Fast Convergent Multiuser Constant Modulus Algorithm for use in Multiuser DS-CDMA Environment

P. Arasaratnam, S. Zhu, A.G. Constantinides

Communication and Signal Processing Group, Dept. Of Electrical and Electronic Engineering,
Imperial college of Science, Technology and Medicine, Exhibition Road, London, SW7 2BT, U.K.
e-mail: a.kieran@ic.ac.uk

ABSTRACT

In a multi-user direct sequence code division multiple access (DS-CDMA) system, retrieval of the transmitted signals with a conventional receiver is difficult when there is chip asynchronism, multipath propagation and the associated near-far problem. The results show that despite these problems an adaptive linear receiver based on a mixed cross correlation and constant modulus algorithm (CC-CMA) has the potential to retrieve all users simultaneously. In this paper, we propose several algorithms based on Quasi-Newton type algorithms. Compared with the conventional CC-CMA algorithm, the proposed algorithms achieve faster convergence at low computational cost. Simulations support the improved convergence properties of the algorithm.

1. INTRODUCTION

In a communication system, channel equalisation/interference cancellation is required to eliminate intersymbol interference (ISI) and multiple access interference (MAI). Blind algorithms, which do not require a training sequence, have the potential to increase channel bandwidth efficiency. Among all blind equalisation algorithms, the constant modulus algorithm (CMA), which exploits the constant modulus property of the signal, appears to be the algorithm of choice due to its computational simplicity [2]. For single-user and multi-user environments with zero mean, circularly symmetric sources, provided that the so-called zero and length condition is satisfied, the CMA algorithm converges globally to a solution where ISI and MAI are removed. To prevent the identical source being repeatedly retrieved in the multi-user environment, a multi-user CMA was proposed in [4]. However, when the adaptation of the CC-CMA algorithm is realised with normal stochastic gradient type algorithm, the rate of convergence to such a solution can be too slow. To mitigate this limitation, We employ an approximation of the Hessian matrix to represent the local curvature of the CC-CMA cost function and hence order of magnitude improved convergence rate. The problem with such approaches is that considerable memory/computational requirements are required for storing, updating, and inverting an estimate of the Hessian of objective. In this paper we propose an alternative type of modified Newton algorithm that implicitly computes a positive definite approximation to the inverse of the Hessian of the objective function using computational/memory requirements which are roughly $O(M)$ as opposed to $O(M^2)$. The proposed algorithm called Block Shanno Cross Correlation Constant Modulus Algorithm, which shows superior convergence speed, better performance and blindly retrieve all multiuser DS-CDMA sources than conventional CMA algorithm which fails when applied to multiuser DS-CDMA environment.

2. SYSTEM MODEL AND CC-CMA ALGORITHM

We assume U users and a real binary phase shift keying (BPSK) modulation scheme with source alphabets $+1$ and -1 . Hence, we need a bank of U parallel equalisers to retrieve all the sources. The signal is received from the output of a multiple input multiple output (MIMO) channel. The channel from the source j to the sensor i is modelled as an FIR filter of order M , i.e. $C_{ij}(z^{-1}) = \sum_{k=0}^M c_{ij}(k)z^{-k}$. Define the source vector at time k as $s(k) = [s_1^T(k) s_2^T(k) \dots s_U^T(k)]^T$ where the i^{th} source vector is $[s_i(k) s_i(k-1) \dots s_i(k-N-M)]^T$. Denote the j^{th} sub-equaliser regressor as $x_j(k) = [x_j(k) x_j(k-1) \dots x_j(k-N)]^T$ and the fractionally spaced equaliser regressor is $x(k) = [x_1^T(k) x_2^T(k) \dots x_R^T(k)]^T$, where N is the order of the sub-equaliser. Define Δ^T as the channel convolution matrix of dimension $R(N+1) \times U(N+M+1)$, which is assumed to be full column rank. Hence, we obtain $x(k) = \Delta^T s(k)$. Denote the l^{th} user equaliser tap vector as $w_l = [w_1^0 \dots w_1^N w_2^0 \dots w_2^N \dots w_R^0 \dots w_R^N]^T$. The equaliser output is given by

$$y_l(k) = w_l^T \Delta^T s(k) \quad (1)$$

where $h_l = \Delta w_l$ is the combined channel l^{th} equaliser impulse response. The CC-CMA algorithm cost function for the l^{th} equaliser is written as

$$J_l = E\left\{\left(y_l^2(k) - 1\right)^2\right\} + \kappa \sum_{m=1}^{l-1} \sum_{\delta=-(N+M)}^{N+M} E\left\{\left(y_l(k) y_m(k - \delta)\right)\right\}^2 \quad (2)$$

where $E\left\{\left(y_l^2(k) - 1\right)^2\right\}$ is the constant modulus cost, $\sum_{m=1}^{l-1} \sum_{\delta=-(N+M)}^{N+M} E\left\{\left(y_l(k) y_m(k - \delta)\right)\right\}^2$ is the cross-correlation between the l^{th} equaliser and the retrieved sources and $\gamma \in \mathbb{R}^+$ is the mixing parameter.

3. SHANNO'S ALGORITHM

Given an objective function $f: \mathbb{R}^M \rightarrow \mathbb{R}$, the classical Newton's algorithm [1, 214-215] updates the filter tap weights as

$$w_j = w_{j-1} H_{newton}^{-1}(w_{j-1}) g(w_{j-1}) \quad (3)$$

where $H_{newton}(w_{j-1}) \in \mathbb{R}^{M \times M}$ is the Hessian of the objective function f , $g(w_{j-1}) \in \mathbb{R}^M$ is the gradient of the objective function f , evaluated at w_{j-1} and j is an iteration index. Shanno's algorithm is a type of modified Newton's algorithm that approximates the inverse of the Hessian $H_{newton}^{-1}(w_{j-1})$ by the Shanno approximation, which is a positive definite matrix

denoted by $H^{-1}(w_{j-1})$. This matrix is implicitly determined by the gradients of the two most recent iterations, a search direction, and a step-size based on a previous iteration. The complexity of the implicit calculation of $H^{-1}(w_{j-1})$ is of $O(M)$, as opposed to $O(M^2)$, for the calculation of $H_{\text{newton}}^{-1}(w_{j-1})$. Shanno's algorithm builds up a trajectory of $w_0, w_1, w_2, \dots$ iteratively according to

$$w_j = w_{j-1} + \gamma_j d_j \quad (4)$$

where d_t is the search direction, written as

$$d_j = -H^{-1}(w_{j-1})g(w_{j-1}) \quad (5)$$

Shanno's approximation of the inverse of the Hessian is given by [1, 218-220].

$$H^{-1}(w_{j-1}) = I - \frac{u_j d_{j-1}^H + d_{j-1} u_j^H}{d_{j-1}^H u_{j+1}} + c_j \frac{d_{j-1} d_{j-1}^H}{d_{j-1}^H u_j} \quad (6)$$

Suppose γ_j is positive real number that satisfies the step size constraints

$$f(w_{j-1} + \gamma_j d_j) \leq f(w_{j-1}) + \alpha \gamma_j g(w_{j-1})^H d_j \quad (7)$$

and

$$g(w_{j-1} + \gamma_j d_j)^H d_j \geq \beta g(w_{j-1})^H d_j \quad (8)$$

where $0 < \alpha < \beta < 1$, [1]. A procedure for calculating the step size is outlined in [1]. If γ_j satisfies the above constraints, the algorithm is guaranteed to converge to a set of critical points [1], [3].

4. BLOCK SHANNO CC-CMA FORMATION AND GRADIENT CALCULATION

In order to apply Shanno's algorithm, we first extend the CC-CMA objective function to admit block processing. Let a block of data be defined as:

$$D_i = [x((i-1)N); x((i-1)N+1); \dots; x(iN-1)] \in C^{M \times N} \quad (9)$$

and define a block CC-CMA objective function

$$f(w) = \frac{1}{4N} \sum_{\Omega=0}^{N-1} [|w_c^H x_{i_1}((i-1)N + \Omega)|^2 - 1]^2 \quad (10)$$

$$+ \frac{\kappa}{N^2} \sum_{\delta=-M}^M \sum_{i_2=i_1}^{N-1} \left[\sum_{\Omega=0}^{N-1} w_c^H x_{i_1}((i-1)N + \Omega) x_{i_2}((i-1)N + \Omega - \delta) w_c \right]^2$$

In (10), $x((i-1)N + \Omega) \in C^M$ represents the received signal vector corresponding to the n th transmitted symbol interval, where $n = (i-1)N + \Omega$, $i \geq 0$, $0 \leq \Omega < N$, and the length of x , i.e. M , is equal to the length of the filter tap. Here, w_c is a complex filter tap weight vector for the l th equaliser. Since the Shanno algorithm requires the input data to be real, we develop an alternative form of the block objective function to

accommodate complex input signals involved with complex channels. First, we define

$$w = \begin{bmatrix} w_c^r \\ w_c^i \end{bmatrix} \quad (11)$$

and then define two kinds of real vectors containing complex data,

$$x_f(n) = \begin{bmatrix} x^r(n) \\ -x^i(n) \end{bmatrix} \quad \text{and} \quad x_b(n) = \begin{bmatrix} x^i(n) \\ -x^r(n) \end{bmatrix} \quad (12)$$

Next, we define the real objective function and its gradient with respect to w as

$$f(w) = \frac{1}{4N} \sum_{n=0}^{N-1} [w^T X_1(n)w - 1]^2$$

$$+ \frac{\kappa}{N^2} \sum_{\delta=-M}^M \sum_{i_1 \neq i_2} \left[\sum_{n=0}^{N-1} w^T X_2(n)w \right]^2 \quad (13)$$

and

$$\nabla f(w) = g(w) = \frac{1}{N} \sum_{n=0}^{N-1} [w^T X_1(n)w - 1] X(n)w$$

$$+ \frac{\kappa}{N^2} \sum_{\delta=-M}^M \sum_{i_1 \neq i_2} \left\{ \left[\sum_{n=0}^{N-1} w^T X_2(n)w \right] \left[\sum_{n=0}^{N-1} X_2(n)w \right] \right\} \quad (14)$$

where

$$X_1(n) = x_{i_1,f}(n) x_{i_1,f}^T(n) + x_{i_1,b}(n) x_{i_1,b}^T(n) \quad (15)$$

$$X_2(n) = x_{i_1,f}(n) x_{i_2,f}^T(n - \delta) + x_{i_1,b}(n - \delta) x_{i_2,b}^T(n) \quad (16)$$

4.1. BSCC-CMA ALGORITHM

Let: i - (Initially set to 0) index for data blocks, $f(w)$ - block CC-CMA objective function defined in (10) evaluated at w_i^j , $g(w_i^j)$ - gradient of block CC-CMA defined in (14) evaluated at w_i^j ;

Step 1: Form a block of data D_i (of size $M \times N$, where N is the block size) from the received signal using (9). Set $j = 0$. Initialise the filter tap weight vector w_i^0 to all elements being zero except one element being unity.

Step 2: Do the following.

- i : $j = j + 1$,
- ii : Calculate the gradient of this block of data at w_i^{j-1} ,
- iii : If j is equal to one, then set $d_i^j = -g(w_i^{j-1})$; else

$$d_i^j = -g(w_i^{j-1}) + a_i^j w_i^j + (b_i^j - c_i^j a_i^j) d_i^{j-1} \quad (17)$$

$$\text{where } w_i^j = g(w_i^{j-1}) - g(w_i^{j-2}), \quad a_i^j = \frac{d_i^{j-1T} g(w_i^{j-2})}{d_i^{j-1T} w_i^j} \quad (18)$$

$$b_i^j = \frac{w_i^{jT} g(w_i^{j-1})}{d_i^{j-1T} w_i^j}, \quad c_i^j = \gamma_i^{j-1} + \frac{|w_i^j|^2}{d_i^{j-1T} w_i^j} \quad (19)$$

iv: Update the filter tap weights $w_i^j = w_i^{j-1} + \gamma_i^j d_i^j$, where γ_i^j is a positive real number that satisfies the constraints in (7) and (8).

v: If $|g(w_i^j)|$ is sufficiently small, then go to Step 3; else, go to the beginning of Step 2. If $|g(w_i^j)|$ is too large, reinitialise, set $j=0$, and go to Step 2.

Step 3: Generate output for this block (D_i), form the next block of data (D_{i+1}), set $w_{i+1}^j = w_i^j$, $j = 0$, $i = i + 1$, and go to Step 2.

The step-size γ^j selection and the multiple checking procedure for the Shanno's algorithm can be updated as [1]. For comparative purposes, we also propose and compare the CC-CMA with two other block-based methods that are variants of the BSCC-CMA algorithm. In (17) if we set $a_i^j = 0$, a block conjugate gradient CC-CMA (BCGCC-CMA) algorithm results [1]. Furthermore, if we set a_i^j and b_i^j both equal to zero, then we obtain a block gradient descent CC-CMA (BGDCC-CMA) algorithm [1].

5. BLOCK STOCHASTIC SHANNO CC-CMA ALGORITHM II (BSSCC-CMA).

In this section we will be looking at another method also based on Shanno's algorithm for less computational cost. The Block Stochastic Shanno CC-CMA (BSSCC-CMA) algorithm also preserves the fast convergence rate of the BSCC-CMA for less computational cost, which is the main problem with BSCC-CMA algorithm. For every block of data, the BSCC-CMA algorithm updates the filter tap weights along specified search directions until a stopping criterion is reached. Next it produces a block of outputs and proceeds to the next block of data. However it is very expensive to let the BSCC-CMA converge all the way before proceeding to the next block of data. Due to the limitation of space, the algorithm can be summarised as follows. For every K samples, a data block of size N is formed, a search direction is determined (using Shanno's Techniques as described above), and a appropriate step size is determined based on the current block of data. Next the filter tap weights are updated once, and block of K outputs are produced. Now instead of converging on each block, we find one search direction per block, and span the search direction across blocks. This eliminates costly inner loops, and reduces complexity significantly. The BSSCC-CMA algorithm above can be easily modified to form Block Stochastic Shanno CC-CMA algorithm. We implemented the so called *back tracking line search* [1] scheme to find appropriate step size.

To provide more control over the balance between the computational complexity and performance, we further introduce the sliding-window type BSSCC-CMA. The sliding-window BSSCC-CMA also takes in N data at a time, but generate only K outputs, $K \leq N$, that is, $K - N$ data elements from the current block will be reused for the next data block. The sliding-window BSSCC-CMA scheme provides control over the balance between the computational complexity and performance. Such flexibility will be very useful in higher data rate system. By adding more overlap between blocks, we increase the convergence rate and lower the residual error floor. However, we also raise the computational complexity.

6. STOCHASTIC NEWTON-LIKE CC-CMA ALGORITHM

Shanno's algorithm derived in previous sections involves only the implicit computation of the Hessian. However, Shanno's algorithm does the conjugate gradient search without fully optimising the step size. The step size is chosen to be within specified range such that the convergence is guaranteed. This feature saves computations as it involves only an implicit computation of the Hessian. There is no need to compute, update and store the inverse of the Hessian, the most costly part of the nonlinear Newton's algorithms. Here, we consider a type of Newton-like CC-CMA (SNLCC-CMA) algorithm based on the work [7] that requires explicit computation, update and storage of inverse of the Hessian. This algorithm also achieves faster convergence but at higher computational cost. The Hessian of the cross-correlation constant modulus algorithm (CC-CMA) is written as:

$$H_{cc-cma}^l(k) = 2E\left\{2\left(|y_l(k)|^2 - R_2\right)x(k)x^H(k)\right\} + 2\kappa \sum_{m=1}^{l-1} \sum_{\delta=-(M+N)}^{N+M} E\left\{y_m^*(k-\delta)x(k)\right\} E\left\{y_m^*(k-\delta)x^H(k)\right\} \quad (20)$$

When it comes to implementation, we can drop the expectation in the gradient, as it done usually. For the Hessian matrix, the expectation can be calculated by time average. If we include a forgetting factor λ in the summation, which places more weight on the most recent data and is useful for tracking for time-varying channels. However, what we need is the inverse of the Hessian matrix. In order to calculate the inverse efficiently, we need the well-known matrix inversion lemma [8].

Denote $\Psi(n) = \sum_{i=1}^n \lambda^{n-i} \left[2|y(i)|^2 - R_2\right] x(i)x^H(i)$ and $\Psi^{-1}(i) = P(n)$. By utilising the woodbury's identity [8] an approximate form for the inverse of H_{cc-cma} can be written as

$$H_l^{-1}(k) = \frac{1 - \lambda^k}{2(1 - \lambda)} \sum_{i=1}^k \Psi^{-1}(k) - \frac{\frac{1-\lambda^k}{2(1-\lambda)} \sum_{i=1}^k \Psi^{-1}(k) \left((b(k)b^H(k)) \cdot I \right) \frac{1-\lambda^k}{2(1-\lambda)} \sum_{i=1}^k \Psi^{-1}(k)}{\kappa^{-1} \left\{ \frac{1-\lambda^k}{1-\lambda} \right\}^2 + b^H(k) \frac{1-\lambda^k}{2(1-\lambda)} \sum_{i=1}^k \Psi^{-1}(k) b(k)}$$

Where the term $(b(k)b^H(k)) \cdot \mathcal{I}$ denotes the diagonal matrix of the term in (22).

$$\sum_{m=1}^{l-1} \sum_{\delta=-(M+N)}^{M+N} \left\{ \sum_{i=1}^k \lambda^{k-i} y_m^*(i-\delta)x(i) \right\} \times \sum_{i=1}^k \lambda^{k-i} y_m^*(i-\delta)x^H(i)$$

Isolating the last term in the summation of $\Psi(n)$, we may write

$$\Psi(n) = \lambda \Psi(n-1) + x(n) \left\{ 2|y(n)|^2 - R_2 \right\} x^H(n) \quad (22)$$

By using the matrix inversion lemma [8], we have

$$P(n) = \lambda^{-1} P(n-1) - \frac{\lambda^{-2} P(n-1) x(n) x^H(n) P(n-1)}{\left\{ 2|y(n)|^2 - R_2 \right\}^{-1} + \lambda^{-1} x^H(n) P(n-1) x(n)} \quad (23)$$

Combining equation (23) and stochastic gradient of CC-CMA, the QNCC-CMA update equation for the l^{th} equaliser at the k^{th} iteration is written as

$$w_l(k) = w_l(k-1) - \mu \left[2 \left(|y_l(k)|^2 - R_2 \right) y_l^*(k) + \kappa \sum_{l=1}^{L-1} \sum_{m=1}^{N+M} E \left\{ y_l^*(k) y_m(k-\delta) \right\} y_m^*(k-\delta) \right] H_l^{-1}(k) x(k) \quad (24)$$

where μ is the scalar step size and its value can be adaptively chosen with respect to the inverse Hessian matrix $H_l^{-1}(k)$.

Computational complexity: Let $p = M(n+1)$, for a p tap space-time equaliser, $7p^2 + 4p + (L-1)(4N+4M+5+p(2N+2M+1)) + 6$ complex multiplications, $4p^2 + 3$ complex divisions and p square root operations are required in each iteration of the SNLCC-CMA algorithm. Compared with the conventional CC-CMA algorithm, which only requires $O(M(N+1))$ multiplications and divisions. By using SNLCC-CMA, fast convergence can be achieved at the expense of increased computational complexity.

Figure 1. Comparison between SNLCC-CMA and CC-CMA (a) Eye diagram of EQ-1 after 300 samples (b) Eye diagram of EQ-2 after 500 samples (c) h1: combined channel + EQ-1 impulse response (d) h2: combined channel + EQ-2 impulse response. (e) Residual error of EQ-1 (f) Residual error of EQ-2.

7. SIMULATIONS

In a QPSK system with source alphabet $(\pm \frac{1}{\sqrt{2}} \pm \frac{1}{\sqrt{2}}j)$, we assume $d=3$ users, $R=4$ sensors twelve random complex channels of order 1 ($M=1$) and three 12 tap (i.e. $N=2$ and $R(N+1)=12$) space-time equalisers adapted with the proposed algorithms. The block size for used for BSCC-CMA, BSGCC-CMA and BGDCC-CMA was 100. The channel convolution matrix Δ^H is full rank square matrix of dimension of 12 and $\kappa = 4$. White Gaussian noise of SNR 30dB is present at the channel output. The forgetting factor λ for the SNLCC-CMA and the step size μ are respectively set to 0.99 and 0.01. The constellation diagrams of the output of the equalizers are shown in Fig 1. (a) and (b). After approximately 700 samples, all equalisers of SNLCC-CMA algorithm give open eye pattern. Comparing the residual error, which is defined as $\frac{\|h_l(k)\|_2^2 - \max(\|h_l(k)\|_2^2)}{\max(\|h_l(k)\|_2^2)}$, in fig 1 (e), (f). SNLCC-CMA algorithm shows much faster convergence than the conventional CC-CMA algorithm. In fact, for equalizer-2, clear open-eye constellation cannot be achieved even with 3900 samples. In Fig. 1. (c), (d), the combined channel + equalizer the three equalizers is given. The different position sections of the largest impulse confirm the retrieval all three sources. That is, equalizer-1, 2 and 3 retrieve

sources with different delays. Fig. 2 (a), (b), (c) shows the ISI of proposed BSCC-CMA, BSGCC-CMA and BGDCC-CMA. All of the proposed algorithms converge within the first 5 blocks. Fig. 3 (a), (b), (c) shows the combined channel + equalizer response for the proposed BSCC-CMA algorithm. Fig.3 (d), (e), (f) shows the output constellation for the BSCC-CMA algorithm.

Fig. 2. Intersymbol Interference between the proposed BSCC-CMA, BSGCC-CMA, BGDCC-CMA for EQ-1, EQ-2 and EQ-3.

8. CONCLUSIONS

In order to overcome the slow convergence of the conventional CC-CMA algorithm, we proposed several algorithm based on Shanno's algorithm and also based on Newton like algorithms. Simulation shows the all of these proposed algorithms outperforms the conventional CC-CMA in terms of their super fast convergence and the compactness of their output constellations.

Fig. 3. BSCC-CMA- (a) h1: combined channel + EQ-1 impulse responses (b) h2: combined channel + EQ-2 impulse responses (c) h3: combined channel + EQ-3 impulse responses (d) Eye diagram of EQ-1 (e) Eye diagram of EQ-2 (d) Eye diagram of EQ-3.

9. REFERENCES

1. R. M. Golden, "Mathematical Methods for Neural Network Analysis and Design," Cambridge, MA: MIT Press, 1996.
2. D. N. Goddard, "Self-recovering equalisation and carrier tracking in two dimensional data communication systems", *IEEE Trans. Commun.*, 1980, COM-28, (11), pp, 1867-1875.
3. D. F. Shanno, "Conjugate Gradient methods with inexact searches," *Mathematics of Operations Research*, vol. 3, pp. 244-256, August 1978.
4. S. Lambotharan, J. A. Chambers and A. G. Constantinides, "Adaptive blind retrieval techniques for multi-user DS-CDMA signals", *IEE Elect Lett.* Apr 1999, vol. 35, (9).
5. Z. Wang and E. M. Dowling, "Block Shanno Constant Modulus Algorithm for Wireless Equalisation," in *Proc. ICASSP*, Atlanta, GA, Apr. 1996, pp. 2678-2681.
6. P. E. Gill, *Practical Optimisation*, New York, Academic Press, 1981.
7. G. Yan and H. Fan, "A Newton-Like algorithm for complex variables with applications in blind equalization", *IEEE Trans. on Signal Processing*, vol. 48, no. 2, pp. 553-556, 2000.
8. S. Haykin, *Communication Systems*, New York, 1983.