A COMPARISON OF THREE ALGORITHMS FOR BLIND EQUALIZATION BASED ON THE CONSTANT MODULUS ERROR CRITERION

T. Schirtzinger, X. Li, and W. K. Jenkins

Department of Electrical and Computer Engineering
Coordinated Science Laboratory
University of Illinois
1308 West Main Street
Urbana, IL 61801 U.S.A.

ABSTRACT

Three constant modulus algorithms (CMA), the fast quasi-Newton CMA, the transform domain CMA, and the genetic search based CMA are proposed in this paper. The performances of these three algorithms are compared with each other via computer simulation. It is shown that the fast quasi-Newton CMA and the transform domain CMA achieve much faster convergence rate than the constant modulus algorithm based on the LMS algorithm. This fact shows that the whitening technique is not only useful but also necessary for the CMA.

1. INTRODUCTION

Adaptive equalizers are widely used in communication systems to remove distortion such as intersymbol interference and multipath reflection introduced through transmission. Conventional adaptive algorithms use a training sequence to adapt the equalizer. The training signal is known to the receiver and transmitted during the start up period. However, in many cases, the use of a training signal is costly and unrealistic. Blind equalization schemes do not require the use of a training signal but instead adapt the equalizer based on restoring a known property of the transmitted signal. The constant modulus algorithm (CMA) [3] is one of the blind equalization algorithms [2] which attempts to restore the modulus of certain communication signals (eg. BPSK, QPSK).

The cost function of the blind equalization algorithms is:

$$\hat{J} = E||y(n)|^p - |\delta|^p|^q, \tag{1}$$

where $y(n)$ is the equalizer output, δ is the expected modulus, and p and q are positive integers. The general blind equalization algorithms for arbitrary p's and q's were introduced in [2]. A special case for $p = 2$ and $q = 2$, which is the best known constant modulus algorithm, was introduced in [3]. The steepest decent search algorithm is usually used to update the filter coefficients for the CMA:

$$W(n+1) = W(n) - \mu\hat{\nabla}(n), \tag{2}$$

where $\hat{\nabla}(n)$ is the instantaneous estimate of the gradient of $\hat{J}$ w.r.t. $W(n)$ and equals:

$$\hat{\nabla}(n) = [|y(n)|^2 - |\delta|^2]y(n)X(n). \tag{3}$$

$W(n)$ is the equalizer weight vector, and $X(n)$ is the input vector of the equalizer. The actual updating equation for the steepest decent CMA can be written as:

$$W(n+1) = W(n) - \mu[|y(n)|^2 - |\delta|^2]y(n)X(n). \tag{4}$$

Two main disadvantages of the constant modulus algorithm are its multimodal error surface and its slow convergence rate. Much work has been done in developing more powerful blind equalization algorithms to overcome the first problem. Some work has also been done in developing fast CMA algorithms. In [4] the lattice gradient search based CMA was introduced. The algorithm converges faster but suffers from a high noise floor [7]. A block frequency domain CMA was introduced in [5], but the algorithm only aims at reducing the computational complexity. In this paper, three different algorithms, the frequency domain CMA, the fast quasi-Newton CMA, and the genetic search based CMA are introduced. A comparative study of these three algorithms suggests that the quasi-Newton CMA results in the best performance.

This work was supported in part by NSF under grant number MIP 91-00212, and in part by the Joint Services Electronics Program under grant number N00014-90-J-1270.

0-7803-2431-5/95 $4.00 © 1995 IEEE

2. THE ALGORITHMS

2.1. The Fast Quasi-Newton CMA

In 1986 the orthogonalized constant modulus algorithm (O-CMA) was introduced [6]. The algorithm updates the equalizer weights by the following equation:

$$W(n+1) = W(n) - \mu[|y(n)|^2 - |\delta|^2]y(n)\mathbf{R}_x^{-1}(n)X(n), \tag{5}$$

where $\mathbf{R}_x(n)$ is the estimate of the autocorrelation matrix of the input data vector $X(n)$. The algorithm uses the matrix inversion lamma to update the inverse of the matrix $\mathbf{R}_x(n)$ and has $O(N^2)$ computational complexity. Note that the Newton algorithm uses an estimate of the Hessian matrix to minimize the cost function. As pointed out in [6], the O-CMA is not a Newton method because $\mathbf{R}_x(n)$ is not equal to the Hessian matrix for the constant modulus error criterion. However, $\mathbf{R}_x^{-1}(n)$ works very well as a whitening transform and can increase the convergence speed of the CMA. Here, we use the same update equation as (5). But instead of using the matrix inversion lamma, the fast quasi-Newton algorithm [8] is used to efficiently update the vector $\mathbf{R}_x^{-1}(n)X(n)$. We will call this algorithm the fast quasi-Newton(FQN) CMA. The FQN algorithm achieves $O(N)$ computational complexity by performing only one matrix inversion for every N input samples. The efficient calculation of $\mathbf{R}_x^{-1}(n)X(n)$ is achieved by using the Levinson-Durbin recursion. Assuming that the input statistics are slowly varying, the estimate of the inverse of the covariance matrix $\mathbf{R}_x^{-1}(k)$ at $k = n$ can be used to calculate the Kalman gains at time steps $k = n+1 \ldots n+N-1$. The following step size μ is used in the fast quasi-Newton CMA algorithm, where δ is a small number relative to the average value of $\mathbf{X}(n)^T\mathbf{R}_x^{-1}(n)X(n)$:

$$\mu = \frac{1}{2[\mathbf{X}(n)\mathbf{R}_x(n)\mathbf{X}(n) + \delta]}. \tag{6}$$

This formulation of step size has been used in other quasi-Newton adaptive algorithms and seems to be nearly optimal [8]. The summary of the fast quasi-Newton CMA algorithm is listed in Table 1 [7].

Definitions:	$\mathbf{A} = diag[1,\ \alpha^{1/2},\ \alpha,\ \ldots,\ \alpha^{M-1/2}]$
	$R(n) = [r_0(n),\ r_1(n), \ldots, r_{M-1}(n)]$
	$\mathbf{R}_x(n) = Toeplitz(R(n))$
Recursion:	$y(n) = \mathbf{W}^T(n)\mathbf{X}(n)$
	$e(n) = d(n) - y(n)$
	$\mu = \frac{1}{2[\mathbf{X}(n)\mathbf{R}_x(n)\mathbf{X}(n)+\delta]}$
	$R(n) = \alpha R(n-1) + x(n)\mathbf{A}\mathbf{X}(n)$
	$\mathbf{S}^{-1} = \mathbf{R}_x^{-1}(n)$,
	for n an integer multiple of N
	Update $\mathbf{S}^{-1}\mathbf{X}(n)$ using FQN algorithm
	$\mathbf{W}(n+1) = W(n) + \mu e(n)\mathbf{S}^{-1}\mathbf{X}(n)$

Table 1: Summary of the FQN-CMA

2.2. The Transform Domain CMA

Inspired by the O-CMA which uses the matrix $\mathbf{R}_x^{-1}(n)$ to orthogonalize the data vector $X(n)$ and achieves fast convergence rate, we introduce the transform domain constant modulus algorithm. In linear adaptive filtering, the fixed orthogonal transforms(eg. DCT and DFT) are used to orthogonalize the inputs to the equalizer. Then, estimates of the power in each channel are made (σ_i^2) and the step size for each channel is formed by dividing the nominal step size (μ) by the power estimate. This has the effects of whitening the input signal, decreasing the eigenvalues spread, and increasing the convergence rate of the algorithm. Because the convergence speed of the CMA is highly related to the conditioning of the input data, a highly colored input signal will produce a slow convergence rate for the CMA. If weightening techniques are used to the input signal, fast convergence can be achieved. As in the linear adaptive filtering, the orthogonal transforms and power normalization are used to whiten the input signal of the constant modulus equalizer. Following is the summary of the transform domain CMA.

$$W(n+1) = W(n) - \mu[|y(n)|^2 - |\delta|^2]y(n)\Lambda^{-2}\hat{X}(n) \tag{7}$$

$$\hat{X}(n) = \mathbf{T}X(n) \tag{8}$$

$$y(n) = W^T(n)\hat{X}(n) \tag{9}$$

$$\Lambda^2 = diag[\sigma_0^2\sigma_1^2 \cdots \sigma_{N-1}^2] + \delta\mathbf{I}, \tag{10}$$

Where,

$$\sigma_i^2(n) = \alpha\sigma_i^2(n-1) + (1-\alpha)|\hat{x}_i(n)|^2, \tag{11}$$

and $\mathbf{T}$ is any linear orthogonal transform matrix. The block diagram of the transform domain filter is shown in Figure 1.

2.3. The Genetic Search Based CMA

The Genetic Algorithm is an optimization technique which emulates the mechanism of natural selection and genetics [9], including structured and randomized information exchange among a population of parameters represented by strings. The ultimate result of the operation is the survival of the fittest parameter set over the problem domain. Because the algorithm does not use the gradient information, it has the ability to achieve global optimality on error surfaces that are not unimodal.

The basic idea of the genetic algorithm is that it works on a population of parameters of the system during the evolution. Each parameter is represented by or coded into fixed length string. The representation of parameters by fixed length strings should cover all possible solutions of the problem domain or give very good approximation to this problem domain. In most of the situations the binary representation is used to code each parameter. An initial population set is first randomly generated. A fitness value is given to each of the solutions among the population represented by fixed length string. Each solution is then assigned a probability measure based on the fitness value which decides the contribution that the solution would make to the next population set. Two basic operations, crossover and mutation of the genetic algorithm, are used to generate offsprings for the next population set. In the crossover, the substrings of two strings randomly selected according to their probability measure from the current population are exchanged with each other, resulting in two new strings. The principle of the crossover operation is to generate a new set of parameters or to reproduce offsprings by using crossover and mutation. The mutation operation occasionally change the value at a particular string position with a probability called the probability of mutation. Its purpose is to insure the diversity of the population during the course of the searching. The ultimate result of the genetic search operations is the survival of the fittest parameter set over the problem domain. Because the algorithm does not use the gradient information, it has the ability to achieve global optimality on error surfaces that are not unimodal. The details of the algorithm can be found in [9]

Some work has been reported on using genetic algorithms in adaptive filtering. But no direct comparison between the genetic algorithms and the classical adaptive filtering algorithms has been reported. In this paper, we apply the genetic algorithm into the constant modulus algorithm and compare its performance with other algorithms. In the algorithm, a population size of $M = 40$ is used. Each equalizer coefficient is encoded into 8-bit binary code. The following performance measure:

$$E_i = \frac{1}{N}\sum_{k=1}^{N}[|y_i(k)|^2 - |\delta|^2|^2], \qquad (12)$$

is used as fitness function. Here N is the block size, over which the mean square errors between the modulus of the equalizer output and the desired modulus are calculated, $y_i(k)$ is the output of the equalizer, and i is the index of the $i-th$ parameter set. The genetic search algorithm is used to generate new sets of parameters at each iteration. The search stops when the optimum parameters are achieved.

3. SIMULATION RESULTS AND DISCUSSIONS

Computer simulations have been used to compare the performance of the three newly introduced constant modulus algorithms and the steepest decent based constant modulus algorithm. The CMA algorithms are used to equalize the distorted signals to compensate the intersymbol interference. The DCT is used for the transform domain CMA. Here the signal set $x(n) \in \{+1, -1\}$ is used. A nonminimum phase channel model taken from an example in [1]:

$$h(n) = \frac{1}{2}[1 + cos(2\pi(n-2)/W] \qquad (13)$$

for $n = 1, 2, 3$, and $h(n) = 0$ for n otherwise, is used here. The simulations were performed for $W = 3.5$ and $W = 3.8$ which correspond to the eigenvalue spread $X(R) = 46.821$ and $X(R) = 192.90$ and are denoted as channel I and II, respectively. A 11-tap FIR filter is used as equalizer. Figures 2 and 3 show the learning curves for the four constant modulus algorithms for the two channel models. From the results we can see that both the fast quasi-Newton CMA and the DCT-CMA have much faster convergence speeds than the steepest decent CMA and the genetic search based CMA. This is not surprising because both the algorithms use whitening techniques to improve the input conditioning. Of all the algorithms the fast quasi-Newton CMA achieves the fastest convergence rate. The genetic search based CMA does not give the satisfactory performance due to the drawback of the inability of achieving high precision representation of the parameters. It is also more computationally complex than the transform domain CMA and the fast quasi-Newton CMA.

4. CONCLUSION

In this paper the three new constant modulus algorithms, the transform domain CMA, the fast quasi-Newton CMA, and the genetic search based CMA are introduced. The computer simulations show that the first two algorithms improve the convergence speed dramatically. This shows that the whitening techniques are not only useful but also necessary for the CMA. The genetic search based CMA converges fast but it also has high noise floor due to its inability of achieving high precision representation of filter parameters. But one important feature of the genetic search based algorithm is that it is able to identify the global optimal, so it can be used first to find the initial values for any other blind equalization algorithms.

5. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory.* Prentice Hall, Englewood Cliffs, NJ, 1991

[2] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Transactions on Communications,* COM-28, pp. 1867-1875, 1980.

[3] J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-28, pp. 349-472, 1983.

[4] R. Gooch, M. Ready, and J. Svoboda, "A lattice-based constant modulus algorithm," *Twentieth Asilomar Conference on Circuits, Systems, and Computers,* pp. 282-286, 1986.

[5] J. J. Shynk, C. K. Chan, and M. R. Petraglia, "Frequency domain implementations of the constant modulus algorithms," *Twenty-Third Asilomar Conference on Signals, Systems, and Computers,* vol. 2, pp. 663-669, 1989.

[6] R. Gooch and J. Lundell, "The CM Array: An adaptive beamformer for constant modulus signals," *International Conference on Acoustics, Speech, and Signal Processing,* vol. 4, pp. 2523-2526, 1986.

[7] T. R. Schirtzinger, *"Designing Adaptive Equalizers Based on the Constant Modulus Error Criterion,"* M.S.E.E. Thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1994.

[8] D. F. Marshall and W. K. Jenkins, "A fast quasi-Newton adaptive filtering algorithm," *IEEE Transactions on Signal Processing,* vol. SP-40, pp. 1652-1662, 1992.

[9] D. H. Goldberg, *Genetic Algorithms In Search, Optimization and Machine Learning.* AddisonWesley Publishing Company, 1989.

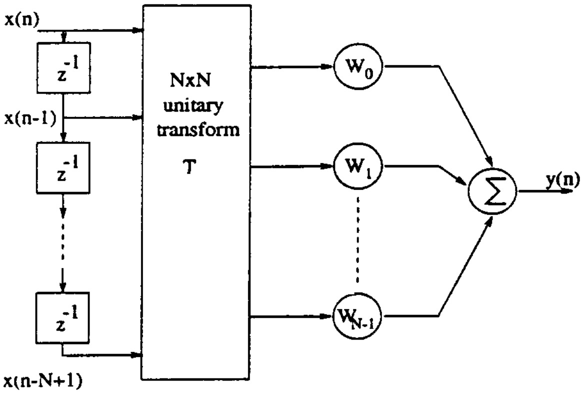

Figure 1: The transform domain filter.

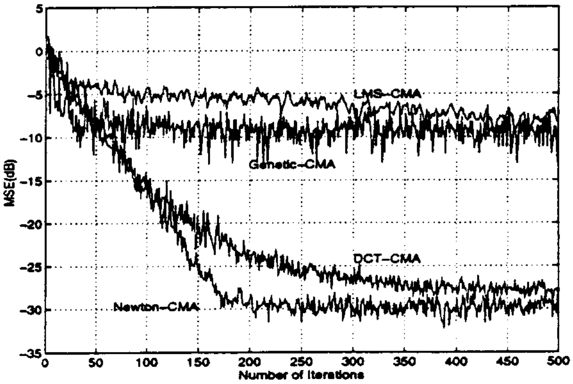

Figure 2: Learning Curves of the four constant modulus algorithms for channel I.

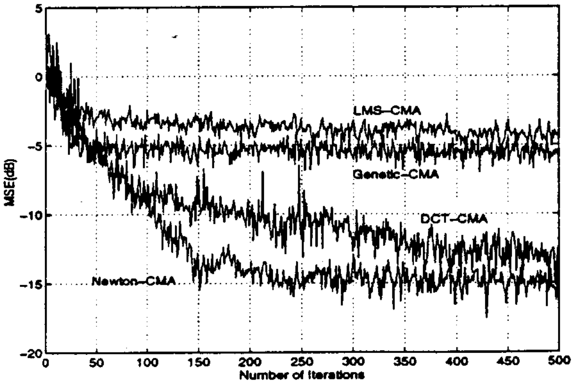

Figure 3: Learning Curves of the four constant modulus algorithms for channel II.