Digital Signal Processing 74 (2018) 111–118

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

An improved variable tap-length algorithm with adaptive parameters

Yufei Han *, Mingjiang Wang, Ming Liu

School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen, HIT Campus of University Town of Shenzhen, 518055, Shenzhen, China

ARTICLE INFO

Article history:
Available online 23 December 2017

Keywords:
Variable tap-length algorithm
Convergence speed
Fluctuation
Fragment-full error
Variable error width

ABSTRACT

Convergence rate and anti-interference ability are significant rules to estimate the variable tap-length algorithm. Moreover conciseness and robustness are important essentials for these algorithms to be utilized in actual application. Following this way, the improved fractional variable tap-length adaptive algorithm that contains variable error width based on fragment-full error and selects the proportion of restricted function by threshold is presented. The proposed algorithm is able to increase the efficiency of the adaptation of error width. Besides, it has specific and reasonable criteria to select restricted factor automatically in different stage. It can improve convergence speed and constrain fluctuation at the same time. The parameters are able to modify reasonably on the basis of different situations. Simulations demonstrate that the proposed algorithm is the nearest to the optimal tap-length on the basis of maintaining good performance.

Introduction

Tap-length is a significant parameter which impacts performance and computational quantity of the adaptive algorithms. However, it is hard to determine the optimum value in traditional algorithm. The small tap-length may cause huge error [1,2]. Large tap-length may cause a great deal of calculation and excess mean square error as well.

To balance the performance and complexity, some scholars put forward variable tap-length algorithm [3–7]. The fractional tap-length least mean square (FT-LMS) algorithm [7] established a most favorable approach among these algorithms. Error width in FT algorithm acts as a significant part in regulating convergence rate and fluctuation of tap-length. Large error width is able to improve convergence rate while small error width decreases convergence rate. Meanwhile, adopting large error width will result in big fluctuation and small error width may lead to undermodeling problem [8]. Therefore some scholars put forward time-varying strategy of error width [8,9]. Nevertheless, these algorithms can not avoid the interference of system error or indicating the exact state of iteration at the same time. These problems will result in poor robustness and low convergence speed. Subsequently an improved method which can determine the error width according to the status of the adaptive filter is proposed.

Moreover, instantaneous value is vulnerable to the noise and noise level highly affects the variation of tap-length. Particularly in high noise circumstance, the fluctuation of tap-length is usually large [10,11]. It leads to decrease the performance. For reducing the unexpected influence of noise and improving robustness, a method that constrains amplitude of the cost function is adopted in FT algorithm [12]. Nevertheless the method of restricting amplitude may decrease the convergence rate at the beginning stage. A hybrid algorithm [13] is proposed to balance this problem. However, the algorithm lacks specific and reasonable criteria to select the scaling factor automatically in different stage. So, an improved method that selects the proportion of limited and unlimited value by comparing the output error with the estimated threshold error is also proposed.

FT algorithm

The partial error is defined as the error of the first M elements of the filter.

$$e_M^{(L)} = d(n) - W_{L,1:M}^T X_{L,1:M}(n) \tag{1}$$

$1 \le M \le L$, W_L and $X_L(n)$ are the filter coefficients and the input vectors in the algorithm. $W_{L,1:M}$ and $X_{L,1:M}$ represent the first M elements of W_L and $X_L(n)$. In FT algorithm, the cost function of tap-length is defined as the square of the fragment error $\xi_M^{(L)} = E\{(e_M^{(L)}(n))^2\}$. Then the algorithm needs to find the most suitable L to satisfy situation.

$$\xi_{L-\Delta}^{(L)} - \xi_L^{(L)} \le \varepsilon \tag{2}$$

* Corresponding author.
E-mail address: 160164242@qq.com (Y. Han).

https://doi.org/10.1016/j.dsp.2017.12.005

Δ is error width and a positive integer less than L. ε is a small positive integer required by the system. Then the minimum L satisfying (2) is chosen as the optimal filter tap-length.

Tap-length is defined as a FT $l_f(n)$ and the iterative procedure of FT is:

$$l_f(n+1) = \left(l_f(n) - \alpha\right) - \gamma * \left[\left(e_{L(n)}^{(L(n))}(n)\right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n)\right)^2\right] \tag{3}$$

α is a positive leakage parameter and its function is to avoid the problem that the iteration tap-length tends to be much larger than the optimal value. γ is step-size of tap-length. When the cumulative change of the fractional order exceeds a certain value, the tap-length updates as

$$L(n+1) = \begin{cases} \lfloor l_f(n) \rfloor & \text{if } |L(n) - l_f(n)| > \delta \\ L(n) & \text{others} \end{cases} \tag{4}$$

where $\lfloor \cdot \rfloor$ represents the nearest integer to $l_f(n)$. δ is a small integer threshold. The weight vector values in normalized least mean square (NLMS) FT algorithm are updated as follows:

$$W_{L(n)}(n+1) = W_{L(n)}(n) + \mu(n) \frac{e_{L(n)}^{(L(n))}(n) X_{L(n)}(n)}{\|X_{L(n)}\|^2 + \beta} \tag{5}$$

$W_{L(n)}$, $e_{L(n)}^{(L(n))}$ and $X_{L(n)}$ represent weight vector value, error signal and the input vector respectively. $\mu(n)$ is the step-size of the filter. β is regularization factor.

Proposed algorithm

The FT algorithm adopts a changeless error width that is obliged to a compromise between convergence rate and fluctuation of tap-length. FT algorithm does not have any adaptive parameters, such as error width. Fixed big error width may cause deviation of tap-length and huge calculation. However, small error width leads to small convergence speed of tap-length. In different experimental environments, the optimal choice of error width can not be guaranteed, which will reduce the performance. In other words, the algorithm should have corresponding error width in different phase.

For improving the performance of FT algorithm, there are already some time-varying strategies of error width so far. The algorithm in [8] adjusts error width with the energy of the instantaneous error as

$$\hat{e}^2(n) = \rho \hat{e}^2(n-1) + (1-\rho)\left(e_{L(n)}^{(L(n))}(n)\right)^2 \tag{6}$$

$$\Delta(n) = \min\left(\Delta_{\max}, \rho \hat{e}^2(n)\right) \tag{7}$$

where λ is a smoothing parameter. At the beginning stage, a large error width is used to make the tap-length update to the optimum as fast as possible. When the error decreases, a small error width is applied to avert big bias near the optimum. Nevertheless, since the instantaneous error is vulnerable to the noise, the energy of the instantaneous error does not reflect well the state of adaptation [9]. So the method in [8] can not perform as expected in the presence of noise.

For avoiding the interference of system error, the algorithm in [9] adopts a novel method as

$$\varepsilon(n) = \beta \varepsilon(n-1) + (1-\beta) e_{L(n)}^{(L(n))}(n) e_{L(n)-1}^{(L(n-1))}(n-1) \tag{8}$$

$$\Delta(n) = \min\left(\Delta_{\max}, C\left|\varepsilon(n)\right|\right) \tag{9}$$

where β is a positive constant. Adopting $\varepsilon(n)$ to calculate error width can eliminate the interference of system noise and improve the convergence performance. Whereas, the error width may be affected by the previous moment value $e(n-1)$. This method may not reflect the current changes accurately and has a certain passive impact on updating of tap-length in the initial stage. It will decrease the convergence speed of tap-length and improve the unexpected fluctuation of tap-length.

Above algorithms do not equip good robust and fast convergence speed at the same time. Thus, we propose a new method which alters error width based on fragment-full error (FE) to solve the contradiction between robustness and convergence speed. In other words, the proposed algorithm can equip adaptive parameter in different phase. Besides, comparing to the previous algorithms [7–9], this method mainly finds and solves some problems comprising overestimation of tap-length, robustness, adaptive ability of parameters and convergence performance. The proposed algorithm exhibits a favorable time-varying strategy of error width to improve performance.

The proposed algorithm in this paper is based on FT algorithm [7] and its improved algorithms [8,9]. We find FE appears in [7] firstly. Then we start to link FE with error width for theoretical analysis. FE can reflect the current iteration accurately and avoids interference of the last moment. Once tap-length is far away from the optimum, FE will be large according to big related error. When algorithm enters the steady state, FE will be small. So FE supplies an effective method to control Δ. The update equation of Δ is described as

$$FE(n) = \left(e_{L(n)}^{(L(n))}(n)\right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n)\right)^2 \tag{10}$$

$$t(n) = \lambda t(n-1) + (1-\lambda) FE(n) \tag{11}$$

$$\Delta(n) = \min\left(\Delta_{\max}, \rho\left|t(n)\right|\right) \tag{12}$$

where $t(0)$ is zero. λ is a smoothing parameter, which should be set close to 1. ρ is a factor and its value can be determined in next section.

At the beginning stage, FE is large and results in a large error width which is used to make the tap-length update to the optimum as fast as possible. When approaching the optimum, FE is small and makes a small error width which is applied to avert big fluctuation of tap-length. Δ will not be zero at last.

By using FE and intermediate variable t to calculate $\Delta(n)$, the proposed algorithm will increase the efficiency of the adaptation. Moreover, the proposed algorithm is easier to implement since it uses less parameter.

Algorithm analysis

Parameter analysis

Firstly, we will determine the value of ρ. It can be obtained according to the following equations.

From equation (12)

$$\Delta(n) = \rho\left|t(n)\right| \tag{13}$$

When $n \to \infty$

$$\Delta(\infty) = \rho\left|t(\infty)\right| \tag{14}$$

$$\Delta(\infty) = \rho\left|E\left\{e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right\}\right| \tag{15}$$

$l_f(\infty)$ can be expressed as follows:

$$l_f(\infty) = \left(l_f(\infty) - \alpha\right) - \gamma * E\left\{e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right\} \tag{16}$$

So, FE can be expressed as:

$$E\left\{e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right\} = -\alpha/\gamma \tag{17}$$