Theta$ and the frequency offset Ω . Accurate and efficient recovery of these parameters is critical for accurate OFDM demodulation which uses the FFT operation.

B. Brute Force CP based Maximum-Likelihood Synchronization

An obvious synchronization method given a block of $M + 1$ received OFDM symbols $\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_M$ is to compute the maximum likelihood estimate of the timing delay \hat{T}_M as

$$\hat{T}_M = \arg \max_{\theta \in \Theta} L_M(\theta), \quad \text{where } L_M(\theta) = \log p(\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_M | \theta), \quad M = 1, 2, \dots \quad (4)$$

denotes the log likelihood function. Due to the Gaussian noise it is straightforwardly shown (see [4]) that the log likelihood (omitting constant terms that are independent of θ and the energy term [4, Eq(4)] that is approximately a constant for large M) is

$$L_M(\theta) = \log p(\mathbf{S}_0, \mathbf{S}_1, \dots, \mathbf{S}_M | \theta) = \frac{1}{M} \sum_{m=0}^{M-1} |G_m(\theta)|, \quad M = 1, 2, \dots \quad (5)$$

where for each $\theta \in \Theta$

$$G_m(\theta) = \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-1} r(k + \theta + m(N + N_{cp}))r^H(k + \theta + (m + 1)(N + N_{cp}) - N_{cp}). \quad (6)$$

Here the superscript H indicates complex conjugate. Notice that $G_m(\theta)$ is merely the sum of N_{cp} consecutive empirical correlations between pairs of samples spaced N samples apart as shown in Figure 1. Thus computing the MLE involves brute force enumeration of the log likelihood $L_M(\theta)$ for all $\theta \in \Theta$ which can be very expensive for large sets Θ and large M .

It is well known that the MLE \hat{T}_M is strongly consistent, i.e., as $M \rightarrow \infty$, $\hat{T}_M \rightarrow T$ with probability one (w.p.1.), where T in (2) denotes the true delay. Indeed since the noise samples $\{g(n)\}$ are i.i.d., the strong law of large numbers (SLLN) applied to (5) yields $\lim_{M \rightarrow \infty} L_M(\theta) = \mathbf{E}\{|G_m(\theta)|\}$, which is the Kullback Leibler information measure [13], [24]. This and the finiteness of Θ implies that as $M \rightarrow \infty$,

$$\hat{T}_M = \arg \max_{\theta \in \Theta} L_M(\theta) \rightarrow \arg \max_{\theta \in \Theta} \mathbf{E}\{|G_m(\theta)|\} = T. \quad (7)$$

At the true delay T , samples of the cyclic prefix and their copies in the current OFDM symbol are perfectly aligned in the summation window. As an example, Figure 2 shows the simulated magnitude plot of $G_m(\theta)$ for 2 consecutive OFDM symbols, with $T = 128$ (true delay), $\Omega = 0.3$ (frequency offset) $N = 512$ and $N_{cp} = 64$ under a AWGN channel with SNR = 10 dB.

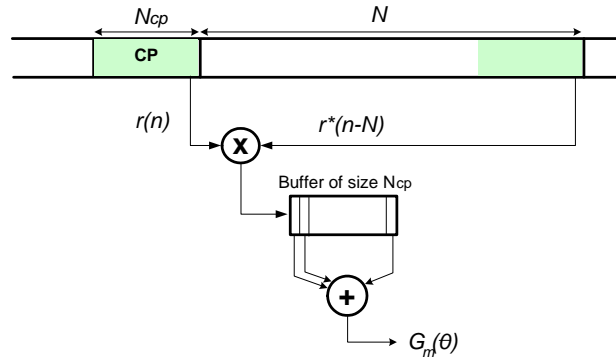


Fig. 1. Computation of the correlation function $G_m(\theta)$ using a shift register of length N_{cp} . It calculates the correlation of two sequences of N_{CP} samples length, separated by N samples, in the received sample sequence $r(n)$.

The frequency error Ω is estimated using the phase of the correlation function at the MLE delay \hat{T}_M [10], and is given by

$$\hat{\Omega}_M = -\frac{1}{2\pi} \angle L_M(\hat{T}_m). \quad (8)$$

The above brute force MLE algorithm is an off-line algorithm and assumes that the true time delay T and frequency offset Ω are constants. It is not suitable for adaptive synchronization where the time delay T and frequency offset Ω vary with time due to fading channels. Furthermore, the complexity of the algorithm is $O((N + N_{cp})M)$ which can become expensive for large M and N .

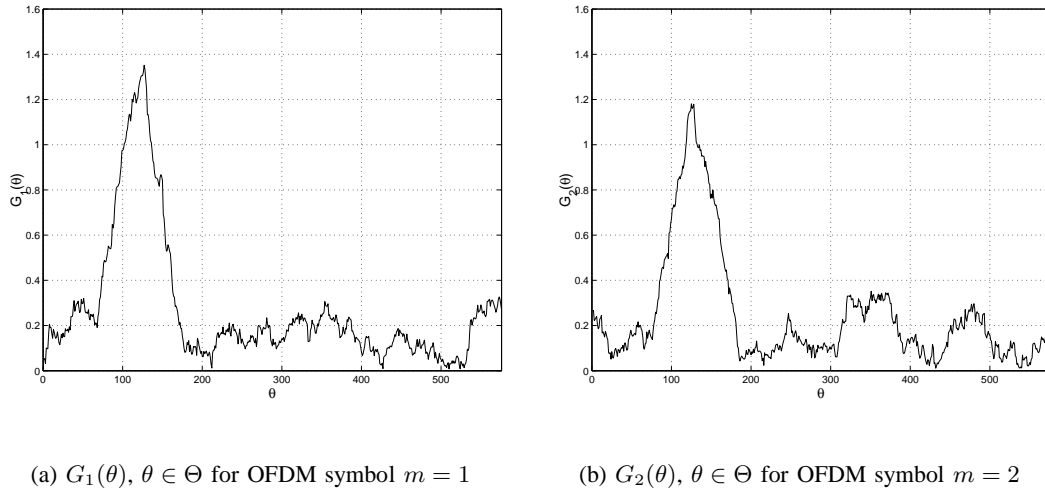


Fig. 2. Simulated magnitude plot $|G_m(\theta)|$ for 2 consecutive OFDM symbols ($T = 128$, $\Omega = 0.3$, $N = 512$, $N_{cp} = 64$ and SNR = 10 dB). Peaks of the $|G_m(\theta)|$ plot provide estimates of the symbol timing.

III. DISCRETE STOCHASTIC APPROXIMATION BASED SYNCHRONIZATION ALGORITHMS

In this section, we formulate the synchronization problem as a discrete stochastic optimization problem and present a recursive discrete stochastic approximation algorithm for solving it. Two versions of the algorithm are given – one is locally convergent, the other is globally convergent. Implementation details of these algorithms for low-complexity OFDM synchronization and the associated computational complexity are also discussed.

A. Discrete Stochastic Optimization

As demonstrated in (7) the task of determining the optimal delay can be formulated as the following discrete stochastic optimization problem:

$$\text{Estimate } T = \arg \max_{\theta \in \Theta} \mathbf{E}\{|G_m(\theta)|\} \quad (9)$$

Let Θ^* denote the set of global optimizers of (9). The uniqueness and strong consistency of the MLE \hat{T}_M , see (7), implies that $\Theta^* = \{T\}$.

The MLE of Sec.II-B can be viewed as an obvious brute force method to compute the estimate \hat{T}_M , and for large M the estimate is *consistent*, i.e., (7) holds. However, the method is highly inefficient since $L_M(\theta)$ needs to be evaluated for each $\theta \in \Theta$. The evaluations of $L_M(\theta)$ for $\theta \neq T$ are wasted because they contribute nothing to the estimation of $L_M(T)$. The idea of discrete stochastic approximation [2]

is to design a recursive algorithm that is both *consistent* and *attracted* to the maximum. That is, the algorithm should spend more time obtaining observations $G_m(\theta)$ in areas of the state space Θ near the minimizer T , and less in other areas. Thus in discrete stochastic approximation the aim is to devise an *efficient* [17, Chapter 5.3] adaptive search (sampling) plan which allows to find the maximizer with as few samples as possible by not making unnecessary observations at non-promising values of θ .

B. Discrete Stochastic Approximation Algorithm for Synchronization

The discrete stochastic approximation algorithm presented below is *consistent* (like the brute force MLE) but also *attracted* to the maximum of (9). In addition it is recursive in that it yields an updated estimate with each new OFDM symbol. The algorithm resembles an adaptive filtering, e.g. least mean square (LMS) algorithm in the sense that it recursively generates a sequence of parameter values where each new parameter value is obtained from the old one by moving in a good direction, and in the sense that it converges to the global (or a local) optimizer of the objective function.

In the algorithm below the following notation is used:

$\hat{\theta}_m \in \Theta$ is a random variable generated by the algorithm at symbol time m and can be thought of as the *state* of the algorithm at symbol time m .

$W_m(\theta)$, $\theta \in \Theta$ is a counter that yields at any symbol time m , the number of times the algorithm state sequence $\{\hat{\theta}_l\}$, $0 \leq l \leq m$ has visited any point $\theta \in \Theta$.

$\hat{\theta}_m^*$ is the estimate of the time delay T generated by the algorithm based on the m observed symbols. It is the main output of the algorithm.

The globally/locally convergent discrete stochastic approximation algorithms given below are based on [1]-[2]. They operate with different search neighborhoods \mathcal{N}_m at each symbol m . For the globally convergent algorithm $\mathcal{N}_m = \Theta - \{\hat{\theta}_m\}$. For the locally convergent algorithm $\mathcal{N}_m = 1$ if $\hat{\theta}_m = 0$, $\mathcal{N}_m = N + N_{cp} - 2$ if $\hat{\theta}_m = N + N_{cp} - 1$, and $\mathcal{N}_m = \{\hat{\theta}_m + 1, \hat{\theta}_m - 1\}$ otherwise, where $\{\hat{\theta}_m\}$ is a sequence generated by the algorithm below.

Algorithm 1: Decreasing Step Size Synchronization Algorithm

Step 0: Initialization. Select a starting point $\hat{\theta}_0 \in \Theta$. Set $W_0(\hat{\theta}_0) = 1$ and $W_0(\theta) = 0$ for all $\theta \in \Theta$, $\theta \neq \hat{\theta}_0$. Set $m = 0$ and $\hat{\theta}_m^* = \hat{\theta}_0$. Go to Step 1.

Step 1: Sampling Step. At symbol time m , given the value of the current state $\hat{\theta}_m$, generate the candidate state $\hat{\theta}'_m \in \mathcal{N}_m$ according to a uniformly distributed independent random variable. Go to Step 2.

Step 2: Evaluation and Acceptance. Given the values of the current state $\hat{\theta}_m$ and the candidate state $\hat{\theta}'_m$, generate the observation $R_m(\hat{\theta}'_m, \hat{\theta}_m) = |G_m(\hat{\theta}'_m)| - |G_m(\hat{\theta}_m)|$ where $G_m(\cdot)$ is defined in (6).

If $R_m > 0$, then update state as $\hat{\theta}_{m+1} = \hat{\theta}'_m$.

Otherwise, let $\hat{\theta}_{m+1} = \hat{\theta}_m$. Go to Step 3.

Step 3: Timing and Frequency Offset Update. Update occupation times as

$$W_m(\hat{\theta}_m) = W_{m-1}(\hat{\theta}_m) + 1, \quad W_m(\theta) = W_{m-1}(\theta) \text{ for all } \theta \in \Theta, \theta \neq \hat{\theta}_m$$

Compute the timing and frequency offset estimates: $\hat{\theta}_m^* = \arg \max_{\theta \in \Theta} W_m(\theta)$, $\hat{\Omega}_m = -\frac{1}{2\pi} \angle G_m(\hat{\theta}_m^*)$.

Set $m = m + 1$ and go to Step 1. \square

The random variable $R_m(\hat{\theta}'_m, \hat{\theta}_m) = |G_m(\hat{\theta}'_m)| - |G_m(\hat{\theta}_m)|$ is used to compare the current state $\hat{\theta}_m$ with the candidate state $\hat{\theta}'_m$. If $R_m(\hat{\theta}'_m, \hat{\theta}_m) > 0$ then the candidate state $\hat{\theta}'_m$ is better than the current state $\hat{\theta}_m$ and Step 2 of the algorithm chooses this candidate state as the new state.

The key difference between the globally and locally convergent algorithms is the neighbourhood structure \mathcal{N}_m . For the locally convergent algorithm only the two neighbours of $\hat{\theta}_m$ are members of \mathcal{N}_m , whereas for the globally convergent algorithm all points in $\Theta - \{\hat{\theta}_m\}$ are possible neighbours.

The main idea behind the above algorithm is that the finite-state sequence $\{\hat{\theta}_m\}$ generated by Steps 1 and 2 is a homogeneous Markov chain with state space Θ . As long as conditions (C1) and (C2) of Sec.IV hold, this Markov chain will spend more time at the set of global maximizers Θ^* (or local maximizer depending on the choice of neighborhood \mathcal{N}_m) than any other $\theta \in \Theta$. In Step 3, $W_m(\theta)$ is a counter that measures the number of times the sequence $\{\hat{\theta}_m\}$ until symbol m has visited $\theta \in \Theta$. Finally, the maximization in Step 3 yields the value $\hat{\theta}_m^* \in \Theta$ where the sequence $\{\hat{\theta}_m\}$ has spent most time. In Sec.IV we prove the above algorithm is consistent and attracted to the set of global (or local) maximizers.

Decreasing Step Size Interpretation of Step 3: It is convenient to map the finite-state sequence $\{\hat{\theta}_m\} \in \Theta$ generated by Algorithm 1 to the sequence $\{\mathbf{X}_m\}$ of unit vectors where

$$\mathbf{X}_m = \mathbf{e}_i \text{ if } \hat{\theta}_m = i, \quad i \in \Theta. \quad (10)$$

Here \mathbf{e}_i is the $N_{cp} + N$ dimensional unit vector with 1 in the i -th position and zeros elsewhere.

Now consider Step 3 of Algorithm 1. The counters $W_m(\theta)$, $\theta \in \Theta$ can be normalized as follows: Define the $N_{cp} + N$ dimensional vector $\hat{\alpha}_m = (\hat{\alpha}_m(1), \dots, \hat{\alpha}_m(N_{cp} + N))^T$ where $\hat{\alpha}_m(\theta) = W_m(\theta)/m$. Then $\hat{\alpha}_m(\theta)$ is an empirical measure of the occupation probability of $\theta \in \Theta$. The update of Step 3 is straightforwardly expressed as

$$\hat{\alpha}_{m+1} = \hat{\alpha}_m + \mu_{m+1} (\mathbf{X}_{m+1} - \hat{\alpha}_m) \quad (11)$$

where $\mu_m = 1/m$ represents a decreasing step size. Then the time delay estimate of Step 3 is given by $\hat{\theta}_m^* = \arg \max_{\theta \in \Theta} \hat{\alpha}_m(\theta)$, i.e., the state with the maximum occupation probability.

Hence Algorithm 1 can be viewed as a decreasing step size algorithm which involves a LMS algorithm (with decreasing step size) in tandem with a random search step and evaluation (Steps 1 and 2) for generating \mathbf{X}_m . Fig. 3 shows a schematic diagram of the algorithm with this LMS interpretation for Step 3.

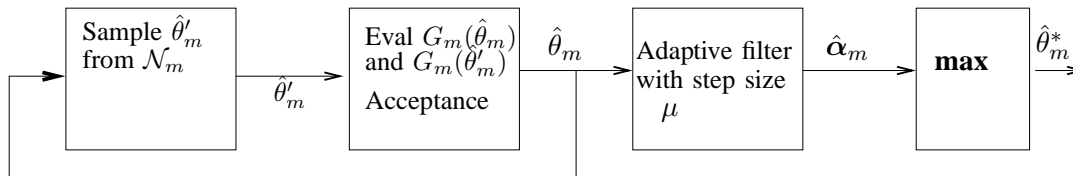


Fig. 3. Schematic of Synchronization Algorithm 1.

Implementation Details: In Step 0, the initial point in the search space $\hat{\theta}_0$ can be initialized as the MLE \hat{T}_1 (4) based on the first two received OFDM symbols $\mathbf{S}_0, \mathbf{S}_1$.

Since $\hat{\theta}_m^*$ does not directly feedback into the algorithm (see Step 3) it does not have to be computed at each time instant. Also it is clearly not necessary to store the sequences R_m and $W_m(\theta)$ for all m . They can be overwritten at each symbol time m . The main memory overhead required for the algorithm is storing the local variables $W_m(\theta)$, $\theta \in \Theta$ which requires $N_{cp} + N$ memory.

C. Computational Requirements

The number of *complex multiplications* to be performed is used as the measure of complexity. The correlation function $G_m(\theta)$ in (6) has to be evaluated for all points within the window of size $N_{cp} + N$. According to (6), evaluation of $G_m(\theta)$ for a given θ involves N_{cp} complex multiplications. However, when $G_m(\theta)$ is evaluated for consecutive points θ within the window $0 \leq \theta \leq N_{cp} + N - 1$, it can be performed as depicted in Figure 1 using a buffer of size N_{cp} . This reduces the computational cost of evaluating $G_m(\theta)$ to one complex multiplication for each θ . Therefore, the complexity C_{ML} of the ML technique becomes $N_{cp} + N$ multiplications per symbol.

$$C_{ML} = N_{cp} + N \text{ per symbol} \quad (12)$$

In the proposed synchronization techniques based on discrete stochastic approximation, correlation function $G_m(\theta)$ has to be evaluated for only two points within a window of $0 \leq \theta \leq N_{cp} + N - 1$. For the

globally convergent algorithm with neighborhood $\mathcal{N}_m = \Theta - \{\hat{\theta}_m\}$, these two points can generally be far apart within the window, and therefore the evaluation of $G_m(\theta)$ will cost $2N_{\text{cp}}$ multiplications. For the locally convergent algorithm with neighborhood $\mathcal{N}_m = \{\hat{\theta}_m + 1, \hat{\theta}_m - 1\}$ the two points are adjacent, and therefore the evaluation of $G_m(\theta)$ will cost only $N_{\text{cp}} + 1$ multiplications. Therefore, the complexities $C_{\text{DSA-global}}$ and $C_{\text{DSA-local}}$ of the global and local search DSA algorithms become

$$C_{\text{DSA-global}} = 2N_{\text{cp}} \text{ per symbol} \quad (13)$$

and

$$C_{\text{DSA-local}} = N_{\text{cp}} + 1 \text{ per symbol}, \quad (14)$$

respectively. In Sec.VI we present numerical examples that illustrate the attraction (self learning) property of the discrete stochastic approximation algorithm in terms of accuracy for a fixed computational cost.

IV. CONVERGENCE ANALYSIS OF SYNCHRONIZATION ALGORITHM

A. Convergence Theorem and Stochastic Ordering Conditions

The main convergence results in [1] for the discrete stochastic approximation based synchronization Algorithm 1 of Sec.III-A can be summarized as follows:

Let $\Theta^* \in \Theta$ denote the set of global optimizers, so that the true time delay $T \in \Theta^*$. Consider the following stochastic ordering assumption. For each $\theta \in \Theta^*$, $\theta' \in \Theta - \Theta^*$, $\zeta \in \Theta$, $\zeta \neq \theta, \theta'$: $R_m(\theta, \theta') = |G_m(\theta)| - |G_m(\theta')|$ satisfies

- (C1) $P(R_m(\theta, \theta') > 0) > P(R_m(\theta', \theta) > 0)$ or equivalently $P(R_m(\theta, \theta') > R_m(\theta', \theta)) > 0.5$.
(C2) $P(R_m(\theta, \theta') > 0) \geq P(R_m(\zeta, \theta') > 0)$.

As described below, (C1) and (C2) are conditions that ensure the synchronization Algorithm 1 is more likely to jump towards the optimum set Θ^* than away from it.

The following theorem summarized from [1] states that under conditions (C1) and (C2), Algorithm 1 is attracted to a global optimum in the sense that the state sequence of estimate $\{\hat{\theta}_m\}$ generated by the algorithm will spend more time in the set Θ^* than any other point $\theta \in \Theta - \Theta^*$.

Theorem 4.1 ([1]): Under assumptions (C1), (C2) and for the neighborhood choice $\mathcal{N}_m = \Theta - \hat{\theta}_m$, the sequence of estimates $\{\hat{\theta}_m^*\}$ generated by Algorithm 1 converges with probability 1 (w.p.1) to the global optimizer Θ^* . Under assumption (C1), for the local neighborhood choice \mathcal{N}_m defined in Sec.III-B, $\{\hat{\theta}_m^*\}$ converges w.p.1 to a local optimizer.

Remark: Recall that $\hat{\theta}_m^*$ is obtained from the maximum occupation time of the state $\hat{\theta}_m$ of Algorithm 1. In terms of this state sequence $\{\hat{\theta}_m\}$, the above theorem can be interpreted as follows: The sequence

$\{\hat{\theta}_m\}$ generated by the synchronization Algorithm 1 is *attracted* to the global optimizer set Θ^* , i.e., $W_m(T) > W_m(\theta)$, (or equivalently the estimates $\{\hat{\theta}_m\}$ visit the set of optimizers Θ^* more often than $\Theta - \Theta^*$). Thus the above algorithm is consistent and attracted to the global maximum. As explained earlier this attraction (efficiency) property is crucially important in discrete stochastic optimization problems where consistent estimators can easily be designed by exhaustive computation. In comparison, the brute force maximum likelihood algorithm spends equal computational effort at each $\theta \in \Theta$ meaning that only $1/(N_{cp} + N)$ of its computational effort is spent at the true time delay. \square

The following interpretation is useful in understanding why the above theorem works, see [1] for the formal proof. The state sequence $\{\hat{\theta}_m\}$ generated by the discrete stochastic approximation Algorithm 1 of Sec.III-A. is a homogeneous Markov chain on the state space Θ . (This follows directly from Steps 1 and 2 of the algorithm). Let $\mathbf{A} = (a_{ij})$ denote the $(N_{cp} + N) \times (N_{cp} + N)$ transition probability matrix of this Markov chain. Providing it is aperiodic irreducible, it is well known from the strong law of large numbers for Markov chains that $\hat{\alpha}_m \rightarrow \alpha$ w.p.1, where $\alpha = (\alpha(1), \dots, \alpha(N_{cp} + N))^T$ is the Perron Frobenius eigenvector (invariant state distribution) [6] of the Markov chain $\{\hat{\theta}_m\}$, i.e. α is a probability vector satisfying

$$\mathbf{A}^T \alpha = \alpha, \quad \sum_{\theta \in \Theta} \alpha(\theta) = 1.$$

The assumptions (C1) and (C2) shape the transition probability matrix $\mathbf{A} = (a_{ij})$ and hence the invariant distribution α as follows: (C1) imposes the condition that $a_{\theta\theta'} < a_{\theta'\theta}$ for $\theta \in \Theta^*$, $\theta' \in \Theta - \Theta^*$, i.e., it is more probable to jump from a candidate outside Θ^* to a candidate in Θ^* than the reverse. (C2) says that $a_{\theta\theta'} < a_{\zeta\theta'}$ for $\theta \in \Theta^*$, $\theta', \zeta \in \Theta - \Theta^*$, i.e. it is less probable to jump out of the global optimum θ to another state θ' compared to any other state ζ . Intuitively, one would expect that such a transition probability matrix would generate a Markov chain with a net inflow towards the set of global maximizers Θ^* , i.e., the algorithm spends more time in Θ^* than other states, i.e., $\alpha(\theta) > \alpha(\theta')$, $\theta \in \Theta^*$, $\theta' \notin \Theta^*$. This indeed is what Theorem 4.1 says and is proved in [1] using algebraic arguments.

B. Verification of Stochastic Ordering Conditions (C1) and (C2) for OFDM model

We now verify the stochastic ordering conditions (C1) and (C2) for the OFDM model (2). Given the fact that for the OFDM model (2), $\{s(n)\}$ is a finite state random signal and $\{g(n)\}$ is a complex-valued Gaussian noise process, it follows that the received signal $\{r(n)\}$ is a complex-valued Gaussian mixture distribution. Then given that $G_m(\theta)$ is a windowed average of $r(\cdot)$ multiplied with a shifted complex conjugate of $r(\cdot)$, it is difficult (if not impossible) to verify analytically the conditions (C1) and (C2) for

finite N_{cp} . (For example, there is no analytical formula for the distribution even for the product of two dependent Gaussian random variables). Hence there are two ways of proceeding: (i) Verify (C1) and (C2) empirically by numerical simulations, see Sec.IV-C. (ii) Show that (C1) and (C2) hold asymptotically. This leads to a two-time scale algorithm – a fast time scale over which the received samples $r(n)$ are obtained, and a slower time scale over which the observations $R_m(\hat{\theta}'_m, \hat{\theta}_m)$ are computed in Algorithm 1. For stochastic gradient algorithms such two-time scale behaviour has been extensively studied, e.g., [21], [14].

In numerical studies we show that the algorithm converges for $N_{cp} = 64$. We have empirically shown via Monte-Carlo simulations that (C1) and (C2) hold for $N_{cp} = 16$ for a wide range of parameter values, see Sec.IV-C. In theorem 4.2 below, we show asymptotically (for sufficiently large N_{cp}) that a much stronger condition than (C1) and (C2) holds.

Theorem 4.2: Let $\theta', \zeta \in \Theta$ such that $|\theta' - T| > \varepsilon_1 N_{cp}$, $|\zeta - T| > \varepsilon_2 N_{cp}$ where $\varepsilon_1, \varepsilon_2$ in $(0, 1)$ are fixed constants. Then as $N_{cp} \rightarrow \infty$, for the OFDM model (2), the following inequalities hold with probability one (w.p.1) for $R_m(\cdot, \cdot)$ defined in Step 2 of Algorithm 1:

$$R_m(T, \theta') > R_m(\theta', T), \quad R_m(T, \theta') > R_m(\zeta, \theta'). \quad (15)$$

Remarks: 1. Condition (C1) requires that $P(R_m(T, \theta') > R_m(\theta', T)) > 0.5$ whereas the above theorem states that $P(R_m(T, \theta') > R_m(\theta', T)) = 1$ which obviously implies (C1). Similarly $R_m(T, \theta') > R_m(\zeta, \theta')$ w.p.1 implies (C2).

2. The choice of $|\theta' - T| > \varepsilon_1 N_{cp}$, $|\zeta - T| > \varepsilon_2 N_{cp}$ allows us to pick points in Θ away from Θ^* depending on the choice of ε_1 and ε_2 – and it is for these points θ', ζ that we verify (C1) and (C2). Actually using a similar proof to that given below, (C1) and (C2) can be shown to hold for the following less restrictive condition on $\theta', \zeta \in \Theta$: $|\theta' - T| > \varepsilon_1 \sqrt{N_{cp}} \log N_{cp}$, $|\zeta - T| > \varepsilon_2 \sqrt{N_{cp}} \log N_{cp}$.

3. The condition $N_{cp} \rightarrow \infty$ can be replaced by the equivalent condition: $N \rightarrow \infty$ and N_{cp}/N is a bounded constant, i.e., large OFDM block size with fixed ratio of prefix samples to data samples.

Proof: For notational convenience, for any $\theta \in \Theta$ denote the quantities in (2) as

$$x_{m, \theta+k} = s(\theta + k + m(N + N_{cp}))e^{j2\pi n\Omega/N}, \quad g_{m, \theta+k} = g(\theta + k + m(N + N_{cp})).$$

Since the cyclic prefix samples of a symbol m are a replica of the last N_{cp} samples of the symbol m , there are three classes of $\theta', \zeta \in \Theta$ we need to consider in order to prove (15):

Case 1: Both the candidate points θ', ζ are within N_{cp} points of the true delay T , i.e., $\theta' = T + M$ and $\zeta = T + L$ where $M, L \leq N_{cp}$. In this case the contributions of $x_{m, \theta'+k}$ and $x_{m+1, \theta'+k}$ in $G_m(\theta')$ are correlated since $x_{m+1, \theta'+k}$ contains the last few samples of symbol m and $x_{m, \theta'+k}$ contains the cyclic

prefix samples of symbol m which is a copy of the last M samples of the symbol m . The correlation of these samples need to be accounted for. Similarly for $\zeta \in \Theta$.

Case 2: The candidate points θ', ζ are more than N_{cp} points away from T , i.e., $\theta' = T + M$ and $\zeta = T + L$ where $M, L > N_{cp}$. In this case the contributions of $x_{m,\theta'+k}$ and $x_{m+1,\theta'+k}$ to $G_m(\theta')$ are iid since they do not contain cyclic prefix samples – this is much easier to handle. Similarly for $\zeta \in \Theta$.

Case 3: Either θ' is within N_{cp} points of T and ζ is more than N_{cp} points away from T or vice versa. Showing (15) for this case straightforwardly follows from cases (i) and (ii).

Case 1: We start with this more difficult case where $\theta' = T + M$ where $M \leq N_{cp}$. Recall that $G_m(\theta')$ in (6) is computed using two OFDM symbols – and each OFDM symbol has N_{cp} cyclic prefix samples and N iid data samples. Thus $N_{cp} - M$ of the terms involving x in the summation (6) corresponding to the cyclic prefix samples and data samples of symbol m are identical, i.e., $x_{m,\theta'+k} = x_{m+1,\theta'+k}$ $1 \leq k \leq N_{cp} - M$, while the remaining terms $x_{m,\theta'+k}, x_{m+1,\theta'+k}, k = N_{cp} - M + 1, \dots, N_{cp}$ corresponding respectively to the data samples from symbol m and cyclic prefix samples from symbol $m + 1$ are statistically independent. Hence, with H denoting complex conjugate,

$$G_m(\theta') = \frac{1}{N_{cp}} \left[\sum_{k=0}^{N_{cp}-M-1} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m+1,\theta'+k-N_{cp}} + g_{m+1,\theta'+k-N_{cp}})^H + \sum_{k=N_{cp}-M}^{N_{cp}-1} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m+1,\theta'+k-N_{cp}} + g_{m+1,\theta'+k-N_{cp}})^H \right] \quad (16)$$

Expanding out $G_m(\theta')$, we define the following variables for notational convenience:

$$\begin{aligned} R_{1:N_{cp}-M}^S &= \Re \left\{ \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-M-1} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m+1,\theta'+k-N_{cp}} + g_{m+1,\theta'+k-N_{cp}})^H \right\} \\ &= \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-M-1} |x_{m,\theta'+k}|^2 + \Re \left\{ \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-M-1} \left(g_{m,\theta'+k} x_{m,\theta'+k}^H + x_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right. \right. \\ &\quad \left. \left. + g_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right) \right\}, \end{aligned}$$

where we have used the fact that the cyclic prefix samples are identical to the last N_{cp} samples of the OFDM symbol, i.e., $x_{m,\theta'+k} = x_{m+1,\theta'+k-N_{cp}}$ $1 \leq k \leq N_{cp} - M$.

$$\begin{aligned} I_{1:N_{cp}-M}^S &= \Im \left\{ \frac{1}{N_{cp}} \sum_{k=1}^{N_{cp}-M} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m,\theta'+k} + g_{m+1,\theta'+k-N_{cp}})^H \right\} \\ &= \Im \left\{ \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-M-1} \left(g_{m,\theta'+k} x_{m,\theta'+k}^H + x_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H + g_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right) \right\}, \end{aligned}$$

$$\begin{aligned}
R_{N_{cp}-M+1:N_{cp}}^N &= \Re \left\{ \frac{1}{N_{cp}} \sum_{k=N_{cp}-M}^{N_{cp}-1} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m+1,\theta'+k-N_{cp}} + g_{m+1,\theta'+k-N_{cp}})^H \right\} \\
&= \Re \left\{ \frac{1}{N_{cp}} \sum_{k=N_{cp}-M}^{N_{cp}-1} \left(x_{m,\theta'+k} \tilde{x}_{m+1,\theta'+k-N_{cp}}^H + g_{m,\theta'+k} x_{m+1,\theta'+k-N_{cp}}^H + x_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right. \right. \\
&\quad \left. \left. + g_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right) \right\}, \\
I_{N_{cp}-M+1:N_{cp}}^N &= \Im \left\{ \frac{1}{N_{cp}} \sum_{t=N_{cp}-M}^{N_{cp}-1} (x_{m,\theta'+k} + g_{m,\theta'+k}) (x_{m+1,\theta'+k-N_{cp}} + g_{m+1,\theta'+k-N_{cp}})^H \right\} \\
&= \Im \left\{ \frac{1}{N_{cp}} \sum_{t=N_{cp}-M}^{N_{cp}-1} \left(x_{m,\theta'+k} x_{m+1,\theta'+k-N_{cp}}^H + g_{m+1,\theta'+k-N_{cp}} x_{m+1,\theta'+k-N_{cp}}^H \right. \right. \\
&\quad \left. \left. + x_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H + g_{m,\theta'+k} g_{m+1,\theta'+k-N_{cp}}^H \right) \right\}.
\end{aligned}$$

Then

$$\begin{aligned}
|G_m(T)|^2 &= \left(R_{1:N_{cp}}^S \right)^2 + \left(I_{1:N_{cp}}^S \right)^2, \\
|G_m(T+M)|^2 &= \left(R_{1:N_{cp}-M}^S + R_{N_{cp}-M+1:N_{cp}}^N \right)^2 + \left(I_{1:N_{cp}-M}^S + I_{N_{cp}-M+1:N_{cp}}^N \right)^2.
\end{aligned}$$

It therefore follows that

$$\begin{aligned}
\mathbf{E} \left(R_{1:N_{cp}}^S \right) &= \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-1} \mathbf{E} |x_{m,\theta'+k}|^2 > \mathbf{E} \left(R_{1:N_{cp}-M}^S \right) = \frac{1}{N_{cp}} \sum_{k=0}^{N_{cp}-M-1} \mathbf{E} |x_{m,\theta'+k}|^2, \\
\mathbf{E} \left(I_{1:N_{cp}}^S \right) &= \mathbf{E} \left(R_{N_{cp}-M+1:N_{cp}}^N \right) = \mathbf{E} \left(I_{1:N_{cp}-M}^S \right) = \mathbf{E} \left(I_{N_{cp}-M+1:N_{cp}}^N \right) = 0.
\end{aligned}$$

Below, for any real number x , $\lceil x \rceil$ denotes the smallest integer that is greater than or equal to x .

Lemma 1: For any fixed $\varepsilon_1, \varepsilon_2$ in $(0, 1)$, as $N_{cp} \rightarrow \infty$, the following inequalities hold with probability 1 (w.p.1) – these imply (15):

$$\begin{aligned}
|G_m(T)| &> |G_m(T + \lceil \varepsilon_1 N_{cp} \rceil)| \\
|G_m(T)| - |G_m(T + \lceil \varepsilon_1 N_{cp} \rceil)| &> |G_m(T + \lceil \varepsilon_2 N_{cp} \rceil)| - |G_m(T + \lceil \varepsilon_1 N_{cp} \rceil)| \tag{17}
\end{aligned}$$

Proof: Using Kolmogorov's strong law of large numbers for iid sequences (since the data samples are iid) we have

$$\begin{aligned}
R_{1:N_{cp}}^S &\rightarrow \beta, \quad R_{1:N_{cp}-\lceil \varepsilon_1 N_{cp} \rceil}^S \rightarrow (1 - \varepsilon_1) \beta, \quad R_{N_{cp}-M+1:N_{cp}}^N \rightarrow 0, \quad \text{w.p.1} \\
I_{1:N_{cp}}^S &\rightarrow 0, \quad I_{1:N_{cp}-M}^S + I_{N_{cp}-M+1:N_{cp}}^N \rightarrow 0, \quad \text{w.p.1}
\end{aligned}$$

where

$$\beta = \mathbf{E} \left(|x_{m,T}|^2 \right) > 0.$$

So,

$$\lim_{N_{cp} \rightarrow \infty} |G_m(T)| = \beta > (1 - \varepsilon_1) \beta = \lim_{N_{cp} \rightarrow \infty} |G_m(T + \lceil \varepsilon_1 N_{cp} \rceil)|, \quad a.s.$$

Similarly,

$$\lim_{N_{cp} \rightarrow \infty} |G_m(T + \lceil \varepsilon_1 N_{cp} \rceil)| - |G_m(T + \lceil \varepsilon_2 N_{cp} \rceil)| = (1 - \varepsilon_1) \beta - (1 - \varepsilon_2) \beta = (\varepsilon_2 - \varepsilon_1) \beta \leq \beta$$

Case 2: The proof for the case when $\theta' = T + M$ where $M > N_{cp}$ is simpler. For this case $x_{m,\theta'+k}$ and $x_{m+1,\theta'+k-N_{cp}}$ in (16) are independent since they only comprise iid data samples and do not include any cyclic prefix samples. Thus $|G_m(T + M)| \rightarrow 0$ as $N_{cp} \rightarrow \infty$ which trivially implies (C1) and (C2).

Finally, if ζ is within N_{cp} points of T and θ' is further than N_{cp} points of T , the same reasoning as above yields $|G_m(T)| - |G_m(\theta')| = \beta$, while $|G_m(\zeta)| - |G_m(\theta')| = \varepsilon_2 \beta$ which implies that (17) holds. (a similar proof holds if ζ is further than N_{cp} points of T and θ' is within N_{cp} points of T .) ■

C. Empirical Verification of (C1) and (C2)

Here we empirically verify (C1) and (C2) for small N_{cp} . We chose $N = 512$, $N_{cp} = 16$, a multipath Rayleigh fading channel (as specified in Example 3 of Sec.VI) so that the delay spread is 4 times N_{cp} , and SNR=10 dB.

To empirically verify (C1), Fig.4 plots empirical estimates of $P(R_m(\theta', T) > 0)$ based on 1000 independent replications for each $\theta' \in [T - 10, \dots, T + 10]$, $\theta = T = 128$. As can be seen from Fig.4, $P(R_m(\theta', T) > 0) < 0.5$, thus empirically verifying C1.

To empirically verify (C2) we chose, $\theta' \in [T - 10, \dots, T + 10]$, $\theta = T = 128$, and $\zeta = T + 1$ (Fig.5), $\zeta = T + 2$ (Fig.6), respectively. Again each of the empirical probability estimates were computed based on 1000 independent replications. It can be seen from Fig.5 and Fig.6 that $P(R_m(\theta', \theta) > 0) < P(R_m(\theta', \zeta) > 0)$, or equivalently, $P(R_m(\theta, \theta') > 0) \geq P(R_m(\zeta, \theta') > 0)$, thus empirically verifying (C2).

V. ADAPTIVE SYNCHRONIZATION ALGORITHM FOR TIME AND FREQUENCY SELECTIVE FADING CHANNELS

So far we have presented discrete stochastic approximation algorithms for estimating the time delay T and frequency offset Ω when these parameters are fixed constants. Here we consider the case where

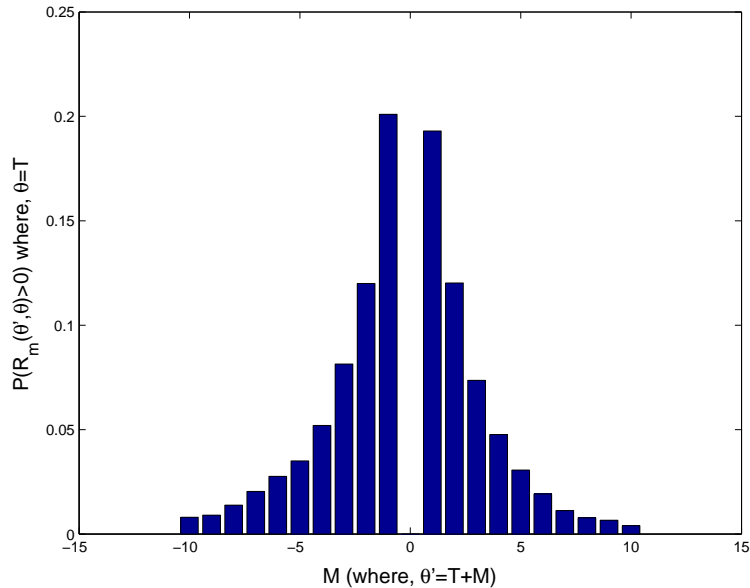


Fig. 4. Empirical verification of (C1) for $N_{cp} = 16$, $\theta' \in [T - 10, \dots, \theta + 10]$, $T = 128$.

due to time and frequency selective fading channel, the true time delay T and the frequency offset Ω are time varying. We denote these as $T_m \in \Theta$ and Ω_m , where m denotes the symbol time. Then the aim is to devise an *adaptive* discrete stochastic approximation algorithm that dynamically tracks these time varying parameters and hence achieve synchronization. Note that since Θ is a finite set, T_m is a finite state process.

From a mathematical point of view, the aim is to devise an adaptive algorithm to adaptively track the time varying discrete valued parameter (delay) by recursively maximizing the Kullback Leibler information measure $\mathbf{E}\{|G_m(\theta)|\}$. Formulation of tracking algorithms based on recursively optimizing the Kullback Leibler information measure are well known, they have been used for recursive maximum likelihood parameter estimation of slowly time-varying state space models in [24], [25] and Hidden Markov Models in [13]. Of course the key difference is that whereas these papers formulate stochastic gradient algorithms for tracking the parameter, here we are dealing with a discrete valued parameter for which gradients do not apply. We present an adaptive (tracking) version of Algorithm 1 to recursively minimize the Kullback Leibler information measure. Naturally since we are in a discrete optimization domain, the efficiency (attraction property) of the adaptive algorithm in tracking the time varying parameter is of key importance.

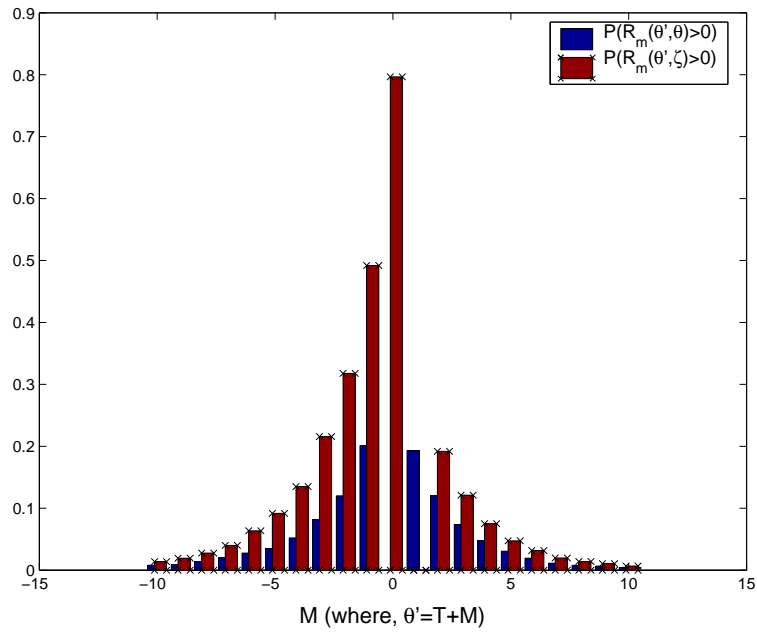


Fig. 5. Empirical verification of (C2) for $N_{cp} = 16$, $\theta' \in [T - 10, \dots, \theta + 10]$, $T = 128$, $\zeta = \theta + 1$.

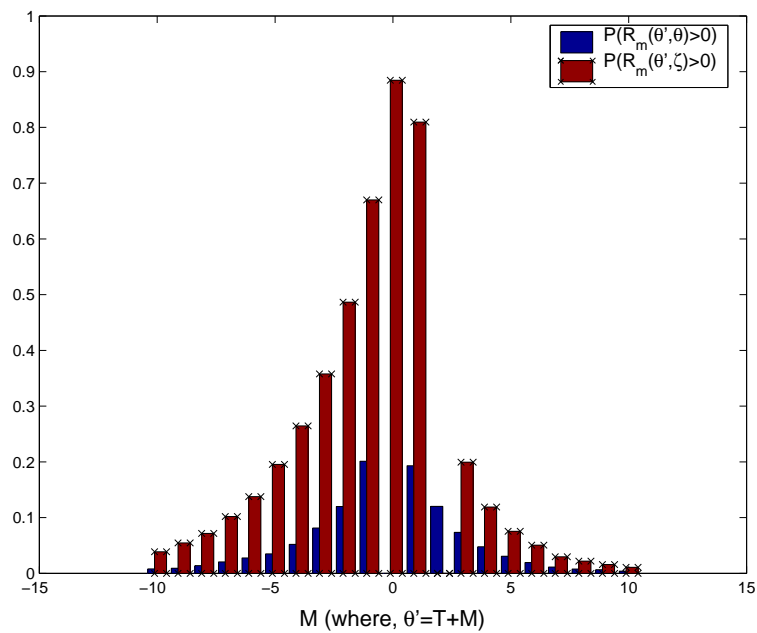


Fig. 6. Empirical verification of (C2) for $N_{cp} = 16$, $\theta' \in [T - 10, \dots, \theta + 10]$, $T = 128$, $\zeta = \theta + 2$.

A. Adaptive Synchronization Algorithm

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

Algorithm 2: Constant step-size Synchronization Algorithm

Steps 0, 1 and 2: Identical to Algorithm 1.

Step 3: Adaptive Tracking of Timing and Frequency Offset. Update occupation probabilities using the following fixed step-size LMS algorithm (cf (11)):

$$\hat{\alpha}_{m+1} = \hat{\alpha}_m + \mu (\mathbf{X}_{m+1} - \hat{\alpha}_m), \quad (18)$$

where the step-size $\mu \ll 1$ is a small positive constant.

Set $\hat{\theta}_m^* = \arg \max_{\theta \in \Theta} \hat{\alpha}_m(\theta)$. $\hat{\Omega}_m = -\frac{1}{2\pi} \angle G_m(\hat{\theta}_m^*)$.

Set $m = m + 1$ and go to Step 1.

Remark: As long as the step size satisfies $0 < \mu < 1$, $\hat{\alpha}_m$ is guaranteed to be a probability vector. To see this, note that $\mathbf{1}^T(\mathbf{X}_{m+1} - \hat{\alpha}_m) = 0$ implying that $\mathbf{1}^T \hat{\alpha}_{m+1} = \mathbf{1}^T \hat{\alpha}_m = 1$. Also rewriting (18) as $(1 - \mu)\hat{\alpha}_m + \mu\mathbf{X}_{m+1}$ implies that all elements of $\hat{\alpha}_{m+1}$ are non-negative.

The constant step size μ essentially introduces an exponential forgetting of the past occupation probabilities and permits us to track slowly time-varying time delays and frequency offsets. Its schematic is as in Fig.3 but with μ now a fixed step size in the LMS algorithm. Thus Algorithm 2 can be viewed as a discrete sampling and evaluation step in tandem with a fixed step-size adaptive filtering algorithm.

Unlike Step 3 of Algorithm 1 which only requires 1 real multiplication, Step 3 of Algorithm 2 requires $N_{cp} + N$ real multiplications at each time instant. In practical implementation, the following asynchronous version of Algorithm 2 can be used: Replace $\hat{\alpha}_m$ in Algorithm 2 with $\tilde{\alpha}_m$ where $\tilde{\alpha}_m$ is updated as

$$\tilde{\alpha}_{m+1} = \tilde{\alpha}_m + \mu \text{diag}(\mathbf{X}_{m+1})(\mathbf{X}_{m+1} - \tilde{\alpha}_m), \quad \tilde{\alpha}_0 = \mathbf{0}. \quad (19)$$

Here $\text{diag}(\mathbf{X})$ denotes the diagonal matrix with diagonal elements \mathbf{X} . Thus only one component of $\tilde{\alpha}_m$ is updated at each time instant which requires one real multiplication. Note that the update times of the j th component of $\tilde{\alpha}_m$ occur at the (random) time instants when the Markov chain \mathbf{X}_m visits the state \mathbf{e}_j – thus the algorithm can be viewed as an asynchronous implementation of Algorithm 2; see [14] and the references therein. Clearly the components of $\tilde{\alpha}_m$ are non-negative but do not sum to one. Let $\bar{\alpha}_m$ be the normalized version of $\tilde{\alpha}_m$, i.e., the elements of $\bar{\alpha}_m$ add to one. It can be shown [12] that $\bar{\alpha}_m$ satisfies the same mean square error results as $\hat{\alpha}$ below. In numerical examples, both Algorithm 2 and its asynchronous implementation (19) yielded virtually identical behaviour.

B. Assumptions for Mean Square Tracking Analysis

In stochastic gradient schemes (e.g., LMS), a typical elementary method for analyzing the tracking performance of an adaptive algorithm is to examine how the algorithm with a constant step size hovers around a constant (time-invariant) parameter, see for example [8]. A more advanced analysis approach, e.g., [21], postulates a *hypermodel* (usually a continuous-valued random walk) for the variation in the true parameter and then analyses how the adaptive algorithm tracks this hypermodel. In this section we extend this tracking analysis to discrete stochastic approximation Algorithm 2. We postulate a discrete-valued hypermodel, namely a finite state Markov chain for the time-varying delay $\{T_m\}$ and then present mean square convergence and probability of error results for the adaptive Algorithm 2. Obviously the analysis of the tracking algorithm for a constant parameter is a trivial case of our analysis below. It is also important to note that the hypermodel assumption is only used for our subsequent tracking analysis, it does not enter the actual implementation of Algorithm 2.

The time-varying delay $\{T_m\}$ has 2 properties:

(i) $\{T_m\}$ is integer valued and belongs to the *finite* state space $\Theta = \{0, \dots, N + N_{cp} - 1\}$. Hence the sequence $\{T_m\}$ is a finite state discrete time stochastic process.

(ii) Due to the correlated nature of the fading channel, $\{T_m\}$ is a correlated finite state process.

Hence it is natural to model $\{T_m\}$ as a finite state Markov chain which nicely captures both of the above properties.

In the tracking analysis of continuous stochastic gradient algorithms (e.g. Least Mean Squares (LMS) adaptive filtering algorithms) [5], [8], it is assumed that the parameter varies slower than the adaptation speed of the tracking algorithm. Here we make a similar assumption for our mean square convergence analysis: we assume that the time evolution of $T_m \in \Theta$ is a slow Markov chain.

We now formalize the above description as the following assumptions:

(M1) Fading Hypermodel: Assume that there is a small parameter $\epsilon > 0$ and that $\{T_m\}$ is a discrete-time homogeneous Markov chain, with state space Θ . The transition probability matrix $\mathbf{P}^\epsilon = (p_{ij}^\epsilon)$ is given by

$$\mathbf{P}^\epsilon = \mathbf{I} + \epsilon \mathbf{Q}, \quad (20)$$

where \mathbf{I} denotes the $(N_{cp} + N) \times (N_{cp} + N)$ identity matrix and $\mathbf{Q} = (q_{ij})$ is a $(N_{cp} + N) \times (N_{cp} + N)$ matrix which is the generator of a continuous-time Markov chain. That is, \mathbf{Q} satisfies $q_{ij} \geq 0$ for $i \neq j$ and $\sum_{j=1}^{N_{cp}+N} q_{ij} = 0$ for each $i \in \Theta$ (thus $q_{ii} < 0$). Assume for simplicity that the initial distribution $P(T_0 = i) = p_{0,i}$ is independent of ϵ for each $i \in \Theta$, where $p_{0,i} \geq 0$ and $\sum_{i \in \Theta} p_{0,i} = 1$.

Remark: The parameter ϵ specifies the degree of “slowness” of the Markov chain dynamics, i.e., it characterizes how slowly the hypermodel evolves with time, see [27] and references therein for the motivation and extensive discussion of two-time scale and singularly perturbed Markov chains. Note that for sufficiently small $\epsilon > 0$, \mathbf{P}^ϵ is a valid transition probability matrix (i.e., each element is non-negative). Then the corresponding Markov chain $\{T_m\}$ is irreducible. It is clear from Theorem 4.1 that Algorithm 1 generates an irreducible aperiodic Markov chain. In what follows, we put this in (M2) for convenience.

(M2) Dynamics of Discrete Stochastic Approximation Algorithm: Given $\{T_m\}$, then $\{\mathbf{X}_m\}$ (or equivalently $\hat{\theta}_m$, see (10)) generated by Algorithm 1 is a Markov chain with the $(N_{cp} + N) \times (N_{cp} + N)$ transition probability matrix $\mathbf{A}_{T_{m+1}} = (a_{ij}(T_{m+1}))$, where the elements

$$a_{ij}(T_{m+1}) = P(\mathbf{X}_{m+1} = \mathbf{e}_j \mid \mathbf{X}_m = \mathbf{e}_i, T_{m+1}). \quad (21)$$

Assume the transition matrix is irreducible and aperiodic. Let α_{T_m} denote the invariant distribution of \mathbf{A}_{T_m} , i.e.,

$$\alpha_{T_m} = \mathbf{A}_{T_m}^T \alpha_{T_m}, \quad \mathbf{1}^T \alpha_{T_m} = 1.$$

Remark: The quantity ϵ being small is to ensure that the Markov chain $\{T_m\}$ and thus the true optimum T_m has slow dynamics, i.e., it jumps infrequently. Note that μ is the step size of the adaptive algorithm for estimating $\hat{\alpha}_m$. 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 (21) 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 slow transition rates.

C. Mean Square Tracking Analysis and Probability of Error

The following theorem gives a mean-squared bound on the tracking error of the occupation probability estimate $\hat{\alpha}_m$ generated by Algorithm 2.

Theorem 1: Under the conditions (C1), (C2), (M1), and (M2), for sufficiently large number of OFDM symbols m , the mean square error of the estimate $\hat{\alpha}_m$ generated by the tracking Algorithm 2 satisfies

$$\mathbf{E}\|\hat{\alpha}_m - \alpha_{T_m}\|^2 = O\left(\mu + \epsilon + \frac{\epsilon^2}{\mu}\right) \quad (22)$$

The proof is given in [26]. Looking at the order of magnitude estimate, to balance the two terms μ and ϵ^2/μ , we need to choose $\mu = O(\epsilon)$. That is, the rate of change of the true parameter can be as fast as the adaptation of the tracking algorithm for the algorithm to successfully track the time varying parameter.

Due to the discrete nature of our problem, it makes sense to give bounds on the probability of error of the estimates $\hat{\theta}_m^*$ generated by Step 3 of Algorithm 2 rather than the mean-squared error. Define the error event E and the probability of error $P(E)$ as

$$E = \{\hat{\theta}_m^* \neq T_m\}, \quad P(E) = P(\hat{\theta}_m^* \neq T_m). \quad (23)$$

Clearly E depends on m ; we suppress the m here for notational simplicity. Based on the mean square error of Theorem 1 the following result holds:

Corollary 1: Under the conditions (C1), (C2), (M1) and (M2) if $\mu = O(\epsilon)$, then for sufficiently large n , the error probability of the estimate $\{\hat{\theta}_m^*\}$ generated by the adaptive Algorithm 2 satisfies

$$P(E) \leq K\mu^{1-2\gamma}, \quad 0 < \gamma < 1/2, \quad (24)$$

where K is a positive constant independent of μ and ϵ , and $0 < \gamma < 1/2$ is an arbitrary constant.

The proof is given below. The main use of the above corollary is as a consistency check: As $\mu \rightarrow 0$, the probability of error $P(E)$ of the tracking algorithm goes to zero. This is similar to analyses of the LMS algorithm where similar type of results are proved.

Proof: The estimate of the maximum generated by the discrete stochastic approximation algorithm at symbol time m is $\hat{\alpha}_m^* = \arg \max_j \hat{\alpha}_m(j)$. Thus the error event E in (23) is equivalent to $E = \{I(\arg \max_{i \in \Theta} \alpha_{T_m}(i) \neq \arg \max_{j \in \Theta} \hat{\alpha}_m(j))\}$, where $I(\cdot)$ denotes the indicator function. Then clearly the complement event $\bar{E} = \{I(\arg \max_{i \in \Theta} \alpha_{T_m}(i) = \arg \max_{j \in \Theta} \hat{\alpha}_m(j))\}$ satisfies

$$\begin{aligned} \bar{E} &\supseteq \{I(|\max_i \alpha_{T_m}(i) - \max_j \hat{\alpha}_m(j)| \leq \min_{i,j} |\alpha_{T_m}(i) - \hat{\alpha}_m(j)|)\} \\ &\supseteq \{I(|\max_i \alpha_{T_m}(i) - \max_j \hat{\alpha}_m(j)| \leq L)\} \end{aligned}$$

where

$$L \leq \min_{i,j} |\alpha_{T_m}(i) - \hat{\alpha}_m(j)| \quad (25)$$

is a positive constant. Then the probability of no error is

$$P(\bar{E}) = P(\arg \max_i \alpha_{T_m}(i) = \arg \max_j \hat{\alpha}_m(j)) > P(|\max_i \alpha_{T_m}(i) - \max_j \hat{\alpha}_m(j)| \leq L)$$

for any sufficiently small positive number L . Then using the above equation and Theorem 1

$$\begin{aligned} P(E) &\leq P(|\max_i \alpha_{T_m}(i) - \max_j \hat{\alpha}_m(j)| > L) \\ &\leq P(\max_i |\alpha_{T_m}(i) - \hat{\alpha}_m(i)| > L) \end{aligned} \quad (26)$$

Applying Chebyshev's inequality to (22) yields for any i ,

$$P(|\alpha_{T_m}(i) - \hat{\alpha}_m(i)| > L) \leq \frac{1}{L^2} K \mu$$

for some constant K . Thus (26) yields

$$P(\max_i |\alpha_{T_m}(i) - \hat{\alpha}_m(i)| > L) \leq \frac{1}{L^2} K \mu \quad (27)$$

It only remains to pick a sufficiently small L . Choose $L = \mu^\gamma$ where $0 < \gamma < \frac{1}{2}$ is arbitrary. It is clear that for sufficiently small μ , L satisfies (25). Then (27) yields $P(E) \leq K \mu^{1-2\gamma}$. ■

VI. NUMERICAL EXAMPLES

We present three classes of examples that illustrate the efficiency and adaptive learning properties of the discrete stochastic approximation Algorithms 1 and 2. The first example considers a scenario similar to that in [10] where as a result of a frequency selective fading channel, the timing delay and frequency offset do not change with time. In the second example we consider the adaptive synchronization case where the timing delay evolves randomly according to a slow Markov chain as a result of a time and frequency selective fading channel. The third example deals with an actual simulation of a time and frequency selective channel. In all three examples, the FFT implementation of the complex-baseband OFDM modulation process with QPSK was used for simulations [9].

Throughout this section the number of subcarriers is $N = 512$ and the cyclic prefix length is $N_{cp} = 64$. In all cases below, the discrete stochastic approximation algorithms were initialized with \hat{T}_1 (which is the MLE based on 2 OFDM symbols, see (4)). The normalized frequency offset was set to $\Omega = 0.3$.

A. Example 1: Frequency Selective Fading Channel

In this example we consider a time-invariant wireless channel with frequency selective fading (FIR channel) in AWGN. We consider a FIR channel model with $I = 3$ multi-paths

$$h(n) = \sum_{i=1}^I a_i \delta(n - n_i). \quad (28)$$

The values $a_1 = 0.9$, $a_2 = 0.36$, $a_3 = 0.29$ and $n_1 = 0$, $n_2 = 6$, $n_3 = 11$ approximate the impulse response of a wireless channel in a warehouse-type indoor environment [9], [10]. The synchronization example considered here is similar to that in [10] and the parameters T and Ω are assumed to be constant over an arbitrary large number of OFDM symbols.

In Sec.IV we proved the efficiency (attraction property) and consistency of the discrete stochastic approximation Algorithm 1 of Sec.III-B – both are asymptotic properties. Here we examine via numerical

examples, the behaviour of Algorithm 1 for short data lengths. by comparing the tradeoff between computational complexity and estimation accuracy between Algorithm 1 and the brute force MLE algorithm. To make a fair comparison of the algorithms we compare their synchronization accuracy for a fixed computational budget – see [7] for an insightful discussion on how to compare simulation based optimization algorithms. From (12) and (14), computing the brute force MLE $\hat{T}_{\bar{M}}$ based on \bar{M} OFDM symbols has equal complexity to computing the locally convergent Algorithm 1 estimate $\hat{\theta}_M^*$ using M OFDM symbols where

$$\bar{M} = \left\lceil \frac{N_{cp} + 1}{N_{cp} + N} M \right\rceil \text{ OFDM symbols}$$

In the simulations we chose $M = 256$, hence $\bar{M} = 29$. In Fig.7 we plot the mean square error in the symbol timing estimate versus average channel SNR for:

- (i) the brute force MLE \hat{T}_{29} .
- (ii) the locally convergent version of Algorithm 1 estimate $\hat{\theta}_{256}^*$.
- (iii) The two symbol MLE \hat{T}_1 see (7), (8).
- (iv) The brute force MLE \hat{T}_{256} .

The average channel SNR was computed as

$$\text{SNR} = 10 \log_{10} \frac{\mathbf{E}\{|p(n)|^2\}}{\mathbf{E}\{|g(n)|^2\}}$$

where $p(n)$, $g(n)$ are defined in (2), (3), respectively. The mean square error in the symbol timing estimate was computed by averaging $(T - \hat{T}_m)^2$ over 25,600 OFDM symbols. The main point to note in Fig.7 is that while (i) and (ii) have the same computational cost, the attraction (learning) property of Algorithm 1 results in a remarkable decrease in mean square error. The plot (iii) of \hat{T}_1 serves as an obvious upper bound for the mean square error. The plot (iv) of the brute force MLE \hat{T}_{256} serves as a lower bound for Algorithm 1, in the sense that if the algorithm had spent all its effort only at the true delay T (i.e., the state $\hat{\theta}_m = T$ for all m), then its MSE would coincide with \hat{T}_{256} .

In Fig.9 we plot a sample path of the estimated time delay sequence $\hat{\theta}_m^*$ generated by the locally convergent version of Algorithm 1 from symbol times $m = 0, 1, \dots, 256$ for a SNR of 20 dB. The algorithm rapidly converges to the true timing delay.

To further illustrate the attraction (learning) property of Algorithm 1, Fig.8 shows the normalized occupation time $\hat{\alpha}_{256}(\theta) = W_{256}(\theta)/256$, $\theta \in \Theta$ computed by Step 3 of Algorithm 1 for a SNR of 20 dB. It shows that the algorithm spends most of its time in the neighbourhood of the true time delay $T = 60$ and does not waste computations at values far away from the true delay. From the Fig.8 it follows that for

the first 256 OFDM symbols the state $\hat{\theta}_m$ of the algorithm spends roughly half its computational effort at the true time delay $T = 60$. In comparison, the brute force MLE spends only 1/256 of its computational effort at the true time delay T . Thus the average SNR performance of Algorithm 1 is similar to the brute force MLE \hat{T}_{128} , yet Algorithm 1 requires only approximately 1/5 of the computational cost.

Due to lack of space we have omitted the results for the globally convergent version of Algorithm 1. It performed similarly to the locally convergent version. Also the frequency offset estimates $\hat{\Omega}_m$ performed similarly to the time-offset estimates – i.e., Algorithm 1 resulted in improved estimates compared to the MLE for comparable computational budget, see [3].

B. Example 2: Time and Frequency Selective Fading Channel

Having illustrated the attraction (learning) property of the discrete stochastic approximation Algorithm 1, here we illustrate the tracking properties of its adaptive counterpart, namely Algorithm 2 for a time-varying channel with Rayleigh fading. In the case of a time and frequency selective fading channel, the synchronization parameters T and Ω can no longer assumed to be constant. Since $T \in \Theta$ is discrete valued, we modelled its evolution due to time selective fading as a Markov chain with slow dynamics as in Sec.V. The $(N_{cp} + N) \times (N_{cp} + N)$ (with $N = 512$, $N_{cp} = 64$) transition probability matrix $\mathbf{P}^\epsilon = \mathbf{I} + \epsilon\mathbf{Q}$ (20) was chosen with $Q_{ij} = 1/(N_{cp} + N)$, $i \neq j$ and $Q_{ii} = -(N_{cp} + N - 1)/(L + N)$. The parameter ϵ quantifies the rate of fading which in turn affects the rate at which the synchronization parameters (T_m and Ω_m) vary with time. We chose $\epsilon = 10^{-3}$ in our simulations below.

The locally convergent version of Algorithm 2 was run with step sizes $\mu = 0.001$, $\mu = 0.02$, $\mu = 0.03$, respectively. Fig.10 shows the time delay estimates generated by Algorithm 2 for these different step sizes for a SNR of 20 dB. As might be expected, small μ resulted in slow tracking but small variation, large step sizes resulted in faster tracking but larger variation around the true parameter.

Fig.11 shows snapshots of the occupation probability vector $\hat{\alpha}_m$ (see Step 3 of Algorithm 2) for symbol times $m = 2400$ and $m = 5000$. The x -axis of these figures have been magnified around the $\hat{\theta}_m^*$ (the value of θ which has the highest occupancy probability $\hat{\alpha}_m(\theta)$) for clarity. As can be seen from Fig.10 and Fig.11, the algorithm satisfactorily tracks the time varying symbol timing $\{T_m\}$ of Fig.10. Also note that the algorithm spends substantially more time at the true symbol timing than any other value of θ – this is reflected by the fact that occupation probabilities $\hat{\alpha}_m(\theta)$ are substantially higher for θ near T_m than other values of $\theta \in \Theta$. This illustrates the adaptive learning capability of the discrete stochastic approximation algorithm.

We have also conducted simulations on how the tracking algorithms perform for different values of

ϵ . As expected, the smaller ϵ is, the less frequently the Markov chain $\{T_m\}$ jumps – hence the tracking algorithm performs better for smaller ϵ and vice versa – we refer the reader to our conference paper [3] for further numerical studies. Finally, we simulated the asynchronous version (19) of Algorithm 2 – it yielded almost identical results.

C. Example 3: Multipath Time-Varying (Rayleigh Fading) Channel

Here we consider a multipath time-varying (i.e. both time and frequency selective) wireless channel. The channel was implemented as a finite impulse response (FIR) filter with time-varying filter coefficients. Each propagation path i is characterized by a fixed delay τ_i and a time-varying amplitude $A_i(t)$. Here $A_i(t) = a_i g_i(t)$ where a_i denotes the amplitude and $g_i(t)$ denotes the Rayleigh fading process. The impulse response of the channel model can be given as

$$h(t, \tau) = \sum_{i=0}^{I-1} A_i(t) \delta(\tau - \tau_i) \quad (29)$$

where I is the total number of propagation paths. The fading function $g_i(t)$ is a complex Gaussian process that is independent for different paths. The function $g_i(t)$ is assumed to be normalized, i.e. $E\{|g_i(t)|^2\} = 1$.

The OFDM system and the channel parameters are as follows:

- Total number of OFDM subcarriers $N = 512$.
- The Cyclic-prefix length is $N_{\text{cp}} = 64$ samples.
- Each OFDM subcarrier is modulated using random data from a 16-QAM constellation.
- The wireless channel has $I = N_{\text{cp}}$ paths, with path delays $0, 1, \dots, N_{\text{cp}} - 1$ samples. The amplitude $a_i g_i(t)$ of each path varies independently of the others, according to a Rayleigh distribution with an exponential power-delay profile.

$$|a_i|^2 = \exp(-i/\tau_{\text{rms}}), \quad \text{for } i = 0, 1, \dots, N_{\text{cp}} - 1$$

where, the RMS delay-spread was set to $\tau_{\text{rms}} = 10$ samples. The phase shift on each path is uniformly distributed over $(0, 2\pi)$.

- The normalized frequency offset was set to $\Omega = 0.3$.

Thus the multipaths are sample-spaced and cover the whole cyclic-prefix length of the OFDM system.

Fig.12 shows a sample path of the time varying true delay for the above multipath fading channel. Note that this true delay evolves very similarly to the delay modelled as a slow Markov chain in Example 2. Fig.12 illustrates the tracking properties of the adaptive discrete stochastic approximation algorithm

for fixed step sizes of $\mu = 0.001$, $\mu = 0.01$ and $\mu = 0.05$. Again it can be seen that with larger step size, the algorithm tracks faster but jump around the true delay.

Fig.13 shows snapshots of the occupation probability vector $\hat{\alpha}_m$ (see Step 3 of Algorithm 2) for symbol times $m = 200$ and $m = 400$. The x -axis of these figures have been magnified around the $\hat{\theta}_m^*$ (the value of θ which has the highest occupancy probability $\hat{\alpha}_m(\theta)$) for clarity. As can be seen from Fig.12 and Fig.13, the algorithm satisfactorily tracks the rapidly time varying symbol timing $\{T_m\}$ of Fig.12. Also note that the algorithm spends substantially more time at the true symbol timing than any other value of θ – this is reflected by the fact that occupation probabilities $\hat{\alpha}_m(\theta)$ are substantially higher for θ near T_m than other values of $\theta \in \Theta$. This illustrates the adaptive learning capability of the discrete stochastic approximation algorithm.

VII. CONCLUSIONS AND EXTENSIONS

We have presented low-complexity discrete stochastic approximation algorithms for time and frequency synchronization of OFDM systems. Due to their recursive nature, the proposed algorithms are useful in slowly-varying channel situations, where the synchronization parameters (T and Ω) to be estimated, are slowly time-varying. Simulation results were presented to illustrate the performance of the algorithms.

Due to the fact that the received signals are Gaussian mixtures, our verification of the stochastic ordering conditions (C1) and (C2) for convergence of the algorithms required N_{cp} to be large. For such large N_{cp} we obtained much stronger results than needed by (C1) and (C2). It is of interest to examine how to verify these conditions for small N_{cp} . It is worthwhile to examine other recently proposed discrete stochastic approximation algorithms such as nested partition methods [23]. Finally, it is also of interest to examine the sensitivity of the OFDM system performance and bit error rate degradation due to time synchronization and frequency offset errors, see [19], [9].

REFERENCES

- [1] S. Andradottir. A global search method for discrete stochastic optimization. *SIAM J. Optimization*, 6(2):513–530, May 1996.
- [2] S. Andradottir. Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modelling and Computer Simulation*, 9(4):349–380, Oct. 1999.
- [3] C. Athaudage and V. Krishnamurthy. A low complexity timing and frequency synchronization algorithm for OFDM systems. In *Proc. IEEE Globecom*, pages 244–248, Nov. 2002.
- [4] J. V. D. Beek, M. Sandell, and P. O. Borjesson. ML estimation of time and frequency offset in OFDM systems. *IEEE Trans. Signal Proc.*, 45(7):1800–1805, July 1997.

- [5] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*, volume 22 of *Applications of Mathematics*. Springer-Verlag, 1990.
- [6] P. Bremaud. *Markov Chains*. Springer-Verlag, 1999.
- [7] P.W. Glynn and W. Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, 40(3), May-June 1992.
- [8] S. Haykin. *Adaptive Filter Theory*. Information and System Sciences Series. Prentice Hall, second edition, 1991.
- [9] T. Keller and L. Hanzo. Adaptive multicarrier modulation: A convenient framework for time-frequency processing in wireless communications. *Proceedings of the IEEE*, 88:611–642, May 2000.
- [10] T. Keller, L. Piazzo, P. Mandarini, and L. Hanzo. Orthogonal frequency division multiplex synchronization techniques for frequency selective fading channel. *IEEE J. Selected Areas Communications*, 19(6):999–1007, June 2001.
- [11] V. Krishnamurthy, X. Wang, and G. Yin. Spreading code optimization and adaptation in CDMA via discrete stochastic approximation. *IEEE Trans. Info Theory (submitted)*, 2002.
- [12] V. Krishnamurthy, X. Wang, and G. Yin. Adaptive spreading code optimization in multiantenna multipath fading channels in CDMA. In *Proc. IEEE Conference on Communications (ICC)*, May 2003.
- [13] V. Krishnamurthy and G. Yin. Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime. *IEEE Trans. Inform. Theory*, 48(2):458–476, February 2002.
- [14] H.J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, 1997.
- [15] M. Luise and R. Reggiannini. Carrier frequency acquisition and tracking for OFDM systems. *IEEE Trans. Commun.*, 44:1590–1598, Nov. 1996.
- [16] R. Nee and R. Prasad. *OFDM for Wireless Multimedia Communications*. Artech House, 2000.
- [17] G. Pflug. *Optimization of Stochastic Models: The Interface between simulation and Optimization*. Kluwer Academic Publishers, 1996.
- [18] M. Sandell, J. V. D. Beek, and P. O. Borjesson. Timing and frequency synchronization in OFDM systems using the cyclic prefix. In *Proc. International Symposium on Synchronization*, pages 16–19, Essen, Germany, Dec. 1995.
- [19] K. Santhanathan and C. Tellambura. Probability of error calculation of OFDM systems with frequency offset. *IEEE Trans. Commun.*, 49(11):1884–1888, 2001.
- [20] T.M. Schmidl and D.C. Cox. Robust frequency and timing synchronization for OFDM. *IEEE Trans. Commun.*, 45(12):1613–1621, Dec. 1997.
- [21] V. Solo and X. Kong. *Adaptive Signal Processing Algorithms – Stability and Performance*. Prentice Hall, N.J., 1995.
- [22] J. Spall. *Introduction to Stochastic Search and Optimization*. Wiley, 2003.
- [23] J.R. Swisher, Jacobson S, H, P.D. Hyden, and L.W. Schruben. A survey of simulation optimization techniques and procedures. In *Proc. 2000 Winter Simulation Conference*, Orlando, Florida, 2000.
- [24] E. Weinstein, M. Feder, and A.V. Oppenheim. Sequential algorithms for parameter estimation based on the Kullback-Leibler information measure. *IEEE Transactions ASSP*, 38(9):1652–1654, 1990.
- [25] E. Weinstein, A.V. Oppenheim, M. Feder, and J.R. Buck. Iterative and sequential algorithms for multisensor signal enhancement. *IEEE Transactions on Signal Processing*, 42(4):846–859, April 1994.
- [26] G. Yin, V. Krishnamurthy, and C. Ion. Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization. *SIAM J. Opt.*, 2004. to appear.
- [27] G. Yin and Q. Zhang. Singularly perturbed discrete-time Markov chains. *SIAM J. Applied Math*, 61:833–854, 2000.

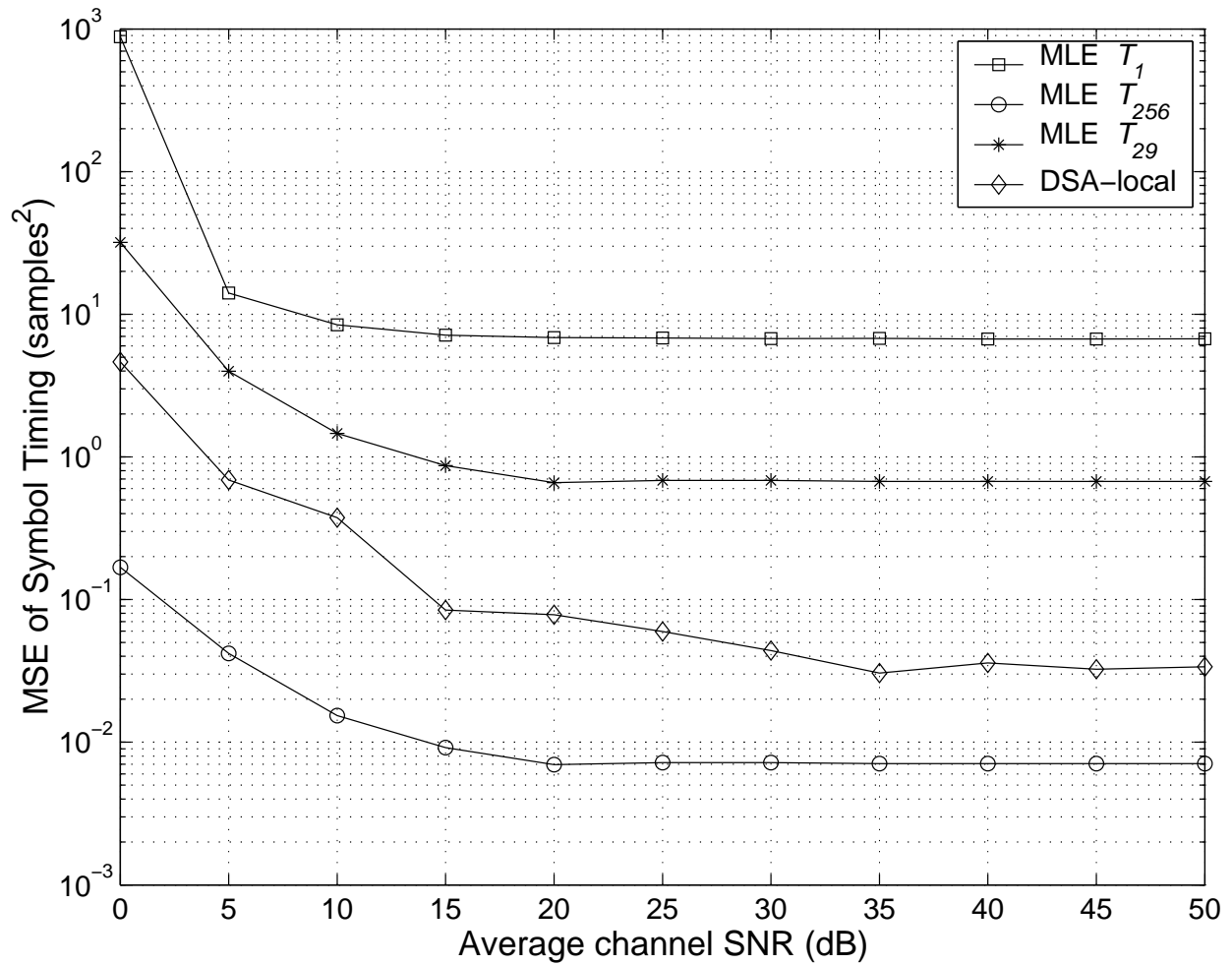


Fig. 7. Mean square symbol timing error versus the average channel SNR comparing the brute force MLE and discrete stochastic approximation Algorithm 1. The MLE estimate \hat{T}_{57} has the same computational complexity as the discrete stochastic approximation algorithm (DSA-local) – yet DSA-local yields an order of magnitude reduction in mean square error. This is due to the attraction (self learning) property of Algorithm 1.

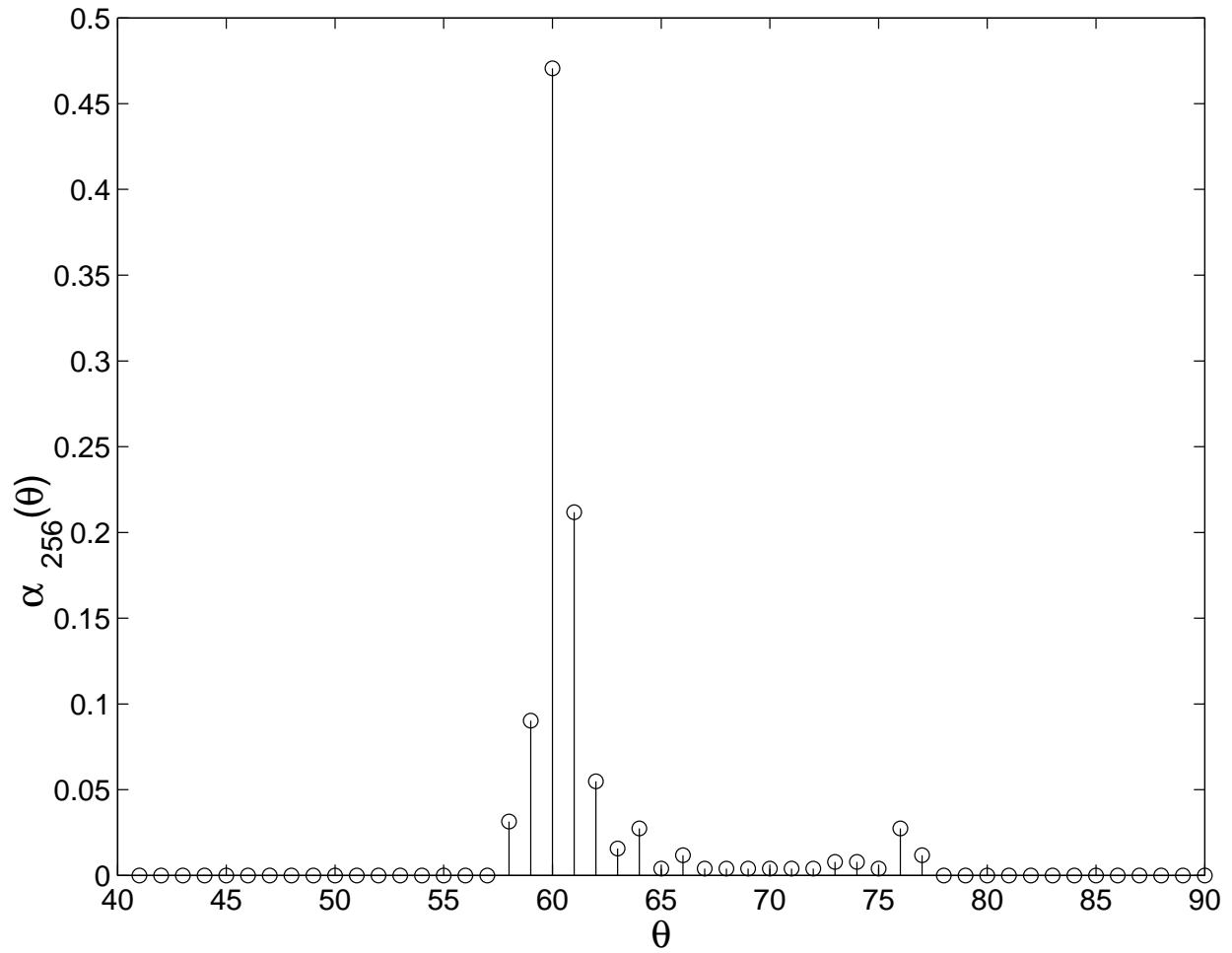


Fig. 8. State occupancy probabilities $\hat{\alpha}_m$ (see (11)) at symbol time $M = 256$ for SNR = 20 dB. For clarity only a subset of $\Theta = \{1 \leq \theta \leq 576\}$ containing high occupation probabilities is shown in the x-axis. These occupation probabilities illustrate the attraction property of Algorithm 1 – the algorithm spends more computational effort near the true time delay $T = 60$, than other values of $\theta \in \Theta$.

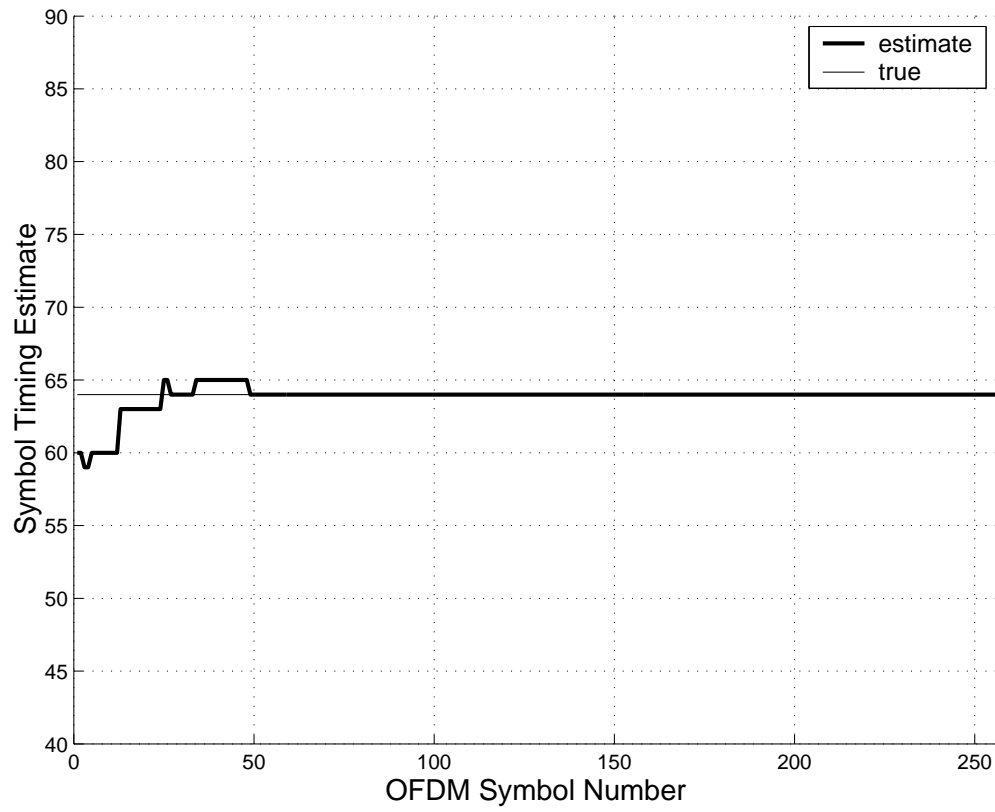


Fig. 9. Sample path time delay estimate $\hat{\theta}_m^*$ generated by locally convergent version of Algorithm 1 for SNR = 20 dB.

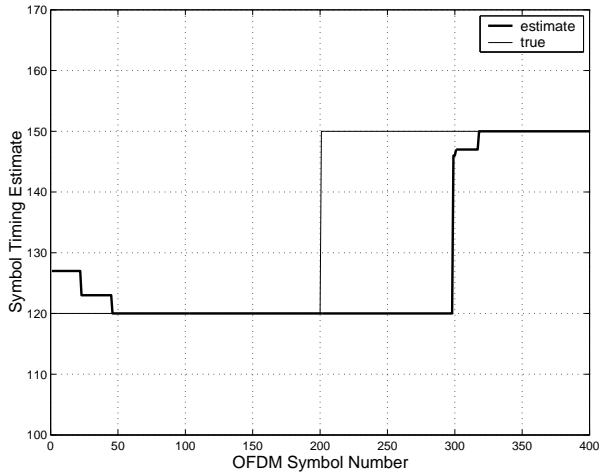
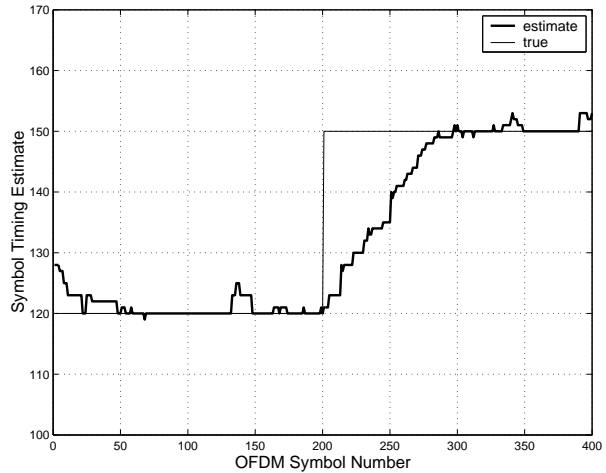
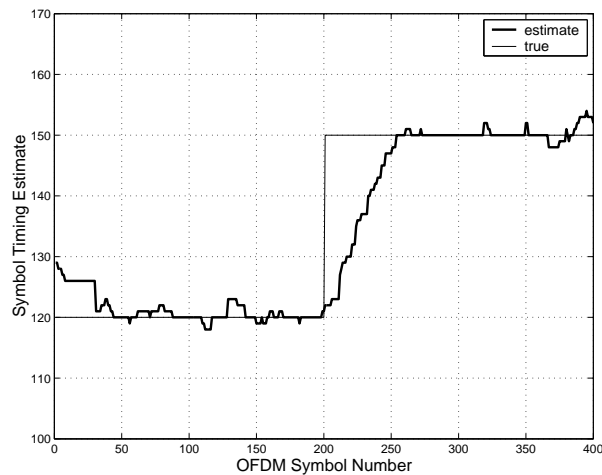
(a) $\mu = 0.001$ (b) $\mu = 0.02$ (c) $\mu = 0.03$

Fig. 10. Estimated time delay $\hat{\theta}_m^*$ generated by adaptive discrete stochastic approximation Algorithm 2 for SNR = 20 dB. Also shown is the sample path of time-varying true symbol timing sequence $\{T_m\}$. The sequence $\{T_m\}$ is modelled as a Markov chain with slow dynamics with transition probability matrix $\mathbf{P} = \mathbf{I} + \epsilon\mathbf{Q}$ given in Sec.VI-B and $\epsilon = 10^{-3}$.

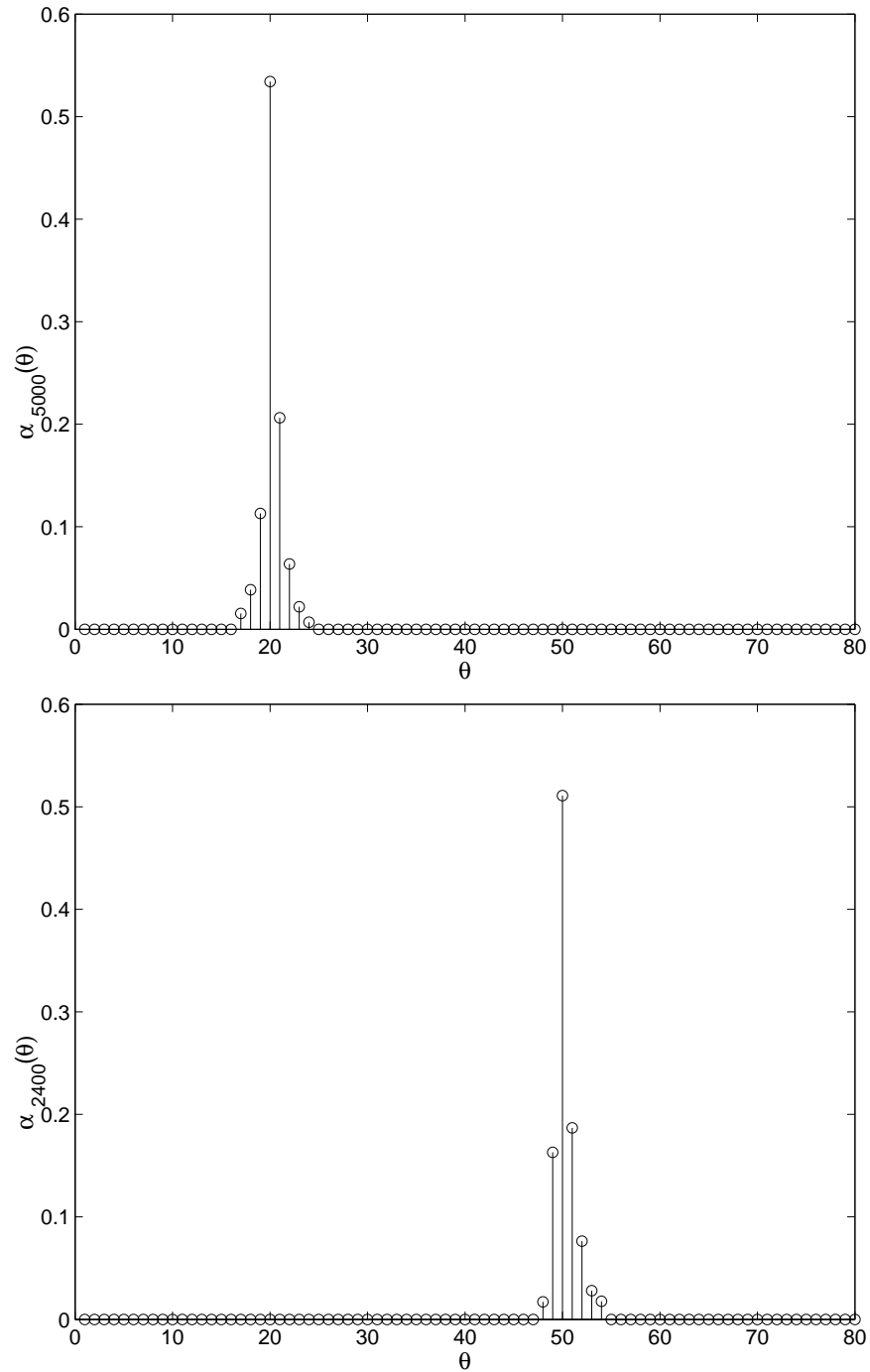


Fig. 11. State occupancy probabilities $\hat{\alpha}_m$ at symbol times $m = 2400$ and $m = 5000$. For clarity only a subset of $\Theta = \{1 \leq \theta \leq 576\}$ containing high occupation probabilities is shown in the x-axis. These two snapshots of the occupation probabilities illustrate the adaptive tracking capability of Algorithm 2 for tracking the time delay sequence $\{T_m\}$ of Fig.10.

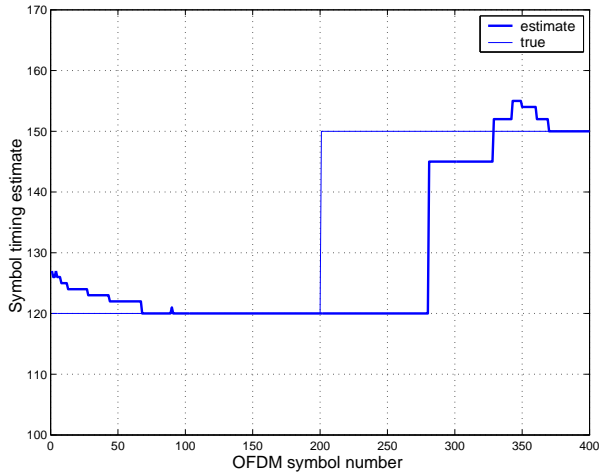
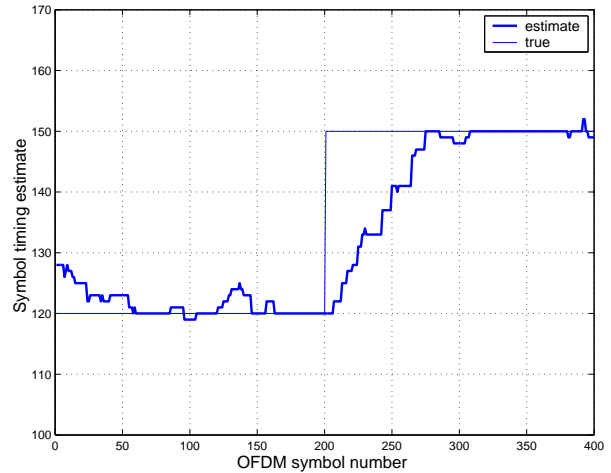
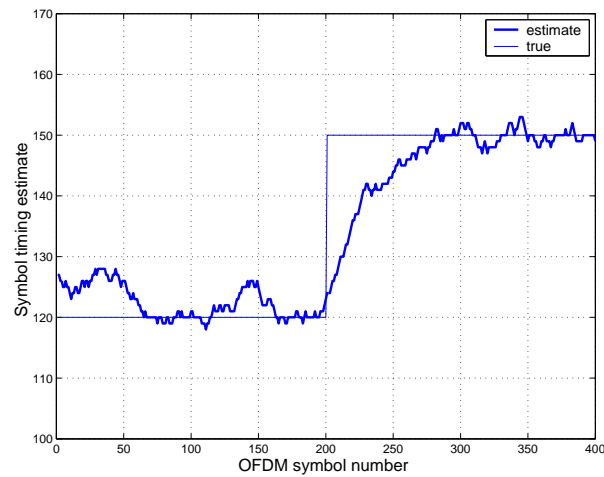
(a) $\mu = 0.001$ (b) $\mu = 0.02$ (c) $\mu = 0.03$

Fig. 12. Estimated time delay $\hat{\theta}_m^*$ generated by adaptive discrete stochastic approximation Algorithm 2. Also shown is the sample path of time-varying true symbol timing sequence $\{T_m\}$. The sequence $\{T_m\}$ is obtained by simulating the time-varying multipath channel with parameters described in Sec.VI-C.

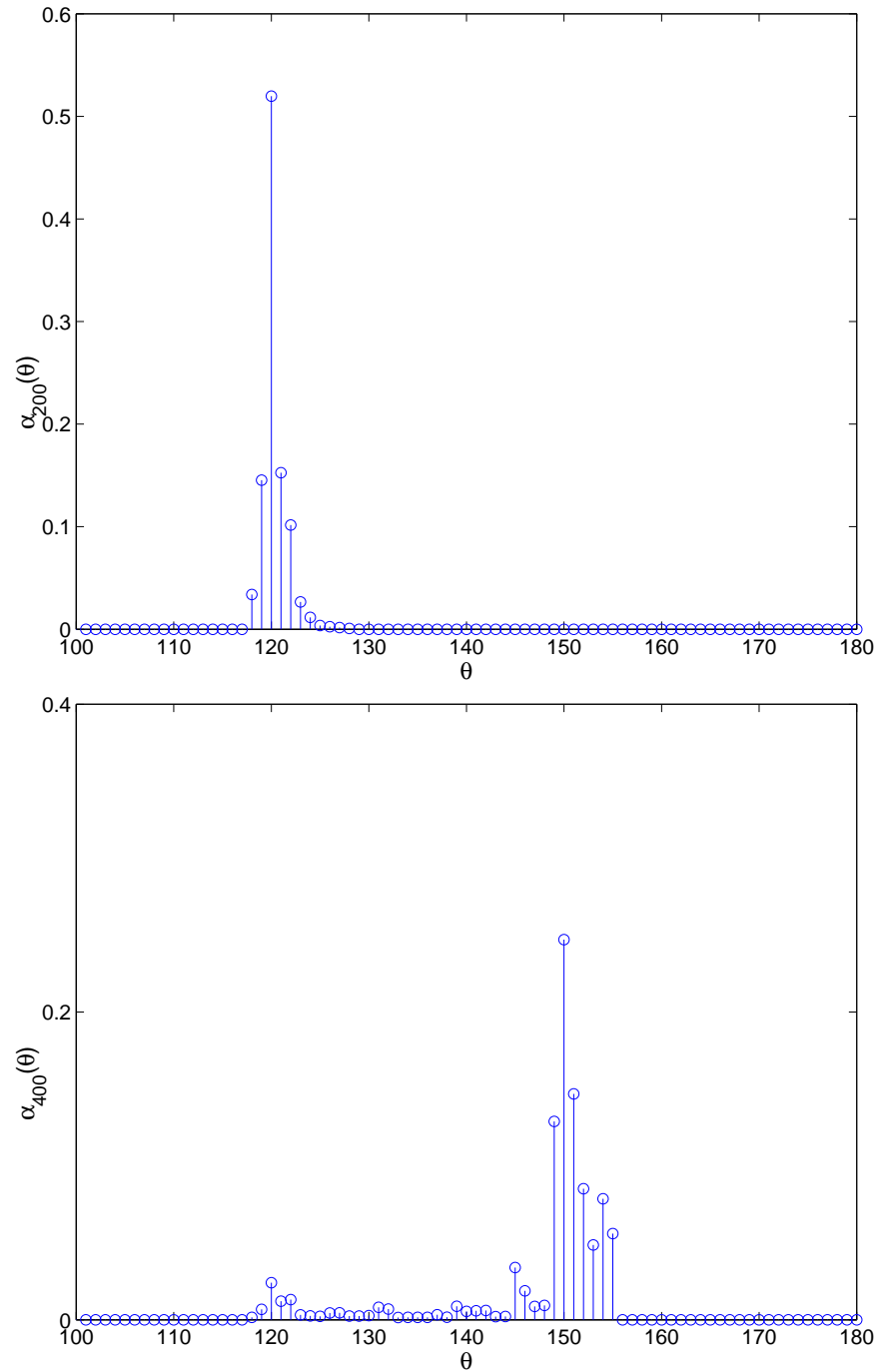


Fig. 13. State occupancy probabilities $\hat{\alpha}_m$ at symbol times $m = 200$ and $m = 400$. For clarity only a subset of $\Theta = \{1 \leq \theta \leq 576\}$ containing high occupation probabilities is shown in the x-axis. These two snapshots of the occupation probabilities illustrate the fast tracking capability of Algorithm 2 for tracking the time delay sequence $\{T_m\}$ of Fig.12 generated by a time-varying multipath channel.