TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.2, February 2014, pp. 1022 ~ 1028
DOI: http://dx.doi.org/10.11591/telkomnika.v12i2.4216

Research on Orthogonal Direction Algorithm using Direction Error

Jun-fang Guo*[1], Yong-feng Zhi[2]
[1]Wenzhou University, Zhejiang 325035, China
[2]Northwestern Polytechnical University, Xi'an 710072, China
*Corresponding author, e-mail: gjf@wzu.edu.cn

Abstract

For the orthogonal direction algorithm, the iteration direction is the direction vector, and the iteration error of the adaptive filter is caused by the input vector. These two directions are not the same one, which leads to more deviation for the iteration error. In this paper, an orthogonal direction algorithm using direction error is presented to settle this problem, which the iteration error is caused by the iteration direction. By setting the weight error to be zero in the direction vector, the optimal step-size is obtained, which improves the adaptive filtering convergence rate.

Keyword*: affine projection, iteration direction, adaptive filtering, system identification.*

1. Introduction

The adaptive filtering techniques are widely used in noise and echo cancellation, equalization, and beam forming. A very popular algorithm is the normalized least mean square (NLMS) algorithm [1], which is computationally very simple. But, for highly colored input signals, this algorithm suffers from slow convergence. Over the past three decades, computationally efficient, rapidly converging adaptive filtering algorithms have been proposed to ameliorate this problem. For example, from the geometric viewpoint of affine subspace projections, the affine projection (AP) algorithm was discovered [2]. The orthogonal direction (OD) was proposed based on the definition of the direction vector [3]. The NLMS with orthogonal correction factors (NLMS-OCF) was proposed based on the idea that the best improvement in weights occurs if successive input vectors are orthogonal to each other [4]. But for these algorithms, the iteration direction and the direction that causes the iteration error are not the same one, which reduce the adaptive filtering convergence rate.

In order to improve the adaptive filtering convergence, a developed AP algorithm was introduced based on the concept of reuse time of the current input vector [5]. In [6], an AP algorithm with exponential smoothing factor was shown using a variable step-size. An AP algorithm with the variable regularization factor was proposed in [7]. An AP algorithm that automatically determined its projection order was presented using the evolutionary method [8]. The affine projection using regressive estimated error was introduced in [9, 10] by redefining the iteration error of the AP algorithm, and its convergence behavior was also analyzed.

The iteration error of the OD algorithm is caused by the input vector, however this algorithm applies updates to the weights in the direction vector, and these two directions are not the same. In order to solve the problem, by analyzing the iteration error in the iteration direction of the adaptive filtering, an orthogonal direction using direction error (OD-DE) algorithm is proposed in this paper. Under a measurement noise free condition, the iteration error is directly caused by the direction vector which is also the iteration direction of the adaptive filter. Also, the optimal step-size is obtained by setting the weight error to be zero in the direction vector, which improves the adaptive filtering convergence performance.

2. OD Algorithm

Figure 1 show an adaptive filter used in the system identification mode. The wide sense stationary input process x_n, which is zero-mean, and the measured output d_n, possibly

Received July 25, 2013; Revised September 5, 2013; Accepted October 4, 2013

contaminated with the measurement noise e_n, is measurable. The measurement noise is zero mean white noise. The input process is converted into input vectors $\mathbf{x}_n$, via a tapped delay line, and are defined as:

$$\mathbf{x}_n = \begin{bmatrix} x_n & x_{n-1} & \cdots & x_{n-N+1} \end{bmatrix}^T \tag{1}$$

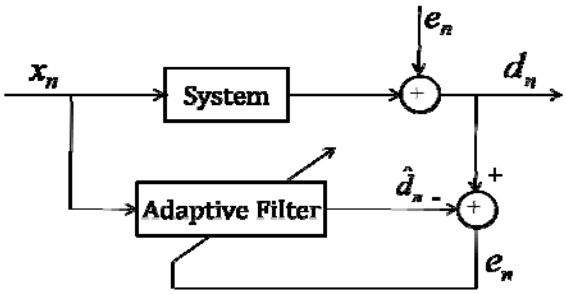

Figure 1. Adaptive System Identification Scenario

The objective is to estimate an N-dimensional weight vector $\mathbf{w}_n$ using the most recent input vectors available at the nth instant. The weight vector is adjusted so that the estimated output $\hat{d}_n$ is as close as possible to the measured output d_n in the mean-square error (MSE) sense. And the output error is defined as:

$$e_n = d_n - \hat{d}_n \tag{2}$$

Where,

$$\hat{d}_n = \mathbf{w}_n^T \mathbf{x}_n \tag{3}$$

The adaptive filter implements the OD algorithm updates of the weight vector [3], as follows:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \frac{\boldsymbol{\varphi}_n}{\boldsymbol{\varphi}_n^T \boldsymbol{\varphi}_n} e_n \tag{4}$$

Where μ is the step-size, and it is used to control the adaptive filtering performance. $\boldsymbol{\varphi}_n$ is the direction vector, which is defined as:

$$\boldsymbol{\varphi}_n = \mathbf{x}_n - \hat{\mathbf{Z}}_{n-1} \hat{\mathbf{b}}_n \tag{5}$$

and the vector $\hat{\mathbf{b}}_n$ is given as:

$$\hat{\mathbf{b}}_n = \left[\hat{\mathbf{Z}}_{n-1}^T \hat{\mathbf{Z}}_{n-1} \right]^{-1} \hat{\mathbf{Z}}_{n-1}^T \mathbf{x}_n \tag{6}$$

The input matrix $\mathbf{X}_n$ is defined as follows:

$$\hat{\mathbf{Z}}_{n-1} = [\boldsymbol{\varphi}_{n-1} \quad \boldsymbol{\varphi}_{n-2} \quad \cdots \quad \boldsymbol{\varphi}_{n-m}] \tag{7}$$

And the initial values is given as $\hat{\mathbf{Z}}_0 = [\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}]$.

In order to analyze the OD algorithm, supposes that there exists a true adaptive filter weight vector $\mathbf{w}^0$ of dimension N such that the corresponding measured output d_n can be written as:

$$d_n = \mathbf{w}^{0T}\mathbf{x}_n + e_n \tag{8}$$

Based on (2), (3) and (8), the corresponding output error signal can be rewritten as:

$$e_n = \tilde{\mathbf{w}}_n^T\mathbf{x}_n + e_n \tag{9}$$

Where $\tilde{\mathbf{w}}_n$ is the weight error at the *n*th instant, and it is defined as:

$$\tilde{\mathbf{w}}_n = \mathbf{w}^0 - \mathbf{w}_n \tag{10}$$

From (9), it can be concluded that, at the *n*th instant, the iteration error is caused by the input vector $\mathbf{x}_n$ for the OD algorithm. But from (4), we find that the iteration direction is the direction vector $\boldsymbol{\varphi}_n$, these two directions are not the same one, which leads to more deviation for the iteration error. In order to improve the adaptive filtering convergence, an OD-DE algorithm will give in the next section. Under the measurement noise free condition, the iteration direction and the direction that causes the iteration error are both the direction vector, i.e. these two direction are the same one.

3. OD-DE Algorithm

Combining (4) and (10), the adaptation equation in error form for the OD algorithm is obtained as:

$$\tilde{\mathbf{w}}_{n+1} = \tilde{\mathbf{w}}_n - m\frac{\boldsymbol{\varphi}_n}{\boldsymbol{\varphi}_n^T\boldsymbol{\varphi}_n}e_n \tag{11}$$

Pre-multiplication of (11) by $\boldsymbol{\varphi}_n^T$ and $\hat{\mathbf{Z}}_{n-1}^T$ respectively, using the results $\boldsymbol{\varphi}_n^T\hat{\mathbf{Z}}_{n-1} = 0$ [3], we have:

$$\boldsymbol{\varphi}_n^T\tilde{\mathbf{w}}_{n+1} = \boldsymbol{\varphi}_n^T\tilde{\mathbf{w}}_n - me_n \tag{12a}$$

$$\hat{\mathbf{Z}}_{n-1}^T\tilde{\mathbf{w}}_{n+1} = \hat{\mathbf{Z}}_{n-1}^T\tilde{\mathbf{w}}_n \tag{12b}$$

From (7) and (12b), we have:

$$\begin{aligned}
\boldsymbol{\varphi}_{n-d}^T\tilde{\mathbf{w}}_{n+1} &= \boldsymbol{\varphi}_{n-d}^T\tilde{\mathbf{w}}_n;\ 1 \le d \le m \\
\boldsymbol{\varphi}_{n-1-t}^T\tilde{\mathbf{w}}_n &= \boldsymbol{\varphi}_{n-1-t}^T\tilde{\mathbf{w}}_{n-1};\ 1 \le t \le m-1 \\
\boldsymbol{\varphi}_{n-2-k}^T\tilde{\mathbf{w}}_{n-1} &= \boldsymbol{\varphi}_{n-2-k}^T\tilde{\mathbf{w}}_{n-2};\ 1 \le k \le m-2 \\
&\vdots \\
\boldsymbol{\varphi}_{n-m-g+1}^T\tilde{\mathbf{w}}_{n-m+2} &= \boldsymbol{\varphi}_{n-m-g+1}^T\tilde{\mathbf{w}}_{n-m+1};\ g = 1
\end{aligned} \tag{13}$$

Based on (13), we obtain:

$$\begin{bmatrix}\boldsymbol{\varphi}_{n-1}^T & \boldsymbol{\varphi}_{n-2}^T & \cdots & \boldsymbol{\varphi}_{n-m}^T\end{bmatrix}\tilde{\mathbf{w}}_n = \begin{bmatrix}\boldsymbol{\varphi}_{n-1}^T\tilde{\mathbf{w}}_n & \boldsymbol{\varphi}_{n-2}^T\tilde{\mathbf{w}}_{n-1} & \cdots & \boldsymbol{\varphi}_{n-m}^T\tilde{\mathbf{w}}_{n-m+1}\end{bmatrix} \tag{14}$$

Using (12a) into (14), yields:

$$\begin{aligned}&\begin{bmatrix}\boldsymbol{\varphi}_{n-1}^T & \boldsymbol{\varphi}_{n-2}^T & \cdots & \boldsymbol{\varphi}_{n-m}^T\end{bmatrix}\tilde{\mathbf{w}}_n = \\ &\quad \begin{bmatrix}\boldsymbol{\varphi}_{n-1}^T\tilde{\mathbf{w}}_{n-1} - \mu e_{n-1} & \boldsymbol{\varphi}_{n-2}^T\tilde{\mathbf{w}}_{n-2} - \mu e_{n-2} & \cdots & \boldsymbol{\varphi}_{n-m}^T\tilde{\mathbf{w}}_{n-m} - \mu e_{n-m}\end{bmatrix}\end{aligned} \tag{15}$$

In order to improve the adaptive filtering performance, the iteration error should cause by the direction that updates the weight. Thus, the iteration error is redefined as:

$$\hat{e}_n = \tilde{\mathbf{w}}_n^T\boldsymbol{\varphi}_n \tag{16}$$

Based on (16), (15) can be rewritten as:

$$\begin{aligned}&\begin{bmatrix}\boldsymbol{\varphi}_{n-1}^T & \boldsymbol{\varphi}_{n-2}^T & \cdots & \boldsymbol{\varphi}_{n-m}^T\end{bmatrix}\tilde{\mathbf{w}}_n = \\ &\quad \begin{bmatrix}\hat{e}_{n-1} - \mu e_{n-1} & \hat{e}_{n-2} - \mu e_{n-2} & \cdots & \hat{e}_{n-m} - \mu e_{n-m}\end{bmatrix}\end{aligned} \tag{17}$$

From (5), (7) and (16), we have:

$$\hat{e}_n = \tilde{\mathbf{w}}_n^T\mathbf{x}_n - \begin{bmatrix}\tilde{\mathbf{w}}_n^T\boldsymbol{\varphi}_{n-1} & \tilde{\mathbf{w}}_n^T\boldsymbol{\varphi}_{n-2} & \cdots & \tilde{\mathbf{w}}_n^T\boldsymbol{\varphi}_{n-m}\end{bmatrix}\hat{\boldsymbol{\beta}}_n \tag{18}$$

Using (17) into (18), yields:

$$\hat{e}_n = \tilde{\mathbf{w}}_n^T\mathbf{x}_n - \begin{bmatrix}\hat{e}_{n-1} - \mu e_{n-1} & \hat{e}_{n-2} - \mu e_{n-2} & \cdots & \hat{e}_{n-m} - \mu e_{n-m}\end{bmatrix}\hat{\boldsymbol{\beta}}_n \tag{19}$$

But in the actual system, the measurement noise needs to be considered. Thus, from (9), the iteration error of the adaptive filtering is approximate as:

$$\hat{e}_n = e_n - \begin{bmatrix}\hat{e}_{n-1} - \mu e_{n-1} & \hat{e}_{n-2} - \mu e_{n-2} & \cdots & \hat{e}_{n-m} - \mu e_{n-m}\end{bmatrix}\hat{\boldsymbol{\beta}}_n \tag{20}$$

So the new update for the OD-DE algorithm is given by:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\frac{\boldsymbol{\varphi}_n}{\boldsymbol{\varphi}_n^T\boldsymbol{\varphi}_n}\hat{e}_n \tag{21}$$

Thus, (2), (5), (6), (7), (20) and (21), in the specified order, constitute the OD-DE algorithm. From (16) and (21), we find that, under the measurement noise free condition, the iteration error $\hat{e}_n$ is caused by the direction vector, which is also the weight update direction for the OD-DE algorithm. Thus, by redefining the iteration error, the OD-DE algorithm should improve the adaptive filtering convergence performance.

Since the past $(m+1)$ direction vectors are orthogonal each other, the inverse of the matrix $\hat{\mathbf{Z}}_{n-1}^T\hat{\mathbf{Z}}_{n-1}$ can be rewritten as [3]:

$$\begin{bmatrix}\hat{\mathbf{Z}}_{n-1}^T\hat{\mathbf{Z}}_{n-1}\end{bmatrix}^{-1} = diag\begin{bmatrix}\dfrac{1}{\boldsymbol{\varphi}_{n-1}^T\boldsymbol{\varphi}_{n-1}} & \dfrac{1}{\boldsymbol{\varphi}_{n-2}^T\boldsymbol{\varphi}_{n-2}} & \cdots & \dfrac{1}{\boldsymbol{\varphi}_{n-m}^T\boldsymbol{\varphi}_{n-m}}\end{bmatrix} \tag{22}$$

So based on (22), the OD-DE algorithm avoids computing the inverse of the matrix, and improves the computation accuracy.

4. Optimal Step-size

In order to obtain the optimal step-size for the OD-DE algorithm, we use the variable step-size μ_n to replace the fixed step-size μ, and we have:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_n \frac{\boldsymbol{\varphi}_n}{\boldsymbol{\varphi}_n^T \boldsymbol{\varphi}_n} \hat{e}_n \tag{23}$$

Suppose at $(n+1)\mathrm{th}$ instant, the optimal step-size μ_n set the weight error to be zero at the direction vector. Thus, from (12a), the optimal step-size μ_n is obtained as:

$$\mu_{n,opt} = \frac{\boldsymbol{\varphi}_n^T \tilde{\mathbf{w}}_n}{e_n} \tag{24}$$

Based on the same reasoning process from (12) to (20), we have:

$$\begin{aligned} \hat{e}_n &= \boldsymbol{\varphi}_n^T \tilde{\mathbf{w}}_n \\ &= e_n - \left[\hat{e}_{n-1} - \mu_{n-1,opt} e_{n-1} \quad \hat{e}_{n-2} - \mu_{n-2,opt} e_{n-2} \quad \cdots \quad \hat{e}_{n-m} - \mu_{n-m,opt} e_{n-m} \right] \hat{\mathbf{b}}_n \end{aligned} \tag{25}$$

From (24) and (25), the optimal step-size can be obtained as:

$$\mu_{n,opt} = 1 - \frac{\left[\hat{e}_{n-1} - \mu_{n-1,opt} e_{n-1} \quad \hat{e}_{n-2} - \mu_{n-2,opt} e_{n-2} \quad \cdots \quad \hat{e}_{n-m} - \mu_{n-m,opt} e_{n-m} \right] \hat{\mathbf{b}}_n}{e_n} \tag{26}$$

Thus, (2), (5), (6), (7), (23), (25) and (26), in the specified order, constitute the OD-DE algorithm with the optimal step-size.

5. Simulation Results

In this section, we show the MSE learning curves from simulations with the optimal step-size OD-DE algorithm. The initial estimate for the weights is $\mathbf{w}_0 = 0$. The system to be identified has a 32-point long impulse with a signal-to-noise ratio 60dB. The simulation results shown are obtained by ensemble averaging over 100 independent trials for each experiment. The initial value for the optimal step-size $\mu_{0,opt}$ is 0.2. The true adaptive filter weight $\mathbf{w}^0$ is chosen to be the maximum entropy vector, i.e. equal weight is given to each of the eigen-directions:

$$\mathbf{w}^0 = \mathbf{1} \tag{27}$$

where $\mathbf{1}$ is a vector of all 1's.

Case 1: Consider an ARMA (1, 2) input signal given by $y_n = z_n + 0.5 z_{n-1} - 0.75 y_{n-1}$ with z_n zero-mean white Gaussian noise. We use the parameter $m = 2$. Figure 2 gives the learning curves of the MSE for the optimal step-size OD-DE algorithm. We observe that the proposed algorithm improves the convergence rate.

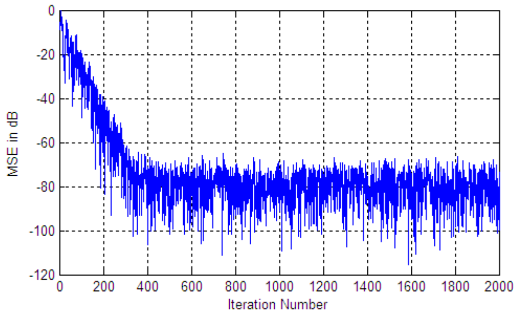

Figure 2. Learning Curves for the Proposed Algorithm with ARMA (1, 2) Input

Case 2: The input is an ARMA (1, 3) signal given by $y_n = 0.95z_n - 0.72z_{n-1} + 0.5z_{n-3} - 0.95y_{n-1}$. We use the parameter $m = 3$. The MSE behavior predicted by the optimal step-size OD-DE algorithm is shown in Figure 3. It can be conclude that the proposed algorithm improves the adaptive filtering performance.

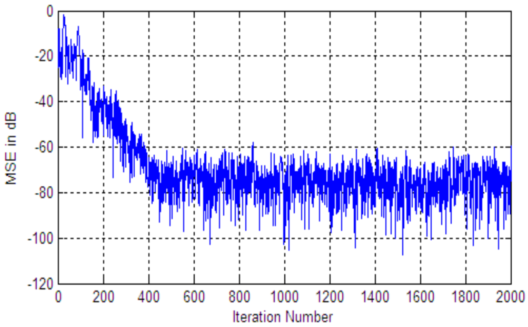

Figure 3. Learning Curves for the Proposed Algorithm with ARMA (1, 3) Input

6. Conclusion

The iteration direction of the OD algorithm is the direction vector, but the iteration error is caused by the input vector, these two directions are not the same one. In this paper, the OD-DE algorithm is presented to solve this problem. Under the measurement noise free condition, the iteration direction and the direction that causes the iteration error are both the direction vector $\boldsymbol{\varphi}_n$, these two directions are the same one. Also, by setting the weight error to be zero in the direction vector $\boldsymbol{\varphi}_n$, the optimal step-size is obtained for the OD-DE algorithm, which improves the adaptive filtering convergence performance.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of Zhejiang Province of China under Grant No. LY12F01016

References

[1] Haykin S. Adaptive Filter Theory, Fourth Edition. Englewood Cliffs, NJ: Prentice-Hall. 2002.
[2] Ozeki K, Umeda T. An Adaptive Filtering Algorithm Using an Orthogonal Projection to an Affine Subspace and Its Properties. *Electronics and Communication in Japan*. 1984; 67-A(5): 19–27.
[3] Rupp M. A Family of Adaptive Filter Algorithms with Decorrelating Properties. *IEEE Transactions on Signal Processing.* 1998; 46(3): 771–775.

[4] Sankaran SG, Louis Beex AA. Fast Generalized Affine Projection Algorithm. *International Journal of Adaptive Control and Signal Processing*. 2000; 14(6): 623–641.
[5] Chang MS, Kong NW, Park PG. An Affine Projection Algorithm Based on Reuse Time of Input Vectors. *IEEE Signal Processing Letters.* 2010; 17(8): 750–753.
[6] Fan Y, Zhang J. Variable Step-Size Affine Projection Algorithm with Exponential Smoothing Factors. *Electronics Letters*. 2009; 45(17): 911–912.
[7] Yin WT, Mehr AS. A Variable Regularization Method for Affine Projection Algorithm. *IEEE Transactions on Circuits and Systems.* 2010; 57(6): 476–480.
[8] Kim SE, Kong SJ, Song WJ. An Affine Projection Algorithm with Evolving Order. *IEEE Signal Processing Letters.* 2009; 16(11): 937–940.
[9] Zhang S, Zhi YF. Affine Projection Algorithm Using Regressive Estimated Error. *ISRN Signal Processing.* 2011; doi: 10.5402.
[10] Zhi YF, Li HX, Li R. Statistical Analysis of Affine Projection Using Regressive Estimated Error Algorithm. *Acta Automatica Sinica.* 2013; 39(3): 244-250.