REDUCED COMPLEXITY NLMS ALGORITHM FOR BLIND ADAPTIVE MULTIUSER DETECTION

Kutluyıl Doğançay and Oğuz Tanrıkulu

School of Electrical & Information Engineering
University of South Australia
Mawson Lakes, SA 5095, Australia
Email: kutluyil.dogancay@unisa.edu.au

Wireless Infrastructure Business Unit
Texas Instruments
Waltham 02451, MA, USA
Email: oguz.tanrikulu@ti.com

ABSTRACT

A reduced complexity normalized least-mean-square (NLMS) algorithm is presented for blind linear adaptive multiuser detection in synchronous direct-sequence CDMA systems. Selective partial updating is employed to reduce the computational complexity of the NLMS algorithm. The basic idea behind selective partial updating is to update only a small number of the adaptive filter coefficients at each iteration as identified by a selection criterion. It is shown that the reduced complexity selective partial update NLMS algorithm not only performs almost as well as the full-update NLMS algorithm in general, but in some cases it is also capable of outperforming the full-update algorithm. The performance of the proposed selective partial update algorithm is illustrated with computer simulations.

1. INTRODUCTION

Code division multiple access (CDMA) using direct sequence spreading is known to have certain advantages over time division multiple access (TDMA) and frequency division multiple access (FDMA) in terms of capacity, voice quality, cost effectiveness and handover performance. To make the most of these advantages, the CDMA systems must perform joint detection of multiple users. While optimal multiuser detection techniques are available [1], their computational complexity is prohibitive even for a moderate number of users. Adaptive suboptimal multiuser detection techniques provide a low-complexity solution with good performance. The adaptive nature of the detector allows the CDMA receiver to keep track of changes in the system such as those caused by users entering or exiting the system.

Several papers have been published in the literature on adaptive multiuser detection. The linear minimum mean-square-error (MMSE) detector that was proposed in [2, 3] can be used in training or blind mode. In blind mode, only the prior knowledge of the signatures and timing of the users to be detected is required by the receiver. The least-mean-square (LMS) algorithm for blind MMSE detection offers a simple, low-complexity solution to multiuser detection.

The concept of partial updating has been studied extensively (see e.g. [4, 5] and the references therein). In this paper we develop a new reduced complexity LMS blind multiuser detection algorithm employing selective partial updates. We show by simulation that in the context of blind adaptive multiuser detection, selective partial updating is capable of improving the performance of the LMS blind multiuser detection algorithm while reducing its complexity significantly.

2. BLIND ADAPTIVE LINEAR MULTIUSER DETECTION

The synchronous DS-CDMA signal model for a K-user system using binary phase shift keying (BPSK) modulation is

$$\boldsymbol{r}(i) = \sum_{k=1}^{K} A_k b_k(i) \boldsymbol{s}_k + \boldsymbol{n}(i) \tag{1}$$

where $\boldsymbol{r}(i)$ is the received signal vector at the ith symbol interval

$$\boldsymbol{r}(i) = [r_0(i), r_1(i), \ldots, r_{N-1}(i)]^T$$

N is the processing gain, A_k is the received complex signal amplitude for user k, $b_k(i) = \pm 1$ is the ith symbol of user k, $\boldsymbol{s}_k$ is the normalized signature (scrambling code) of user k ($\|\boldsymbol{s}_k\| = 1$)

$$\boldsymbol{s}_k = \frac{1}{\sqrt{N}} [c_{0,k}, c_{1,k}, \ldots, c_{N-1,k}]^T$$

and $\boldsymbol{n}(i) \sim \mathcal{N}_c(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_N)$ is the additive complex Gaussian channel noise. The signature signal $\boldsymbol{s}_k$ can be an m-sequence or Gold sequence with period N. The latter has better cross-correlation properties [6].

A linear receiver aims to detect individual users from the received signal $\boldsymbol{r}(i)$ by means of linear filtering. A linear receiver for one of the users, say user l ($1 \le l \le K$), has the following form [7]

$$\begin{aligned} z_l(i) &= \boldsymbol{w}_l^H \boldsymbol{r}(i) \\ \hat{b}_l(i) &= \operatorname{sign}(\Re\{A_l^* z_l(i)\}) \end{aligned}$$

where $\boldsymbol{w}_l$ is the linear receiver, $z_l(i)$ is the linear receiver output and $\hat{b}_l(i)$ is the detected symbol. The output of the linear receiver can be written as

$$\begin{aligned} z_l(i) = \underbrace{A_l(\boldsymbol{w}_l^H \boldsymbol{s}_l) b_l(i)}_{\text{desired signal}} + \underbrace{\sum_{\substack{1 \le k \le K \\ k \ne l}} A_k (\boldsymbol{w}_l^H \boldsymbol{s}_k) b_k(i)}_{\text{multiple access interference}} \\ + \underbrace{\boldsymbol{w}_l^H \boldsymbol{n}(i)}_{\text{noise}}. \end{aligned} \tag{2}$$

An important performance measure of multiuser detection is the signal-to-interference-plus-noise ratio (SINR) at the receiver output:

$$\operatorname{SINR}(\boldsymbol{w}_l) = \frac{|A_l(\boldsymbol{w}_l^H \boldsymbol{s}_l)|^2}{\sum_{\substack{1 \le k \le K \\ k \ne l}} |A_k(\boldsymbol{w}_l^H \boldsymbol{s}_k)|^2 + \sigma^2 \|\boldsymbol{w}_l\|^2}. \tag{3}$$

It is desirable to carry out multiuser detection in a blind mode (i.e., without the knowledge of other users' signatures) in order to save bandwidth especially in the downlink where the mobile users may not be aware of dynamic changes in the user traffic. A simple blind multiuser detection method can be formulated based on the solution of the following constrained optimization problem:

$$\boldsymbol{m}_l = \arg\min_{\substack{\boldsymbol{w}\in\mathbb{C}^N \\ \boldsymbol{w}^H\boldsymbol{s}_l=1}} E\{\|\boldsymbol{w}^H\boldsymbol{r}(i)\|^2\} \tag{4}$$

which is referred to as *minimum-output-energy* (MOE) detection [3]. In (4) $\boldsymbol{m}_l$ is the blind linear receiver that demodulates user l and is given by

$$\boldsymbol{m}_l = E^{-1}\{\boldsymbol{r}(i)\boldsymbol{r}^H(i)\}\boldsymbol{s}_l \tag{5}$$

which is termed the direct matrix inversion blind linear MMSE detector.

Decomposing $\boldsymbol{m}_l$ into orthogonal components

$$\boldsymbol{m}_l = \boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l, \quad \boldsymbol{s}_l \perp \boldsymbol{P}_l\boldsymbol{x}_l$$

with a projection matrix that is orthogonal to $\boldsymbol{s}_l$

$$\boldsymbol{P}_l = \boldsymbol{I}_N - \boldsymbol{s}_l\boldsymbol{s}_l^H$$

such that $\boldsymbol{P}_l\boldsymbol{m}_l = \boldsymbol{P}_l\boldsymbol{x}_l$, (4) can be re-written as an unconstrained optimization problem [7]:

$$\boldsymbol{m}_l = \boldsymbol{s}_l + \boldsymbol{P}_l \arg\min_{\boldsymbol{x}\in\mathbb{C}^N} E\{\|(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x})^H\boldsymbol{r}(i)\|^2\}. \tag{6}$$

The LMS algorithm solving (6) is given by [7]

$$\begin{aligned} \boldsymbol{x}_l(i+1) &= \boldsymbol{x}_l(i) - \mu_{\text{LMS}}\big((\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^H\boldsymbol{r}(i)\big)^* \\ &\quad \boldsymbol{P}_l\boldsymbol{r}(i) && \text{(7a)} \\ z_l(i) &= (\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^H\boldsymbol{r}(i) && \text{(7b)} \\ \hat{b}_l(i) &= \operatorname{sign}(\Re\{A_l^* z_l(i)\}) && \text{(7c)} \end{aligned}$$

where μ_{LMS} is the LMS step-size.

3. BLIND MULTIUSER NLMS ALGORITHM

For the sake of simplicity, we will assume that the A_k and $\boldsymbol{n}(i)$ are real-valued, and $A_k > 0$. The solution of the following constrained optimization problem then gives the normalized LMS (NLMS) algorithm for blind adaptive multiuser detection:

$$\min_{(\boldsymbol{s}_l+\boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i)=0} \|\boldsymbol{x}_l(i+1) - \boldsymbol{x}_l(i)\|^2. \tag{8}$$

This optimization problem aims to modify $\boldsymbol{x}_l(i)$ in a minimal fashion so that the new vector $\boldsymbol{x}_l(i+1)$ minimizes the output energy for the current input. Note that the constraint of the optimization problem given by $(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i) = 0$ can never be satisfied, which leads to a compromise solution that brings $(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i)$ close to zero as in MOE detection while minimizing the change in $\boldsymbol{x}_l(i)$. The above optimization problem will provide the basis for the selective partial update algorithm derived in the next section.

Using a Lagrange multiplier, the cost function to be minimized can be written as

$$\begin{aligned} J(\boldsymbol{x}_l(i+1)) &= (\boldsymbol{x}_l(i+1) - \boldsymbol{x}_l(i))^T(\boldsymbol{x}_l(i+1) - \boldsymbol{x}_l(i)) \\ &\quad + \lambda(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i). \end{aligned} \tag{9}$$

Taking the gradient with respect to $\boldsymbol{x}_l(i+1)$ and λ and setting it equal to zero gives

$$\begin{aligned} \frac{\partial J(\boldsymbol{x}_l(i+1))}{\partial \boldsymbol{x}_l(i+1)} &= 2\boldsymbol{x}_l(i+1) - 2\boldsymbol{x}_l(i) + \lambda\boldsymbol{P}_l\boldsymbol{r}(i) = \boldsymbol{0} && \text{(10a)} \\ \frac{\partial J(\boldsymbol{x}_l(i+1))}{\partial \lambda} &= (\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i) = 0. && \text{(10b)} \end{aligned}$$

Solving (10a) for $\boldsymbol{x}_l(i+1)$, we obtain

$$\boldsymbol{x}_l(i+1) = \boldsymbol{x}_l(i) - \frac{\lambda}{2}\boldsymbol{P}_l\boldsymbol{r}(i). \tag{11}$$

Substituting this into (10b) gives

$$\begin{aligned} \left(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i) - \frac{\lambda}{2}\boldsymbol{P}_l\boldsymbol{r}(i)\right)^T \boldsymbol{r}(i) &= 0 && \text{(12a)} \\ (\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i) - \frac{\lambda}{2}\boldsymbol{r}^T(i)\boldsymbol{P}_l\boldsymbol{r}(i) &= 0 && \text{(12b)} \\ \frac{\lambda}{2} &= \frac{(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i)}{\boldsymbol{r}^T(i)\boldsymbol{P}_l\boldsymbol{r}(i)}. && \text{(12c)} \end{aligned}$$

Note that for real A_k and $\boldsymbol{s}_k$, $\boldsymbol{P}_l^T = \boldsymbol{P}_l$ and $\boldsymbol{P}_l^2 = \boldsymbol{P}_l$. Using (11) and (12c), we obtain the following NLMS algorithm:

$$\boxed{\boldsymbol{x}_l(i+1) = \boldsymbol{x}_l(i) - \mu_{\text{NLMS}}\frac{z_l(i)\boldsymbol{P}_l\boldsymbol{r}(i)}{\boldsymbol{r}^T(i)\boldsymbol{P}_l\boldsymbol{r}(i)}} \tag{13}$$

where μ_{NLMS} is a step-size. The only difference between the LMS and NLMS algorithms is that the NLMS has a normalized step-size with normalization factor $\boldsymbol{r}^T(i)\boldsymbol{P}_l\boldsymbol{r}(i) = \|\boldsymbol{P}_l\boldsymbol{r}(i)\|^2$.

4. SELECTIVE PARTIAL UPDATING

To apply selective partial updating, segment $\boldsymbol{x}_l(i)$ and $\boldsymbol{P}_l$ into B equal-size sub-blocks:

$$\boldsymbol{x}_l(i) = \begin{bmatrix} \boldsymbol{x}_{l,1}(i) \\ \vdots \\ \boldsymbol{x}_{l,B}(i) \end{bmatrix}, \quad \boldsymbol{P}_l = [\boldsymbol{P}_{l,1} \quad \cdots \quad \boldsymbol{P}_{l,B}]. \tag{14}$$

The selective partial update (SPU) version of the NLMS algorithm is given by the solution of the following constrained optimization problem:

$$\min_{1\le j\le B} \min_{(\boldsymbol{s}_l+\boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i)=0} \|\boldsymbol{x}_{l,j}(i+1) - \boldsymbol{x}_{l,j}(i)\|^2. \tag{15}$$

The above optimization problem is motivated by the earlier work on selective partial updating [5]. For a given sub-block j, the cost function to be minimized is

$$\begin{aligned} J(\boldsymbol{x}_{l,j}(i+1)) &= \|\boldsymbol{x}_{l,j}(i+1) - \boldsymbol{x}_{l,j}(i)\|^2 \\ &\quad + \lambda(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i). \end{aligned} \tag{16}$$

Taking the gradient with respect to $\boldsymbol{x}_{l,j}(i+1)$ and λ and setting it to zero gives

$$\begin{aligned} \frac{\partial J(\boldsymbol{x}_{l,j}(i+1))}{\partial \boldsymbol{x}_{l,j}(i+1)} &= 2\boldsymbol{x}_{l,j}(i+1) - 2\boldsymbol{x}_{l,j}(i) \\ &\quad + \lambda\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i) = \boldsymbol{0} && \text{(17a)} \\ \frac{\partial J(\boldsymbol{x}_{l,j}(i+1))}{\partial \lambda} &= (\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i+1))^T\boldsymbol{r}(i) = 0. && \text{(17b)} \end{aligned}$$

Solving (17a) for $\boldsymbol{x}_{l,j}(i+1)$, we get

$$\boldsymbol{x}_{l,j}(i+1) = \boldsymbol{x}_{l,j}(i) - \frac{\lambda}{2}\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i). \tag{18}$$

Substituting this into (17b) gives

$$\left(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i) - \frac{\lambda}{2}\boldsymbol{P}_{l,j}\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i)\right)^T \boldsymbol{r}(i) = 0 \tag{19a}$$

$$(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i) - \frac{\lambda}{2}\boldsymbol{r}^T(i)\boldsymbol{P}_{l,j}\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i) = 0 \tag{19b}$$

$$\frac{\lambda}{2} = \frac{(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i)}{\boldsymbol{r}^T(i)\boldsymbol{P}_{l,j}\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i)}. \tag{19c}$$

Using (18) and (19c), we obtain the following adaptation algorithm:

$$\boldsymbol{x}_{l,j}(i+1) = \boldsymbol{x}_{l,j}(i) - \mu_{\mathrm{SPU}}\frac{(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i)}{\|\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i)\|^2}\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i) \tag{20}$$

where μ_{SPU} is a step-size.

The sub-block that has the minimum update among the B sub-blocks is

$$j = \arg\min_{1\le m\le B}\left\|\frac{(\boldsymbol{s}_l + \boldsymbol{P}_l\boldsymbol{x}_l(i))^T\boldsymbol{r}(i)}{\boldsymbol{r}^T(i)\boldsymbol{P}_{l,m}\boldsymbol{P}_{l,m}^T\boldsymbol{r}(i)}\boldsymbol{P}_{l,m}^T\boldsymbol{r}(i)\right\|^2 \tag{21a}$$

$$= \arg\min_{1\le m\le B}\frac{1}{\|\boldsymbol{P}_{l,m}^T\boldsymbol{r}(i)\|^2}. \tag{21b}$$

Thus, the sub-block selected for update at the ith iteration is simply given by

$$\boxed{j = \arg\max_{1\le m\le B}\|\boldsymbol{P}_{l,m}^T\boldsymbol{r}(i)\|^2.} \tag{22}$$

It was shown in [8] that replacing the normalization factor $\|\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i)\|^2$ in (20) with $\|\boldsymbol{P}_l\boldsymbol{r}(i)\|^2$ maximizes the convergence speed. Thus, the SPU-NLMS algorithm for blind multiuser detection is given by

$$\boxed{\begin{aligned}\boldsymbol{x}_{l,j}(i+1) &= \boldsymbol{x}_{l,j}(i) - \mu_{\mathrm{SPU}}\frac{z_l(i)\boldsymbol{P}_{l,j}^T\boldsymbol{r}(i)}{\|\boldsymbol{P}_l\boldsymbol{r}(i)\|^2},\\ j &= \arg\max_{1\le m\le B}\|\boldsymbol{P}_{l,m}^T\boldsymbol{r}(i)\|^2.\end{aligned}} \tag{23}$$

It is interesting to note that for $B = N$ the coefficient sub-blocks become individual coefficients. This is the preferred way of selective partial updating since it results in the fastest convergence for partial updates. The case of $B = 1$ is equivalent to the full-update NLMS. It is also possible to consider multiple sub-blocks for updating.

The computational complexity of the full-update NLMS algorithm in (13) is $3N$ multiplications for computing the output $z_l(i)$, and $4N$ multiplications and one division for the update term per iteration. The SPU-NLMS algorithm in (23) has the same complexity as the NLMS for computing $z_l(i)$, but requires $2N(1 + 1/B) \approx 2N$ multiplications and one division for the update term per iteration.

5. SEQUENTIAL PARTIAL UPDATING

A zero overhead alternative to selective partial updating is sequential partial updating [9]. Sequential partial updating does not select the coefficient sub-block to be updated in an "intelligent" way. It simply updates the sub-blocks in a sequential manner. The sequential-partial-update NLMS algorithm can be defined by simply replacing the selection criterion in (22) with

$$j = \{k + 1 : (i - k) \bmod B = 0,\ 0 \le k \le B - 1\}. \tag{24}$$

The convergence speed of a sequential-partial-update algorithm is $1/B$th that of the full-update algorithm, which is often far too slow.

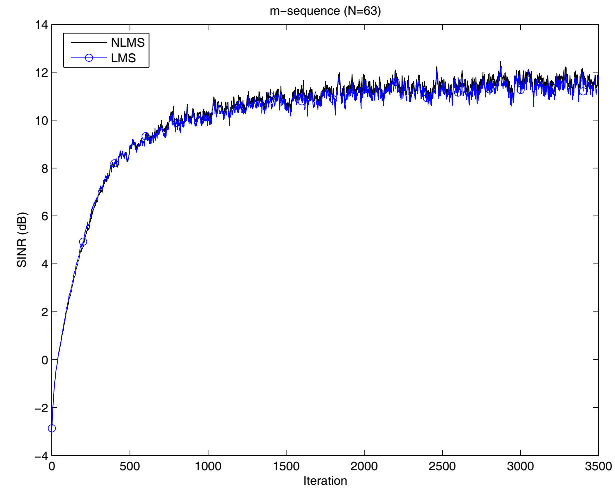

Figure 1: Comparison of LMS and NLMS algorithms.

6. SIMULATION STUDIES

In the simulations the signature sequences are short codes obtained from randomly selected m-sequences with period $N = 63$. The system has $K = 10$ users. The user to be detected is user 1 (i.e., $l = 1$). The system has six 10 dB multiple access interferences (MAIs) and three 20 dB MAIs, i.e., $A_k^2/A_1^2 = 10$ for $k = 2, \ldots, 7$, and $A_k^2/A_1^2 = 100$ for $k = 8, 9, 10$. The desired signal to ambient noise ratio is 20 dB. Fig. 1 shows the SINR performance of the LMS and NLMS algorithms for step-sizes $\mu_{\mathrm{LMS}} = 4.5 \times 10^{-4}$ and $\mu_{\mathrm{NLMS}} = 0.15$ averaged over 100 simulations. The blind LMS and NLMS multiuser detection algorithms are observed to have almost identical performance. In the case of trained multiuser detection the NLMS algorithm is known to converge faster than the LMS algorithm. For blind multiuser detection, this performance advantage appears to vanish.

Simulations were carried out to compare the performance of the full-update NLMS and SPU-NLMS algorithms for processing gains $N = 63$ and $N = 127$. The number of sub-blocks was set equal to the number of coefficients ($B = N$). This means that only one coefficient out of N is updated at each iteration. This is normally expected to yield the worst-case performance for partial updating. However it was observed that the SPU-NLMS algorithm not only performed as well as the full-update NLMS algorithm, but it also outperformed the full-update NLMS algorithm in some cases. The simulation results are shown in Figs. 2–4.

The convergence performance of the sequential-partial-update NLMS (Seq-NLMS) algorithm is compared with that of the SPU-NLMS in Fig. 5. Both algorithms update one out of 63 coefficients. The superior performance of selective partial updating is clearly visible.

7. CONCLUSION

A complexity reduction method was proposed for blind linear adaptive multiuser detection algorithms based on selec-

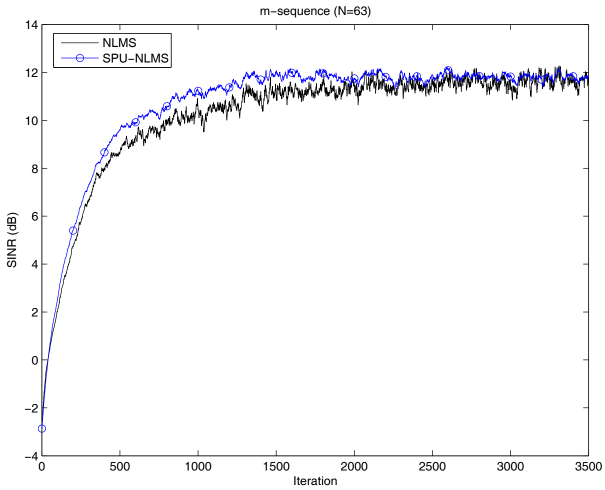

Figure 2: Comparison of full-update NLMS and SPU-NLMS for the same steady-state SINR.

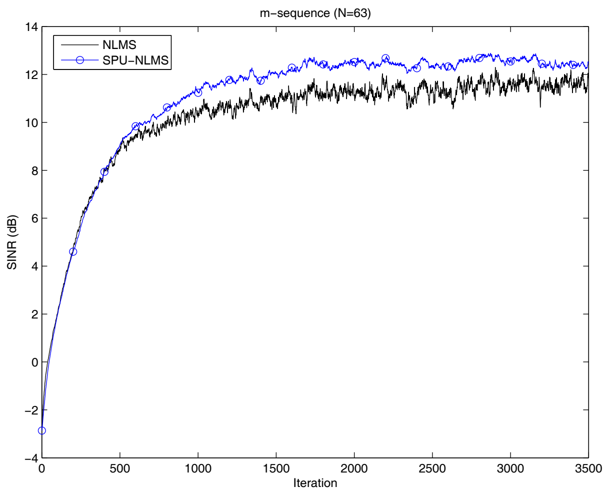

Figure 3: Comparison of full-update NLMS and SPU-NLMS for the same initial converge rate.

tive partial updating. Contrary to initial expectation, no significant performance loss was observed even when only one filter coefficient was updated out of N. In some cases, the convergence performance even improved compared with full updating. Future work will investigate the observed performance improvement by resorting to averaging analysis of the LMS cost function.

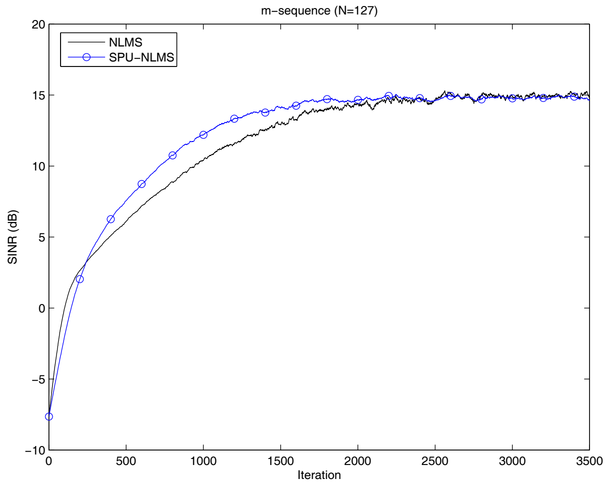

Figure 4: Comparison of full-update NLMS and SPU-NLMS for larger processing gain.

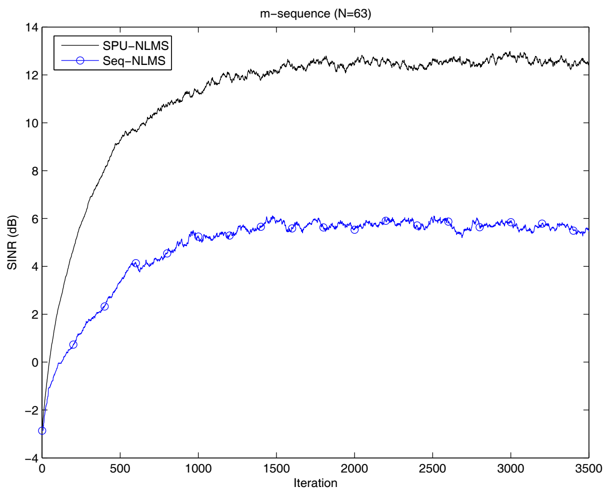

Figure 5: Comparison of SPU-NLMS and Seq-NLMS.

REFERENCES

[1] S. Verdu, "Minimum probability of error for asynchronous gaussian multiple-access channels," *IEEE Trans. on Information Theory*, vol. 32, no. 1, pp. 85–96, January 1986.

[2] U. Madhow and M. Honig, "MMSE interference suppression for direct-sequence spread-spectrum CDMA," *IEEE Trans. on Commun.*, vol. 42, no. 12, pp. 3178–3188, December 1994.

[3] M. L. Honig, U. Madhow, and S. Verdu, "Blind adaptive multiuser detection," *IEEE Trans. on Information Theory*, vol. 41, no. 4, pp. 944–960, July 1995.

[4] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. on Circuits and Systems II*, vol. 44, no. 3, pp. 209–216, March 1997.

[5] K. Doğançay and O. Tanrıkulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. on Circuits and Systems II*, vol. 48, no. 8, pp. 762–769, August 2001.

[6] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. on Information Theory*, vol. 13, pp. 619–621, October 1967.

[7] X. Wang and H. V. Poor, *Wireless Communication Systems: Advanced Techniques for Signal Reception*. Upper Saddle River, NJ: Prentice Hall, 2004.

[8] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Trans. on Signal Processing*, vol. 52, no. 4, pp. 938–949, April 2004.

[9] S. M. Kuo and J. Chen, "Multiple microphone acoustic echo cancellation system with the partial adaptive process," *Digital Signal Processing*, vol. 3, pp. 54–63, 1993.