A Robust Block-Shanno Adaptive Blind Multiuser Receiver for DS-CDMA Systems

Ayman Elnashar, Said Elnoubi, and Hamdi El-mikati

***Abstract*—A robust block adaptive blind multiuser receiver based on block-Shanno constant modulus algorithm (BSCMA) with a Quadratic Inequality (QI) constraint on the weight vector norm is developed in this paper. Quadratic constraint has been a widespread methodology to improve the robustness of least-mean square detectors. The BSCMA is realized using a modified Newton's algorithm without inverse of Hessian matrix estimation. The partition linear interference canceller (PLIC) structure with multiple constraints is invoked to identify the multiple access interference (MAI). The Lagrange multiplier methodology is exploited to solve the QI constraint problem. Simulations are conducted in rich multipath environment under imperfect power control to authenticate the robustness of the proposed detector.**

Index Terms*—*DS/CDMA, Block-Shanno, Newton's algorithm, multiuser detection, constrained optimization, quadratic inequality constraint

I. Introduction

Constant Modulus algorithm (CMA) or Godared is an effective technique for blind receiver design in communications systems [1]. It has been shown in [2], that the CMA can perform analogous to the non-blind receivers if capture of desired user minima can be guaranteed. The traditional CMA, like the celebrated LMS algorithm, involves a constant step-size that controls the speed of convergence [3]. The selection of the CMA step-size is case sensitive and can affect the algorithm convergence. In addition to this, it was explored in [4] that the recursive implementation of constant modulus based algorithms need more robustness against mismatch and perturbations errors. These errors may be caused by imperfect covariance matrix estimation, improper initialization of the weight vector, signature vector mismatch, etc. Robustness can be added by incorporating a quadratic inequality (QI) constraint on the weight vector norm. CMA-based detectors with QI constraint on the weight vector norm are proposed in [4]. Unfortunately, the robust approach proposed in [4] uses RLS-like algorithm with $O\left(M^2\right)$ to estimate the blind multiuser receiver. Moreover, it was demonstrated there, that the constant modulus LMS-like algorithm with $O\left(M\right)$ complexity can not offer any performance improvement by incorporating the QI constraint.

A modified version from block Shanno's algorithm with $O\left(M\right)$ complexity has been shown to offer good convergence speed at low cost in [5] and [6]. However, the proposed BSCMA in [5], [6] is notorious to suffer from sensitivity to step-size selection and there is no clear vision was given to step-size update. More importantly, the algorithm involves a gradient vector norm check step and in this case, if the norm starts to increase, the algorithm stops the recursive adaptation and start with the initial weight vector and hence this block of data will not benefit from previous updates. Moreover, the BSCMA is more sensitive to the number of iterations required inside every block of data and no clear break point was determined to stop iteration inside the block. In this paper, we apply the QI constraint on the weight vector norm in an attempt to enhance the performance of BSCMA. The QI constraint will oversee the weight vector norm as well as the gradient vector norm and hence no need to check the gradient vector norm increase. Additionally, the iteration inside block can continue without affecting algorithm stability due to weight vector norm constraint.

The proposed variable loading (VL) technique in [4], [7] is exploited to estimate the optimum diagonal loading value. The BSCMA algorithm is used to update the adaptive vector of PLIC structure. The PLIC structure with multiple constraints is employed to identify the MAI and hence assist in avoiding interference capture [8]. In addition, the different forms of BSCMA algorithms; block-conjugate gradient CMA (BCGCMA) and block gradient descent CMA (BGDCMA), are investigated. The resistance of BSCMA-based algorithms against near-far effect is investigated and assessed.

II. System Model

In the uplink of DS-CDMA system we have k mobile user's transmitting simultaneously to the base station, where each user symbols are assumed for simplicity and without loss of generality to be BPSK with arbitrary power and timing. Each user's symbol is broaden by a spreading waveform $\boldsymbol{c}_j = \left[c_j(0) \quad \cdots \quad c_j(L-1)\right]^T$ of length *L,* where chips period $T_c = T/L$ and T is the symbol period. Each user signal is assumed to pass through FIR channel including any attenuation, multipath, asynchronism, pulse shaping filter, and front end receiver filter. As a consequence, the *j*th user FIR channel $\boldsymbol{g}_j(t)$ is the chip waveform of the *j*th user that has

Ayman Elnashar is with the Egyptian Company for Mobile Services (MobiNil), WTC, Cairo, Egypt (phone: 0020-12-2219802; e-mail: aelnashar@mobinil.com).

Said Elnoubi is with the Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt (e-mail: selnoubi@hotmail.com).

Hamdi El-mikati is with the Department of Electronics and Electrical Communication, Mansoura University, Mansoura, Egypt (e-mail: h.elmikati@ieee.org).

been filtered at the transmitter and receiver and distorted by the multipath channel. Let a_j stand for the amplitude of *j*th user signal, m_j be the number of multipath components, $\alpha_{j,m}$ be the amplitude of the user signal scattered in *m*th path, $\delta_{j,m}$ be the multipath delay spread of this path. Then $g_j(t)$ can be written as follows:

$$g_j(t) = a_j \sum_{m=0}^{m_j} \alpha_{j,m}\varphi_j(t-\delta_{j,m}) \quad (1)$$

The received signal is the superposition of all transmitted signals plus noise and is sampled at chip rate. As a result, we can write the sampled received signal $\boldsymbol{x}(n)$ as:

$$\boldsymbol{x}(n) = \sum_{j=1}^{K}\sum_{l=-\infty}^{\infty} \boldsymbol{s}_j(l).\boldsymbol{h}_j(n-lL-\tau_j) + \boldsymbol{w}(n) \quad (2)$$

The channel noise $\boldsymbol{w}(n)$ is assumed to be zero mean Gaussian and independent from source symbols. The effective signature waveform $\boldsymbol{h}_j(n)$ of j^{th} user represents the spreading sequence of that user transmitted through channel characterized by impulse response $\boldsymbol{g}_j(t)$, τ_j stands for user delay, and $\boldsymbol{s}_j(n)$ denotes transmitted data bits of user j.

$$\boldsymbol{h}_j(n) = \sum_{m=-\infty}^{\infty} \boldsymbol{c}_j(m).\boldsymbol{g}_j(n-m) \quad (3)$$

Without loss of generality, we will assume that the required user is user number one and is used as the timing reference ($\tau_1 = 0$). We can write the effective signature waveform of the required user $\boldsymbol{h}_1$ in matrix form as follows:

$$\boldsymbol{h}_1 = \boldsymbol{C}_1 \cdot \boldsymbol{g}_1 \quad (4)$$

The matrix $\boldsymbol{C}_1$ is consisting of shifted versions from signature waveform of the involved user. The channel vector $\boldsymbol{g}_1$ is $N_g \times 1$ vector obtained from (1).

III. Robust BSCMA Algorithm

A. Background

A chip-rate linear receiver can be designed by collecting N_f samples from the received signal vector and the detector $\boldsymbol{f}(n)$ is $N_f \times 1$ vector consisting of the weights. The output signal of this detector is given by:

$$y(n) = \boldsymbol{f}^H(n)\boldsymbol{x}(n) \quad (5)$$

In [1], Godared proposed the CM principle that minimizes the dispersion of the receiver output about dispersion constant. For BPSK, the dispersion constant is one and hence

$$J(\boldsymbol{f}) \triangleq E\left\{\left(\left|\boldsymbol{f}^H\boldsymbol{x}\right|^2 - 1\right)^2\right\} \quad (6)$$

We must first extend the CM cost function (6) to permit block processing analogous to [5], [6]. Take a block from the received vector signal with length $N_f \times M$, where M is the block length (i.e. number of bits from the received signal), thus:

$$\boldsymbol{D}_i = \left[\boldsymbol{x}\left((i-1)M\right) \vdots \boldsymbol{x}\left((i-1)M+1\right) \vdots \cdots \vdots \boldsymbol{x}\left(iM-1\right)\right] \in R^{N_f \times M} \quad (7)$$

The block CMA objective function is defined as [6]:

$$\Psi(\boldsymbol{f}) = \frac{1}{4M}\sum_{l=0}^{N-1}\left[\left|\boldsymbol{w}^H\boldsymbol{x}\left((i-1)M+l\right)\right|^2 - 1\right]^2 \quad (8)$$

The length of $\boldsymbol{x}\left((i-1)M+l\right)$ is equal to the detector length N_f.

The objective function is amended to fulfill the real requirement of Shanno's algorithm as follows:

$$\Psi(\boldsymbol{f}) = \frac{1}{4M}\sum_{n=0}^{N-1}\left[\boldsymbol{f}^H\boldsymbol{X}(n)\boldsymbol{f} - 1\right]^2 \quad (9)$$

and

$$\nabla_f\Psi(\boldsymbol{f}) = \frac{1}{M}\sum_{n=0}^{N-1}\left[\boldsymbol{f}^H\boldsymbol{X}(n)\boldsymbol{f} - 1\right]\boldsymbol{X}(n)\boldsymbol{f} \quad (10)$$

where $\boldsymbol{X}(n)$ is the current estimate of the covariance data matrix and for real data samples $\boldsymbol{X}(n) = \boldsymbol{x}^T(n)\boldsymbol{x}(n)$ whereas for complex samples data we can adopt the approach proposed in [6].

The PLIC structure [8] is invoked to identify the MAI interference by dividing the weight vector $\boldsymbol{f}(n)$ as follows

$$\boldsymbol{f} = \boldsymbol{f}_{c(N_f\times 1)} - \boldsymbol{B}_{N_f\times N_a;(N_a=N_f-N_g)}\boldsymbol{f}_{a(N_a\times 1)} \quad (11)$$

$$\boldsymbol{f}_c = \boldsymbol{C}_1\left(\boldsymbol{C}_1^H\boldsymbol{C}_1\right)^{-1}\boldsymbol{g} \quad (12)$$

In order to prevent cancellation of desired source from occurring a blocking matrix $\boldsymbol{B}$ is inserted to ensure the orthogonality between upper and lower branches and satisfies: $\boldsymbol{B}^H\boldsymbol{C}_1 = \boldsymbol{0}$, $\boldsymbol{B}^H\boldsymbol{B} = \boldsymbol{I}$.

The Newton's algorithm is exercised to updates the filter tap weights as follows [9], [10]:

$$\bar{\boldsymbol{f}}_a(j) = \boldsymbol{f}_a(j-1) - \boldsymbol{H}_{newton}^{-1}\left(\boldsymbol{f}_a(j-1)\right)\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right) \quad (13)$$

Where $\boldsymbol{H}_{newton}^{-1}\left(\boldsymbol{f}_a(j-1)\right)$ is the Hessian of the objective function $\Psi(\boldsymbol{f})$ and $\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right)$ is the gradient of the cost function with respect to $\boldsymbol{f}_a$ evaluated at the iteration block index $j-1$. Shanno's approximation is used to approximate the inverse of the Hessian matrix using $O(M)$ complexity as follows

$$\boldsymbol{H}^{-1}\left(\boldsymbol{f}_a(j-1)\right) = \boldsymbol{I} - \frac{\boldsymbol{u}(j)\boldsymbol{d}^H(j-1) - \boldsymbol{d}^H(j-1)\left[c(j)\boldsymbol{d}^H(j-1) - \boldsymbol{u}^H(j)\right]}{\boldsymbol{d}^H(j-1)\boldsymbol{u}(j)} \quad (14)$$

where

$$c(j) = \delta(j-1) + \frac{\left|\boldsymbol{u}(j)\right|^2}{\boldsymbol{d}^H(j-1)\boldsymbol{u}(j)} \quad (15)$$

$$\boldsymbol{u}(j) = \boldsymbol{g}\left(\boldsymbol{f}_a(j)\right) - \boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right) \quad (16)$$

$$\boldsymbol{d}(j) = -\boldsymbol{H}^{-1}\left(\boldsymbol{f}_a(j-1)\right)\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right) \quad (17)$$

and $\delta(j-1)$ is the step size at $j-1$ block iteration .

Therefore, the adaptive weight vector can be updated as follows:

$$\boldsymbol{f}_a(j) = \boldsymbol{f}_a(j-1) + \delta(j)\boldsymbol{d}(j) \qquad (18)$$

The step size must satisfy the following constraints to guarantee convergence [6]

$$\Psi\left(\bar{\boldsymbol{f}}_a(j)\right) \le \Psi\left(\boldsymbol{f}_a(j-1)\right) + \alpha\delta(j)\boldsymbol{g}^H\left(\boldsymbol{f}_a(j-1)\right)\boldsymbol{d}(j) \qquad (19)$$

$$\left(\boldsymbol{g}\left(\boldsymbol{f}_a(j)\right)\right)^T\boldsymbol{d}(j) \ge \sigma\left(\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right)\right)^T\boldsymbol{d}(j) \qquad (20)$$

A procedure for selecting the step-size according to the above constraints is reported in [5].

Substituting from (14)-(16) into (17) and after some manipulations with real-valued form, the following update equation of $\boldsymbol{d}(j)$, is attained:

$$\boldsymbol{d}(j) = \boldsymbol{d}(j-1)e(j) + \left(a(j)-1\right)\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right) \qquad (21)$$

where

$$a(j) = \frac{\boldsymbol{u}(j)\boldsymbol{d}^T(j-1)}{\boldsymbol{d}^T(j-1)\boldsymbol{u}(j)} \qquad (22)$$

$$e(j) = \frac{\left[\boldsymbol{u}^T(j) - c(j)\boldsymbol{d}^T(j-1)\right]\boldsymbol{g}\left(\boldsymbol{f}_a(j-1)\right)}{\boldsymbol{d}^T(j-1)\boldsymbol{u}(j)} \qquad (23)$$

If we set $a(j)=0$, the BCGCMA algorithm is obtained, and if we set $a(j)=e(j)=0$, we get the BGDCMA algorithm.

B. Quadratically Constraint BSCMA Receiver

The QI constraint can be applied on the block adaptive weight portion $\boldsymbol{f}_a(j)$. Consequently, the robust weight vector $\bar{\boldsymbol{f}}_a(j)$ can be acquired from the solution of the following constrained optimization problem:

$$\Phi(\boldsymbol{f}) = \frac{1}{4M}\sum_{n=0}^{N-1}\left[\bar{\boldsymbol{f}}_a^T(j)\boldsymbol{Z}(n)\bar{\boldsymbol{f}}_a(j) - 1\right]^2$$

$$\textit{Subject to} \quad \bar{\boldsymbol{f}}_a^T(j)\bar{\boldsymbol{f}}_a(j) \le \beta^2 \qquad (24)$$

The constrained value is $\beta^2 = \delta^2 - \boldsymbol{f}_c^H\boldsymbol{f}_c$ where $\delta^2 = t.\left(\|\boldsymbol{f}_c\|^2\right)$ and t is set to a suitable value and $\boldsymbol{Z}(n) = \boldsymbol{B}^T\boldsymbol{X}(n)\boldsymbol{B} = \boldsymbol{z}(n)\boldsymbol{z}^T(n)$ is the sample blocked covariance data matrix of the lower branch from PLIC structure and $\boldsymbol{z}(n)$ is the output of the blocking matrix.

Unfortunately, no closed form solution can be obtained for the above optimization problem. Alternatively, the BSCMA is invoked to update the weight vector and the Lagrange methodology is employed to solve the QI constraint. The new cost function and the gradient vector will be represented, respectively, as follows:

$$\Phi(\bar{\boldsymbol{f}}_a) = \frac{1}{4M}\sum_{n=0}^{N-1}\left[\bar{\boldsymbol{f}}_a^T(j-1)\boldsymbol{Z}(n)\bar{\boldsymbol{f}}_a(j-1) - 1\right]^2 + \frac{1}{2}\lambda.t\left(\bar{\boldsymbol{f}}_a^T(j-1)\bar{\boldsymbol{f}}_a(j-1) - \beta^2\right) \qquad (25)$$

$$\bar{\boldsymbol{g}}\left(\bar{\boldsymbol{f}}_a(j-1)\right) = \frac{1}{M}\sum_{n=0}^{N-1}\left[\bar{\boldsymbol{f}}_a^T(j-1)\boldsymbol{Z}(n)\bar{\boldsymbol{f}}_a(j-1) - 1\right]\bullet \boldsymbol{Z}(n)\bar{\boldsymbol{f}}_a(j-1) + \lambda(j)\bar{\boldsymbol{f}}_a(j-1) \qquad (26)$$

Therefore,

$$\bar{\boldsymbol{d}}(j) = -\boldsymbol{H}^{-1}\left(\bar{\boldsymbol{f}}_a(j-1)\right)\left[\bar{\boldsymbol{g}}\left(\bar{\boldsymbol{f}}_a(j-1)\right) - \lambda(j)\bar{\boldsymbol{f}}_a(j-1)\right] \qquad (27)$$

Using (27) into the primary update equation (18), we get:

$$\bar{\boldsymbol{f}}_a(j) = \boldsymbol{f}_a(j) + \lambda(j)\delta(j)\boldsymbol{H}^{-1}\left(\bar{\boldsymbol{f}}_a(j-1)\right)\bar{\boldsymbol{f}}_a(j-1) \qquad (28)$$

In order to simplify the computation of diagonal loading term (i.e. avoid the computation of Hessian matrix), the QI constraint term in (25) is updated using the steepest descent algorithm instead of Newton's algorithm with the same step-size $\delta(j)$. Therefore, the update equation (28) is simplified as

$$\bar{\boldsymbol{f}}_a(j) = \boldsymbol{f}_a(j) + \lambda(j)\delta(j)\bar{\boldsymbol{f}}_a(j-1) \qquad (29)$$

The constraint $\bar{\boldsymbol{f}}_a^T(j)\bar{\boldsymbol{f}}_a(j) \le \beta^2$ should be met, therefore using (29), we solve for $\lambda(j)$ as follows [7]:

$$\lambda(j) = \left[b \pm \sqrt{b^2 - 4ac}\right]\Big/2a \qquad (30)$$

where

$$a = \delta^2(j)\left\|\bar{\boldsymbol{f}}_a(j-1)\right\|^2$$
$$b = 2\delta(j)\boldsymbol{f}_a^T(j)\bar{\boldsymbol{f}}_a(j-1) \qquad (31)$$
$$c = \left\|\boldsymbol{f}_a(j)\right\|^2 - \beta^2$$

By substituting from (31) and (18) into $b^2 - 4ac \ge 0$, the following inequality is obtained

$$\left[(\bar{\boldsymbol{f}}_a(j-1) + \delta(j)\boldsymbol{d}(j))^T\bar{\boldsymbol{f}}_a(j-1)\right]^2 \ge \left\|\bar{\boldsymbol{f}}_a(j-1)\right\|^2 . \left[(\bar{\boldsymbol{f}}_a(j-1) + \delta(j)\boldsymbol{d}(j))^T(\bar{\boldsymbol{f}}_a(j-1) + \delta(j)\boldsymbol{d}(j)) - \beta^2\right] \qquad (32)$$

After some manipulations to (32), we get the following inequality [4],

$$\delta(j) \le \frac{\beta\left\|\bar{\boldsymbol{f}}_a(j-1)\right\|}{\sqrt{\left\|\bar{\boldsymbol{f}}_a(j-1)\right\|^2\left\|\boldsymbol{d}(j)\right\|^2 - \boldsymbol{d}^T(j)\bar{\boldsymbol{f}}_a(j-1)\bar{\boldsymbol{f}}_a^T(j-1)\boldsymbol{d}(j)}} \qquad (33)$$

Therefore, this upper bound on $\delta(j)$ guarantees real positive roots in (30) and consequently the optimal loading level can be obtained. Additionally, the two constraints (19) and (20) should be considered as well to guarantee BSCMA convergence.

C. Block Processing and Adaptive implementation

The proposed adaptive algorithm based on the block Shanno's algorithm and the VL technique is summarized in Table I. There are two main loops in the algorithm. The outer loop is for each block of data and the inner loop is repeated over the same block of data until certain number of iterations or the norm of the gradient vector is sufficiently small. A block of data with length $N_f \times M$ is selected and the sample covariance matrix of the lower branch from the PLIC structure $\boldsymbol{Z}(n)$ is commuted. The adaptive vector $\bar{\boldsymbol{f}}_a(1,1)$ is initialized with the required user signature waveform multiplied by the blocking matrix ($\boldsymbol{B}^T c_1$). On the other hand, the final weight vector $\bar{\boldsymbol{f}}_a(i,j)$ of each block is used as the initial vector to the next block (i.e. $\bar{\boldsymbol{f}}_a(i+1,1) = \bar{\boldsymbol{f}}_a(i,j)$). There is no need to but constraint on the gradient vector whereas the quadratic constraint on $\boldsymbol{f}_a$ will oversee the gradient vector norm increase. The gradient vector is estimated, taken into consideration the previous diagonal loading term $\lambda(j-1)\bar{\boldsymbol{f}}_a(i,j-1)$. The vector $\boldsymbol{d}(j)$ is computed according to (17) and then the initial adaptive vector is updated using (18). If the norm constraint on $\boldsymbol{f_a}(i,j)$ is not met, therefore the VL technique is invoked to fulfill the quadratic inequality constraint. Unfortunately, we will not be able to find an optimum step-size value due to the cost function (25) is not a quadratic equation and hence no global minimum for it with the step-size. Alternatively, the procedure provided in [5] is used in addition to (33) is added to guarantee positive diagonal loading. After convergence of certain bock of data the output of this block is calculated.

As shown in Table I, the total amount of the required computations for the proposed robust diagonal loading technique is about $O(4N_a)$ for each block-iteration. Therefore, the added complexity due to the robust technique is of order $O(N_a)$. As the complexity of the BSCMA is about $O(28.6N_a + 11N_a/M)$ [6], hence the total complexity of the proposed robust BSCMA is of order $O(28.6N_a + 15N_a/M)$ per output point from every block of data with length M.

IV. Simulations Results

In this section the performance of the block Shanno's algorithm and its two subsidiaries algorithms (i.e. BCGCMA and BGDCMA) will be compared with the corresponding developed robust algorithms. The robust techniques are referred, respectively, as BSCMA w. VL, BCGCMA w. VL and BGDCMA w. VL. Five asynchronous users in a multipath Rayleigh fading channel with 5 multipath components are simulated. Gold codes are exercised. Code length is 31 chips and detector length N_f is assumed to span one bit duration (i.e. 31 samples). The channel length (max delay spread) is assumed to be 10 delayed components and multipath delays are randomly distributed. The block length is 100 bits and the max iterations inside the block are 25 iterations.

Two scenarios are simulated; the first scenario assumes perfect power control (i.e. equal user powers). In the second scenario, users have equal power except that the required user is 10 dB less than other users to model the near-far effect. The performance of the six detectors is assessed in terms of output signal-to-interference plus noise ratio (SINR) and bit error rate (BER) versus block iterations. Figures 1-4 show the SINR and BER for the two scenarios, respectively. The figures show that the quadratically constraint BSCMA w. VL algorithm and its variants: BCGCMA w. VL and BGDCMA w. VL exhibit a considerable improvement over corresponding non-robust algorithms. At perfect power control the three robust algorithms perform almost the same in terms of BER. The BGDCMA present the worst convergence speed. For both scenarios, the convergence of the robust algorithms are almost attained after about 50 block iterations which means 2 block of data are required for convergence. On the other hand, the non-robust algorithms require at least 150 block iterations to attain the same convergence point of the corresponding robust algorithms. Finally, in the second scenario, the steady state behavior of the proposed robust algorithms is superior over the corresponding non-robust algorithms in terms of BER and SINR.

TABLE I
SUMMARY OF THE ROBUST BSCMA RECEIVER

- Initialization:
  - $\rho^2 = t.\left(\|\boldsymbol{f_c}\|^2\right), \beta^2 = \rho^2 - \|\boldsymbol{f}_c\|^2, \varepsilon = 0.1$
  - $\lambda(0) = 0\,; \boldsymbol{d}(0) = \boldsymbol{g}\left(f_a(0)\right), \alpha = 0.25, \sigma = 0.5$
- For $i = 1, 2, \cdots, \lceil N/M \rceil$ outer loop on block basis
  - $D_i = \left[\boldsymbol{x}\left((i-1)M\right) \vdots \boldsymbol{x}\left((i-1)M+1\right) \vdots \cdots \vdots \boldsymbol{x}\left(iM-1\right)\right]$
  - $\boldsymbol{Z}(i) = \sum_{n=(i-1)M}^{iM-1} \boldsymbol{z}(n)\boldsymbol{z}^T(n), \boldsymbol{z}(n) = \boldsymbol{B}^T\boldsymbol{x}(n)$
    - if $(i=1)$ $\overline{\boldsymbol{f}}_{\boldsymbol{a}}(1,1) = \boldsymbol{B}^T c_1$
    - Else $\overline{\boldsymbol{f}}_{\boldsymbol{a}}(i,1) = \overline{\boldsymbol{f}}_{\boldsymbol{a}}(i,j)$
  - $j = 0$
  - For $j = 1, 2, \cdots,$ until converge or max number of iterations reach
    - $\overline{\boldsymbol{g}}\left(\overline{f}_a(i,j-1)\right) = \frac{1}{M}\sum_{n=0}^{N-1}\left[\overline{f}_a^{\,T}(i,j-1)\boldsymbol{Z}(n)\overline{f}_a(i,j-1) - 1\right] \bullet \boldsymbol{Z}(n)\overline{\boldsymbol{f}}(i,j-1)_a - \lambda(j-1)\overline{\boldsymbol{f}}_a(i,j-1)$
    - If $j = 1$
      - $\boldsymbol{d}(j) = \overline{\boldsymbol{g}}\left(\overline{\boldsymbol{f}}_a(i,j-1)\right)$
    - Else
      - $\boldsymbol{d}(j) = \boldsymbol{d}(j-1)e(j) + \left(a(j)-1\right)\overline{\boldsymbol{g}}\left(\overline{\boldsymbol{f}}_a(i,j-1)\right)$
    - $\boldsymbol{f}_a(i,j) = \overline{\boldsymbol{f}}_a(i,j-1) + \delta(j)\boldsymbol{d}(j)$
    - *if* $\left(\|\boldsymbol{f}_a(i,j)\|^2 > \beta^2\right)$
      - $c = \|\boldsymbol{f}_a(i,j)\|^2 - \beta^2, N_a$
      - $b = 2\delta(j)\boldsymbol{f}_a^T(i,j)\overline{\boldsymbol{f}}_a(i,j-1),\ 2N_a$
      - $a = \delta^2(j)\|\overline{\boldsymbol{f}}_a(i,j-1)\|^2, N_a$
      - $\lambda(j) = -b - \left(\sqrt{b^2-4ac}\right)\Big/2a$
      - $\overline{\boldsymbol{f}}_a(i,j) = \boldsymbol{f}_a(i,j) + \lambda(j)\delta(j)\overline{\boldsymbol{f}}_a(i,j-1)$
    - *else*
      - $\overline{\boldsymbol{f}}_{\boldsymbol{a}}(i,j) = \boldsymbol{f}_a(i,j)\,;\ \lambda(j) = 0$
    - End if
    - If $\left[\Phi\left(\overline{\boldsymbol{f}}_a(i,j)\right) \le \Phi\left(\overline{\boldsymbol{f}}_a(i,j-1)\right) + \alpha\delta(j)\left(\overline{\boldsymbol{g}}\left(\overline{\boldsymbol{f}}_a(i,j-1)\right)\right)^T\overline{\boldsymbol{d}}(j)\right]$
      - $\delta(j+1) = \delta(j) + \varepsilon\delta(j)$
    - Else If $\left[\left(\overline{\boldsymbol{g}}\left(\overline{\boldsymbol{f}}_a(j)\right)\right)^T\overline{\boldsymbol{d}}(j) \ge \sigma\left(\overline{\boldsymbol{g}}\left(\overline{\boldsymbol{f}}_a(j-1)\right)\right)^T\overline{\boldsymbol{d}}(j)\right]$
      - $\delta(j+1) = \delta(j) - \varepsilon\delta(j)$
    - else if
    - $\delta(j) > \frac{\beta\|\overline{\boldsymbol{f}}_a(i,j)\|}{\sqrt{\|\overline{\boldsymbol{f}}_a(i,j)\|^2\|\boldsymbol{d}(j)\|^2 - \boldsymbol{d}^T(j)\overline{\boldsymbol{f}}_a(i,j)\overline{\boldsymbol{f}}_a^T(i,j)\boldsymbol{d}(j)}} = \delta(j+1)$
    - otherwise
      - $\delta(j+1) = \delta(j)$
    - End if
  - End for j loop
- $\boldsymbol{f}(i,j) = \boldsymbol{f}_c - \boldsymbol{B}\overline{\boldsymbol{f}}_a(i,j)$
- $\boldsymbol{y}\left(((i-1)M):(iM-1)\right) = \boldsymbol{x}\left(((i-1)M):(iM-1)\right)\boldsymbol{f}(i,j)$
- End for i loop

V. Conclusions

In this paper, we have proposed a new robust block adaptive blind multiuser detector based on the BSCMA with a QI constraint on the weight vector norm. The weight vector is updated using the BSCMA algorithm and concurrently a QI constraint is imposed on weight vector norm. The required amount of diagonal loading is precisely computed using a VL technique with low computational complexity. The complexity of the robust proposed receiver is still of order $O(M)$ complexity. It is shown from the simulations that the new proposed detector outperforms the BSCMA algorithm in both robustness, and steady state performance. Moreover, we have incorporated the robust technique with both BCGCMA and BGDCMA algorithms and a performance improvement over corresponding non-robust algorithms is reported as well. Future work may include practical application in 3G systems.

References

[1] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, Vol. COMM-28, pp. 1867-1875, Nov. 1980.

[2] H. H. Zeng, L. Tong, and C. R. Johnson, Jr., "Relationships between the constant modulus and Wiener receivers," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1523-1538, July 1998.

[3] S. R. Nelatury, and S. S. Rao "Increasing the speed of convergence of the constant modulus algorithm for blind channel equalization" *IEEE Trans. On Comm.*, Vol. 50, No. 6, pp.872-876, June 2002.

[4] A. Elnashar, S. Elnoubi, and H. Elmikati "A robust linearly constrained CMA for DS/CDMA systems" has been accepted for presentation at *IEEE WCNC*, New Orleans, LA, USA, March 2005.

[5] Z. Wang and E. M. Dowling, "Block-Shanno constant modulus algorithm for wireless equalizations," *Proc. ICASSP, Atlanta*, GA, USA, April 1996, pp. 2678-2681.

[6] U. G. Jani, E. M. Dowling, R. M. Golden, Z. Wang, "Multiuser interference suppression using block Shanno constant modulus algorithm," *IEEE Trans. On Sig. Proc*. Vol., 48, No. 5, May 2000.

[7] A. Elnashar, S. Elnoubi, and H. Elmikati "A Robust Quadratically Constrained Adaptive Blind Multiuser Receiver for DS/CDMA Systems," *ISSSTA 2004*, Sydney, Australia, pp.164-168, Aug 2004.

[8] J. B. Schodorf and D. B. Williams, "A Constrained Optimization Approach to Multi-user Detection," *IEEE Trans. Signal Processing.*, vol. 45, no. 1, pp. 258-262, Jan. 1997.

[9] R. M. Golden, Mathematical Methods for Neural Networks Analysis and Design, Cambridge, MA: MIT Press, 1996.

[10] Simon Haykin, Neural Networks: A Comprehensive Foundation, Second Edition, Prentice Hall, Inc. 1999.

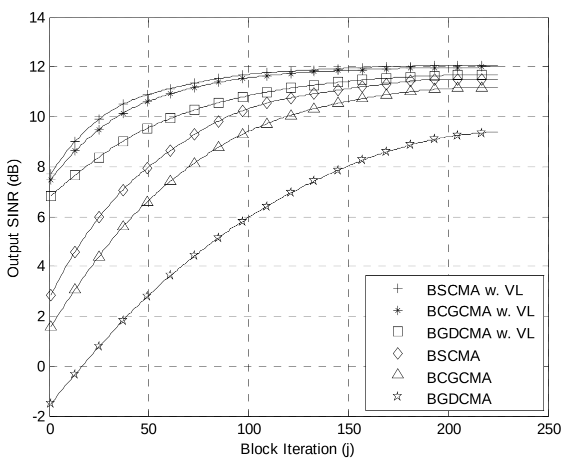

Figure 1. SINR versus block iterations for first scnario

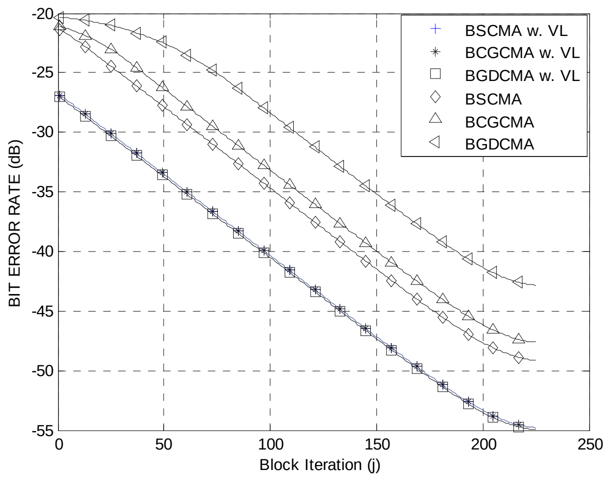

Figure 2. BER versus block iterations for first scnario

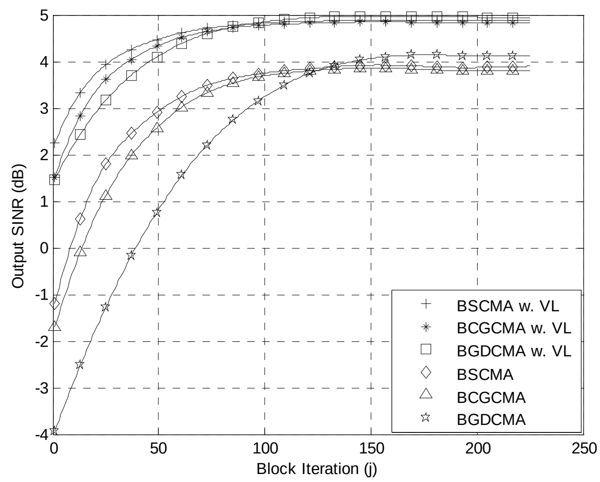

Figure 3. SINR versus block iterations for second scnario

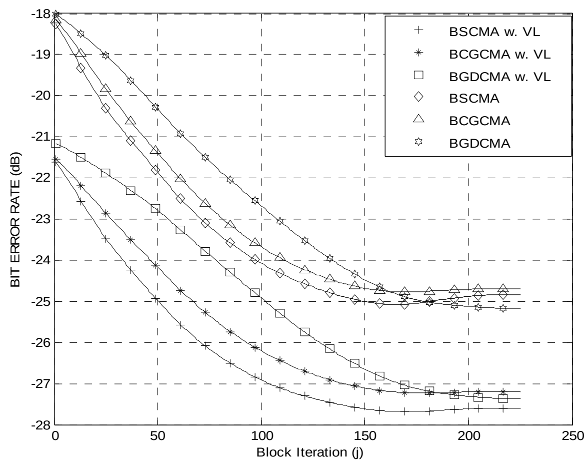

Figure 4. BER versus block iterations for second scnario