Novel Variable Step Size Blind Adaptive Algorithms for Smart Antenna Applications

ZHIMIN DU, PENG WAN, LU GAO, WEILING WU
School of Information Engineering
Beijing University of Posts and Telecommunications
Xitucheng Road 10#, Beijing 100876
P. R. CHINA

Abstract: Novel blind adaptive algorithms, two Variable Step Size Constant Modulus Algorithms (VSS-CMAs) and two Variable Step Size Constrained Minimum Output Energy (VSS-CMOE) algorithms, are proposed in this paper for smart antenna applications. In our algorithms there is no need to worry about whether the step size works, since they automatically and adaptively choose the best step size which minimizes the cost function along the gradient descent direction. Simulation results show that our new algorithms obviously outperform conventional methods both in convergence rate and steady-state performance.

Key Words: Variable Step Size Gradient Descent Method, CMA, Constrained CMA, CMOE, Smart Antenna

1 Introduction

Facing the soaring demand for mobile communications, the use of smart antenna arrays to exploit “the last frontier”—spatial resource has recently attracted considerable attention. The objective of adopting smart antenna system at a base station is to enhance the communication capacity and communication quality as well as the cell coverage.

Blind adaptive algorithms initially for equalization have been studied for more than twenty years. Constant Modulus Algorithm (CMA) [1], as a special case of Godard algorithm, was widely studied in the past years. Its application into smart antenna applications was proposed in [2][3] etc. It is shown that there are many undesired stationary points when using CMA. Therefore, dedicated initialization is needed to guarantee proper convergence. Additionally, the choice of appropriate step size for stochastic gradient type CMA is quite difficult. In this paper we try to solve these two problems by using variable step size gradient descent method.

In Section 2, signal model used in this work is described. Section 3 derives new variable step size CMA and CMOE algorithms. In Section 4 various simulations are done to compare their performance. Finally, Section 5 summarizes the conclusions of this paper.

2 Signal Model

For the sake of presenting convenience, here we use a quite simple signal model which doesn’t take the delay spread into account. And we only concern spatial domain processing. In fact delay spread always exists in wireless channel. A space-time equalizer structure then can be used to combat it. By rearranging the data vector and increasing the dimension of input vector and weight vector, one can easily extend our new algorithms into that scenario.

Consider a system with one desired user and K-1 co-channel interfering users. The base station uses an array of M sensors to receive up-link signals. The complex envelope representation of the M×1 input data vector x(t) can be modeled as:

\[ x(t) = a_k \sqrt{P_t} h_k(t) + \sum_{k=2}^{K} a_k \sqrt{P_t} h_k(t - \tau_k) + n(t) \quad (1) \]

where \( h_k(t) \), \( P_t \) and \( \tau_k \) are the transmitted symbol, power and time delay of the \( k \)th user, \( a_k \) is the normalized steering vector of user \( k \). Its matrix form after symbol-rate sampling then is

\[ x = A \sqrt{P} b + n \quad (2) \]

where \( A = [a_1, \ldots, a_K] \) is a \( M \times K \) matrix whose columns are users’ steering vectors with unit
energy, $\sqrt{P} = \text{diag}(\sqrt{P_1}, ..., \sqrt{P_K})$ is the signal amplitude matrix, $b = [b_1 \ldots b_K]^T$ is the data symbol vector, and $\mathbf{n}$ is Gaussian noise vector with covariance $E(\mathbf{n} \mathbf{n}^H) = \sigma^2 \mathbf{I}$. Herein and thereafter, lowercase with boldface denotes column vector. Capital letter with boldface denotes matrix. Superscript $\ast$, $T$, $H$ denote complex conjugate, transpose and complex conjugate transpose, respectively.

3 Derivation of New Algorithms

3.1 VSS-CMA

As we know the cost function of (2,2) CMA can be expressed as

$$J(\mathbf{w}) = E\left( \left\| \mathbf{w}^H \mathbf{x} \right\|^2 - \xi^2 \right)$$

(3)

, where $\mathbf{w}$ is the weight vector of smart antenna, $\xi$ is a parameter. The conventional stochastic gradient algorithm then is:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - u \mathbf{w}^H(n) \mathbf{x} (\mathbf{w}^H(n) \mathbf{x} - \xi^2) \mathbf{x} \mathbf{w}(n) \mathbf{x}$$

(4)

, where $u$ is a positive constant. In Variable Step Size-CMA (VSS-CMA), we try to use a relatively optional step size rather than a constant one. By substituting equation (4) into (3), differentiating the cost function with respect to $u$, setting it to 0 and solving the new equation, we get three solutions:

$$u_1 = \frac{1}{\left\| \mathbf{x} \right\|^2} \left( \left\| \mathbf{x} \right\|^2 - \xi^2 \right)$$

$$u_2 = \frac{1}{\left\| \mathbf{x} \right\|^2} \left( \left\| \mathbf{x} \right\|^2 + \xi^2 \right)$$

$$u_3 = \frac{1}{\left\| \mathbf{x} \right\|^2} \left( \left\| \mathbf{x} \right\|^2 - \xi^2 \right)$$

(5)

(6)

(7)

, where $y = \mathbf{w}^H(n) \mathbf{x}$. Since

$$\frac{\partial^2 J}{\partial u_i^2} \bigg|_{u_i = u_{a_1}} = 8 \xi^2 \left\| \mathbf{x} \right\|^4 \left( \left\| \mathbf{x} \right\|^2 - \xi^2 \right)^2 \left\| \mathbf{x} \right\|^2 \geq 0$$

(8)

and $u_2$ is always positive, we use it as the step size. So in VSS-CMA the weight vector is updated according to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{\mathbf{x} \mathbf{x}^H} \left( \left\| \mathbf{x} \right\|^2 - \xi^2 \right) \mathbf{y} \mathbf{y}^H \mathbf{x}$$

(10)

3.2 VSS-CCMA

To escape from the ill conditioned initialization, new constrained item can be added when we know desired user’s steering vector. That is the Constrained Constant Modulus Algorithm (CCMA) [4]. The new cost function is:

$$J(\mathbf{w}) = E \left( \left\| \mathbf{w}^H \mathbf{x} \right\|^2 - \xi^2 \right)^2$$

subject to $\mathbf{w}^H \mathbf{a}_i = 1$ (11)

According to the constrained LMS algorithm, the weight updating equation can be:

$$\mathbf{w}(n+1) = \mathbf{P}_a \mathbf{w}(n) - u \left( \left\| \mathbf{y} \right\|^2 - \xi^2 \right) \mathbf{y} \mathbf{P}_a^+ \mathbf{x} + \mathbf{a}_i$$

(12)

, where $\mathbf{P}_a = \mathbf{I} - \mathbf{a}_i \mathbf{a}_i^H$. By using the same method we can get the three solutions $u$ as:

$$u_1 = \frac{1}{\left\| \mathbf{y} \right\|^2 - \xi^2 \left\| \mathbf{y} \right\|^2 \mathbf{P}_a^+ \mathbf{x}$$

$$u_2 = \frac{1}{\left\| \mathbf{y} \right\|^2 + \xi^2 \left\| \mathbf{y} \right\|^2 \mathbf{P}_a^+ \mathbf{x}$$

$$u_3 = \frac{1}{\left\| \mathbf{y} \right\|^2 - \xi^2 \left\| \mathbf{y} \right\|^2 \mathbf{P}_a^+ \mathbf{x}$$

(13)

(14)

(15)

Since

$$\frac{\partial^2 J}{\partial u_1^2} \bigg|_{u_1 = u_{a_1}} = 8 \xi^2 \left\| \mathbf{y} \right\|^4 \left( \left\| \mathbf{y} \right\|^2 - \xi^2 \right)^2 \left\| \mathbf{y} \right\|^2 \left\| \mathbf{x} \right\|^2 \geq 0$$

(16)

$$\frac{\partial^2 J}{\partial u_2^2} \bigg|_{u_2 = u_{a_1}} = -4 \xi^2 \left\| \mathbf{y} \right\|^4 \left( \left\| \mathbf{y} \right\|^2 - \xi^2 \right)^2 \left\| \mathbf{y} \right\|^2 \left\| \mathbf{x} \right\|^4 \leq 0$$

(17)

and $u_1$ is always positive, so in VSS-CCMA we update $\mathbf{w}$ according to:

$$\mathbf{w}(n+1) = \mathbf{P}_a \mathbf{w}(n) - \frac{1}{\left\| \mathbf{y} \right\|^2 \mathbf{P}_a^+ \mathbf{x} \mathbf{y} \mathbf{P}_a^+ \mathbf{x} + \mathbf{a}_i$$

(18)

3.3 VSS-CMOE1

As we will see in section 4, though VSS-CCMA is robust to incorrect initialization it is sensitive to the constant $\xi$. When the power of desired user can’t be estimated correctly it may obviously degrade the performance. Constrained Minimum Output Energy (CMOE) algorithm [5] on the other hand doesn’t depend on the estimated constant. It minimizes the following cost function

$$J(\mathbf{w}) = \mathbf{w}^H E(\mathbf{x} \mathbf{x}^H) \mathbf{w}, \text{ subject to } \mathbf{w}^H \mathbf{a}_i = 1.$$ (19)

The corresponding stochastic gradient algorithm is:

$$\mathbf{w}(n+1) = \mathbf{P}_a \mathbf{w}(n) - u \mathbf{y} \mathbf{P}_a^+ \mathbf{x} + \mathbf{a}_i$$

(20)

By using the same method we can determine the best $u$ in this case as

$$u = \frac{1}{\mathbf{x} \mathbf{x}^H} \mathbf{y} \mathbf{y}^H \mathbf{x}$$

(21)

So in VSS-CMOE1 the updating equation is

$$\mathbf{w}(n+1) = \mathbf{P}_a \mathbf{w}(n) - \frac{1}{\mathbf{x} \mathbf{x}^H} \mathbf{y} \mathbf{y}^H \mathbf{x} \mathbf{a}_i$$

(22)
3.4 VSS-CMOE2

To improve the steady-state performance of VSS-CMOE1 we construct a RLS type cost function as

\[ J(w(n)) = \sum_{i=1}^{n} x^H(i)x^{H}(i)w(n), \text{ subject to } w^a_i = 1 \]

The stochastic gradient algorithm then is

\[ w(n+1) = P_n^a w(n) - u P_n^a \sum_{i=1}^{n} x^H(i)x^H(i)w(n) + a_i \]  \hspace{1cm} (23)

Substituting equation (23) into the cost function, differentiating it with respect to \( u \), setting the result to 0 and solving the new equation, we get the best \( u \) as:

\[ u = \frac{w^H(n)R(n)P_n^aR(n)w(n)}{w^H(n)R(n)P_n^aR(n)P_n^aR(n)w(n)} \]  \hspace{1cm} (24)

, where \( R(n) = \sum_{i=1}^{n} x^H(i)x^H(i) \). That is the new algorithm VSS-CMOE2.

4 Simulation Results

A 4-element uniform linear array with \( \lambda/2 \) spacing is deployed in all the simulations. There are totally 5 users: the desired user’s signal reaches the array at \( 0^\circ \) while interfering users’ DOAs are \( -60^\circ, -45^\circ, 30^\circ \) and \( 65^\circ \), respectively. The SNR is set to be \( P_i/\sigma^2 = 20dB \). The power of the received signals are set to be \( P_i = 1 \) and \( P_i/P_i = SIR(dB), i = 2, \ldots, 5 \). The modulation scheme is BPSK for all users.

![Fig.1 Ensemble-average SINR gain curves for different algorithms with correct initialization](image1.png)

Fig.1 Ensemble-average SINR gain curves for different algorithms with correct initialization

Firstly, we compare the convergence behavior of different algorithms with correct initialization.

Figure 1 plots the transient behavior of them when \( w \) is initialized with desired user’s steering vector \( a_j \) and \( \xi = -20dB \). Here the SINR gain is calculated by averaging 100 independent random runs’ output SINR with respect to the spatial matched filter’s output (\( w = a_i \)). It’s shown from fig. 1 that in case of correct initialization all algorithms converge to the desired user, but our variable step size algorithms converge faster than the stochastic gradient algorithms though they use almost the biggest available \( u \). Another observation is that there is only slight difference between the VSS-CMA and VSS-CCMA with correct initialization.

Secondly, we study the effect of incorrect initialization. In this case \( w \) is initialized with user 2’s steering vector \( a_j \). That is figure 2. Obviously VSS-CCMA, VSS-CMOE1 and VSS-CMOE2 still converge to the desired user while VSS-CMA converge to the undesired user (user 2). Experiments with random initialization show that VSS-CCMA, VSS-CMOE1 and VSS-CMOE2 always converge to the desired user even with very low SIR. Another observation is that different initialization has almost no effect on steady-state performance of VSS-CCMA, VSS-CMOE1 and VSS-CMOE2.

![Fig.2 Ensemble-average SINR gain curves for different algorithms with incorrect initialization](image2.png)

Fig.2 Ensemble-average SINR gain curves for different algorithms with incorrect initialization

In VSS-CCMA the optimal constant \( \xi \) should equal to \( \sqrt{P_i} \). This requires the accurate estimation of desired user’s power which is difficult in real application. So in figure 3 we study the effect of different \( \xi \) to VSS-CCMA, where \( w \) is initialized with \( a_2 \) and \( \xi = -20dB \). It’s shown that the value of \( \xi \) has slight effect on convergence rate but obvious effect on steady-state performance. Smaller and bigger value both degrade the steady-state performance.
performance, but a bigger value (in the same ratio) degrades the performance more seriously. Compare equation (18) and (22) we can see that VSS-CMOE1 is a special case of VSS-CCMA with $\xi = 0$. It explains in one way why VSS-CCMA always outperforms VSS-CMOE1.

![Graph](image)

Fig.3 Effect of the CONSTANT in VSS-CCMA

Figure 4 compares the steady-state performance of VSS-CMOE2 and VSS-CCMA with different input SIR, where the measurement of steady-state performance difference is calculated by subtracting the SINR gain of VSS-CCMA from that of VSS-CMOE2. It shows that when input SIR is low VSS-CMOE2 outperforms VSS-CCMA. But when input SIR increases VSS-CCMA has a much bigger SINR gain.

![Graph](image)

Fig.4 Steady-state performance comparison between VSS-CMOE2 and VSS-CCMA

5 Conclusions

In this paper, we developed four new adaptive algorithms for blind array processing. All of them use the variable step size gradient descent method to adaptively choose the relatively optimal step size. It was shown from simulation that our new algorithms converge much faster than the conventional fixed step size algorithms. VSS-CCMA, VSS-CMOE1 and VSS-CMOE2 can guarantee convergence to the desired user with random initialization due to the additional constraint item. VSS-CCMA has the best performance when input SIR is not very low, but it is sensitive to the estimated constant $\xi$. When input SIR decreases, VSS-CMOE2 can outperform VSS-CCMA.

References: