NEW ALGORITHM FOR BLIND ADAPTIVE EQUALIZATION BASED ON CONSTANT MODULUS CRITERION

Y. J. Kou, W.-S. Lu, and A. Antoniou

Department of Electrical and Computer Engineering
University of Victoria, Victoria, B.C., Canada V8W 3P6
{ykou, wslu,aantoniou}@ece.uvic.ca

ABSTRACT

Constant modulus (CM) based algorithms for blind channel equalization are well known for their effectiveness and simplicity. Recently, new CM-based equalization algorithms with improved performance have been proposed. In this paper, a new blind adaptive CM equalization algorithm using a quasi-Newton optimization method is proposed. Simulation results are presented which demonstrate that the proposed algorithm leads to an improved convergence rate as well as reduced computational complexity relative to those of some existing algorithms.

1. INTRODUCTION

Adaptive channel equalization techniques have been widely used in communication systems to deal with intersymbol interference (ISI) caused by channel distortion or multipath transmission. Conventional equalization algorithms require the transmission of a training signal to update the parameters of the equalizer. This inevitably reduces channel capacity. In addition, the inclusion of a training signal increases the complexity of the transceiver significantly. Therefore, blind adaptive equalization algorithms that do not require a training phase are often preferred. Among various blind equalization algorithms, constant-modulus (CM) based algorithms are well known for their effectiveness and simplicity [1]. However, these are usually implemented in terms of gradient based algorithms which are usually quite slow [2][3]. Recently, several improved CM-based blind adaptive equalization algorithms have been proposed. In [4], a blind equalization algorithm based on stochastic gradient decent minimization of order-α Renyi's entropy was proposed. A fast recursive constant modulus algorithm (RCMA) based on the recursive least square (RLS) algorithm was proposed in [5]. These algorithms reduce the time required for convergence at the cost of increased computational complexity.

In this paper, a new blind adaptive CM equalization algorithm using a quasi-Newton optimization method [6] is derived. Simulation results are presented to demonstrate that the proposed algorithm outperforms the algorithms in [1][5] in terms of convergence rate and achieves reduced compuatational complexity relative to that of RCMA algorithm.

2. PROBLEM FORMULATION

Consider the digital communication system depicted in Fig. 1, where a_i, d_i, x_i and n_i represent the CM input signal, channel output signal, received signal, and additive white Gaussian noise (AWGN), respectively. In communication systems, the channel characteristics are far from ideal and a channel equalizer is ofter needed to combat ISI especially in the case of wireless communications.

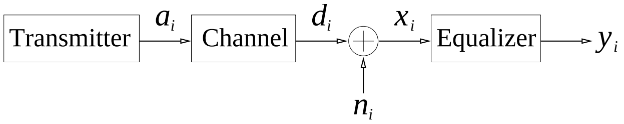

Fig. 1. Block diagram of a digital communication system.

The output signal of the equalizer in Fig. 1 can be expressed as

$$y_i = \mathbf{w}^H \mathbf{x}_i \tag{1}$$

where $\mathbf{x}_i = [x_{i-n+1}\ x_{i-n+2}\ \cdots\ x_i]^T$ is a block of input samples available at time instant i, $\mathbf{w}$ is an n-dimensional weight vector, and n is the length of the equalizer. If perfect equalization is achieved, then y_i has a constant instantaneous modulus. The aim of a constant-modulus based equalizer, therefore, is to minimize modulus variations of sequence y_i. Mathematically, the optimization problem for the equalizer can be formulated as

$$\underset{\mathbf{w}}{\text{minimize}} \quad f(\mathbf{w}) = \sum_{i=1}^{N} (|y_i|^2 - 1)^2 \tag{2}$$

where N is the length of sequence y_i.

3. QUASI-NEWTON ALGORITHM

The objective function in (2) is a fourth-order polynomial of variable $\mathbf{w}$ and is not in general convex. A commonly used optimization method to solve the problem in (2) is the steepest descent method (SDM) [6]. In each iteration, the SDM uses the gradient $\nabla f(\mathbf{w})$ to compute a search direction which in conjunction with a line search step determines the next iterate. Various least-mean-squares (LMS) algorithms for the channel equalization are essentianlly different implementations of the SDM proposed in the past [7]. A serious drawback of the SDM is its slow convergence, especially when the condition number of the Hessian matrix of

The authors are grateful to Micronet, NCE Program, and the Natural Sciences and Engineering Research Council of Canada for supporting this work.

$f(\mathbf{w})$ is large. The Newton method along with Hessian matrix manipulation to ensure its positive definiteness solves the problem in (2) significantly faster at the cost of a considerable increase in computational complexity [6]. The main computational burden in the Newton method is the evaluation of the inverse of a possibly modified Hessian matrix of $f(\mathbf{w})$. Recursive least-squares algorithms are essentially adaptive implementations of the Newton method [7]. The class of quasi-Newton methods, which does not require the evaluation of the Hessian matrix and its inverse, offers a quadratic convergence rate with much reduced computational effort relative to that of the Newton method. Moreover, because the approximate inverse of the Hessian matrix is always positive definite, quasi-Newton algorithms are descent algorithms in that the objective function decreases monotonically as iteration continues. One of the most frequently used quasi-Newton algorithms is the Broyden-Fletcher-Goldfrab-Shanno (BFGS) algorithm [6] which is summarized below.

Table 1. BFGS algorithm

```
Step 1
  Input initial $\mathbf{w}_0$ and stopping tolerance $\epsilon$.
  Set $k = 0$ and $\mathbf{S}_0 = \mathbf{I}_n$.
  Compute $\mathbf{g}_k = \nabla f(\mathbf{w}_k)$.
Step 2
  Set $\mathbf{d}_k = -\mathbf{S}_k\mathbf{g}_k$ and find $\alpha_k$ that minimizes $f(\mathbf{w}_k + \alpha\mathbf{d}_k)$.
  Set $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha\mathbf{d}_k$.
Step 3
  If $\|\alpha_k\mathbf{d}_k\| < \epsilon$, output $\mathbf{w}^\star = \mathbf{w}_{k+1}$ and stop.
  Otherwise go to Step 4.
Step 4
  Compute $\mathbf{g}_{k+1}$ and set $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.
  Update matrix $\mathbf{S}_k$ using
      $\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{\delta_k\delta_k^T}{\delta_k^T\gamma_k} - \frac{\mathbf{S}_k\gamma_k\gamma_k^T\mathbf{S}_k}{\gamma_k^T\mathbf{S}_k\gamma_k}$    (3)
  Set $k = k + 1$ and repeat from Step 2.
```

From (2), it is clear that the optimized weight vector $\mathbf{w}$ depends on the data set $\{\mathbf{x}_k,\ k = 1, \ldots, N\}$. If we refer to this data block as *block l*, then the minimizer of the problem in (2) can be denoted as $\mathbf{w}_l^\star$. In the next section, we derive an explicit expression for $f(\mathbf{w})$ for a complex-valued input signal and weights, an efficient line search method, and an adaptive implementation of the equalizer that generates a good approximation of $\mathbf{w}_l^\star$ in real time.

4. NEW ADAPTIVE CM EQUALIZER

4.1. Data structure

For real-time channel equalization, the input data is processed block by block. A new data block for the next round of processing is generated by including a certain number of new input samples, say N_v while discarding N_v old samples. The data structure is illustrated in Fig. 2, where N_B is the size of one data block.

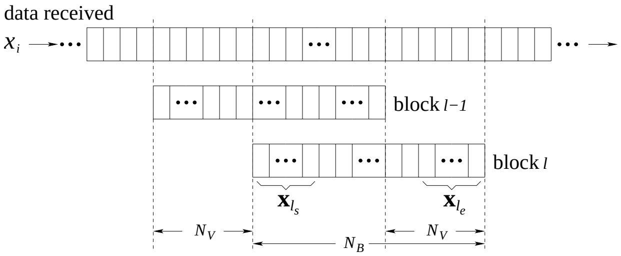

Fig. 2. Data structure of the adaptive CM equalization algorithm.

4.2. Objective function

When the lth block of data is processed, vector $\mathbf{x}_1$ in (2), which represents the first n samples of the data, assumes the form $[x_{lN_v+1} \ \cdots \ x_{lN_v+n}]^T$ and vector $\mathbf{x}_N$ in (2), which represents the last n samples of the data, is given by $[x_{lN_v+N_B-n+1} \ \cdots \ x_{lN_v+N_B}]^T$. Therefore, the objective function for the lth data block in (2) becomes

$$f_l(\mathbf{w}) = \sum_{i=l_s}^{l_e} \left(\mathbf{w}^H\mathbf{x}_i\mathbf{x}_i^H\mathbf{w} - 1\right)^2 \tag{4}$$

where $l_s = lN_v + n$ and $l_e = lN_v + N_B$. Since both $\mathbf{w}$ and $\mathbf{x}_i$ are complex-valued, in general, we need to write $\mathbf{w} = \mathbf{w}_r + j\mathbf{w}_j$ and $\mathbf{x}_i = \mathbf{x}_{r,i} + j\mathbf{x}_{j,i}$, and express (4) as

$$\begin{aligned} f_l(\hat{\mathbf{w}}) &= \sum_{i=l_s}^{l_e} \left[\left(\hat{\mathbf{w}}^T\hat{\mathbf{x}}_{1,i}\right)^2 + \left(\hat{\mathbf{w}}^T\hat{\mathbf{x}}_{2,i}\right)^2 - 1\right]^2 \\ &= \sum_{i=l_s}^{l_e} \left(A_i^2 + B_i^2 - 1\right)^2 \end{aligned} \tag{5}$$

where $A_i = \hat{\mathbf{w}}^T\hat{\mathbf{x}}_{1,i},\ B_i = \hat{\mathbf{w}}^T\hat{\mathbf{x}}_{2,i}$, and

$$\hat{\mathbf{w}} = \begin{bmatrix}\mathbf{w}_r \\ \mathbf{w}_j\end{bmatrix},\ \hat{\mathbf{x}}_{1,i} = \begin{bmatrix}\mathbf{x}_{r,i} \\ \mathbf{x}_{j,i}\end{bmatrix},\ \hat{\mathbf{x}}_{2,i} = \begin{bmatrix}-\mathbf{x}_{j,i} \\ \mathbf{x}_{r,i}\end{bmatrix}.$$

The gradient of $f_l(\hat{\mathbf{w}})$ is found to be

$$\mathbf{g}_l(\hat{\mathbf{w}}) = 4\sum_{k=l_s}^{l_e} \left(A_i^2 + B_i^2 - 1\right)\left(A_i\hat{\mathbf{x}}_{1,i} + B_i\hat{\mathbf{x}}_{2,i}\right). \tag{6}$$

Using the BFGS algorithm, the weight vector is updated to

$$\hat{\mathbf{w}}_{k+1} = \hat{\mathbf{w}}_k + \alpha_k\hat{\mathbf{d}}_k \tag{7a}$$

with

$$\hat{\mathbf{d}}_k = -\mathbf{S}_k\mathbf{g}_l(\hat{\mathbf{w}}_k). \tag{7b}$$

where $\hat{\mathbf{w}}_k$ denotes the current weight vector, matrix $\mathbf{S}_{k+1}$ is obtained using (3), and α_k is a positive scalar that minimizes $f_l(\hat{\mathbf{w}}_k + \alpha\hat{\mathbf{d}}_k)$. The process of finding α_k is known as *line search* and can in the present case be carried out accurately and efficiently as described next.

4.3. Line search

We can write $f_l(\hat{\mathbf{w}}_k + \alpha\hat{\mathbf{d}}_k)$ as a fourth-order polynomial of scalar variable α as

$$f_l(\hat{\mathbf{w}}_k + \alpha\hat{\mathbf{d}}_k) \ = \ c_4\alpha^4 + c_3\alpha^3 + c_2\alpha^2 + c_1\alpha + c_0 \tag{8}$$

where

$$
\begin{aligned}
c_4 &= \sum_{i=l_s}^{l_e} (C_i^2 + D_i^2)^2 \\
c_3 &= \sum_{i=l_s}^{l_e} 4(A_iC_i + B_iD_i)(C_i^2 + D_i^2) \\
c_2 &= \sum_{i=l_s}^{l_e} \left[2(C_i^2 + D_i^2)(A_i^2 + B_i^2 - 1) + 4(A_iC_i + B_iD_i)^2\right] \\
c_1 &= \sum_{i=l_s}^{l_e} 4(A_iC_i + B_iD_i)(A_i^2 + B_i^2 - 1) \\
c_0 &= \sum_{i=l_s}^{l_e} (A_i^2 + B_i^2 - 1)^2
\end{aligned}
$$

and $C_i = \hat{\mathbf{d}}_l^T \hat{\mathbf{x}}_{1,i}$, $D_i = \hat{\mathbf{d}}_l^T \hat{\mathbf{x}}_{2,i}$. The minimizer α_k must satisfy the condition $df_l(\hat{\mathbf{w}}_k + \alpha\hat{\mathbf{d}}_k)/d\alpha = 0$, i.e.,

$$4c_4\alpha^3 + 3c_3\alpha^2 + 2c_2\alpha + c_1 = 0. \qquad (9)$$

If there is only one real root for (9), then it is the minimizer α_k. Otherwise, there are three real roots for (9), two of which satisfy the second-order sufficiency condition

$$6c_4\alpha^2 + 3c_3\alpha + c_2 > 0. \qquad (10)$$

and we choose the one that achieves the absolute minimum value of the function $f_l(\hat{\mathbf{w}}_k + \alpha\hat{\mathbf{d}}_k)$ as α_k.

4.4. Adaptive implementation

A real-time implementation of the algorithm can be carried out as follows. At any given instant, a block of input samples of size N_B is used to construct vectors $\{\mathbf{x}_i,\ i = l_s, \ldots, l_e\}$ in (4) and a certain number of iterations of the BFGS algorithm are applied to minimize $f_l(\hat{\mathbf{w}})$ in (4) to obtain an improved weight vector $\hat{\mathbf{w}}_l^\star$. This approximate solution is utilized for channel equalization for the next N_v samples periods. The data set is then updated by including N_v new samples and excluding N_v old samples, and the BFGS algorithm is applied again. The initial weight vector $\hat{\mathbf{w}}_0$ and matrix $\mathbf{S}_0$ for a given data block l are the vector $\hat{\mathbf{w}}_{l-1}^\star$ and matrix $\mathbf{S}_k$ obtained in the previous iteration for the proceeding data block, respectively, except the case of $l = 0$ in that a reasonable $\hat{\mathbf{w}}_0$ and $\mathbf{S}_0 = \mathbf{I}_{2n}$ are utilized. The algorithm is summarized in Table 2.

Table 2. Blind CM equalization adaptation algorithm

```
Outer Loop:
Step $\mathcal{O}1$
  Set $l = 0$ and $\mathbf{S}_0 = \mathbf{I}_{2n}$.
  Input an initial $\hat{\mathbf{w}}_0$ and parameters $N_B$ and $N_v$.
Step $\mathcal{O}2$
  Form data block $l$.
  Inner Loop:
  Step $\mathcal{I}1$
    Set the maximum number of iterations to $k_{max}$.
    Set $k = 0$.
    Compute $\mathbf{g}_l(\hat{\mathbf{w}}_k)$ using (6).
  Step $\mathcal{I}2$
    Compute $\hat{\mathbf{d}}_k$ using (7b).
    Find $\alpha_k$ using the method in Sec. 4.3.
    Update $\hat{\mathbf{w}}_k$ to $\hat{\mathbf{w}}_{k+1}$ using (7a).
  Step $\mathcal{I}3$
    Compute $\mathbf{g}_l(\hat{\mathbf{w}}_{k+1})$ and update $\mathbf{S}_k$ to $\mathbf{S}_{k+1}$ using (3).
  Step $\mathcal{I}4$
    If $k + 1 = k_{max}$,
    then set $\hat{\mathbf{w}}_l^\star = \hat{\mathbf{w}}_{k+1}$ and go to Step $\mathcal{O}3$.
    Otherwise, go to Step $\mathcal{I}5$.
  Step $\mathcal{I}5$
    Set $k = k + 1$ and go to Step $\mathcal{I}2$.
Step $\mathcal{O}3$
  Set $\hat{\mathbf{w}}_0 = \hat{\mathbf{w}}_l^\star$, $\mathbf{S}_0 = \mathbf{S}_{k+1}$.
  Set $l = l + 1$ and repeat from Step $\mathcal{O}2$.
```

5. SIMULATIONS

The proposed equalization algorithm was applied to the baseband communication system shown in Fig. 1 and its performance was evaluated and compared with that of the LMS algorithm proposed in [1] and the RCMA algorithm proposed in [5]. A commonly used performance measure for equalization algorithms is the residual ISI which is defined as

$$ISI = 10\log_{10}\frac{\|\mathbf{f}\|_2^2 - \|\mathbf{f}\|_\infty^2}{\|\mathbf{f}\|_\infty^2}$$

where $\mathbf{f} = \mathbf{h} * \mathbf{w}$ denotes the convolution of the channel impulse response and the weight vector of the equalizer, and $\|\mathbf{f}\|_2$ and $\|\mathbf{f}\|_\infty$ represent the 2-norm and infinity-norm of vector $\mathbf{f}$, respectively. In all simulations, $k_{max} = 1$ was used.

Example 1: Each two consecutive input bits were mapped into a four-quadrature amplitude-modulation (4-QAM) symbol and then the modulation symbols were transmitted through a channel with the channel impulse response given by $\mathbf{h} = [0.815\ 0.419\ 0.419]^T$ [5]. Additive white Guassian noise (AWGN) was added at the output of the channel and the signal-to-noise ratio (SNR) of the received signal was set to 20 dB. At the receiver, a baud-spaced equalizer was implemented. For all algorithms, the length of the equalizer n was chosen to be 9 and the equalizer was initialized with $\mathbf{w}_0 = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$. For the proposed algorithm, N_B and N_v were chosen to be 100 and 20, respectively. The step size was set to 5×10^{-3} for the LMS algorithm, and the forgetting factor was set to 0.99 for the RCMA algorithm. The performance of the equalization algorithms was evaluated and averaged over 50 trials. The residual ISI of the proposed algorithm versus the sample index is plotted in Fig. 3 as the solid curve while those of the algorithms [1] and [5] are plotted as the dashed and dot-dashed curves, respectively. It is observed that the proposed algorithm reaches convergence at -16 dB residual ISI within 200 samples, whereas the algorithms in [1] and [5] require 1300 and 600 samples, respectively, to achieve the same level of residual ISI. The amount of computation per sample required by the proposed algorithm is approximately 25% less than that required by the RCMA algorithm in [5].

Example 2: This example is concerned with a telephone channel with channel impulse response vector $\mathbf{h} = [0.04\ -0.05\ 0.07\ -0.21\ -0.5\ 0.72\ 0.36\ 0.21\ 0.03\ 0.07]^T$ [8]. Using the same settings as in Example 1, the proposed algorithm and the algorithms

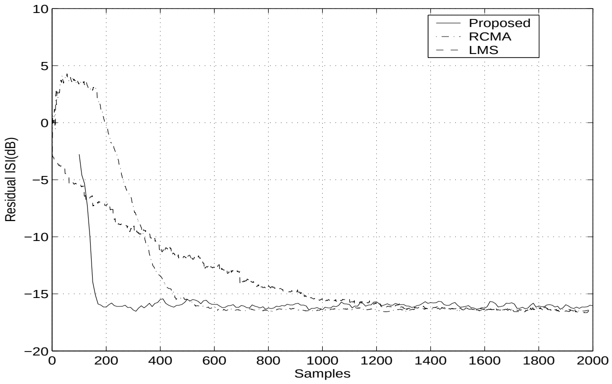

Fig. 3. Performance comparison of various equalization algorithms.

in [1][5] were applied to equalize the channel. The performance of the equalization algorithms was evaluated and averaged over 50 trials. The residual ISI of the proposed algorithm versus the sample index is plotted in Fig. 4 as a solid curve while those of the algorithms [1] and [5] are plotted as dashed and dot-dashed curves, respectively. It is observed that the proposed algorithm reaches convergence at -20 dB residual ISI within 200 samples, whereas the algorithms in [1] and [5] require 600 and 500 samples, respectively, to achieve the same level of performance. The constellations of the input and output signals of the equalizer are plotted in Fig. 5(a) and 5(b), respectively. As can be observed, the constellation of the input signal is fairly disturbed which would lead to a poor bit-error-rate. The application of the proposed algorithm led to a discernible constellation for the output signal.

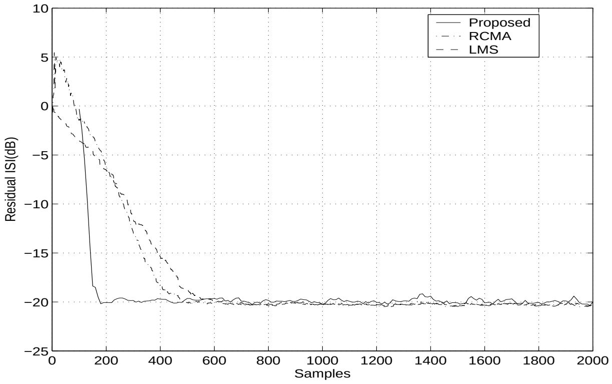

Fig. 4. Performance comparison of various equalization algorithms.

6. CONCLUSIONS

An efficient CM adaptation algorithm for blind channel equalization has been proposed. The algorithm is based on a quasi-Newton optimization method where the computation intensive update of the inverse of the Hessian matrix is carried out using the rank-two BFGS formula. As a result, the proposed adaptation algorithm offers fast convergence rate and reduced computational complexity relative to those of some existing algorithms.

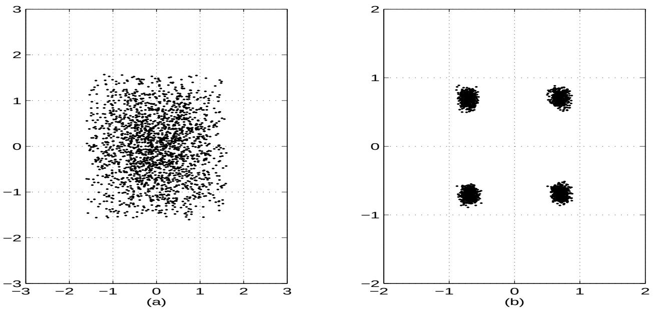

Fig. 5. (a) Constellation of the received signal before equalization; (b) Constellation of the signal after equalization.

7. REFERENCES

[1] J. R. Treichler and M. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 459-472, Apr. 1983.

[2] J. R. Treichler, V. Wolff, and C. R. Johnson, Jr., "Observed misconvergence in the constant modulus adaptive algorithm," *Proc. of 25rd Asilomar Conference on Signals, Systems and Computers*, pp. 663-667, Nov. 1991.

[3] C. R. Johnson, Jr., P. Schniter, T. J. Endres, J. D. Behm, and R. A. Casas, "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE*, vol. 86, no. 10, pp. 1927-1950, Oct. 1998.

[4] I. Santamaria, C. Pantaleon, L. Vielva, and J. C. Principe, "A fast algorithm for adaptive blind equalization using order-α Renyi's entropy," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol.3, p.p. 2657-2660, May 2002.

[5] R. Pickholtz and K. Elbarbary, "The recursive constant modulus algorithm; a new approach for real-time array processing," *Proc. of 27rd Asilomar Conference on Signals, Systems and Computers*, pp. 627-632, Nov. 1993.

[6] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1987.

[7] S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice-Hall, 2000.

[8] J. Proakis, *Digital Communications*, 4th ed., McGraw-Hill, 2000.