Journal of Theoretical and Applied Information Technology
31st January 2016. Vol.83. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL LOW-COMPLEXITY SUB-BLOCK NLMS-DRMTA ALGORITHM FOR BLIND MULTIUSER DETECTION IN WCDMA SYSTEMS

RAED S.H. AL-MUSAWI
Department of Electronic and Communications Engineering, University of Kufa, Najaf, Iraq.
Email: Raaed.almusawi@uokufa.edu.iq

ABSTRACT

In this paper, we present a novel adaptive beamforming algorithm for multiuser detection in wideband code division multiple access (WCDMA) systems. The algorithm is based on a new sub-block normalized least mean squares optimization in conjunction with the despread-respread technique. We denote this new algorithm using the acronym Sub-Block NLMS-DRMTA, which stands for sub-block normalized least mean squares despread-respread multitarget array. A novel adaptive step size rule has been proposed based on the variance of the array input signal, which has resulted in enhanced robustness to estimation errors and very low computational complexity. The proposed algorithm has been benchmarked against exiting block-based and sample-based DRMTA algorithms. The bit error rate (BER) performance of our algorithm outperformed the performance of the block-based least squares despread-respread multitarget constant modulus array (LS-DRMTCMA) algorithm by a wide margin. In comparison to the sample-based (Sample NLMS-DRMTA) algorithm, the proposed algorithm achieved superior or similar BER performance at a markedly reduced computational complexity and significantly faster convergence. In addition, the proposed algorithm exhibited significantly better robustness compared to that of the sample-based recursive least squares (Sample RLS-DRMTA) algorithm at low signal-to-noise ratio (SNR) and/or high number of users. Moreover, the proposed algorithm has substantially lower complexity than the Sample RLS-DRMTA algorithm. Thus, the Sub-Block NLMS-DRMTA algorithm represents a viable approach for achieving both improved BER performance and fast convergence at very low computational complexity.

Keywords: *DRMTA, Smart Antennas, NLMS-DRMTA, CDMA Systems, Computational Complexity*

1. INTRODUCTION

Smart antenna technology has been widely employed in 3G and 4G networks as an effective means to increase the capacity of the cellular system and enhance service quality. Smart antenna systems use an adaptive beamforming algorithm to steer the main beam towards the target user and null in the directions of other users simultaneously accessing the channel. The adaptive algorithm can be used to update the weight vector on a specific time basis in a way that maximizes the received energy of the desired user signal, while minimizing multiuser interference. Significant effort has been dedicated to the design and development of novel adaptive algorithms characterized by improved bit error rate (BER) performance, faster convergence and reduced computational complexity [1-4].

In wideband code division multiple access (WCDMA) systems, all users share the same frequency through the wireless channel, resulting in co-channel interference from other active users. Thus, the adaptive algorithm (i.e., multi-target algorithm) should able to simultaneously identify and extract each desired user signal. Moreover, ideally, adaptive multi-target algorithms should be able to generate a reference signal at the receiver without using a training sequence, as this reduces the spectral efficiency of the system.

Many algorithms based on the despread-respread technique have been reported in extant literature. For sake of completeness, some pertinent works are briefly discussed here. Rong et al. [5,6] have proposed a couple of blind algorithms, namely least squares despread-respread multi-target array (LS-DRMTA) and least squares despread-respread multi-target constant modulus array (LS-DRMTCMA). These algorithms utilize the spreading sequence of each user to generate a reliable reference signal. However, neither LS-DRMTA nor LS-DRMTCMA algorithm achieve satisfactory performance in terms of BER and signal to interference noise ratio (SINR). As a result, neither is capable of effectively increasing the cellular system capacity. Thus, the main

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

advantage of these block-based algorithms is fast convergence.

Du et al. [7] developed the block recursive least squares despread-respread multitarget array (Block RLS-DRMTA) that uses a combination of sliding window weighting function, along with the variable step size steepest gradient method for updating the weight vector. The main advantage of this algorithm is faster convergence relative to the LS-DRMTA and LS-DRMTCMA algorithms. On the other hand, Block RLS-DRMTA can only achieve better performance when the wireless channel varies slowly over time. Moreover, the performance of this algorithm is closely related to the degree of cross-correlation among the different codes used in CDMA systems. The Block RLS-DRMTA is non-robust to estimation errors. Consequently, any error that occurs in estimating the received bit, either due to low level of SINR or due to high cross-correlation between users' codes, will propagate and significantly degrade the performance.

To increase the robustness of the Block RLS-DRMTA, Labib [8] proposed the Sample-based RLS-DRMTS, which updates the weight vector sample-by-sample instead of performing the block-based update. Updating the weight vector on a sample basis is posited to improve the performance in dynamic channels. Furthermore, the robustness would be increased, as any error that occurs would propagate through a specific number of sample indices, instead of through the same number of block (bit) indices. However, these improvements come at the expense of greatly increased computational complexity.

Labib et al. [9] proposed another algorithm for applications in CDMA systems, namely Block Affine Projection-DRMTA (Block APA-DRMTA), which also uses the despread-respread technique to generate the reference signal, followed by the Affine Projection Algorithm for updating the weight vector. The main advantages of this algorithm stem from good performance in dynamic channels and robustness against changes in the mobile network. Moreover, it is characterized by lower computational complexity compared to the Sample RLS-DRMTA, but exceeds that of the Block RLS-DRMTA. The main drawback of this algorithm is slower convergence as compared to both the Block RLS-DRMTA and the Sample-DRMTA algorithms.

Labib [8] proposed a Block-based Conjugate Gradient-DRMTA (Block CG-DRMTA) algorithm that updates the weight vector on bit-by-bit basis. The main benefits of this algorithm are low computational complexity and robustness to changes in the wireless network. However, this algorithm suffers from slow convergence rate.

Mohanned [10] proposed a Sample-based Normalized Least Mean Square-DRMTA (Sample NLMS-DRMTA) algorithm, which uses two variable step sizes to update the weight vector on a sample basis. This algorithm exhibits very good robustness to changes in the mobile network and is characterized by enhanced tracking capabilities in a dynamic channel. The main advantage of the Sample NLM-DRMTA is its significantly better BER performance relative to the aforementioned alternatives. In addition, it is less computationally complex than the Sample RLS-DRMTA, but inferior to the block-based algorithms. The drawback of this algorithm is slower convergence rate than that achieved by the Sample RLS-DRMTA.

Yasin et al. [11] used the concepts Kernel Affine Projection Algorithm (KAPA) proposed in [12] for adaptive beamforming in smart antenna systems. In their work, the authors presented comparison analyses between KAPA and RLS algorithms in terms of normalized array factor and mean square error performance. In [13], authors proposed Kalman filter based adaptive beamforming algorithms, including Kalman LMS, Kalman RLS and Conjugate Gradient Method (CGM). Their findings demonstrated that the CGM algorithm has a low computational complexity and good convergence rate. Reddy et al. [14] proposed an improved LMS adaptive beamforming algorithm using the windowing technique adopted from digital signal processing theory. Using this widowing technique, the authors reported simulation results indicating enhancements in the half-power beam width, leading to improvement in mitigating co-channel interferences.

In this study, we propose a new blind adaptive beamforming algorithm for multitarget detection in CDMA systems. The proposed algorithm utilizes the despread-respread technique to generate the reference signal before applying a sub-block version of the NLMS algorithm to generate the weight vector. To speed up the convergence of the NLMS algorithm, each block of data is divided into smaller blocks, as in the case of the Affine Projection algorithm, allowing a low-complexity NLMS to be applied to each of the smaller blocks to generate the weight vector. The proposed algorithm, denoted as Sub-Block NLMS-DRMTA, is expected to achieve fast convergence and significantly lower computational complexity than

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the Sample NLMS-DRMTA and Sample RLS-DRMTA, along with enhanced BER performance.

This paper is organized as follows. Following a brief overview of extant literature, which was presented above, in order to reveal the state of the current research efforts and provide context for the present study, Section 2 provides an overview of the system model, and briefly explains the basics of the smart antenna system. The new proposed blind multitarget algorithm is presented in Section 3, along with several existing algorithms considered in this study for comparison. Finally, Section 4 discusses the simulation results, and study conclusions are given in Section 5.

2. SIGNAL MODEL

In this study, we assume a direct sequence DS-CDMA system with N active users. A uniform linear array (ULA) antenna of M elements and inter-element spacing denoted as d are located at the base station. We further assume presence of N user signals incident upon the ULA antenna with direction of arrivals (DOAs) vector expressed as $\boldsymbol{\theta} = [\theta_0, \dots, \theta_{N-1}]^T$. The complex envelope representation of the $M \times 1$ input vector $\mathbf{x}(t)$ is given by [9]

$$\mathbf{x}(t) = \sum_{i=1}^{N} \mathbf{a}(\theta_i) s_i(t) + \mathbf{n}(t) \qquad (1)$$

where $\mathbf{n}(t)$ is the $M \times 1$ additive white Gaussian noise (AWGN) vector and $\mathbf{a}(\theta_i)$ is the $M \times 1$ direction vector of the ULA, expressed as:

$$\overline{\boldsymbol{a}}(\theta) = \left[1 \; e^{j(kd\,sin(\theta))} \dots e^{j(M-1)(kd\,sin(\theta))}\right]^T \qquad (2)$$

where $k = 2\pi/\lambda$ is the phase propagation factor, λ is the wavelength, and $d = \lambda/2$ is the inter-element spacing. The superscript T denotes the transpose and $s_i(t)$ is the i^{th} user signal whose complex envelope representation is expressed as [9]:

$$s_i(t) = \sqrt{2P_i} c_i(t-\tau_i) b_i(t-\tau_i) e^{-j\varphi_i} \qquad (3)$$

where P_i is the signal power, $c_i(t)$ is the spreading waveform, $b_i(t)$ is the information waveform, τ_i is the time delay and φ_i is the phase of user i signal.

Let us define $\mathbf{X}(n)$, which is the block of the array input signal during the n^{th} bit and is obtained by collecting K samples during the n^{th} bit. The block size of K is equal to the length of the spreading code multiplied by the number of samples per chip. Hence, $\mathbf{X}(n)$ is $M \times K$ matrix which can be represented as:

$$\mathbf{X}(n) = \left[\mathbf{x}(1+nK), \mathbf{x}(2+nK), \dots, \mathbf{x}\big((1+n)K\big)\right] \qquad (4)$$

The output of the array antenna for the i^{th} user is given by (as shown in Figure 1):

$$\boldsymbol{y}_i(n) = \boldsymbol{\omega}_i^H(n)\mathbf{X}(n) \qquad (5)$$

where $\boldsymbol{\omega}_i$ is the $M \times 1$ weight vector of the i^{th} user and the superscript H denotes the Hermitian transpose of the vector.

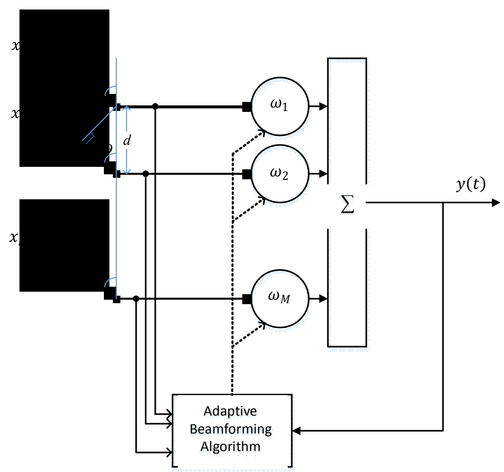

Figure 1: Smart Antenna System.

3. DRMTA ALGORITHMS

The main objective of DRMTA algorithms is to generate the reference signal using the despread-respread technique. In order to detect the i^{th} user data bit, the received signal is despread by multiplying it by the time-delayed version of the spreading code of user i, $c_i(t-\tau_i)$, allowing the bit decision to be made. Finally, a reliable reference signal is generated by respreading the estimated data bit through multiplying it by the same delayed version of the spreading code $c_i(t-\tau_i)$ of the i^{th} user.

3.1 Block LS-DRMTCMA

The block-based LS-DRMTCMA [6] is an improvement with respect to LS-DRMTA [5], achieved through combining the constant modulus (CM) property of the output signal with the despread-respread technique. The LS-DRMTCMA algorithm attempts to adapt the weight vector ω_i to minimize the cost function $J(\boldsymbol{\omega}_i(n))$, expressed as [6]:

$$J(\boldsymbol{\omega}_i(n)) = \|\boldsymbol{y}_i(n) - \boldsymbol{r}_i(n)\|^2 = \|\boldsymbol{\omega}_i^H(n)\mathbf{X}(n) - \boldsymbol{r}_i(n)\|^2 \qquad (6)$$

where $\boldsymbol{r}_i(n)$ is the reference signal generated during the n^{th} bit using the despread-respread technique and it is a column vector of M samples. The LS-DRMTCMA algorithm can be described via the following set of equations:

$$\boldsymbol{y}_i(n) = \boldsymbol{\omega}_i^H(n)\mathbf{X}(n) \qquad (7)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\hat{b}_{i,n} = \text{sgn}\left(\text{Re}(\boldsymbol{y}_i(n)\boldsymbol{c}_i^H)\right) \quad (8)$$

$$\boldsymbol{r}_i(n) = \alpha_{PN}\, \boldsymbol{r}_{i,PN}(n) + (1-\alpha_{PN})\, \boldsymbol{r}_{i,CM}(n) \quad (9)$$

where α_{PN} is a positive constant less than 1 and $\boldsymbol{r}_{i,PN}$ is expressed as:

$$\boldsymbol{r}_{i,PN}(n) = \hat{b}_{i,n}\boldsymbol{c}_i \quad (10)$$

with $\boldsymbol{r}_{i,CM}$ denoting the constant modulus property, given by:

$$\boldsymbol{r}_{i,CM}(n) = \left[\frac{\boldsymbol{y}_i(1+nK)}{|\boldsymbol{y}_i(1+nK)|}\ \frac{\boldsymbol{y}_i(2+nK)}{|\boldsymbol{y}_i(2+nK)|} \cdots \frac{\boldsymbol{y}_i(K+nK)}{|\boldsymbol{y}_i(K+nK)|}\right] \quad (11)$$

Finally, the new weight vector is given by:

$$\boldsymbol{\omega}_i(n+1) = [\mathbf{X}(n)\mathbf{X}^H(n)]^{-1}\mathbf{X}(n)\boldsymbol{r}_i^H(n) \quad (12)$$

where $\hat{b}_{i,n}$ denotes the estimated n^{th} bit, $\boldsymbol{c}_i$ is the spreading sequence of the length of K samples and sgn[·] denotes the sign function.

3.2 Sample RLS-DRMTA

The objective of the sample-based RLS-DRMTA [10] is to minimize the cost function

$$J(\boldsymbol{\omega}_i(h)) = \sum_{j=1}^{h} \lambda^{h-j}\|\boldsymbol{\omega}_i^H(j)\mathbf{x}(j) - \boldsymbol{r}_i(j)\|^2 \quad (13)$$

where h is the sample index, $\boldsymbol{\omega}_i$ is the $M \times 1$ weight vector of user i and $\mathbf{x}(h)$ is the $M \times 1$ input data vector, with M denoting the number of antenna elements. Finally, λ is the forgetting factor that weighs the past input data at the antenna array and is set to 0.99 in our simulations.

The following set of equations describes the algorithm:

Compute the autocorrelation matrix $\mathbf{R}$ for the input data as:

$$\mathbf{R}(h+1) = \lambda\mathbf{R}(h) + \mathbf{x}(h)\mathbf{x}^H(h) \quad (14)$$

Compute the cross-correlation vector z between the input data and the reference signal using the expression:

$$\boldsymbol{z}_i(h+1) = \lambda\boldsymbol{z}_i(h) + \mathbf{x}(h)\mathbf{r}_i^H(h) \quad (15)$$

Update the error vector $\boldsymbol{e}_i$ by applying the following equation:

$$\boldsymbol{e}_i(h+1) = \{\lambda\mathbf{I} - \mu_i(h)\mathbf{R}(h)\}\boldsymbol{e}_i(h) + \mathbf{x}(h)\{\boldsymbol{\omega}_i^H\mathbf{x}_i(h) - \mathbf{r}_i(h)\} \quad (16)$$

Compute the variable step size μ using the expression:

$$\mu_i(h+1) = \frac{\mathbf{e}_i^H(h+1)\boldsymbol{e}_i(h+1)}{\mathbf{e}_i^H(h+1)\mathbf{R}(h+1)\boldsymbol{e}_i(h+1)} \quad (17)$$

Update the next-sample weight vector as:

$$\boldsymbol{\omega}_i(h+1) = \{\mathbf{I} - \mu_i(h+1)\mathbf{R}(h+1)\}\boldsymbol{\omega}_i(h) + \mu_i(h+1)\boldsymbol{z}_i(h+1) \quad (18)$$

3.3 Sample-based NLMS-DRMTA

The Sample-based NLMS-DRMTA [8] aims to adapt the weight vector to minimize the cost function

$$J(\boldsymbol{\omega}_i(h)) = \sum_{j=1}^{h}\|\boldsymbol{\omega}_i^H(j)\mathbf{x}(j) - \boldsymbol{r}_i(j)\|^2 \quad (19)$$

where h is the sample index.

The Sample NLMS-DRMTA employs two variable step sizes to ensure smoother convergence and achieve enhanced BER performance. The first step size, $\mathbf{x}_{max}$, is a $1\times M$ vector whose entries represent the maximum value in each row of the array input signal $\mathbf{X}(n)$ during the n^{th} bit, expressed as:

$$\mathbf{x}_{max}(n) = \left[\text{MAX}\left(\mathbf{X}(1,K)\right), \text{MAX}\left(\mathbf{X}(2,K)\right), \ldots, \text{MAX}\left(\mathbf{X}(N,K)\right)\right] \quad (20)$$

The second variable step size $u_i(h)$ is inversely proportional to the energy of the output signal $\boldsymbol{y}_i(h)$ at the h^{th} sample and is evaluated as:

$$u_i(h) = \frac{1}{\boldsymbol{y}_i(h) * \boldsymbol{y}_i(h)^H + \mu_i(n)} \quad (21)$$

where $\mu_i(n)$ is given by:

$$\mu_i(n) = \frac{1}{\text{MAX}(\boldsymbol{y}_i(n))} \quad (22)$$

Moreover, $\boldsymbol{y}_i(n)$ is the output of the array antenna during the n^{th} bit, given in Eq. (5), whereas $\boldsymbol{y}_i(h)$ is the output obtained during processing of the h^{th} sample, expressed as:

$$\boldsymbol{y}_i(h) = \boldsymbol{\omega}_i{}^H(h)\,\mathbf{x}(h) \quad (23)$$

Next, the error signal is generated by adopting:

$$\boldsymbol{e}_i(h) = \boldsymbol{r}_i(h) - \boldsymbol{y}_i(h) \quad (24)$$

where $\boldsymbol{r}_i$ is the reference signal generated using Eq. (10). Finally, the new weight vector for the next sample is given by:

$$\boldsymbol{\omega}_i(h+1) = \boldsymbol{\omega}_i(h) + u_i(h) * \frac{e_i{}^H(h)}{\mathbf{x}_{\max}(n) * \mathbf{x}_{max}^H(n)} * \mathbf{x}(h) \quad (25)$$

3.4 Sub-block NLMS-DRMTA

In this section, we propose a new adaptive multiuser algorithm for applications in CDMA systems. We denote this new algorithm Sub-block Normalized Least Mean Squares-DRMTA (Sub-block NLMS-DRMTA). As the name implies, the new algorithm employs the despread-respread technique to generate the reference signal before applying a simplified version of the NLMS algorithm to generate the weight vector. First, the reference signal $r_i(n)$ is generated, as in the case of the Sample NLMS-DRMTA algorithm, after which $r_i(n)$ is divided into smaller blocks in order to apply a low-complexity NLMS optimization to

Journal of Theoretical and Applied Information Technology
31st January 2016. Vol.83. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

each of the smaller blocks and finally generate the weight vector $\omega_i(n)$. Figure 2 shows the block diagram of the Sub-block NLMS-DRMTA algorithm for user i.

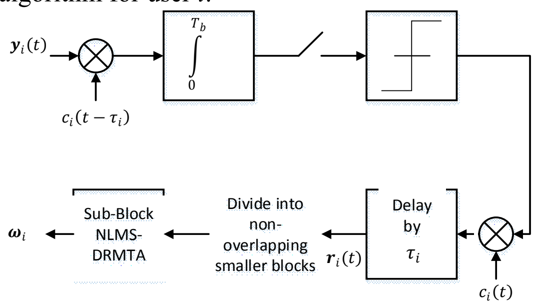

Figure 2: Block diagram of The Proposed Sub-block NLMS-DRMTA Algorithm.

The array input signal $\mathbf{X}$, given by Eq. (4), is divided into smaller blocks, each of size $p = K/P$, where P is the number of blocks in each bit. This division forms the matrix $\mathbf{G}_j$ of size $M \times p$. $\mathbf{G}_1$ is formed as:

$$\mathbf{G}_1 = [\mathbf{x}(p), \mathbf{x}(p-1), \dots, \mathbf{x}(1)\,] \tag{26}$$

Several approaches can be adopted to form the next matrix $\mathbf{G}_2$, which will determine the algorithm performance, convergence behavior, as well as computational complexity. In the approach employed here, the next block is formed by changing the entire block and taking the next p samples without any overlaps between adjacent blocks. Our experiments have revealed that this method results in the best compromise between performance and computational complexity. Thus, $\mathbf{G}_2$ is given as:

$$\mathbf{G}_2 = [\mathbf{x}(2p), \mathbf{x}(2p-1), \dots, \mathbf{x}(p+1)\,] \tag{27}$$

and $\mathbf{G}_j$ is generated as:

$$\mathbf{G}_j = [\mathbf{x}(jp), \mathbf{x}(jp-1), \dots, \mathbf{x}(jp-p+1)\,] \tag{28}$$

where $j = 1, 2, \dots, P$. Our simulation experiments have revealed that increasing P would lead to a higher computational complexity without significant improvements in the system performance. On the other hand, small P would result in a very low computational complexity without sacrificing performance or reducing the convergence speed. It is also worth noting that, if $P = 1$, $\mathbf{G}_1$ would represent the array input signal $\mathbf{X}(n)$ during the n^{th} bit, whereby each bit is represented by a single block. In this case, the weight vector is updated once per every bit and the algorithm is referred to as block-based.

During the processing of the n^{th} bit, the reference signal is generated as in Eq. (10), and the algorithm can be summarized with the following steps:

$$\mathbf{r}_{i,j} = [r_i(jp), r_i(jp-1), \dots, r_i(jp-p+1)\,] \tag{29}$$

$$\mathbf{y}_{i,j} = \omega_{i,j-1}^H \mathbf{G}_j \tag{30}$$

$$\mathbf{e}_{i,j} = \mathbf{r}_{i,j} - \mathbf{y}_{i,j} \tag{31}$$

$$\mathbf{g}_{i,j} = \mathrm{Var}(\mathbf{G}_j) \tag{32}$$

where $\mathbf{g}_{i,j}$ is the $1 \times M$ vector computed through evaluation the variance of each column of $\mathbf{G}_j$,

$$\mu_{i,j} = \alpha \frac{\sqrt{\mathbf{g}_{i,j}}}{\mathbf{g}_{i,j}\mathbf{g}_{i,j}^H} \tag{33}$$

$$\boldsymbol{\omega}_{i,j} = \boldsymbol{\omega}_{i,j-1} + \mu_{i,j}\mathbf{G}_j\mathbf{e}_{i,j}^H \tag{34}$$

where $\boldsymbol{\omega}_{i,j}$ is the weight vector for user i pertaining to iteration j. The last weight vector $\boldsymbol{\omega}_{i,j}$ is used with the incoming block $\mathbf{G}_{j+1}$ to form the next output vector $\mathbf{y}_{i,j+1}$ during the processing of the n^{th} bit. At the next bit, the last updated weight vector is used to calculate the beamformer output. The constant α is an optimization factor introduced to adjust the step size μ in order to guarantee both smooth convergence and maximum robustness to estimation errors for each value of P. Based on our simulations, the optimum value of α is 2 and 0.5, for $P = 1$ and $P = 2$, respectively.

4. SIMULATION RESULTS

In this section, we will investigate the performance of the proposed algorithm and study the performance impact of the number of blocks P in an AWGN channel. To demonstrate the effectiveness of the proposed algorithm in comparison to existing alternatives considered in this study, we have performed several simulation experiments, allowing us to measure and compare important performance metrics. As a part of this endeavor, we evaluated the BER performance, convergence speed, and computational complexity of each algorithm. The findings enable us to highlight the advantages and shortcomings of the proposed algorithm in comparison to LS-DRMTCMA, Sample RLS-DRMTA and Sample NLMS-DRMTA. In our simulation experiments, we assumed a ULA of eight elements with $d = \lambda/2$ denoting the inter-element spacing. Each user is assigned a Walsh code spreading sequence of length equal to 16 chips per bit. The sampling rate is taken to be 4 samples per chip, resulting in $K = 64$ total number of samples per bit. In addition, perfect power control is assumed, and each user is assigned a unitary power. The final assumption is that no multipath or fading effects in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the channel exist and the time delay of each user is perfectly estimated.

4.1 Experiment 1 − BER Evaluation

In this experiment, we evaluate the BER performance of the proposed algorithm against SNR for a couple of values of $P \in \{1, 2\}$, as shown in Figure 3. The number of users is assumed to be twice the number of the array elements. It can be seen from Figure 3 that, as the SNR increases, the BER decreases. More importantly, nearly similar BER performance is observed for $P \in \{1, 2\}$ over the entire range of SNR considered in this experiment, i.e., $-4 \leq \text{SNR} \leq 8$ dB. Hence, splitting the bit into two blocks $(P = 2)$ would not effectively improve the BER performance. Consequently, it is logical to choose $P = 1$ because it results in the lowest computational complexity of the proposed algorithm. As previously noted, in this case, the algorithm is referred to as block-based.

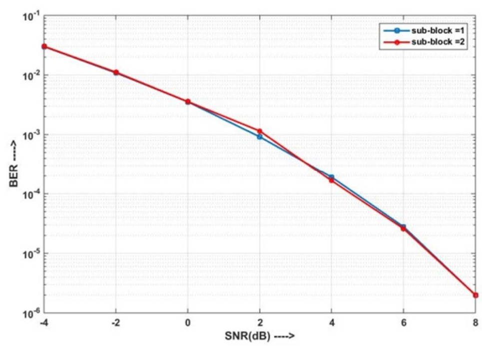

Figure 3: BER vs. SNR for the Sub-block NLMS-DRMTA in AWGN with $N = 16$ users.

Figure 4 shows BER performance versus number of users in an AWGN channel when the SNR is equal to 4 dB. Similarly, to investigate the impact of the number of sub-blocks, the BER is evaluated at two values of $P \in \{1, 2\}$. It can be seen that, as the number of users increases, the BER increases as well. When the number of users ranges from 8 to 12, both values of P result in the same BER, with somewhat superior performance in the case of $P = 1$. On the other hand, when the number of users exceeds 12, $P = 2$ is more advantageous, as it results in lower BER.

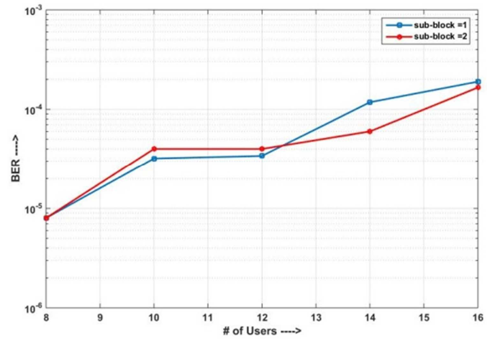

Figure 4: BER vs. No. of Users in AWGN with SNR = 4 dB.

4.2 Experiment 2 – Convergence-Time Evaluation

In order to demonstrate the convergence behavior of the proposed algorithm and the effect of the choice of P on the convergence speed, we applied our proposed algorithm to an environment with 16 users. Two SNR levels are considered in this experiment, namely 4 dB and 6 dB. In addition, even though 200 bits are transmitted, only the first 60 bits are plotted because this is sufficient to reveal the steady state behavior of the proposed algorithm. Figure 5 depicts the signal to interference noise ratio (SINR) versus number of bits with $P \in \{1, 2\}$ at $\text{SNR} = 6$ dB. It can be seen that $P = 1$ results in a marginally faster convergence rate and reaches the level of 0.5 dB after 10 bits. The proposed algorithm converges to $\text{SINR} \approx 1$ dB after 17 bits for both values of P. Similar observations can be noted when the SNR is reduced to 4 dB, as shown in Figure 6. The only noteworthy exception is that the steady-state SINR slightly below 1 dB is obtained after 20 bits.

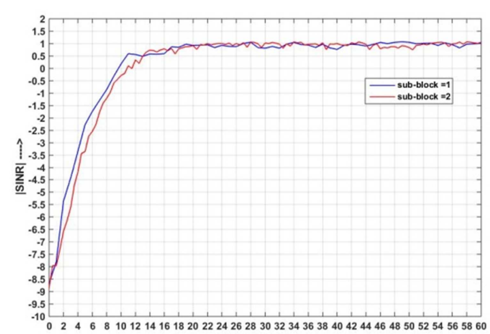

Figure 5: SINR (dB) vs. No. of Bits in AWGN with SNR = 6 dB and N = 16 Users.

Journal of Theoretical and Applied Information Technology
31st January 2016. Vol.83. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

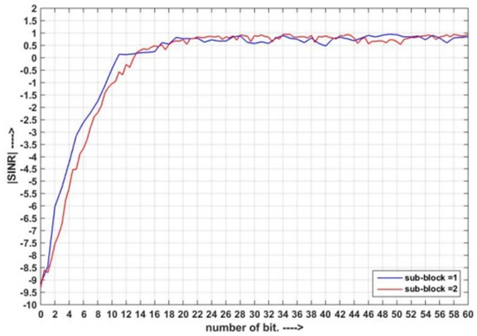

Figure 6: SINR (dB) vs. No. of Bits in AWGN with SNR = 4 dB and N = 16 Users.

In this investigation, we have not considered higher values of P because our aim is to reduce computational complexity. As we highlighted previously, $P = 2$ would not effectively deliver either better BER performance or faster convergence as compared to the block-based case $(P = 1)$. In contrast, using $P = 1$ results in the lowest computational complexity because all the calculations required to update the weight vector are carried out once per bit. Quantitative analysis of the computational cost is presented in Section 5.

4.3 BER Comparison

In this section, we will evaluate the BER performance of the proposed algorithm relative to that of Block LS-DRMTCMA, Sample RLS-DRMTA and Sample NLMS-DRMTA. The conditions stated in experiment 1 also apply here.

Figure 6 shows the BER performance versus SNR in an AWGN with the number of users equals 16. It can be seen that the Block LS-DRMTCMA has the worst BER performance in comparison to all algorithms. In addition, the Sample RLS-DRMTA has very poor performance at low SNR duo to the propagation of the estimation errors through future decisions. However, at SNR $> 4\ dB$, the BER curve declines sharply. Hence, it is evident that our proposed algorithm consistently achieves superior BER performance relative to that of the Sample RLS-DRMTA over the entire range of SNR. Moreover, our proposed algorithm can rival the performance of the Sample NLMS-DRMTA and even delivers lower BER over a wide range of SNR values (SNR ≤ 6 dB). At SNR > 6 dB, the Sample NLMS-DRMTA exhibits slight advantage over our algorithm. It is obvious that, with $P = 1$, our proposed algorithm gives consistently better BER performance in comparison to all algorithms. Moreover, the proposed algorithm delivers the best robustness to estimation errors that increase when the noise level is very high, resulting in significant performance degradation. It follows that the proposed adaptive step size rule based on evaluating the variance of the input array signal provides the best update increment for generating the new weight vector.

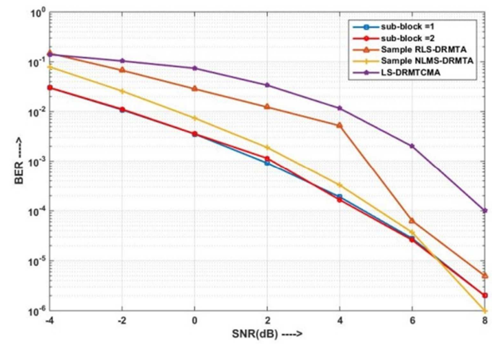

Figure 7: BER vs. SNR in AWGN with N = 16 Users.

Figure 8 depicts the BER performance versus the number of users that results from using each algorithm at the SNR level of 4 dB. Once again, the LS-DRMTCMA algorithm exhibits the worst BER, as it does not utilize the information carried in the previous data. In addition, the constant modulus property of the output signal does not help in generating a reliable reference signal, particularly when the number of users exceeds that of array elements.

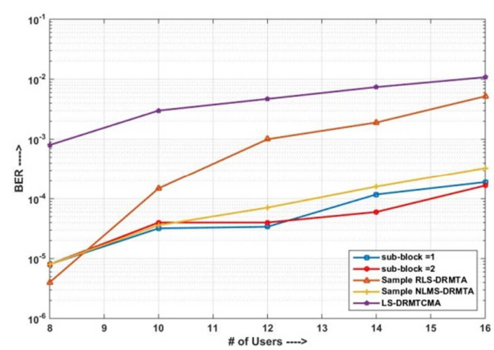

Figure 8: BER vs. No. of Users in AWGN with SNR = 4 dB.

When the number of users exceeds 9, the level of cross-correlation between Walsh codes becomes rather high. Therefore, when Sample RLS-DRMTA algorithm is applied, any error in detecting the present or previous samples would propagate, leading to significant degradation of the BER performance, as shown in Figure 9. The Sample NLMS-DRMTA algorithm is more robust to estimation errors and exhibits better performance than the two aforementioned algorithms because it

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

employs two adaptive step sizes when updating the weight vector on the sample basis. Finally, the proposed algorithm with $P \in \{1, 2\}$ outperforms the Sample NLMS-DRMTA, whereby the advantage becomes particularly noticeable when the number of users exceeds 10, as shown in Figure 9.

On the other hand, when the number of users decreases, the level of cross-correlation among users significantly declines, resulting in marked improvement in the Sample RLS-DRMTA performance. In particular, it can be seen that, when the number of users declines to 8, the Sample RLS-DRMTA starts to exhibit good performance, which is marginally better than that of the proposed algorithm. It should be noted that the Sample RLS-DRMTA can achieve good performance when the number of users is less than the number of array elements.

The BER performance of the different algorithms when the input SNR = 6 dB is shown in Figure 7. As can be seen, when the number of users exceeds 10, the proposed algorithm can achieve superior performance compared to the Sample NLMS-DRMTA and Sample RLS-DRMTA, irrespective of the number of users. The reason for this advantage is that the proposed algorithm is more robust to estimation errors that occur more frequently when the number of users (interferers) increases. When the number of users decreases below 10, the BER declines sharply and the proposed algorithm achieves the fastest decrement rate.

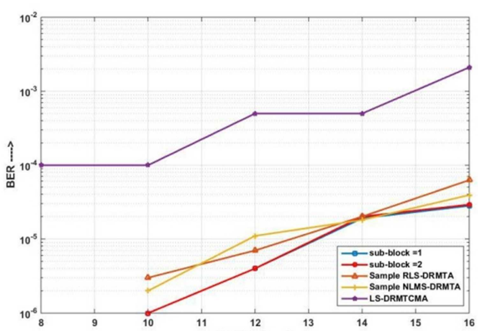

Figure 9: BER vs. No. of Users in AWGN with SNR = 6 dB.

By comparing graphs given in Figure 8 and 9, we can observe that, as the number of users decreases, the BER performance for all algorithms fails to improve as much as it does when SNR = 6 dB. This lack of improvement can be explained by the receiver performance that, at such low SNR, becomes mainly limited by the AWGN noise, rather than the user interference. Therefore, it is the noise level rather than the number of users that determines the BER at low values of SNR. On the other hand, at higher levels of SNR, the BER performance is primarily affected by the number of users (interferers), rather than the noise level. In both cases, our proposed algorithm almost outperforms the three algorithms with respect to BER.

4.4 Convergence Time Comparison

In this experiment, we compare the convergence behavior of each algorithm based on the output SINR versus number of bits. The SINR is evaluated at two levels of SNRs—4dB and 6dB. The number of users is assumed to be twice the number of the array elements, i.e., $N = 16$. As in the previous experiment, although 200 bits are transmitted, only the first 60 bits are plotted to reveal the convergence behavior (as all the algorithms converge to steady state within the first 60 bits, and no additional information is obtained by including additional bits).

The convergence times for the algorithms at SNR = 6 dB are shown in Figure 10. It can be seen that the LS-DRMTCMA yields the lowest output SINR because it does not utilize the information carried in the previous data. Moreover, the Sample RLS-DRMTA results in a slightly larger level of SINR than the LS-DRMTCMA. Finally, the Sample NLMS-DRMTA achieves marginally larger SINR than the RLS-DRMTA. On the other hand, the proposed algorithm exhibits the best performance in terms of large SINR. With regard to the convergence time, the Sample RLS-DRMTA has the fastest convergence rate, as it converges after approximately 1.75 bits, which is equivalent to 150 iterations. The LS-DRMTA also has a fast convergence rate, in the range of 3 bits. Finally, as approximately 17 bits, the proposed algorithm offers a good convergence rate compared to the Sample NLMS-DRMTA algorithm, which converges after 34 bits.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

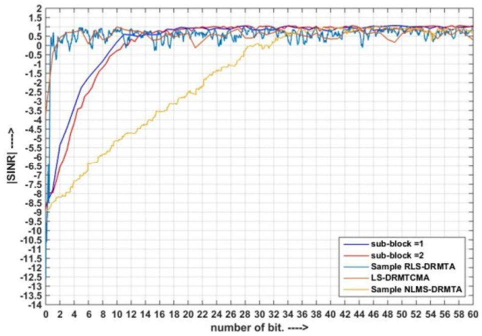

Figure 10: SINR (dB) vs. No. of bits with SNR = 6 dB and N = 16 Users.

Figure 11 shows the convergence curves at low SNR level of 4 dB. As can be seen, the LS-DRMTCMA is characterized by the worst performance in terms of low SINR. The RLS-DRMTA algorithm also produces a low level of SINR, albeit slightly greater than that yielded by the LS-DRMTCMA. On the other hand, the Sample NLMS-DRMTA slowly converges to a larger SINR. Once again, the proposed algorithm outperforms all other algorithms and results in the largest SINR. With respect to convergence speed, at 2 bits and 3 bits, respectively, the Sample RLS-DRMTA and LS-DRMTCMA provide similar and reasonably fast convergence rates. The proposed algorithm exhibits a fairly fast convergence speed; however, the convergence time is still in the range of 20 bits. On the other hand, at 35 bits, the Sample NLMS-DRMTA is characterized by the slowest convergence rate.

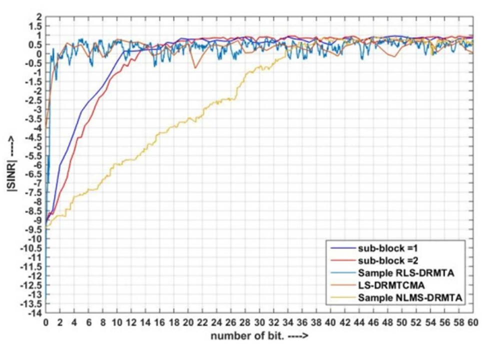

Figure 11: SINR (dB) vs. No. of bits with SNR=4 dB and N=16 Users.

From Figure 10 and 11, it is apparent that the Sample NLMS-DRMTA suffers from very slow convergence rate compared to other algorithms. Though the Sample RLS-DRMTA offers very fast convergence, it is achieved at the expense of a significantly increased computational complexity. On the other hand, the LS-DRMTCMA yields the lowest SINR level, which in turn significantly degrades the BER performance. Thus, the proposed algorithm offers the largest output SINR at a reasonably fast convergence rate and requires low computational complexity, as will be discussed in the following section.

5. COMPUTATIONAL COMPLEXITY

In this section, we will compare the different algorithms in terms of the computational complexity based on the simulation running time of each algorithm. In this experiment, the number of antenna elements is set to $M = 8$, the number of samples in one bit is set to $K = 64$, and the number of users is taken to be 16. Our simulations were run on an HP® laptop equipped with Intel® Core-i7 3610QM (@2.30GHz) processor and 8 GB of RAM. The simulation code is written in MATLAB® (2014b). In each run, we recorded the execution time required to process 200 bits of data. Table 1 provides the running time of each algorithm averaged over 100 runs.

From the results given in Table 1, we can see that the block-based algorithms, LS-DRMTCMA and our proposed algorithm with $P = 1$, have the shortest execution times, confirming that they have the lowest computational complexity of all algorithms. This superior performance is a result of the block-based algorithms being able to handle the entire bit in one iteration. In each iteration, the LS-DRMTCMA algorithm requires the calculation of matrix inversion of size $(M \times M)$, where $M = 8$ in our experiments. On the other hand, the proposed algorithm with $P = 1$ involves the calculation of the variance of each column of the $(M \times K)$ input array signal $\mathbf{X}$ during each iteration. In case of $P = 2$, where each bit is divided into two $(M \times K/2)$ sub-blocks, the calculation of the variance vector is repeated twice per bit, which results in increasing the computational complexity by twofold.

On the other hand, sample-based algorithms are very computationally expensive because they process only one sample during each iteration. Hence, one bit requires $K = 64$ iterations to complete. As can be seen from Table 1, the Sample RLS-DRMTA is the most computationally complex algorithm, as it requires the longest execution time of all algorithms. This computational complexity arises because it involves the calculation of the $(M \times M)$ correlation matrix $\mathbf{R}$ and the $(M \times 1)$ cross-correlation vector for every sample. The Sample NLMS-DRMTA is less complex and requires almost half the running time of the Sample

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RLS-DRMTA. However, the Sample NLMS-DRMTA algorithm does not require the calculation of neither matrix inversion nor correlation matrix. Instead, it involves vector multiplications and a simple routine that searches for the maximum value in a vector of length M. Still, it is important to note that these calculations should be repeated on a sample basis.

Table 1 The Simulation Running Time For The Diffeent Algorithms

Algorithm	Running Time (ms)
LS-DRMTCMA	37.2
Sub-Block NLMS-DRMTA ($P = 1$)	39.4
Sub-Block NLMS-DRMTA ($P = 2$)	77.8
Sample NLMS-DRMTA	432.2
Sample RLS-DRMTA	918.9

6. CONCLUSIONS

In this research, the impact of the number of sub-blocks on the performance of the proposed algorithm has been investigated in an AWGN channel. We have demonstrated that, for $P \in \{1, 2\}$, the proposed algorithm achieved nearly equal BER performances and similar convergence speeds. However, in case of $P = 1$, its computational complexity is nearly half that pertaining to $P = 2$. In comparison to the Block LS-DRMTCMA, our proposed algorithm with $P = 1$ exhibited far superior BER performance with comparable computational complexities. When compared to the Sample NLMS-DRMTA, our proposed algorithm delivered equally or slightly better BER performance at nearly one tenth of the computational complexity. In addition, the proposed algorithm converges significantly faster than the Sample NLMS-DRMTA. With respect to the Sample RLS-DRMTA, the proposed algorithm exhibited better robustness to estimation errors, resulting in significantly better BER performance when the number of users exceeds the number of array elements or at low values of SNR. Although the Sample RLS-DRMTA converges rapidly, the computational complexity is considerably higher than that of the Sub-block NLMS-DRMTA. Indeed, the proposed algorithm is about one twentieth less computationally expensive than the Sample RLS-DRMTA. Thus, our proposed algorithm provides a low computational complexity approach to achieve high BER performance with convergence speed that enables quick adaptation to dynamic channels.

REFERENCES:

[1] Rangisetty, N. and K.K. Rao, Analysis of Blind adaptive multiuser detection receivers. Transactions on Machine Learning and Artificial Intelligence, 2015. 2(6): p. 01.

[2] Saxena, P. and A. Kothari, Performance analysis of adaptive beamforming algorithms for smart antennas. IERI Procedia, 2014. 10: p. 131-137.

[3] Hakam, A., et al., Robust interference suppression using a new LMS-based adaptive beamforming algorithm. in Mediterranean Electrotechnical Conference (MELECON), 2014, 17th IEEE. 2014.

[4] Razia, S., T. Hossain, and M.A. Matin, Performance analysis of adaptive beamforming algorithm for smart antenna system. in International Conference on Informatics, Electronics & Vision (ICIEV), 2012.

[5] Rong, Z., et al., Simulation of multitarget adaptive array algorithms for wireless CDMA systems. in Vehicular Technology Conference, 1997, 47th IEEE. 1997.

[6] Rong, Z., et al., Despread-respread multi-target constant modulus array for CDMA systems. Communications Letters, IEEE, 1997. 1(4): p. 114-116.

[7] Du, Z., P. Gong, and W. Wu, Block-based RLS despread respread multitarget array: algorithm and performance. Proc. IEEE VTC, 2001. 1: p. 190-193.

[8] Labib, M., "Blind Adaptive Techniques used in Smart Antennas for CDMA Systems", M.Sc. Thesis, Carleton University, 2003.

[9] Labib, M., M. El-Tanany, and R. Goubran. A novel multitarget adaptive array algorithm for wireless CDMA systems using block affine projection. in Vehicular Technology Conference, VTC2004-Fall, 60th IEEE. 2004.

[10] Mohannad, A. M., "Investigation of Adaptive Signal Processing Algorithms for Mobile Communication Using Smart Antennas", M.Sc. Thesis, University of Technology, 2013.

[11] Yasin, M., P. Akhtar, and M.J. Khan, Tracking Performance of RLS and KAPA Algorithms for a Smart Antenna System. World Applied

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sciences Journal, 2010. 10(Special Issue of Computer & Electrical Engineering): p. 1-9.

[12] Liu, W. and J. Príncipe, Kernel affine projection algorithms. EURASIP Journal on Advances in Signal Processing, 2008. 1: p. 784292.

[13] Kranthi Kumar, G., et al., Adaptive Beam Forming Using KLMS, KRLS and CGM Algorithms. IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), 2012. 3(3): p. 41-48.

[14] Reddy, J.G., T.K. Teja, and B.S. Seshu, A Novel Approach for Interference Suppression Using an Improved LMS Based Adaptive Beam forming Algorithm. 2015.

[15] Li, J. and X.-D. Zhang, Blind adaptive multiuser detection based on affine projection algorithm. Signal Processing Letters, IEEE, 2005. 12(10): p. 673-676.

[16] Zhang, D., K. Wang, and X. Zhang, Blind adaptive affine projection algorithm-based multiuser detector over a multipath fading channel. Signal Processing, 2010. 90(6): p. 2102-2106.