Block-Shanno Minimum Bit Error Rate Beamforming

Tamer A. Samir, Said Elnoubi, and Ayman Elnashar

Abstract—**Beamforming is a key technology in smart antenna systems that can increase capacity and coverage and mitigate multipath propagation in mobile radio communication systems. The most popular criterion for linear beamforming is the minimum mean square error (MMSE). However, the mean square error (MSE) cost function is not optimal in terms of the bit error probability performance of the system. A class of adaptive beamforming algorithms has been proposed based on minimizing the bit error rate (BER) cost function directly. Unfortunately, the popular least minimum BER (LMBER) stochastic beamforming algorithm suffers from low convergence speed. Gradient Newton algorithms have been proposed to speed up the convergence rate and enhance performance but at the expense of complexity. In this paper, a block processing objective function for the minimum BER (MBER) is formulated, and a nonlinear optimization strategy, which produces the so-called block-Shanno MBER (BSMBER), is developed. A complete discussion for the complexity calculations of the proposed algorithm is demonstrated. Simulation scenarios are carried out in a multipath Rayleigh-fading direct-sequence code-division multiple access (DS-CDMA) system to explore the performance of the proposed algorithm. Simulation results show that the proposed algorithm offers good performance in terms of convergence speed, steady-state performance, and even system capacity compared to other MBER- and MSE-based algorithms.**

Index Terms—**Antennas, array signal processing, communication system performance, fading channels, mean square error (MSE) methods.**

I. Introduction

OVER the last two decades, an increasing demand for boosting the capacity of wireless systems has been observed to cater to the exponential increase in the number of subscribers. Since bandwidth is a valuable resource, this demand has to be satisfied by exploiting the existing resources more efficiently. Code-division multiple access (CDMA) is one of the most promising techniques for efficient sharing of the frequency spectrum. It is most likely to be applied to many future mobile systems. Although spread-spectrum systems are naturally resistant to interference, it has been noted that [1] with heavily loaded systems, the systems become interference limited rather than noise limited. As a result, the cell radius shrinks (i.e., cell breathing). Many researchers have focused their work on addressing this problem. They have produced a variety of methodologies. Spatial processing with adaptive antenna arrays has been considered by many researchers. It has shown real promise for increasing capacity and coverage and for mitigating multipath propagation of mobile radio communication systems [2], [3]. Adaptive beamforming can separate signals transmitted in the same carrier frequency, provided that they are separated in the spatial domain [4].

Wiener-solution-based beamformers indirectly minimize bit error rate (BER) by optimizing other cost functions [e.g., signal-to-interference-plus-noise ratio or mean square error (MSE)], which may result in suboptimal BER performance [5]–[11]. Minimum MSE (MMSE) beamforming is classically performed by minimizing the MSE between the desired and actual array outputs. MMSE is characterized by its good performance and amenability to adaptive implementation. However, MMSE does not guarantee the absolute minimum BER (MBER) [12], [13].

A fundamental goal in any digital communications system is to minimize the BER directly. The stochastic gradient approximate MBER (AMBER) [12] and the stochastic gradient least MBER (LMBER) [13] are two of the most successful and suitable algorithms for adaptive implementation. However, those MBER algorithms usually require long training sequences to converge to the steady state at minimum BER performance. Another main drawback for those stochastic MBER algorithms is the constant step size, which needs to be chosen very carefully to guarantee fast convergence to the minimum BER.

Stochastic gradient Newton algorithms, which require fewer training samples and, hence, speed up convergence rate, are proposed in [14]. They optimize the step size by calculating the Hessian function. The Newton-AMBER and Newton-LMBER are similar to the well-known least mean square Newton algorithm [15] and employ the cost functions used in the AMBER and the LMBER, respectively.

Therefore, the step size has to be set carefully, depending on the amount of interference and the channel-fading coefficients to ensure optimum convergence speed. In practice, wireless channels change with time. Hence, a variable step size is recommended to compensate for the changes in channel characteristics. The Shanno algorithm [16] provides an excellent choice, as it uses inexact searches to converge to the optimum step size. The main idea of the Shanno algorithm is to formulate a block-processing objective function and present a nonlinear optimization for its rapid minimization. A modified version of the Shanno algorithm has been deployed to optimize some cost functions, e.g., the constant modulus objective function

Manuscript received May 14, 2007; revised November 2, 2007 and December 22, 2007. The review of this paper was coordinated by Dr. A. Ghrayeb.

T. A. Samir is with Etihad Etisalat (Mobily), Riyadh 11423, Saudi Arabia (e-mail: tsamir@yahoo.com; t.mesallam@mobily.com.sa).

S. Elnoubi is with the Department of Electrical Engineering, Alexandria University, Alexandria 21544, Egypt (e-mail: saidelnoubi@yahoo.com).

A. Elnashar is with Emirates Integrated Telecommunications Company PJSC (du), Network Development DMC14, Dubai Media City, Dubai 502666, UAE (e-mail: ayman.elnashar@du.ae; nashar_eg@yahoo.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2008.917237

[17]–[19]. It offers good convergence speed with variable step size at low computational load.

The main contribution of this paper is the deployment of the modified version of the Shanno algorithm to minimize the BER cost function and a detailed discussion on the calculation of its complexity (see also [20] and [21]). Comparative analyses are conducted among the MMSE [11], the LMBER [13], the Newton-LMBER [14], and the proposed block-Shanno MBER (BSMBER) algorithms in terms of convergence rate, steady-state BER, computational complexity, and system capacity.

The rest of this paper is organized as follows. Section II describes a DS-CDMA system model with an antenna array in front. In Section III, the BER cost function is derived, and then, some of the existing stochastic gradient MBER algorithms are outlined. Section IV introduces the proposed BSMBER algorithm. Section V provides simulation results and comparative analyses. Finally, the conclusion is given in Section VI.

II. System Model

A K-user DS-CDMA communication system that employs an M-element uniform linear antenna array (ULA) at the base station is considered. The received continuous-time baseband signal at antenna element m can be modeled as follows:

$$r^{(m)}(t) = \sum_{n=-\infty}^{\infty} \sum_{k=1}^{K} b_k(n) h_k^{(m)}(t - nT_b) + \sigma_\eta \eta^{(m)}(t) \quad (1)$$

where T_b is the bit interval, $b_k(n)$ is the nth data bit of the kth user, $h_k^{(m)}(t)$ is the effective signature waveform of the kth user at antenna element m, $\eta^{(m)}(t)$ is the additive white Gaussian noise (AWGN) at antenna m with unit power spectral density, and σ_η is the noise power spectral density. It is assumed that the data symbols are independent equally likely random variables from a finite alphabet that are independent from $\eta^{(m)}(t)$. The effective signature waveform of user k at antenna element m can be modeled in the following form:

$$h_k^{(m)}(t) = \sum_{l=0}^{L-1} c_k(l) \cdot g_k^{(m)}(t - lT_c) \quad (2)$$

where L is the spreading factor, $T_c = T_b/L$ is the chip interval, $c_k(l)$ is the lth chip in the spreading sequence of the kth user, and $g_k^{(m)}(t)$ is the chip waveform of the kth user at antenna element m that has been distorted by the multipath propagation and front-end and transmit filters.

For M antennas, assuming that the propagating signals are plane waves, multiple copies of the transmitted signal can be collected with different times of arrival due to the reflection from the surrounding obstacles. The channel is assumed to be time invariant and elliptically modeled [22]. Therefore, the distorted chip waveform of the kth user can be modeled as follows:

$$\begin{aligned} \boldsymbol{g}_k(t) &= [\, g_k^1(t) \quad \cdots \quad g_k^M(t) \,] \\ &= \sigma_k \sum_{f=0}^{F_k-1} \alpha_{k,f} \cdot \psi_k(t - \tau_{k,f}) \boldsymbol{a}_{k,f} \end{aligned} \quad (3)$$

where σ_k is the amplitude of the kth user signal, F_k is the number of multipath components that are associated with the kth user, $\alpha_{k,f}$ is the proportion of the kth user's amplitude that is scattered in the fth path, which is assumed to be Rayleigh distributed, $\tau_{k,f}$ is the time delay of this path, $\psi_k(t)$ is the original chip waveform that has been filtered on the receivers' and the transmitters' filters, and $\boldsymbol{a}_{k,f} = \boldsymbol{a}(\theta_{k,f})$ is the array response vector to the fth path of the kth user, where $\theta_{k,f}$ is the direction of arrival (DOA) of the impinging signal of the kth user's fth ray and is given by

$$\boldsymbol{a}(\theta_{k,f}) = \left[1 \quad e^{-j\left\{2\pi \frac{d}{\lambda} \sin\theta_{k,f}\right\}} \quad \ldots \quad e^{-j\left\{2\pi \frac{d}{\lambda}(M-1)\sin\theta_{k,f}\right\}}\right]^T \quad (4)$$

where d is the spacing between the antenna elements, which is usually assumed to be $\lambda/2$, in which λ is the wavelength.

Without loss of generality, we assume that the desired user is the first user and that the receiver is synchronized to that user (i.e., $\tau_{1,0} = 0$). In general, the code sequence $\boldsymbol{c}_1 = [\, c_1(0) \quad \cdots \quad c_1(L-1)\,]^T$ is known for the desired user. The multipath channel parameters of the desired user and multipath ray DOAs and all the parameters for the other users are assumed to be unknown.

Assuming that the received signal is sampled at the chip rate, the output from each array element is placed into a vector with length $[(L + L_s - 1)N]$, where N is the number of bits, and L_s is the multipath delay spread in chips. The sampled received signal for user 1 at antenna element m is formatted as follows:

$$\boldsymbol{r}^{(m)}(n) = b_1(n)\boldsymbol{h}_1^{(m)} + \sum_{k=2}^{K} b_k(n)\boldsymbol{h}_k^{(m)} + \sigma_\eta \boldsymbol{\eta}^{(m)}(n) \quad (5)$$

where $\boldsymbol{h}_k^{(m)}$ and $\boldsymbol{\eta}^{(m)}(n)$ are the sampled signature waveform vectors for user k and the sampled AWGN vector, respectively, at antenna element m with length $[(L + L_s - 1)N]$.

For M antennas, we expand (5) from vector form to matrix form as follows:

$$\boldsymbol{R}(n) = b_1(n)\boldsymbol{H}_1 + \sum_{k=2}^{K} b_k(n)\boldsymbol{H}_k + \sigma_\eta \boldsymbol{\Gamma}(n) \quad (6)$$

where $\boldsymbol{R}(n) = [\,(\boldsymbol{r}^{(1)}(n))^T \quad \cdots \quad (\boldsymbol{r}^{(M)}(n))^T\,]^T$ represents the sampled received signals to the antenna array (i.e., sampled array observations), $\boldsymbol{H}_k = [\,(\boldsymbol{h}_k^{(1)})^T \quad \cdots \quad (\boldsymbol{h}_k^{(M)})^T\,]^T$ is the space–time signature matrix of the kth user, $\boldsymbol{\Gamma}(n) = [\,(\boldsymbol{\eta}^{(1)}(n))^T \quad \cdots \quad (\boldsymbol{\eta}^{(M)}(n))^T\,]^T$ is the sampled noise matrix, and n is the sample index.

For convenience, the complete DS-CDMA system that employs an antenna array at the receiver is illustrated in Fig. 1. At the receiver side, the sampled received signal from each antenna element is multiplied by a matched filter at each antenna branch. The output signal after the matched filter bank, i.e., $\boldsymbol{x}(n) = [\, x^{(1)}(n) \quad \cdots \quad x^{(M)}(n)\,]^T$, with size $M \times 1$, is then multiplied by the complex-valued beamformer weight vector, i.e., $\boldsymbol{w} = [\, w_1 \quad \cdots \quad w_M\,]^T$, with size $M \times 1$ to produce the output statistics of the antenna array as follows:

$$y(n) = \boldsymbol{w}^H \boldsymbol{x}(n) \quad (7)$$

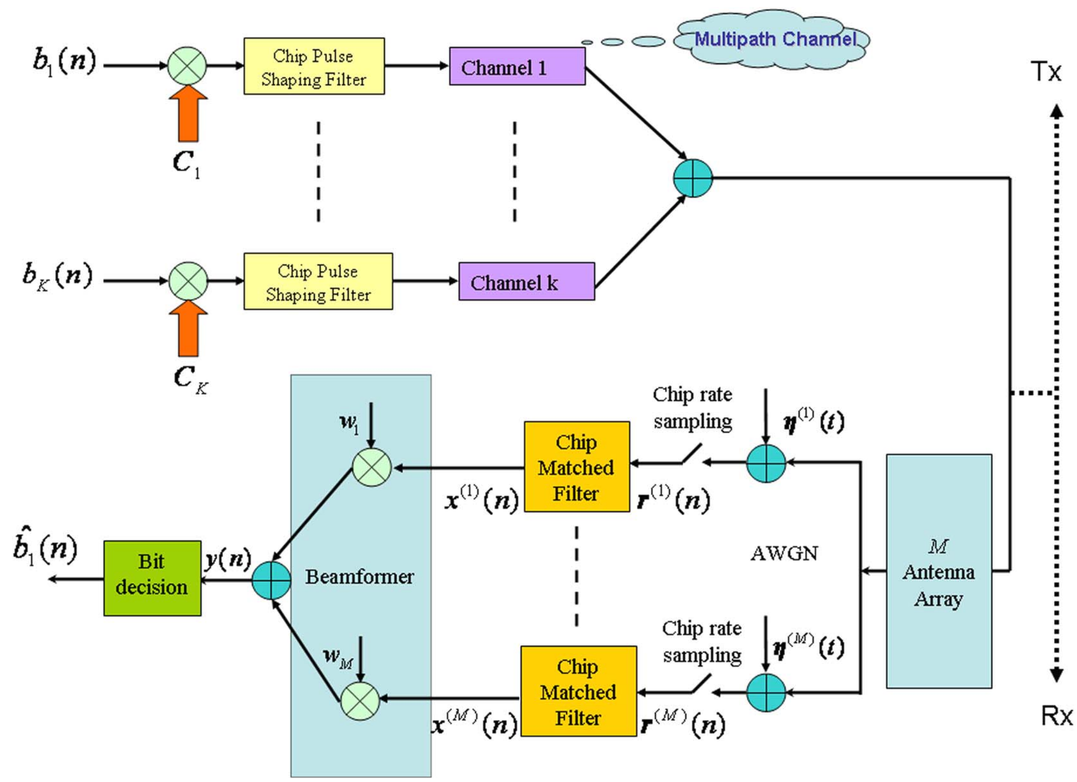


Fig. 1. DS-CDMA system model with an antenna array.

where $(\cdot)^H$ stands for the Hermitian. The beamformer output $y(n)$ can be expressed as

$$y(n) = \boldsymbol{w}^H \left(\overline{\boldsymbol{x}}(n) + \sigma_\eta \tilde{\boldsymbol{\eta}}(n)\right) = \overline{y}(n) + \overline{\eta}(n) \tag{8}$$

where $\overline{\boldsymbol{x}}(n)$ is the noiseless matched filter output vector, $\overline{\eta}(n) = \boldsymbol{w}^H \tilde{\boldsymbol{\eta}}(n)$ is Gaussian with zero mean and variance $E[|\overline{\eta}(n)|^2] = 2\sigma_\eta^2 \boldsymbol{w}^H \boldsymbol{w}$, and $\tilde{\boldsymbol{\eta}}(n)$ is the noise vector after being multiplied by the matched filter.

The bit decision is made according to

$$\hat{b}(n) = \operatorname{sgn}\left(\operatorname{Re}\{y(n)\}\right) = \operatorname{sgn}\left(y_R(n)\right) \tag{9}$$

where $\operatorname{sgn}(\cdot)$ denotes the sign function, and $y_R(n) = \operatorname{Re}\{y(n)\}$ is the real part of the beamformer output $y(n)$.

III. Stochastic Gradient MBER Algorithms

To develop a minimum BER beamformer for a binary system, we start by forming its BER cost function. To derive the BER cost function for a linear detector with weight vector $\boldsymbol{w}$, let us define the following signed variable:

$$y_s(n) = \operatorname{sgn}\left(b(n)\right) y_R(n). \tag{10}$$

Here, $y_s(n)$ is an error indicator for the binary decision, i.e., if it is positive, then we have a correct decision, else if it is negative, then an error occurred.

Furthermore, let $N_b = 2^K$ be the number of possible transmitted bit sequences b_v of $b(n)$, where $1 \le v \le N_b$. Hence, the first element of b_v is the desired user data. The noiseless array output signal $\overline{\boldsymbol{x}}(n)$ takes values from the signal set $\chi \triangleq \{\overline{\boldsymbol{x}}_v = b_v \boldsymbol{h},\ 1 \le v \le N_b\}$. Similarly, the noiseless beamformer's output, i.e., $\overline{y}(n)$, takes values from the scalar set $Y(\boldsymbol{w}) \triangleq \{\overline{y}_v(\boldsymbol{w}) = \boldsymbol{w}^H \overline{\boldsymbol{x}}_v,\ 1 \le v \le N_b\}$. Thus, the real part of the beamformer's output $\overline{y}_R(n)$ can only take values from the set $Y_R(\boldsymbol{w}) \triangleq \{\overline{y}_{R,v}(\boldsymbol{w}) = Re\{\overline{y}_v(\boldsymbol{w})\},\ 1 \le v \le N_b\}$.

Therefore, the probability density function (pdf) of the error indicator, i.e., $y_{\text{sign}}(n)$, is a mixed sum of Gaussian distributions [13], i.e.,

$$P(y_s) = \frac{1}{N_b \sqrt{2\pi\sigma_\eta^2 \boldsymbol{w}^H \boldsymbol{w}}} \times \sum_{v=1}^{N_b} \exp\left(-\frac{\left(y_s - \operatorname{sgn}\left(b(n)\right) y_{R,v}(\boldsymbol{w})\right)^2}{2\sigma_\eta^2 \boldsymbol{w}^H \boldsymbol{w}}\right). \tag{11}$$

Hence, the error probability of the beamformer $\boldsymbol{w}$, i.e., the BER cost function, is given by

$$P_E(\boldsymbol{w}) = \frac{1}{N_b \sqrt{2\pi\sigma_\eta^2 \boldsymbol{w}^H \boldsymbol{w}}} \sum_{v=1}^{N_b} \int_{q(\boldsymbol{w})}^{\infty} \exp\left(-\frac{u^2}{2}\right) du = \frac{1}{N_b} \sum_{v=1}^{N_b} Q\left(q(\boldsymbol{w})\right) \tag{12}$$

where $Q(\cdot)$ is the Gaussian error function given by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp\left(-\frac{y^2}{2}\right) dy \tag{13}$$

$$\begin{aligned} q(\boldsymbol{w}) &= \frac{\operatorname{sgn}(b(n))\, y(n)}{\sigma_\eta \sqrt{\boldsymbol{w}^H \boldsymbol{w}}} \\ &= \frac{\operatorname{sgn}(b(n)) \operatorname{Re}\left\{\boldsymbol{w}^H \boldsymbol{x}(n)\right\}}{\sigma_\eta \sqrt{\boldsymbol{w}^H \boldsymbol{w}}}. \end{aligned} \tag{14}$$

In practice, the set of Y is not available. A widely used approach for approximating the pdf is known as the kernel density or the Parzen window-based estimate [23]–[25]. A kernel density estimate is known to produce a reliable pdf estimate with short data records. Given a block of Z training samples $\{\boldsymbol{x}(Z), b(Z)\}$, a kernel density estimate of the pdf, which was defined in (11), is given by

$$\begin{aligned} \hat{P}(y_s) = &\frac{1}{Z\sqrt{2\pi\rho^2 \boldsymbol{w}^H \boldsymbol{w}}} \\ &\times \sum_{i=1}^{Z} \exp\left(-\frac{(y_s - \operatorname{sgn}(b(i))\, y_R(i))^2}{2\rho^2 \boldsymbol{w}^H \boldsymbol{w}}\right) \end{aligned} \tag{15}$$

where the radius parameter ρ is related to the noise standard deviation σ_η [23]–[25]. Therefore, the block BER cost function could be derived from the kernel density estimate of the pdf as follows:

$$P_E(\boldsymbol{w}) = \frac{1}{Z} \sum_{z=1}^{Z} Q\left(q_Z(\boldsymbol{w})\right) \tag{16}$$

where

$$q_Z(\boldsymbol{w}) = \sum_{i=1}^{Z} \frac{\operatorname{sgn}(b(i)) \operatorname{Re}\left\{\boldsymbol{w}^H \boldsymbol{x}(i)\right\}}{\sigma_\eta \sqrt{\boldsymbol{w}^H \boldsymbol{w}}}. \tag{17}$$

After deriving the BER cost function, we now define the optimization problem. Our objective is to minimize the BER of the system. Hence, the MBER beamforming solution is defined as

$$\boldsymbol{w}_{\text{MBER}} = \arg\min_{\boldsymbol{w}} P_E(\boldsymbol{w}). \tag{18}$$

To solve the optimization problem in (18), a gradient estimate of the BER cost function stated in (16) with respect to $\boldsymbol{w}$ should be estimated. An iterative gradient optimization algorithm can be used. As a result, the block gradient of the BER cost function, i.e., $P_E(\boldsymbol{w})$, is given by

$$\begin{aligned} \nabla P_E(\boldsymbol{w}) &= \frac{\partial P_E(\boldsymbol{w})}{\partial \boldsymbol{w}} \\ &= \frac{1}{2Z\sqrt{2\pi}\rho\sqrt{\boldsymbol{w}^H \boldsymbol{w}}} \\ &\quad \times \sum_{i=1}^{Z} \exp\left(\frac{-(\boldsymbol{x}(i))^2}{2\rho^2 \boldsymbol{w}^H \boldsymbol{w}}\right) \cdot \operatorname{sgn}(\boldsymbol{x}(i)) \cdot \boldsymbol{x}(i). \end{aligned} \tag{19}$$

Alternatively, a sample-by-sample estimate for the BER cost function can be given as [26]

$$P_E(\boldsymbol{w}, z) = \frac{1}{\sqrt{2\pi\rho^2 \boldsymbol{w}^H \boldsymbol{w}}} \exp\left(-\frac{(y_s - \operatorname{sgn}(b(i))\, y(i))^2}{2\rho^2 \boldsymbol{w}^H \boldsymbol{w}}\right). \tag{20}$$

Hence, the instantaneous gradient of $P_E(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ is given by

$$\nabla P_E(\boldsymbol{w}) = \frac{1}{2\sqrt{2\pi}\rho\sqrt{\boldsymbol{w}^H \boldsymbol{w}}} \exp\left(\frac{-(\boldsymbol{x}(i))^2}{2\rho^2 \boldsymbol{w}^H \boldsymbol{w}}\right) \operatorname{sgn}(\boldsymbol{x}(i))\, \boldsymbol{x}(i). \tag{21}$$

After we have introduced the objective optimization problem, we now briefly present the stochastic gradient LMBER and the stochastic gradient Newton-LMBER algorithms. Both algorithms are introduced to emphasize the superiority of the proposed BSMBER algorithm over the other algorithms.

A. Stochastic Gradient LMBER

In [13], Chen developed the LMBER algorithm, which is one of the most popular stochastic gradient algorithms that minimize the cost function of the BER. The LMBER algorithm is preferred due to its simplicity, low complexity, and good performance. The LMBER beamforming seeks the minimization of the cost function in (20). Consequently, its update equation is given by

$$\begin{aligned} \boldsymbol{w}(i+1) &= \boldsymbol{w}(i) + \mu \nabla P_E(\boldsymbol{w}) \\ &= \boldsymbol{w}(i) + \mu \frac{\operatorname{sgn}(\boldsymbol{x}(i))}{\sqrt{2\pi}\rho\sqrt{\boldsymbol{w}^H \boldsymbol{w}}} \exp\left(\frac{-(\boldsymbol{x}(i))^2}{2\rho^2 \boldsymbol{w}^H \boldsymbol{w}}\right) \boldsymbol{x}(i). \end{aligned} \tag{22}$$

By normalizing the weight vector to unit length, i.e., $\boldsymbol{w}^H \boldsymbol{w} = 1$, the update equation reduces to

$$\boldsymbol{w}(i+1) = \boldsymbol{w}(i) + \mu \frac{\operatorname{sgn}(\boldsymbol{x}(i))}{\sqrt{2\pi}\rho} \exp\left(\frac{-(\boldsymbol{x}(i))^2}{2\rho^2}\right) \boldsymbol{x}(i). \tag{23}$$

The step size μ and the kernel width ρ are the two algorithm parameters that have to be appropriately set to ensure faster convergence and small steady-state BER maladjustment.

B. Stochastic Gradient Newton-LMBER

A direct approach to ensure convergence, irrespective of signal input energy, is the calculation of a suitable step size value using the function and the gradient. This usually requires finding estimates of the Hessian of the objective function. The gradient Newton algorithm [14] incorporates second-order statistics of input signals. It usually has a faster convergence rate than the standard gradient technique, but it requires a higher computational complexity. In practice, only estimates of the covariance matrix and the gradient vector are available. These estimates can be applied to Newton's formula to provide an update equation for the gradient Newton-LMBER as follows:

$$\begin{aligned}\boldsymbol{w}(i+1) &= \boldsymbol{w}(i) + \mu \boldsymbol{H}\boldsymbol{e}(\boldsymbol{w}, i) \\ &= \boldsymbol{w}(i) + \mu \boldsymbol{R}_{xx}^{-1}(i)\nabla P_E(\boldsymbol{w})\end{aligned} \tag{24}$$

where $\boldsymbol{H}e(\boldsymbol{w}, i)$ is the Hessian matrix, and $\boldsymbol{R}_{xx} = (1/N)\sum_{n=1}^{N} \boldsymbol{x}(n) \cdot \boldsymbol{x}^H(n)$ is the covariance matrix of the received signal. The convergence factor μ is introduced to protect the algorithm from divergence due to the use of the noisy estimates of the covariance matrix and the gradient vector. Substituting (22) into (24), the Newton-LMBER update equation can be derived as follows:

$$\boldsymbol{w}(i+1) = \boldsymbol{w}(i) + \mu \boldsymbol{R}_{xx}^{-1}(i) \frac{\operatorname{sgn}(\boldsymbol{x}(i))}{\sqrt{2\pi}\rho} \exp\left(\frac{-(\boldsymbol{x}(i))^2}{2\rho^2}\right) \boldsymbol{x}(i). \tag{25}$$

IV. BSMBER

Stochastic gradient algorithms choose constant step size. The constant step size depends on the interference as well as the channel coefficients. To guarantee convergence of the LMBER algorithm, the step size choice is the bottleneck for the algorithm to converge. The Newton algorithm optimizes the step size to guarantee convergence. Hence, it requires computing, updating, and storing the inverse of the Hessian matrix. These operations on the Hessian matrix are the most costly part of the Newton algorithm. As a result, the need to implement a new algorithm to take advantage of the Newton algorithm while maintaining linearity in complexity has motivated this work. The Shanno algorithm is a memoryless modified Newton algorithm, which conducts the conjugate gradient type search without fully optimizing the step size; alternatively, it is chosen to be within a specified range such that the convergence is guaranteed. The higher bound of the step size must satisfy the following inequality [16]:

$$\begin{aligned}P_E(\boldsymbol{w}(i)) &< P_E(\boldsymbol{w}(i-1)) \\ &\quad + \alpha\mu \nabla P_E(\boldsymbol{w}(i-1))^T \boldsymbol{d}(\boldsymbol{w}(i-1)).\end{aligned} \tag{26}$$

In addition, the lower bound of the step size must satisfy the following inequality [16]:

$$\nabla P_E(\boldsymbol{w}(i))^T \boldsymbol{d}(\boldsymbol{w}(i)) > \beta \nabla P_E(\boldsymbol{w}(i-1))^T \boldsymbol{d}(\boldsymbol{w}(i-1)) \tag{27}$$

where α and β are constants, and $\boldsymbol{d}(\boldsymbol{w}(i))$ is the search direction vector. The search direction vector is a linear combination of the negative gradient, the gradient difference between the current gradient and the previous gradient, and the previous search direction. It is defined as

$$\boldsymbol{d}(i) = -\nabla P_E(\boldsymbol{w}(i)) + a(i)\boldsymbol{u}(i) + [b(i) - c(i)a(i)]\, \boldsymbol{d}(i-1) \tag{28}$$

where $\boldsymbol{u}(i)$ is the gradient difference between the current gradient and the previous gradient and is defined as

$$\boldsymbol{u}(i) = \nabla P_E(\boldsymbol{w}(i)) - \nabla P_E(\boldsymbol{w}(i-1)) \tag{29}$$

$$a(i) = \frac{\boldsymbol{d}^T(i-1)\nabla P_E(\boldsymbol{w}(i))}{\boldsymbol{d}^T(i-1)\boldsymbol{u}(i)} \tag{30}$$

$$b(i) = \frac{\boldsymbol{u}^T(i)\nabla P_E(\boldsymbol{w}(i))}{\boldsymbol{d}^T(i-1)\boldsymbol{u}(i)} \tag{31}$$

$$c(i) = \mu(i-1) + \frac{|\boldsymbol{u}(i)|^2}{\boldsymbol{d}^T(i-1)\boldsymbol{u}(i)}. \tag{32}$$

Now, the search direction vector in the Shanno algorithm, which is linear in complexity, will replace the Hessian matrix, which is quadratic in complexity, in the Newton algorithm. This feature reduces calculations, as it involves only an implicit calculation of the Hessian matrix [18].

As a result, the weight update equation for the Shanno algorithm is given by

$$\boldsymbol{w}(i) = \boldsymbol{w}(i-1) + \mu \boldsymbol{d}(i). \tag{33}$$

Another advantage of the Shanno algorithm is its quick and efficient ability to minimize nonlinear objective functions. The BER cost function is a nonlinear function; thus, any minimization process can lead to a local minimum and not to the global minimum [27]. Hence, the Shanno algorithm is the best algorithm to optimize the BER cost function.

The proposed BSMBER algorithm processes the data on a block-by-block basis, i.e., it takes in a block of data and iterates until the convergence tolerance criterion is matched.

To use the block-Shanno algorithm, we should convert the problem from complex data to real data format. First, we define

$$\boldsymbol{w}(i) = \boldsymbol{w}_r(i) + j\boldsymbol{w}_i(i) \tag{34}$$

$$\boldsymbol{x}(i) = \boldsymbol{x}_r(i) + j\boldsymbol{x}_i(i) \tag{35}$$

where $j = \sqrt{-1}$. Then, we define the following new vectors:

$$\boldsymbol{w}_c(i) = \begin{bmatrix} \boldsymbol{w}_r(i) \\ \boldsymbol{w}_i(i) \end{bmatrix}$$

$$\boldsymbol{x}_c(i) = \begin{bmatrix} \boldsymbol{x}_r(i) \\ \boldsymbol{x}_i(i) \end{bmatrix}. \tag{36}$$

Now, the detected signal y_c is given by

$$y_c = \boldsymbol{w}_c^T \boldsymbol{x}_c. \tag{37}$$

The proposed BSMBER algorithm is summarized in Table I. First, we initialize the main algorithm parameters. Then, we perform the matched filter operation for the received signal. The algorithm consists of two main loops. The outer loop is for each block of data, and the inner loop is repeated over the same block of data until a certain number of iterations is reached (e.g., 25 iterations), the norm of the gradient vector is sufficiently small, or the gradient difference becomes zero to prevent dividing by zero. In the main loop, we formulate a block of data (100 bits) from the output of the matched filter

TABLE I
SUMMARY OF THE BSMBER ALGORITHM

- Initialization
  - $i = 1$, μ=0.1, $\nabla\mu$=0.1μ, ε=0.1, α=0.25, β=0.5, ρ=30σ_η,
    $Z = 100$, $\boldsymbol{w}(0) = \dfrac{\boldsymbol{x}(0)}{\|\boldsymbol{x}(0)\|^2}$.
- Matched filter process; $x^{(m)} = \boldsymbol{r}^{(m)}\boldsymbol{c}_1$.
- Outer loop (for $i = 1 : \lfloor N/Z \rfloor$, where $\lfloor . \rfloor$ is the floor).
  - *Form a block of data from the received signals.*
  - *Convert the complex data into real as in (34)–(36).*
    - Now, the vector size become $2M$ rather than M.
  - *Initialize* $\kappa = 1$, $P_E = 1$, $\nabla P_E = \boldsymbol{1}_M$, *where* $\boldsymbol{1}_M$ *is a vector of size* $M \times 1$ *with all elements equal one.*
  - *Inner loop (while* $\kappa \le 25$ *or* $\|\nabla P_E\| < \varepsilon$ *or* $u = 0$ *)*
    - Calculate the cost function BER and the gradient vector over the block from (15) and (19), respectively.
      - Complexity of (19) is $2MZ$.
    - If $\kappa = 1$ then $\boldsymbol{d} = -\nabla P_E$ else calculate $\boldsymbol{d}$ from (28)-(32).
      - Complexity of 29, 30, and 31 are $4M$, $4M$, and $2M$, respectively.
    - Check the direction vector $\boldsymbol{d}$, if $\dfrac{|\boldsymbol{d}^T(\kappa)\nabla P_E(\kappa)|}{\|\boldsymbol{d}^T(\kappa)\| \cdot \|\nabla P_E(\kappa)\|} < \varepsilon$ then reset $\boldsymbol{d}$ to be equal to $-\nabla P_E$.
      - Complexity is $2M$.
    - Check the step size μ as in (26) and (27). If it falls outside the boundaries, then increase or decrease the step size as $\mu(i) = \mu(i-1) \pm \Delta\mu$.
      - Complexity of (26) and (27) are $2M$ and $4M$, respectively.
    - Update the weight vector as in (33).
    - Transfer $\boldsymbol{w}$ to the complex form again.
    - Determine the detected signals for all the block of data as in (36) in order to be used for calculating the BER cost function.
      - Complexity is $2MZ$.
    - Increment the iteration number $\kappa = \kappa + 1$
  - *end of inner loop*
- end of outer loop

TABLE II
SIMULATION PARAMETERS

SNR	15 and 30 dB
Step-size (μ)	0.0975
Kernel radius (ρ)	$= 30\sigma_\eta$
Frame length	500 and 1000 bits (for 1st and 2nd scenarios, respectively)
Spreading factor	31 chip Gold Code
The pulse shape	raised cosine ($\alpha = 0.22$)
Modulation	BPSK
Antenna Elements	5 element ULA with half wavelength spacing between omni elements
Number of users	5
user power distribution	1st scenario: equal power 2nd scenario: desired user -10dB
Max. channel length	10 chips
Number of multirays	3
Max. number of multiray delay	5 chips
Standard deviation of the multipath	0.3
Channel	synchronous Rayleigh fading
Noise	Complex AWGN

operation and convert it to a real format, as in (34)–(36); then, we initialize the inner loop iteration index κ, the BER cost function, and the gradient vector to unity. In the inner loop, we compute the BER cost function and the gradient vector from (15) and (19), respectively. Then, we compute the search direction vector from (28), and unless we are in the beginning of the inner loop, we set it to be equal to the negative of the gradient. After that, we check the search direction vector; if it is in the wrong direction, we reset it back to the negative of the gradient. Then, we check the step size and adjust it according to its boundary. If its value is higher than the higher bound in (26), then we decrease its value. If its value is lower than the lower bound in (27), then we increase its value. Else, we keep its value constant. Then, we compute the weight update vector from (33) and convert it back to the complex form. At the end of the inner loop, we determine the detected signal by multiplying the computed weight vector by the received signal to use it to calculate the BER cost function again and increment the inner loop iteration index. The inner loop iterates until any of the stop criteria is reached. After that, we go back to the main loop and form another block of data, and so on. These processes iterate until we finish all the incoming data. As we can observe, the algorithm keeps tracking the received signal with different channel and interference parameters rather than having a constant step size that may force the algorithm to diverge in case the channel and interference parameters change.

A detailed calculation of the BSMBER complexity is shown in Table I. Given that multiplication operations are more complex than addition operations, we base our complexity calculations on multiplication operations only. Assume that all vectors have the same length M. From a mathematical point of view, multiplying two vectors results in a complexity of M. Going back to our algorithm, converting the vectors from complex format to real format results in doubling the vectors length, i.e., $2M$. Following the algorithm complexity (see Table I), we will find that the total number of multiplications is $4M \cdot Z + 18M$. This number of multiplications is required to determine Z bits. Hence, the BSMBER algorithm needs $(4M \cdot Z + 18M)/Z$ multiplication operations to detect one bit.

V. COMPUTER SIMULATIONS AND PERFORMANCE ANALYSIS

In this section, we present simulation results that illustrate and compare the performance of different MBER beamforming techniques in addition to MMSE beamforming. We employ the elliptical channel model described in Section II. The simulation parameters are described in Table II. The DOA angles were randomly generated between $\pm 60^\circ$, i.e., three sectors cell, and the multipath for the same user have DOA angles between $\pm 10^\circ$ from the main path. One of the interferers is separated with only 10° from the desired user. The DOA angles are unknown to the receiver. The simulations were done in two scenarios. The first scenario assumed perfect power control (equal power distributions). The second scenario assumed that the desired user's power is less than the interferers' power by 10 dB to model the near–far effect.

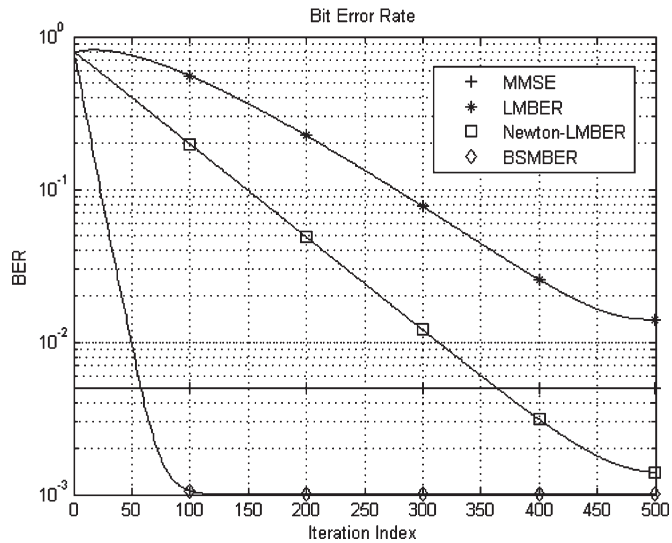


Fig. 2. BER versus iterations for the first scenario (equal power distributions) at SNR = 15 dB.

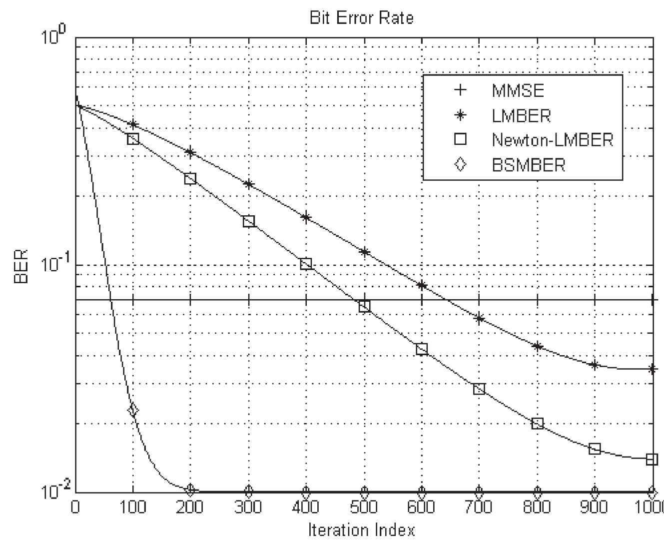


Fig. 4. BER versus iterations for the second scenario (the desired user power is 10 dB below interferers) at SNR = 15 dB.

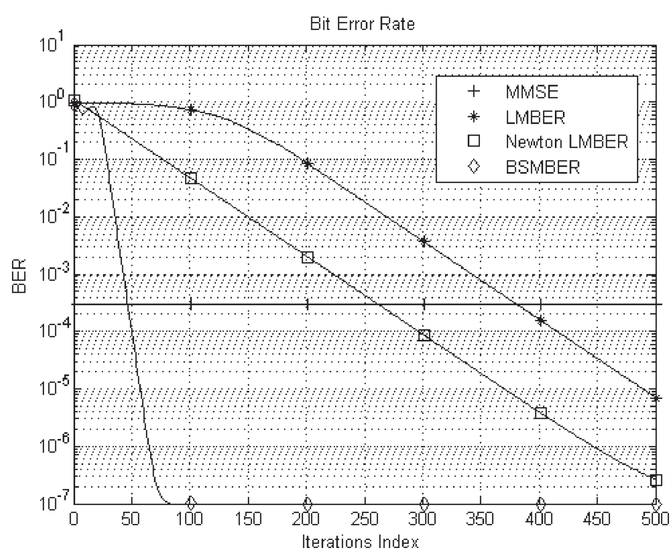


Fig. 3. BER versus iterations for the first scenario (equal power distributions) at SNR = 30 dB.

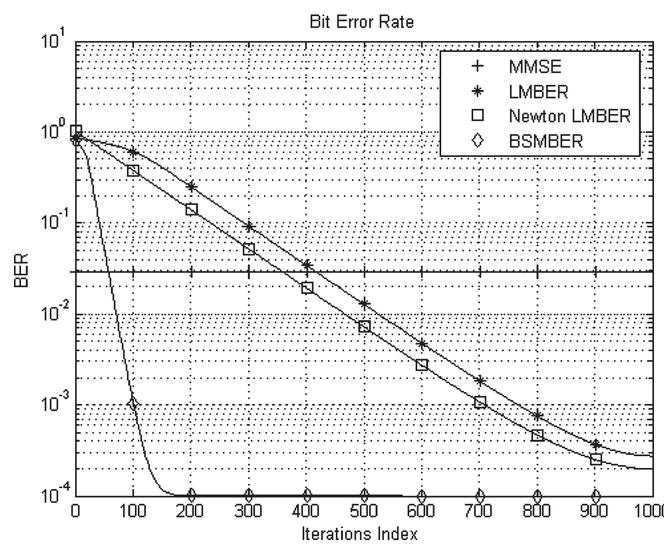


Fig. 5. BER versus iterations for the second scenario (desired user power is 10 dB below interferers) at SNR = 30 dB.

A. Convergence Rate Comparison

In this section, a study of the convergence rate of the algorithms to the steady state during adaptive implementation is conducted. Figs. 2 and 3 illustrate the BER versus the iteration index for the first scenario (equal power distributions) at signal-to-noise ratios (SNRs) equal to 15 and 30 dB, respectively. First, we can observe the enhancement in steady-state BER performance when using the MBER-based techniques over the MMSE techniques. In addition, it can be seen in Fig. 2 that the LMBER and the Newton-LMBER algorithms converge after 500 iterations, i.e., they need 500 bits to reach the steady state, whereas the proposed BSMBER algorithm converges after 100 iterations, i.e., it converges at the fourth block. Hence, the proposed algorithm requires 400 bits to reach the steady state at an SNR of 15 dB. From Fig. 3, LMBER and Newton-LMBER algorithms converge after 500 iterations, i.e., they need 500 bits to reach the steady state, whereas the proposed BSMBER algorithm converges after 75 iterations, i.e., it converges at the third block. Hence, the proposed algorithm requires 300 bits to reach the steady state.

Figs. 4 and 5 illustrate the BER versus the iteration index for the second scenario (the desired user power is 10 dB below interferers) at SNRs equal to 15 and 30 dB, respectively. In addition, for Fig. 4, it can be seen that the LMBER and the Newton-LMBER algorithms converge to the steady state after 1000 iterations, i.e., they require 1000 bits to reach the steady state, whereas the proposed BSMBER algorithm converges after 200 iterations, i.e., it converges at the eighth block. Hence, the proposed algorithm requires 800 bits to reach the steady state. For Fig. 5, it can be seen that the LMBER and the Newton-LMBER algorithms converge to the steady state after 1000 iterations, i.e., they require 1000 bits to reach the steady state, whereas the proposed BSMBER algorithm converges after 175 iterations, i.e., it converges at the seventh block. Hence, the proposed algorithm requires 700 bits to reach the

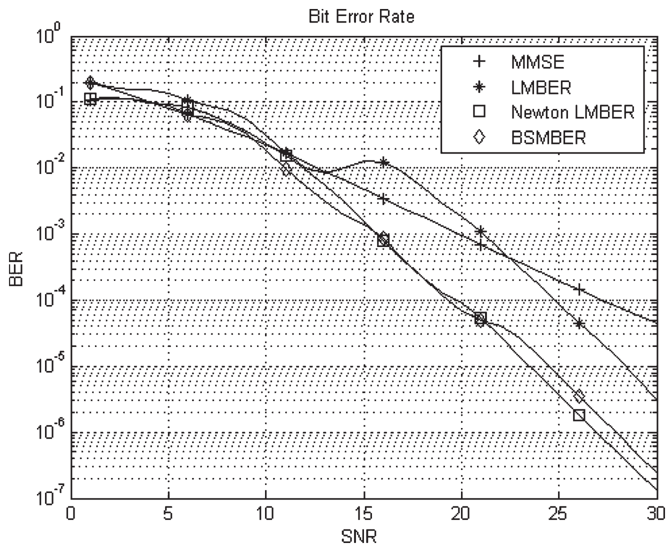


Fig. 6. BER versus SNR for the first scenario (equal power distributions).

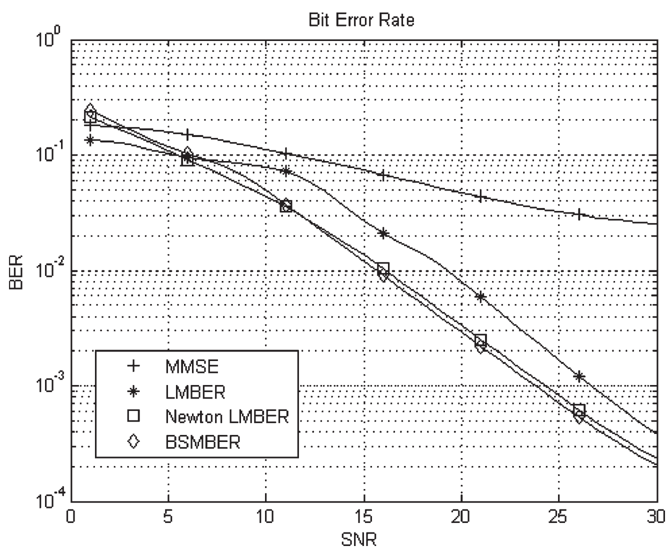


Fig. 7. BER versus SNR for second scenario (desired user power is 10 dB below interferers).

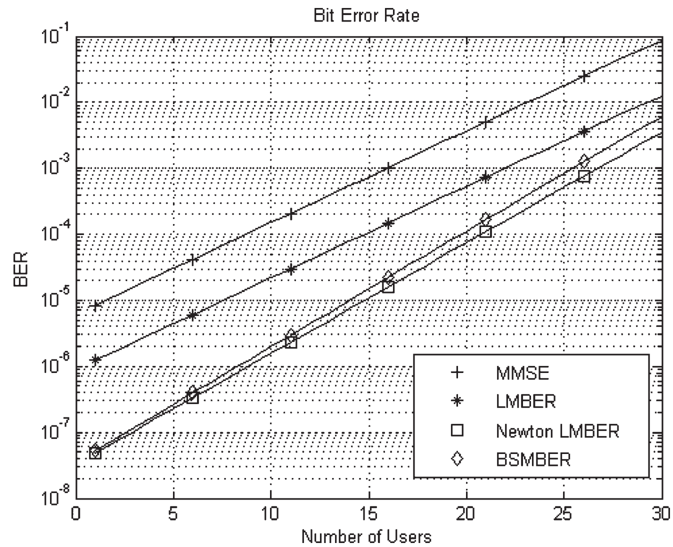


Fig. 8. BER versus number of users for the first scenario (equal power distributions) at SNR = 30 dB.

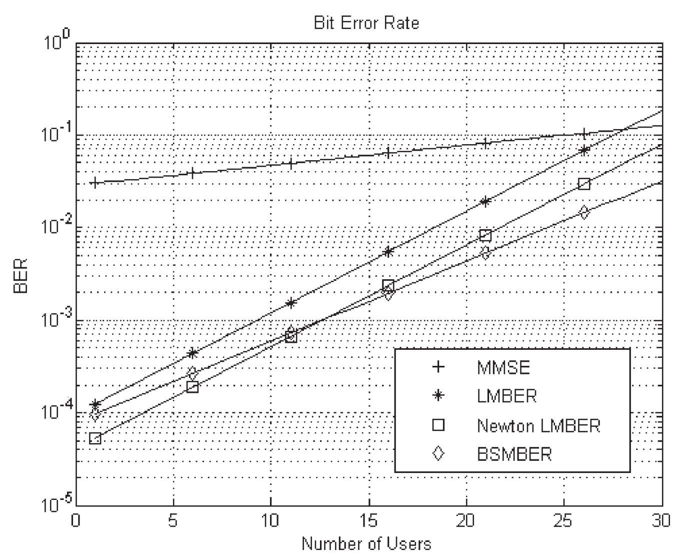


Fig. 9. BER versus number of users for the second scenario (desired user power is 10 dB below interferers) at SNR = 30 dB.

steady state. In addition, the proposed algorithm has the best BER steady-state performance competence.

B. BER Performance Versus SNR Comparison

In this section, BER performance is compared against the SNR of the desired user. For an SNR between 1 and 30 dB, the algorithms are run for 500 bits (five blocks in the BSMBER) in the first scenario and for 1000 bits (ten blocks in the BSMBER) in the second scenario. The average BER based on the Q function is determined after the steady state is reached. Figs. 6 and 7 compare the BER performance against the SNR for the MMSE, LMBER, and Newton-LMBER beamformers and the proposed BSMBER algorithm with the first and second scenarios, respectively. It is evident that the proposed algorithm almost performs similarly to the Newton algorithm. However, the proposed algorithm requires an $O(M)$ complexity compared to the Newton algorithm, which requires an $O(M^2)$ complexity.

C. BER Performance Versus Number of Subscribers

In this section, BER performance is evaluated when the number of users is increased from 1 to 30 at an SNR equal to 30 dB. The algorithms are run for 500 bits (five blocks with the BSMBER) in the first scenario and for 1000 bits (ten blocks with the BSMBER) in the second scenario. The average BER based on the Q function is determined after the algorithms converge. Figs. 8 and 9 show the BER performance for the first and second scenarios, respectively, and demonstrate that the proposed algorithm exhibits the best performance in terms of low BER at a high number of subscribers. The MMSE beamformer is dramatically affected in the second scenario. MMSE algorithms are not robust against the near–far effect; meanwhile, the MBER algorithms keep good performance when the system suffers from the near–far effect, and this agrees with the results in [13].

TABLE III
COMPUTATIONAL COMPLEXITY COMPARISON

Algorithm	Computational complexity
MMSE	$O(M^2)$
LMBER	$O(M)$
Newton-LMBER	$O(3M^2+(5/2)M)$
BSMBER	$O(4M+18M/Z)$

D. Computational Complexity

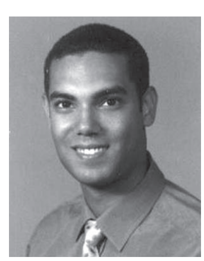
In this section, we compare all the previous algorithms in terms of computational complexity [20]. Since addition is much easier than multiplication, we focus on multiplication to compare computational complexities. Table III illustrates the number of multiplications required to complete a single iteration, i.e., detecting one bit. The proposed BSMBER maintains the linearity in complexity, however; its performance is at least similar to the Newton algorithm with a quadratic complexity [20].

VI. CONCLUSION

In this paper, a block-Shanno algorithm for minimizing the BER cost function has been proposed. A comparison between this algorithm and the popular MMSE, LMBER, and Newton-LMBER algorithms is conducted in terms of BER performance and computational complexity. The BER performance is compared against the iteration index during adaptive implementation, the SNR, and the number of subscribers. The computer simulations show that the MBER-based algorithms are optimum in terms of BER performance compared to MMSE algorithms. The proposed BSMBER yields the best convergence speed while maintaining linear complexity. Moreover, the overall performance of the proposed algorithm is similar to the Newton algorithm.

REFERENCES

[1] A. Viterbi, "When not to spread spectrum—A sequel," *IEEE Commun. Mag.*, vol. 23, no. 4, pp. 12–17, Apr. 1985.
[2] M. Haardt and Q. Spencer, "Smart antennas for wireless communications beyond the third generation," *Comput. Commun.*, vol. 26, no. 1, pp. 41–45, Jan. 2003.
[3] L. C. Godara, "Application of antenna arrays to mobile communications, Part II: Beam-forming and direction-of-arrival considerations," *Proc. IEEE*, vol. 85, no. 8, pp. 1195–1245, Aug. 1997.
[4] H. M. Jones, R. A. Kennedy, B. D. Hart, and P. B. Rapajic, "Limits on linear detector performance for close users in a wireless multipath environment," in *Proc. ICASSP*, May 2001, vol. 5, pp. 2953–2956.
[5] S. Kwon, I. Oh, and S. Choi, "Adaptive beamforming from the generalized eigenvalue problem with a linear complexity for a wideband CDMA channel," in *Proc. IEEE Veh. Technol. Conf.*, 1999, pp. 1890–1894.
[6] F. Alam, D. Shim, and B. D. Woerner, "A new low-complexity beamformer-rake receiver for WCDMA," in *Proc. IEEE ICC*, New York, Apr. 2002, vol. 1, pp. 160–164.
[7] B. D. Van Veen, "Adaptive convergence of linearly constrained beamformers based on the sample covariance matrix," *IEEE Trans. Signal Process.*, vol. 39, no. 6, pp. 1470–1473, Jun. 1991.
[8] L. J. Horowitz, H. Blatt, W. G. Brodsky, and K. D. Senne, "Controlling adaptive antenna arrays with the sample matrix inversion algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, no. 6, pp. 840–847, Nov. 1979.
[9] Y. Li, J. H. Winters, and N. R. Sollenberger, "Spatial-temporal equalization for IS-136 TDMA systems with rapid dispersive fading and cochannel interference," *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1182–1194, Dec. 1999.
[10] J. Litva and T. K. Lo, *Digital Beamforming in Wireless Communications*. Boston, MA: Artech House.
[11] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, no. 12, pp. 2143–2159, Dec. 1967.
[12] C. Yeh, R. Lopes, and J. R. Barry, "Approximate minimum bit-error rate multiuser detection," in *Proc. Global Commun. Conf.*, 1998, pp. 3590–3595.
[13] S. Chen, N. N. Ahmad, and L. Hanzo, "Adaptive minimum bit-error rate beamforming," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 341–348, Mar. 2005.
[14] R. C. de Lamare and R. Sampaio-Neto, "Adaptive multiuser receivers for DS-CDMA using minimum BER gradient-Newton algorithms," in *Proc. 13th IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 15–18, 2002, vol. 3, pp. 1290–1294.
[15] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*. Boston, MA: Kluwer.
[16] D. F. Shanno, "Conjugate gradient method with inexact searches," *Math. Oper. Res.*, vol. 3, no. 3, pp. 244–256, Aug. 1978.
[17] Z. Wang and E. M. Dowling, "Block-Shanno constant modulus algorithm for wireless equalizations," in *Proc. ICASSP*, Apr. 1996, pp. 2678–2681.
[18] U. G. Jani, E. M. Dowling, R. M. Golden, and Z. Wang, "Multiuser interference suppression using block Shanno constant modulus algorithm," *IEEE Trans. Signal Process*, vol. 48, no. 5, pp. 1503–1506, May 2000.
[19] A. Elnashar, S. Elnoubi, and H. Elmikati, "A robust Block-Shanno adaptive blind multiuser receiver for DS-CDMA systems," in *Proc. IST Mobile Wirel. Commun. Summit*, Dresden, Germany, Jun. 19–23, 2005.
[20] T. A. Samir, S. Elnoubi, and A. Elnashar, "Class of minimum bit error rate algorithms," in *Proc. 9th ICACT*, Feb. 12–14, 2007, vol. 1, pp. 168–173.
[21] T. A. Samir, S. Elnoubi, and A. Elnashar, "Block Shanno MBER algorithm," in *Proc. ISSPA*, 2007, pp. 1–4.
[22] Y. S. Song, H. M. Kwon, and B. J. Min, "Computationally efficient smart antennas for CDMA wireless communications," *IEEE Trans. Veh. Technol.*, vol. 50, no. 6, pp. 1613–1628, Nov. 2001.
[23] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1066–1076, Sep. 1962.
[24] B. W. Silverman, *Density Estimation*. London, U.K.: Chapman & Hall.
[25] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*. London, U.K.: Oxford Univ. Press.
[26] B. Mulgrew and S. Chen, "Adaptive minimum-BER decision feedback equalizers for binary signaling," *Signal Process.*, vol. 81, no. 7, pp. 1479–1489, Jul. 2001.
[27] J. Li, F. Chen, and G. Wei, "Local minimum-BER linear multiuser detector," in *Proc. ISCIT*, Oct. 12–14, 2005, vol. 1, pp. 511–514.



Tamer A. Samir was born in Egypt in 1976. He received the B.S. and M.S. degrees from Alexandria University, Alexandria, Egypt, in 1998 and 2001, respectively.

From January 1999 to January 2003, he was an O&M Engineer with BAPETCo. (Shell Egypt), Cairo, Egypt. He was a Project Manager with NASCo. (Motorola K.S.A.), Riyadh, Saudi Arabia, from January 2003 to September 2006. Since September 2006, he has been a Project and Development Senior Engineer with the Department of Mobile Network Development (MND), Etihad Etisalat (Mobily), Riyadh. He was responsible for radio access network (RAN) development and implementation. He has been engaged in the design, planning, and implementation of the Mobily UMTS network, HSDPA broadband evolution, and the DVB-H pilot network. His research interests include digital signal processing for wireless communications, cellular systems, multiuser detection, smart antenna, array processing, beamforming, MIMO systems, and OFDM systems.

Mr. Samir has been a Technical Reviewer of the IEEE Communications Society.

Said Elnoubi was born in Alexandria, Egypt, in September 1951. He received the B.S. and M.S. degrees from Alexandria University in 1974 and 1977, respectively, and the Ph.D. degree from Southern Methodist University, Dallas, TX, in 1980, all in electrical engineering.

From October 1974 to December 1977, he was an Instructor with Alexandria University. From January 1978 to December 1980, he was a Research Assistant with Southern Methodist University, where he was a Postdoctoral Fellow from January 1981 to August 1981. From September 1981 to August 1987, he was an Assistant Professor of electrical engineering with the University of Illinois, Chicago. From September 1987 to June 1989, he was an Associate Professor with the Department of Electrical Engineering, Alexandria University. From August 1989 to May 1990, he was a Visiting Professor with Wichita State University, Wichita, KS. From 1990 to 1994, he was a Member of Technical Staff with MITRE Corporation, McLean, VA. He is currently a Professor and the Chairman of the Department of Electrical Engineering, Alexandria University. He has been a Technical Reviewer for the *Proceedings of the IEE* and journals on selected areas in communications. His research interests include digital modulation, mobile radio and wireless communications, spread spectra, OFDMA, and air/ground radio communications.

Dr. Elnoubi has been a Technical Reviewer for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON INFORMATION THEORY, and *Wireless Communications*. He was an Assistant Editor of Communications for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He received the Egypt National Incentive Award for Engineering Science and the Salah Amer Incentive Award for Electronic Science in 1989. He received the Egypt National Superiority Award for Engineering Science in 2003 and the Alexandria University Distinction Award for Engineering Science in 2004. He is listed in the Marquis *Who's Who in the World*.

Ayman Elnashar was born in Egypt in 1972. He received the B.S. degree in electrical engineering from Alexandria University, Alexandria, Egypt, in 1995 and the M.Sc. and Ph.D. degrees in electrical communications engineering from Mansoura University, Mansoura, Egypt, in 1999 and 2005, respectively.

From March 1997 to March 1998, he was an RF Design Engineer with Systel (Motorola Egypt). From March 1998 to February 2000, he was an RF Systems Engineer with the Wireless Laboratory of the Egypt Electrical Authority (EEA), Mansoura. From March 2000 to June 2005, he was with the Egyptian Company for Mobile Services (MobiNil), Cairo, Egypt. From June 2005 to January 2008, he was a Chief Development Engineer with the Department of Mobile Network Development (MND), Etihad Etisalat (Mobily), Riyadh, Saudi Arabia, where he was responsible for radio access network (RAN) development and implementation. He has been engaged in the design, planning, and implementation of the Mobily UMTS network, HSDPA broadband evolution, and the DVB-H pilot network. Since January 2008, he has been with Emirates Integrated Telecommunications Company PJSC "du" as a Mobile Access Network Director. He is responsible for the Mobile Access Network development and implementation. His research interests include digital signal processing for wireless communications, cellular systems, multiuser detection, smart antennas, array processing, beamforming, and robust adaptive detection.

Dr. Elnashar has been a Technical Reviewer of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, *IET Signal Processing*, and the *EURASIP Journal on Applied Signal Processing*.