



Variable Tap-Length Algorithm with Mixed Parameter

Yufei Han , Mingjiang Wang * and Yun Lu 

Key Laboratory of Shenzhen Internet of Things Terminal Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China; hanyufei_hit@163.com (Y.H.); luyun@stu.hit.edu.cn (Y.L.)

* Correspondence: mjwang@hit.edu.cn

Received: 23 April 2018; Accepted: 4 June 2018; Published: 6 June 2018



Featured Application: The method can be applied to active noise control (ANC). As all actual environments are unknown, the tap-length considerably influences ANC. The proposed algorithm can determine the optimal tap-length to optimize computation time and avoid the under-modeling problem. The effect of noise reduction can be improved to a certain extent.

Abstract: Convergence speed and steady-state performance are important factors to estimate the variable tap-length algorithm. The step-size of tap-length (ST) and the error width considerably impact the performance of the fractional variable tap-length algorithm. A mixed fractional variable tap-length algorithm that simultaneously applies time-varying ST and error width strategies is presented in this paper. The ST and the error width are analyzed in detail. This method uses the same variable to control both aspects, so their changes are relatively consistent. The parameters have strong adaptive ability at different stages. The proposed algorithm can solve some defects of the previous algorithms, including slow convergence rate, weak steady-state performance and robustness. A novel mean square deviation (MSD) gradient experiment is performed. Experimental results show good agreement with the theoretical analysis.

Keywords: adaptive algorithm; variable tap-length; convergence speed; steady-state performance; error width; step-size of tap-length

1. Introduction

The adaptive algorithm involves considerable investigation into signal processing, being widely applied to diverse fields including active noise control, beamforming, and echo canceling [1,2]. The tap-length is a special parameter of the adaptive algorithm. It plays a decisive role in the performance of transient and steady-state. Too small a tap-length will cause divergence [3,4]. To avoid this situation, the tap-length is usually assigned a large value in general algorithms. Nevertheless, a large tap-length also results in a large amount of computation and poor steady-state performance. Thus, finding a method to adjust tap-length is necessary. Many researchers have proposed variable tap-length algorithms. Gu et al. [5] presented a gradient descent algorithm that adjusts tap-length according to the negative gradient direction of squared estimation error. A segmented filter algorithm [6] separated the entirety into some parts and the tap-length updated in one part based on the difference of the error in the previous part. Among these variable tap-length algorithms, the fractional variable tap-length (FT) algorithm [7] is the most favored strategy.

The step-size of tap-length (ST) is important to the performance of the FT algorithm. Large ST causes fluctuations and fast convergence speed. Conversely, small ST causes the under-modeling problem and good steady-state performance [8]. Generally, there is a tradeoff between stability and convergence speed. However, ST is usually fixed in the FT algorithm. The error width of the FT algorithm also influences convergence rate and steady-state performance. Large error width

improves convergence rate but results in poor steady-state performance. Small error width causes little fluctuation but decreases convergence performance [9]. Thus, many researchers proposed time-varying strategies [10,11]. Nevertheless, these algorithms cannot avoid the interference of system error or indicate the exact parameter at the same time. More importantly, they are only concerned with either ST or error width and ignore the effect of the other parameters. Coordination of these two parameters is lacking. These algorithms cannot provide the good performance.

Algorithms in [12,13] improved a certain performance compared with algorithms in [10,11]. However, they only concerned with either ST or error width and also ignored the effect of the other one. There is no reasonable analysis or adjustment strategy for the other parameter in algorithms [12,13]. Their performance may be affected by this. Thus, a mixed variable tap-length algorithm that uses the time-varying strategies of ST and error width is proposed in this paper. The proposed algorithm combines algorithms in [12,13] and balances them reasonably. ST and error width of the proposed algorithm all have good adaptive ability. The significant difference between the proposed algorithm and the algorithms in [12,13] is the performance in high noise environment. As ST and error width can meet the corresponding values at different stages, the proposed algorithm can perform well even in the high noise environment. By using a suitable and unified strategy to control ST and error width, the proposed algorithm improves convergence rate and steady-state performance. The accurate tap-length of the proposed algorithm makes the accurate estimate of the corresponding weight coefficients. Moreover, the proposed algorithm avoids interference of the system error and provides a good level of robustness. Some experiments were performed to compare the performance.

The rest of this paper is organized as follows: Section 2 introduces the least mean square (LMS) algorithm, the influence of tap-length and the FT-LMS algorithm. The proposed mixed variable tap-length algorithm and its performance analysis are introduced in Sections 3 and 4, respectively. The detailed simulation experiments and results are outlined in Section 5. The conclusions are summarized in Section 6.

2. Variable Tap-Length Algorithm

2.1. Least Mean Square Algorithm

The LMS algorithm is composed of an input signal \mathbf{X} , an error signal $e(n)$, an output signal $y(n)$ and a weight coefficient vector \mathbf{W} . The full equation is as follows:

$$y(n) = \mathbf{W}(n)^T \mathbf{X} \quad (1)$$

$$e(n) = (\mathbf{W}_{opt} - \mathbf{W}(n))^T \mathbf{X} + v(n) \quad (2)$$

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \mu e(n) \mathbf{X} \quad (3)$$

where \mathbf{W}_{opt} is the optimal weight. $v(n)$ is the noise signal, and μ is the step-size. When $\mu = \mu' = \frac{\mu}{\mathbf{X}^T \mathbf{X} + \beta}$, it becomes the normalized least mean square algorithm (NLMS). β is a small positive number multiplied by the unit matrix. The length of the weight coefficient vector \mathbf{W} is the tap-length. The lengths of the optimal weight \mathbf{W}_{opt} and actual weight \mathbf{W} are L_{opt} and L , respectively. When the tap-length L is much smaller than L_{opt} , the algorithm diverges. Similarly, when the tap-length L is much larger than L_{opt} , we can infer from Equation (2) that $e(n)$ increases as well. As the computational complexity of the LMS algorithm is proportional to the tap-length L , a large tap-length L also requires a large amount of computation. To maintain the same performance level, a smaller tap-length requires less computation. Therefore, the tap-length L is a tradeoff among stability, computation, and convergence speed.

2.2. Mean Square Deviation Gradient

Next, the mean square deviation (MSD) gradient is used to discuss the effect of different tap-lengths

on the algorithm. For a more intuitive comparison, we analyzed the MSD gradient first. Define \mathbf{p} as follows:

$$\mathbf{p}(n) = \mathbf{W}_{opt} - \mathbf{W}(n) \quad (4)$$

Taking \mathbf{W}_{opt} to subtract Equation (3), Equation (3) becomes as follows:

$$\mathbf{p}(n) = \mathbf{p}(n-1) - \mu e(n)\mathbf{X} \quad (5)$$

Using $t(n)$ to replace $E\{\|\mathbf{p}(n-1)\|^2\} - E\{\|\mathbf{p}(n)\|^2\}$, the equation becomes as follows:

$$t(n) = E\{\|\mathbf{p}(n-1)\|^2\} - E\{\|\mathbf{p}(n)\|^2\} = E\{\|\mathbf{p}(n-1)\|^2\} - E\{\|\mathbf{p}(n-1) - \mu\mathbf{X}e(n)\|^2\} \quad (6)$$

Equation (6) can be simplified as follows:

$$t(n) = 2\mu E\{e(n)^2\} - 2\mu\sigma_v^2 - M\mu^2\sigma_x^2 E\{e(n)^2\} \quad (7)$$

where σ_v^2 is the variance of the noise and σ_x^2 is the variance of the input. The MSD gradient is related to tap-length and noise signal according to Equation (7). Next, an experiment was used to compare the MSD gradient of different tap-lengths.

Figure 1 plots the MSD gradient of different tap-lengths (8, 12, 16, and 20) in an identification system. This system was set up with the best tap-length of 16. The reference input was obtained by passing white Gaussian noise through a transfer system of $H(z) = 0.35 + z^{-1} + z^{-2}$. Transfer system $H(z)$ was referenced from Yu and Cowan [7]. The length 16 of the identification system was adopted from the white Gaussian random sequence with zero-mean and unity variance. The signal-to-noise ratio (SNR) of 10 dB was obtained by adding white Gaussian interference noise. Different tap-lengths had different MSD gradient performance. In the first half of the iteration, the red line ($M = 16$) had the largest $t(n)$. In the latter half, although the black line ($M = 8$) had a fluctuating and negative MSD gradient, part of the black line was larger than the red line. This is because under-modeling appears in the black line. As its tap-length is smaller than the optimum value, the difference between its coefficient and the optimum is very large. Large and fluctuant MSD results in a large MSD gradient that is incorrect. Thus, the closer to optimum the tap-length is, the larger the amplitude of MSD gradient that will be obtained. For achieving the fastest convergence rate and the best tracking capability, the tap-length should be close to the optimum. In this case, MSD will undergo the largest decrease at each iteration and also guarantees the achievement of the lowest possible steady-state MSE [14]. Furthermore, a variable tap-length potentially gains significant computational savings. For an unknown environment, an efficient variable tap-length algorithm provides better results. The fractional variable tap-length algorithm is a favorable strategy, as explained below.

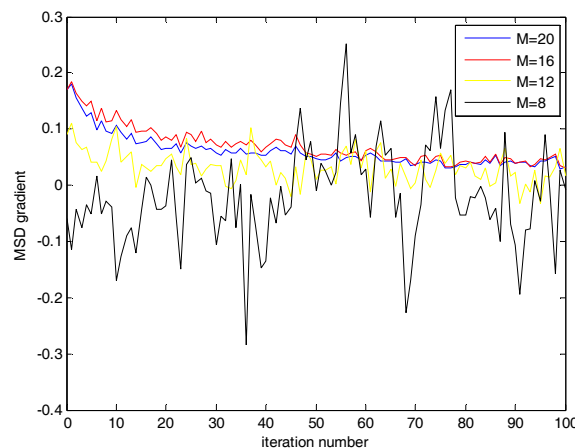


Figure 1. Mean square deviation (MSD) gradient of different tap-lengths.

2.3. Fractional Variable Tap-Length Algorithm

Fractional variable tap-length means that the tap-length varies in the form of fraction. The sectional error refers to M length in front of the filter:

$$e_M^{(L)} = d(n) - \mathbf{W}_{L,1:M}^T \mathbf{X}_{L,1:M} \quad (8)$$

where $1 \leq M \leq L$; \mathbf{W}_L and \mathbf{X}_L are the weight coefficients and the input in the algorithm, respectively. $\mathbf{W}_{L,1:M}$ and $\mathbf{X}_{L,1:M}$ represent the first M elements of the weight coefficients and the input, respectively. The cost function of the tap-length is the square of the fragment error $\zeta_M^{(L)} = E \left\{ \left(e_M^{(L)}(n) \right)^2 \right\}$. Then, the algorithm outputs the most compatible L to meet the requirements.

$$\zeta_{L-\Delta}^{(L)} - \zeta_L^{(L)} \leq \varepsilon \quad (9)$$

where Δ is error width. It is a positive integer less than L . ε is a small positive integer required by the system. The minimum L satisfying Equation (9) is chosen as the optimal filter tap-length.

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

$$L_f(n+1) = \left(L_f(n) - \alpha \right) - \gamma(n) * \left[\left(e_{L(n)}^{(L(n))}(n) \right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n) \right)^2 \right] \quad (10)$$

where α is a positive leakage parameter. Its function is to avoid the problem that the tap-length is much larger than the optimal value. $\gamma(n)$ is the step-size of tap-length (ST). It is fixed in FT algorithm. When the cumulative change of the fractional tap-length exceeds a certain value, the tap-length updates as:

$$L(n+1) = \begin{cases} \lfloor L_f(n) \rfloor & \text{if } |L(n) - L_f(n)| > \delta \\ L(n) & \text{others} \end{cases} \quad (11)$$

where $\lfloor \cdot \rfloor$ represents the nearest integer to $L_f(n)$, and δ is a small integer threshold. The weight vector values in the FT algorithm are updated as follows:

$$\mathbf{W}_{L(n)}(n+1) = \mathbf{W}_{L(n)}(n) + \mu' e_{L(n)}^{(L(n))}(n) \mathbf{X}_{L(n)} \quad (12)$$

where $\mathbf{W}_{L(n)}$, $e_{L(n)}^{(L(n))}$, and $\mathbf{X}_{L(n)}$ represent the weight vector, error, and the input vector, respectively. μ' is the normalized step-size. It can be calculated from $\frac{\mu}{\mathbf{X}_{L(n)}^T \mathbf{X}_{L(n)} + \beta}$; μ is step-size. β is the regularization factor. It is a small positive number multiplied by the unit matrix.

3. Mixed Parameter Fractional Variable Tap-Length Algorithm

Equation (10) is the updating equation of tap-length in the transient state. As α is a small positive leakage parameter, the core method to increase or decrease the tap-length is $\gamma \left[\left(e_{L(n)}^{(L(n))}(n) \right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n) \right)^2 \right]$. A large value of the core part is needed to quickly change the tap-length. Thus, large ST γ and error width Δ are required to increase the convergence speed of tap-length in the transient state. In the steady-state, the updating equation of tap-length can be expressed as follows [8]:

$$L_f(\infty) = L_{opt} + \Delta - \frac{\alpha}{\sigma_x^2 \sigma_c^2} - \frac{\Delta \mu \sigma_v^2}{(2 - \mu) L_{opt} \sigma_x^2 \sigma_c^2} \quad (13)$$

where σ_x^2, σ_v^2 are the variances of the input signal and noise signal, respectively. σ_c^2 is the variance of the tail elements of the optimal weight \mathbf{W}_{opt} . For the determined noise signal, a small ST and error

width are needed to eliminate the last three items in the steady-state. Thus, small ST and error width allow tap-length to avoid the large fluctuations in the steady-state.

From the above, we determined that ST and error width have similar trends. In the beginning, ST and error width should be large to increase convergence speed and avoid the under-modeling problem. In the steady-state, small ST and error width are needed to improve steady-state performance and avoid overestimation of tap-length. In other words, to improve performance, the FT algorithm should simultaneously have the same time-varying strategies of ST and error width. Therefore, a novel method which adjusts ST and error width based on the fragment-full error (FE) is proposed in this study. The proposed algorithm reasonably combines ST and error width so that they coordinate and cooperate with each other. The complete iteration formula is:

$$FE(n) = \left(e_{L(n)}^{(L(n))}(n) \right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n) \right)^2 \quad (14)$$

$$q_1(n) = \lambda q_1(n-1) + (1-\lambda)|FE(n)| \quad (15)$$

$$q_2(n) = \lambda q_2(n-1) + (1-\lambda)FE(n) \quad (16)$$

$$\gamma(n) = \rho_1 q_1(n) \quad (17)$$

$$\Delta(n) = \min(\Delta_{max}, \rho_2 |q_2(n)|) \quad (18)$$

where $q_1(0)$ and $q_2(0)$ are zero. λ is a smoothing parameter and should be close to 1. ρ_1 and ρ_2 are the factors and their values can be referenced in Section 4. As the changing trends of ST and error width are the same, we used the same variable FE to make their changes consistent. Full error and fragment error all contain the noise signal. Subtracting them to obtain FE can eliminate the influence of the noise signal. In other words, FE is noise-free. Thus, FE reflects the state accurately and avoids external interference. Once the tap-length is far from the optimum, FE will be large. In the steady state, FE is small. Thus, FE acts as an effective method to adjust ST and error width. In the initial stage, a large FE brings a large ST and error width to avoid the suboptimal value of tap-length and improve the convergence speed. In the steady-state, a small FE produces a corresponding ST and error width to avoid fluctuation and improve the steady-state performance. We also add the restricted function introduced by Han et al. [13] into the proposed algorithm.

$$\zeta(n) = \frac{C * \left(\left(e_{L(n)}^{(L(n))}(n) \right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n) \right)^2 \right)}{\left[1 + \left(\left(e_{L(n)}^{(L(n))}(n) \right)^2 - \left(e_{L(n)-\Delta}^{(L(n))}(n) \right)^2 \right)^2 \right]^{0.5}} \quad (19)$$

$$LFF(n) = \begin{cases} \lambda_1 * \zeta(n) + (1-\lambda_1) * FE(n) & \text{if } e(n) * e(n) < EST \\ \lambda_2 * \zeta(n) + (1-\lambda_2) * FE(n) & \text{else} \end{cases} \quad (20)$$

$$L_f(n+1) = (L_f(n) - \alpha) - \gamma(n) * LFF(n) \quad (21)$$

where EST is the estimated value of MSE in deficient and sufficient tap-length situations. Its value can be obtained from previous studies [15,16]. λ_1 and λ_2 are scalar mixing parameters and satisfy $0 \leq \lambda_1 < \lambda_2 \leq 1$.

The proposed algorithm can prevent large fluctuations which are caused by interference noise. This method stabilizes the change of the tap-length. Additionally, the restricted function allows the algorithm to adapt to the changing environments better. It also increases the robustness.

In summary, the proposed algorithm reasonably adjusts the values of ST and error width according to different states. Due to characteristics of the proposed algorithm, it can adapt to changing circumstances and return more appropriate ST and error width than other algorithms.

The proposed algorithm solves many problems experienced by previous algorithms [10–13], including steady-state performance, overestimation of tap-length, adaptive ability of parameters, and convergence performance.

4. Performance Analysis

4.1. Parameter Analysis

In this section, the analysis of the proposed algorithm and general guidelines for the parameters ρ_1 and ρ_2 are exhibited. Given the information provided by Zhang et al. [8] and Han et al. [13], we deduced the reasonable values of ρ_1 and ρ_2 . Based on the analysis provided by Zhang et al. [8], the low bound and the up bound of γ are as follows:

$$\gamma_l = \frac{\alpha}{(\Delta - k\delta)\sigma_x^2\sigma_c^2 - \frac{\Delta\mu'\sigma_v^2}{(2-\mu')L_{opt}}} \quad (22)$$

$$\gamma_u = \frac{-\frac{K_2\alpha}{K_3} - \sqrt{\frac{K_2^2\alpha^2}{K_3^2} + \frac{4(\eta^2\delta^2 + \alpha^2)}{K_3}}}{2} \quad (23)$$

$$K_1 = \frac{\Delta\mu'\sigma_v^2}{(2-\mu')L_{opt}} \quad (24)$$

$$K_2 = 2 + 4\sigma_v^2 + \frac{4L_{opt}K_1}{\Delta} \quad (25)$$

$$K_3 = 2K_1K_2 - 2K_1\sigma_c^2((\Delta - 1)\sigma_x^2 + 3) \quad (26)$$

where σ_c^2 is the variance of the tail sequence of the optimal weight. k and η are small positive numbers which can be selected according to the requirement of the steady tap-length oscillation. The values of γ_l and γ_u are also the guidelines of ST in all FT algorithms. These two values are coincident in different FT algorithms. The fixed γ is selected by these two parameters in FT algorithms. Then, Equation (15) can also be expressed as:

$$q_1(n) = \lambda^n q_1(0) + (1 - \lambda) \sum_{i=1}^n \lambda^{n-i} |\text{FE}(i)| \quad (27)$$

Taking the expectation on both sides, Equation (27) becomes:

$$E\{q_1(n)\} = \lambda^n q_1(0) + (1 - \lambda) \sum_{i=1}^n \lambda^{n-i} E\{|\text{FE}(i)|\} \approx \lambda^n q_1(0) + (1 - \lambda^n) E\{|\text{FE}(n)|\} \quad (28)$$

In the steady-state, $n \rightarrow \infty$, as λ is smaller than one, and Equation (28) becomes:

$$E\{q_1\{\infty\}\} = E\{|\text{FE}(\infty)|\} \quad (29)$$

The values of $|\text{FE}(n)|$ and $\text{FE}(n)$ can be expressed from Equation (10).

$$|\text{FE}(n)| = \left| e_{L(n)}^{(L(n))}(n)^2 - e_{L(n)-\Delta}^{(L(n))}(n)^2 \right| = \left| \frac{L_f(n+1) - L_f(n) + \alpha}{-\gamma(n)} \right| \quad (30)$$

$$\text{FE}(n) = e_{L(n)}^{(L(n))}(n)^2 - e_{L(n)-\Delta}^{(L(n))}(n)^2 = \frac{L_f(n+1) - L_f(n) + \alpha}{-\gamma(n)} \quad (31)$$

In addition, in the steady-state, $n \rightarrow \infty$, the mean values of Equations (30) and (31) are expressed as follows:

$$E\{|FE(\infty)|\} = E\left\{\left|e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right|\right\} = E\left\{\left|\frac{L_f(\infty) - L_f(\infty) + \alpha}{-\gamma(\infty)}\right|\right\} = \frac{\alpha}{\gamma(\infty)} \quad (32)$$

$$E\{FE(\infty)\} = E\left\{e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right\} = E\left\{\frac{L_f(\infty) - L_f(\infty) + \alpha}{-\gamma(\infty)}\right\} = -\frac{\alpha}{\gamma(\infty)} \quad (33)$$

where $\gamma(\infty)$ is the value of ST in the steady-state. Its value is equal to the fixed γ of FT algorithm. $\gamma(\infty)$ and α are larger than zero. Using the bound of γ , the range of ρ_1 is obtained:

$$\frac{\gamma_l}{E\{q_1(\infty)\}} < \rho_1 < \frac{\gamma_u}{E\{q_1(\infty)\}} \quad (34)$$

$$E\{q_1(\infty)\} = E\left\{\left|e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right|\right\} = \frac{\alpha}{\gamma(\infty)} \quad (35)$$

Then, the value of ρ_2 is analyzed. When $n \rightarrow \infty$, the steady-state $\Delta(\infty)$ can be expressed as follows:

$$\Delta(\infty) = \rho_2 |E\{q_2(\infty)\}| \quad (36)$$

where $\Delta(\infty)$ can be chosen by the requirement of the system. Too large value of $\Delta(\infty)$ leads to a large tap-length bias in the steady-state. Thus, it should be a small integer value. The value of $|E\{q_2(\infty)\}|$ can be deduced as follows:

$$q_2(n) = \lambda^n q_2(0) + (1 - \lambda) \sum_{i=1}^n \lambda^{n-i} FE(i) \quad (37)$$

$$E\{q_2(n)\} = \lambda^n q_2(0) + (1 - \lambda) \sum_{i=1}^n \lambda^{n-i} E\{FE(i)\} \approx \lambda^n q_2(0) + (1 - \lambda^n) E\{FE(i)\} \quad (38)$$

$$|E\{q_2(\infty)\}| = |E\{FE(\infty)\}| \quad (39)$$

The steady-state of q_2 can be obtained:

$$|E\{q_2(\infty)\}| = \left|E\left\{e_{L(\infty)}^{(L(\infty))}(\infty)^2 - e_{L(\infty)-\Delta}^{(L(\infty))}(\infty)^2\right\}\right| = \frac{\alpha}{\gamma(\infty)} \quad (40)$$

In the steady-state, the values of $E\{q_1(\infty)\}$ and $|E\{q_2(\infty)\}|$ are related with $\gamma(\infty)$ and α . The range of $\gamma(\infty)$ is between 0.1 and 1. However, the value of α is a few percent. It is far less than the value of $\gamma(\infty)$. Thus, $E\{q_1(\infty)\}$ and $|E\{q_2(\infty)\}|$ are small in the steady-state. ρ_2 is expressed as follows:

$$\rho_2 = \Delta(\infty) \frac{\gamma(\infty)}{\alpha} \quad (41)$$

From the above, the values of γ_l and γ_u are required by all FT algorithms and they should be calculated first. Then the fixed γ can be obtained. Thus, the most important parameters which are used to calculate ρ_1 and ρ_2 are $\gamma(\infty)$, $\Delta(\infty)$ and α . $\gamma(\infty)$ is equal to the fixed γ of FT algorithm. $\Delta(\infty)$ and α are the general parameters for all FT algorithms. Once the general parameters that are shown in Section 5 are confirmed, the values of ρ_1 and ρ_2 will be calculated.

4.2. Performance Analysis

In the FT algorithm, a large error width causes a fluctuation and overestimation of tap-length. At the other extreme, a small error width may cause a slow convergence rate and the under-modeling problem. To begin with, a larger error width is needed to converge the tap-length to the optimal interval quickly. According to Zhang et al. [8], the optimal solution of the tap-length tends to be close to $L_{opt} + \Delta$. In the steady-state, the error width should be as small as possible so that the tap-length is

close to the optimal solution. Similarly, we infer that ST has the same characteristic. Thus, ST and error width should be large enough to speed up convergence in the transient state. In the steady-state, ST and error width should be reduced to improve the steady-state performance. Given these characteristics, their changing trends should be the same. The proposed algorithm uses the same FE to control ST and error width so that they are relatively consistent. The main theoretical difference between the proposed algorithm and algorithm [13] is that the proposed algorithm adds the analysis and changing strategy of the other important parameter ST. ST and error width of the proposed algorithm all have good adaptive ability. They can meet the corresponding values at different stages.

In the next step, we analyzed the characteristic of FE at different stages according to Han et al. [13]. In the transient state, \mathbf{W}_{opt} , \mathbf{X}_{Lopt} , and \mathbf{P} are defined as follows:

$$\mathbf{W}_{opt} = \begin{bmatrix} \mathbf{W}_{L(n)-\Delta}^* \\ \overline{\mathbf{W}}_{L(n)-\Delta+1:L(n)}^* \\ \overline{\overline{\mathbf{W}}}_{L(n)+1:L_{opt}}^* \end{bmatrix} \quad (42)$$

$$\mathbf{X}_{Lopt} = \begin{bmatrix} \mathbf{x}_{L(n)-\Delta}^* \\ \overline{\mathbf{x}}_{L(n)-\Delta+1:L(n)}^* \\ \overline{\overline{\mathbf{x}}}_{L(n)+1:L_{opt}}^* \end{bmatrix} \quad (43)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{W}_{L(n)-\Delta} - \mathbf{W}_{L(n)-\Delta}^* \\ \mathbf{W}_{L(n)-\Delta+1:L(n)} - \overline{\mathbf{W}}_{L(n)-\Delta+1:L(n)}^* \\ \mathbf{0} - \overline{\overline{\mathbf{W}}}_{L(n)+1:L_{opt}}^* \end{bmatrix} = \begin{bmatrix} \mathbf{v}^* \\ \overline{\mathbf{v}}^* \\ \overline{\overline{\mathbf{v}}}^* \end{bmatrix} \quad (44)$$

Then we rewrite $e_{L(n)}^{(L(n))}(n)$ and $e_{L(n)-\Delta}^{(L(n))}(n)$.

$$e_{L(n)}^{(L(n))}(n) = -\mathbf{x}_{L(n)-\Delta}^{*T} \mathbf{v}^* - \overline{\mathbf{x}}_{L(n)-\Delta+1:L(n)}^{*T} \overline{\mathbf{v}}^* - \overline{\overline{\mathbf{x}}}_{L(n)+1:L_{opt}}^{*T} \overline{\overline{\mathbf{v}}}^* + v(n) \quad (45)$$

$$e_{L(n)-\Delta}^{(L(n))}(n) = -\mathbf{x}_{L(n)-\Delta}^{*T} v^* + \overline{\mathbf{x}}_{L(n)-\Delta+1:L(n)}^{*T} \overline{\mathbf{W}}_{L(n)-\Delta+1:L(n)}^* - \overline{\overline{\mathbf{x}}}_{L(n)+1:L_{opt}}^{*T} \overline{\overline{\mathbf{v}}}^* + v(n) \quad (46)$$

where $v(n)$ is the noise signal. The expression of FE in the transient state is:

$$E\{e_{L(n)}^{(L(n))}(n)^2 - e_{L(n)-\Delta}^{(L(n))}(n)^2\} = \sigma_x^2 E\{\overline{\mathbf{v}}^{*T} \overline{\mathbf{v}}^*\} - \sigma_x^2 E\{\overline{\mathbf{W}}_{L(n)-\Delta+1:L(n)}^{*T} \overline{\mathbf{W}}_{L(n)-\Delta+1:L(n)}^*\} \quad (47)$$

$e_{L(n)}^{(L(n))}(n)^2$ and $e_{L(n)-\Delta}^{(L(n))}(n)^2$ all have noise signal $v(n)$. As we subtract the former from the latter to calculate FE, FE is immune to the noise signal. It includes two elements that are associated with the weight vector at the initial stage. The former varies, and the latter does not. Once tap-length is far from the optimum value, the former will increase. Then, FE makes ST and error width increase accordingly. Thus, convergence speed is improved in the transient stage.

Similarly, according to Han et al. [13], the estimated energy of FE in the steady-state is:

$$E\{e_{L(n)}^{(L(n))}(n)^2 - e_{L(n)-\Delta}^{(L(n))}(n)^2\} = \sigma_x^2 E\{\overline{\overline{\mathbf{v}}}^{*T} \overline{\overline{\mathbf{v}}}^*\} \quad (48)$$

where $\overline{\overline{\mathbf{v}}}^* = \overline{\overline{\mathbf{W}}}_{L(n)-\Delta+1:L(n)}^* - \mathbf{0}$ and $\mathbf{W}_{L(n)} = \begin{bmatrix} \mathbf{W}_{1:L_{opt}}^* \\ \overline{\mathbf{W}}_{L_{opt}+1:L(n)-\Delta}^* \\ \overline{\overline{\mathbf{W}}}_{L(n)-\Delta+1:L(n)}^* \end{bmatrix}$ in the steady-state. The expression

of FE contains only one part which is associated with excessive weight in the steady-state. As excessive weight is always small, FE is also small in the steady-state. It makes a small ST and error width in the steady-state. Thus, the steady-state performance improves.

The proposed algorithm adopts the same variable FE to adjust ST and error width so that they are relatively consistent. According to the analysis of Equations (47) and (48), FE is noise-free and optimizes ST and error width at the corresponding stage. Once tap-length is far from the optimum, a large FE will bring a large ST and error width. It can avoid a suboptimal value of tap-length and improve the convergence speed. In the steady-state, small FE results in a small ST and error width to avoid fluctuation and improve the steady-state performance. The proposed algorithm equips adaptive parameters in different phases.

5. Simulations and Results

In this section, the proposed method compares with the FT algorithm [7] and other algorithms [11–13] via simulation. FT Algorithm 1 adopts the minimum error width and FT Algorithm 2 adopts the maximum error width. To compare the performance fairly, the experimental condition is similar to that used by Yu and Cowan [7] and Han et al. [13]. The reference input was obtained by passing white Gaussian noise through a transfer system of $H(z) = 0.35 + z^{-1} + z^{-2}$. Transfer system $H(z)$ was referenced from Yu and Cowan [7]. Most experimental conditions used this transfer system. To compare these algorithms fairly, we have adopted the same systems as them. SNR of 20 dB and 0 dB were obtained by adding white Gaussian interference noise. Small-scale system and large-scale system are the typical experimental systems. However, it is difficult to compare the performance of tap-length and MSE in small-scale experimental environment. The tap-length is too small and the changing range of tap-length is very limited. The effect of contrast is not significant. Moreover, the dynamic ranges of error width and ST are relatively small. In this case, the comparison of performance cannot be very prominent. As there is rarely small-scale experimental condition except [7], it is also hard to establish a fair and universal small-scale experimental environment. Thus, the large-scale system is mainly adopted in this section. The experiment used two different unknown systems to evaluate the performance. We define:

$$\mathbf{W}_1 = \sum_{i=1}^{150} a_i z^{-i}, \mathbf{W}_2 = \sum_{i=1}^{80} b_i z^{-i} \quad (49)$$

$$\mathbf{W}_3 = \sum_{i=1}^{200} c_i z^{-i}, \mathbf{W}_4 = \sum_{i=1}^{100} d_i z^{-i} \quad (50)$$

where a_i , b_i , c_i and d_i are adopted from the white Gaussian random sequence with zero-mean and unity variance. We used $\mathbf{W}_{opt} = \mathbf{W}_1$ for $n < 10,000$ and $\mathbf{W}_{opt} = \mathbf{W}_2$ for $10,000 \leq n < 20,000$ in the first unknown system simulation, $\mathbf{W}_{opt} = \mathbf{W}_3$ for $n < 10,000$ and $\mathbf{W}_{opt} = \mathbf{W}_4$ for $10,000 \leq n < 20,000$ in the second unknown system simulation.

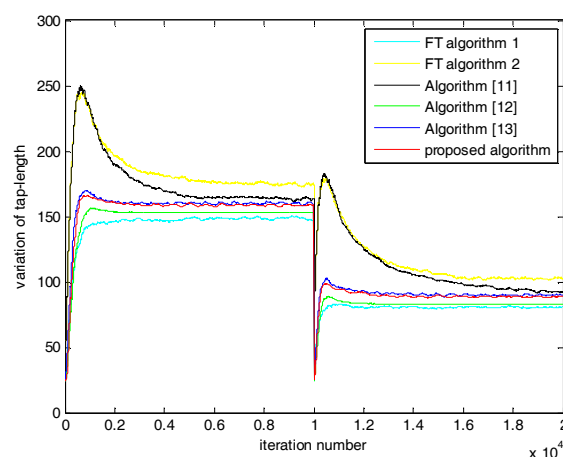
The general parameters (α , μ , ε , β , δ , γ , $\gamma(\infty)$, $\Delta(\infty)$ and Δ_{max}) are chosen first according to the algorithm in [7] and algorithms in [11–13]. Every parameter is selected fairly. These general parameters have no difference for all algorithms. These general parameters are used to calculate the particular parameters in each algorithm. The detailed parameters are listed in the Table 1. The following figures were acquired from 100 and 500 independent experiments with SNR = 20 dB and SNR = 0 dB respectively.

Table 1. Parameters of the compared algorithms.

Algorithm	SNR = 20 dB	SNR = 0 dB
General	$\alpha = 0.03, \mu = 1, \varepsilon = 1, \beta = 0.1, \delta = 1, \gamma = 0.5,$ $\gamma(\infty) = 0.5, \Delta(\infty) = 1, \Delta_{max} = 20$	$\alpha = 0.03, \mu = 0.3, \varepsilon = 1, \beta = 0.1, \delta = 1,$ $\gamma = 0.5, \gamma(\infty) = 0.5, \Delta(\infty) = 1, \Delta_{max} = 50$
FT Algorithm 1	$\gamma = 0.5, \Delta = 1$	$\gamma = 0.5, \Delta = 1$
FT Algorithm 2	$\gamma = 0.5, \Delta = 20$	$\gamma = 0.5, \Delta = 50$
Algorithm [11]	$\gamma = 0.5, \Delta_{max} = 20, \Delta(\infty) = 1$	$\gamma = 0.5, \Delta_{max} = 50, \Delta(\infty) = 1$
Algorithm [12]	$\Delta = 4, \rho = 2, C = 0.02$	$\Delta = 4, \rho = 2, C = 0.02$
Algorithm [13]	$\gamma = 0.5, \rho = 16, \Delta(\infty) = 1, C = 0.5,$ $\lambda_1 = 0.9, \lambda_2 = 0.95$	$\gamma = 0.5, \rho = 16, \Delta(\infty) = 1, C = 0.5,$ $\lambda_1 = 0.9, \lambda_2 = 0.95$
Proposed algorithm	$\rho_1 = 0.01, \rho_2 = 16, \gamma(\infty) = 0.5,$ $\Delta(\infty) = 1, C = 0.5, \lambda_1 = 0.9, \lambda_2 = 0.95$	$\rho_1 = 0.01, \rho_2 = 16, \gamma(\infty) = 0.5,$ $\Delta(\infty) = 1, C = 0.5, \lambda_1 = 0.9, \lambda_2 = 0.95$

5.1. Case 1: SNR = 20 dB

We show the variation of tap-length in Figures 2 and 3. As FT Algorithm 1 adopted the minimum error width, the under-modeling problem appeared. At the beginning, FT Algorithm 2 and the algorithm introduced by Mayyas et al. [11] had a large convergence rate. However, severe overestimation appeared. Tap-lengths of the algorithms introduced by Han et al. [12,13] were in the optimum range. However, the accurate ST and error width of the proposed algorithm made the corresponding tap-length estimate more accurately. The proposed algorithm maintained their advantages and mitigated their disadvantages. Before 1000, 10,500, 1300 and 10,400 iterations in Figures 2 and 3, the slopes of the proposed algorithm were larger than that of the algorithm in [12]. The variation of tap-length in the proposed algorithm was relatively faster than that in the algorithm in [12]. The algorithm in [13] had a little overestimation at the beginning of iteration. The peak of the algorithm [13] was larger than that of the proposed algorithm. The performance of the proposed algorithm is the balance of these two algorithms in [12,13]. The proposed algorithm slightly improved the initial convergence speed compared with the algorithm in [12] and improved the overestimation of tap-length compared with the algorithm in [13].

**Figure 2.** Tap-length variation of different algorithms in the first experimental system.

Next, we compared MSD performance. The variation of tap-length correlated with MSD performance. Given the under-modeling problem in FT Algorithm 1 shown in Figures 2 and 3, FT Algorithm 1 had the worst MSD performance. Similarly, lacking the varying strategy of ST or error width, MSD of FT Algorithm 2 and that of Mayyas et al. [11] were weaker than the other algorithms. Han and coworkers' algorithms [12,13] had their respective strategies of error width or ST. However, their performance was also slightly affected by the lack of the other parameters. In the first half of Figures 4 and 5, MSD of Han and coworkers' algorithm [12] was stable at 4000 and 5000 iterations.

Its MSD convergence speed was slower and its MSD was larger than the proposed algorithm. The MSD of Han and coworkers' algorithm [13] fluctuated in the second half of Figure 5. As the proposed algorithm used strategies of ST and error width, it improved the stability and convergence performance compared with the other algorithms.

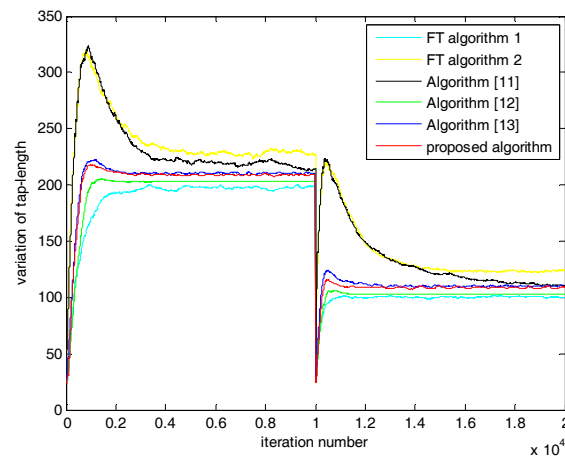


Figure 3. Tap-length variation of different algorithms in the second experimental system.

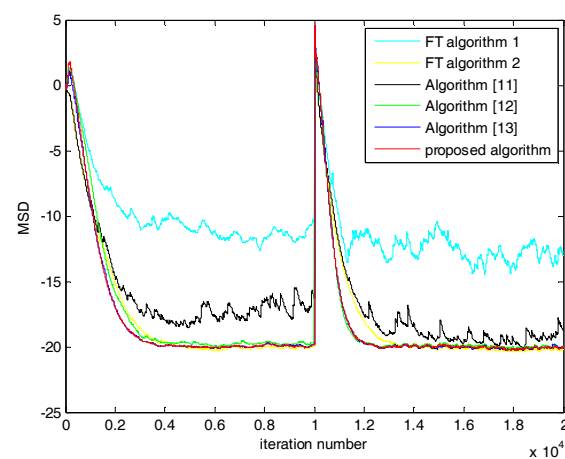


Figure 4. MSD of different algorithms in the first experimental system.

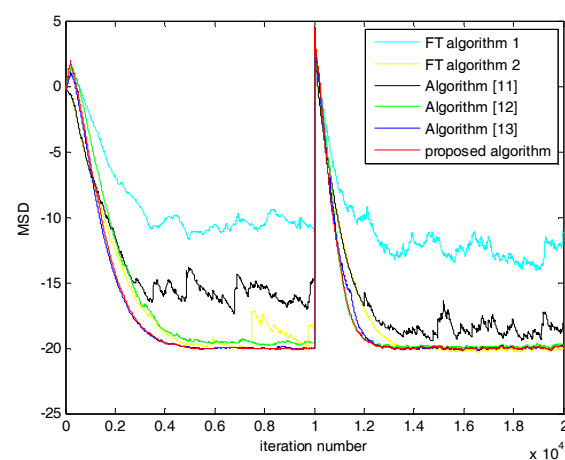


Figure 5. MSD of different algorithms in the second experimental system.

5.2. Case 2: SNR = 0 dB

The tracking abilities of Han and coworkers' algorithm [12], FT Algorithm 1, and FT Algorithm 2 deteriorated to a certain extent in high noise condition. Under-modeling, large fluctuations and large overestimations of tap-length all appeared in these algorithms. Due to the lack of appropriate error width, the under-modeling problem appeared in Han and coworkers' algorithm [12]. From the Figures 6 and 7, the effect of error width on algorithm is greater than ST in a high noise environments. The error width acts as a decisive role in a high noise environment. These algorithms performed poorly in the SNR = 0 dB environment.

On the contrary, the algorithms of Mayyas et al. [11] and Han et al. [13], and the proposed algorithm were close to the optimum value. However, the performance of the algorithms of Mayyas et al. [11] and Han et al. [13] were also affected by the lack of reasonable choice for ST. The tap-length of the proposed algorithm converged at 1000, 11,000, 2000, and 10,500 iterations in Figures 6 and 7, respectively. All other algorithms converged after these iterations. Moreover, the proposed algorithm did not fluctuate after completing convergence. The other algorithms still fluctuated. Although the environment changed, the proposed algorithm demonstrated good tracking ability and avoided large overestimations of tap-length.

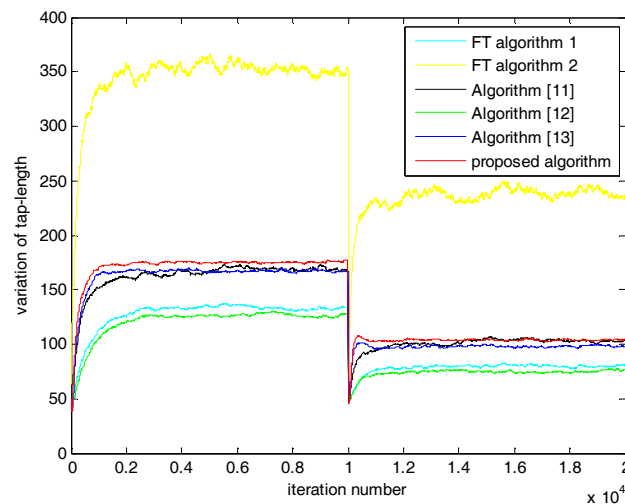


Figure 6. Tap-length variation of different algorithms in SNR = 0 dB.

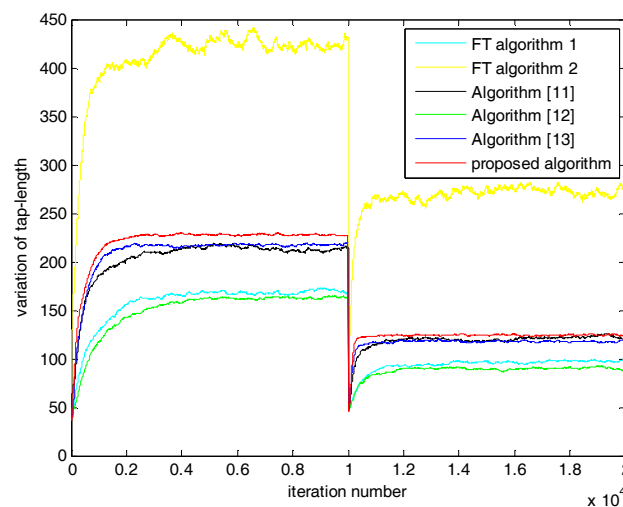


Figure 7. Tap-length variation of different algorithms in SNR = 0 dB.

In Figures 8 and 9, the MSD performance of these algorithms is generally poor except the proposed algorithm. MSD performance of the proposed algorithm is low and stable. This is because accurate tap-length of the proposed algorithm made accurate estimate of the corresponding tap-coefficients. Moreover, the reasonable parameters improved the convergence and steady-state performance of the proposed algorithm. According to the analysis in Section 4, the proposed algorithm was not affected by noise. It has been confirmed by the experimental results of the high noise environment. It also has good robustness. It can adapt to changes of the environment and returns more appropriate ST and error width than the other algorithms.

Moreover, we also experimented with the MSD gradient. We chose an iteration period to compare MSD gradient. Figures 10 and 11 plot the MSD gradient of different variable tap-length algorithms in SNR = 20 dB and SNR = 0 dB.

Different algorithms had different MSD gradient performance. MSD gradient is $t(n)$ from Equation (5). It indicates the MSD difference between two moments in each iteration. The larger is the MSD gradient, the better is the performance that will be obtained. In Figures 10 and 11, although the proposed algorithm fluctuated, its peak and bottom values were larger than those of the other algorithms. To better compare these algorithms, we calculated the mean values of the MSD gradient (Table 2). As shown in Table 2, although these variable tap-length algorithms improved MSD gradient to a certain extent, the proposed algorithm still had the largest MSD gradient. In combination with the MSD results and MSD gradient, the proposed algorithm led MSD to undergo the largest descent at each iteration. It also guaranteed the achievement of the best possible MSD performance.

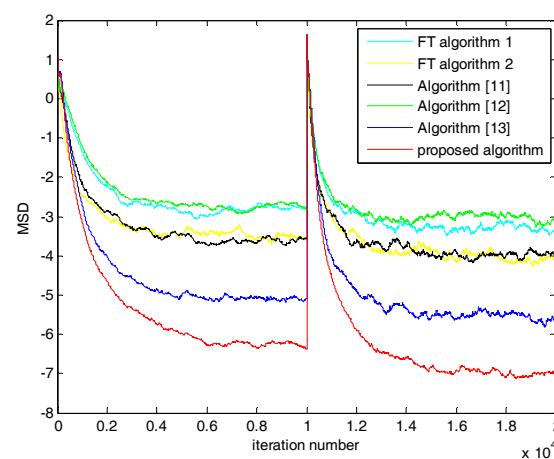


Figure 8. MSD of different algorithms in SNR = 0 dB.

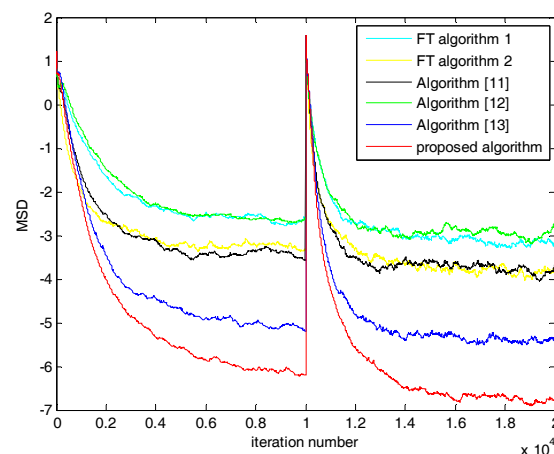


Figure 9. MSD of different algorithms in SNR = 0 dB.

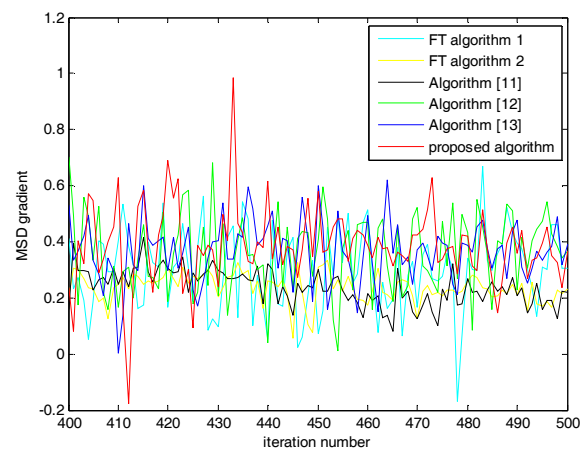


Figure 10. MSD gradient of different algorithms in SNR = 20 dB.

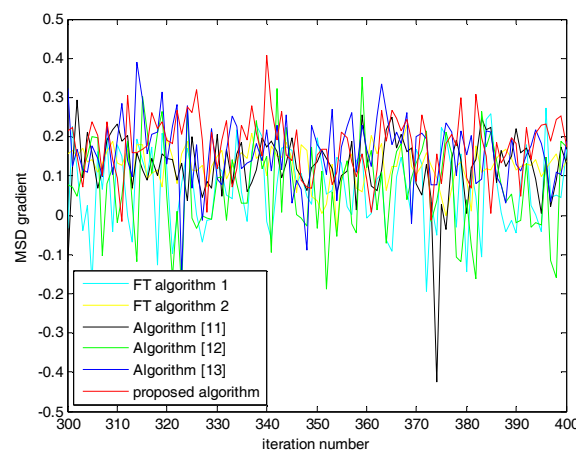


Figure 11. MSD gradient of different algorithms in SNR = 0 dB.

Table 2. Mean value of MSD gradient in different SNR environments.

Mean Value	SNR = 20 dB	SNR = 0 dB
FT Algorithm 1	0.2977	0.0701
FT Algorithm 2	0.2365	0.1259
Algorithm [11]	0.2390	0.1244
Algorithm [12]	0.3624	0.0700
Algorithm [13]	0.3660	0.1613
Proposed algorithm	0.3872	0.1774

The experimental results agreed with the theoretical analysis. The proposed algorithm simultaneously improved the convergence and steady-state performance.

6. Conclusions

A mixed variable tap-length algorithm which adopts the time-varying strategies of ST and error width is presented in this paper. ST and error width were analyzed in detail. The proposed method used the same variable FE to adjust ST and error width. This method ensured that their changes are relatively consistent. The proposed algorithm reasonably combined ST and error width so that they coordinated and cooperated. The convergence and steady-state performance of the proposed algorithm is improved. The overestimation problem of tap-length was also optimized. At the same time, the restricted function was added to avert large fluctuations of tap-length. The proposed algorithm can

adapt to changing environments. It also increased the robustness. The experimental results agreed well with the theoretical analysis.

Author Contributions: Y.H. and M.W. conceived the research theme and designed the experiments; Y.H. and Y.L. performed the experiments; Y.H. analyzed the experimental data and wrote the manuscript; M.W. contributed materials/analysis tools and assisted to supervise the field activities. All authors read and approved the final manuscript.

Funding: This work was supported by the Shenzhen Fundamental Research Project under Grant JCYJ20170412151226061, and Grant JCYJ20170808110410773.

Acknowledgments: The authors would like to thank all the editors and reviewers for valuable feedback on this manuscript. They provided much help and suggestions in this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Magliacano, D.; Viscardi, M.; Dimino, I.; Concilio, A. Active vibration control by piezoceramic actuators of a car floor panel. In Proceedings of the 23rd International Congress on Sound and Vibration (ICSV), Athens, Greece, 10–14 July 2016.
2. Qu, Y.; He, D. An adaptive remaining useful life prediction method for hybrid ceramic bearing. In Proceedings of the 2016 International Conference on Mechanics Design, Manufacturing and Automation (MDM 2016), Suzhou, China, 14–15 May 2016.
3. Haykin, S. *Adaptive Filter Theory*, 3rd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 1996.
4. Sayed, A.H. *Fundamentals of Adaptive Filtering*; Wiley: New York, NY, USA, 2003.
5. Gu, Y.; Tang, K.; Cui, H. LMS algorithm with gradient descent filter length. *IEEE Signal Process. Lett.* **2004**, *11*, 305–307. [[CrossRef](#)]
6. Riera-Palou, F.; Noras, J.M.; Cruickshank, D.G.M. Linear equalisers with dynamic and automatic length selection. *Electron. Lett.* **2001**, *37*, 1553–1554. [[CrossRef](#)]
7. Yu, G.; Cowan, C.F.N. An LMS style variable tap-length algorithm for structure adaptation. *IEEE Trans. Signal Process.* **2005**, *53*, 2400–2407.
8. Zhang, Y.; Li, N.; Chambers, J.A.; Sayed, A.H. Steady-state performance analysis of a variable tap-length LMS algorithm. *IEEE Trans. Signal Process.* **2008**, *56*, 839–845. [[CrossRef](#)]
9. Li, N.; Zhang, Y.; Zhao, Y.; Hao, Y. An improved variable tap-length LMS algorithm. *Signal Process.* **2009**, *89*, 908–912. [[CrossRef](#)]
10. Xu, D.J.; Yin, B.; Wang, W.; Zhu, W. Variable tap-length LMS algorithm based on adaptive parameters for TDL structure adaption. *IEEE Signal Process. Lett.* **2014**, *21*, 809–813.
11. Mayyas, K.; Abuseba, H.A. A new variable length NLMS adaptive algorithm. *Digit. Signal Process.* **2014**, *34*, 82–91. [[CrossRef](#)]
12. Han, Y.; Wang, M.; Zhao, B. Variable tap-length NLMS algorithm with adaptive parameter. *IEEE Trans. Fundam. Electron. Commun. Comput. Sci.* **2017**, *100*, 1720–1723. [[CrossRef](#)]
13. Han, Y.; Wang, M.; Liu, M. An improved variable tap-length algorithm with adaptive parameters. *Digit. Signal Process.* **2018**, *74*, 111–118. [[CrossRef](#)]
14. Almeida, S.J.M.D.; Bermudez, J.C.M.; Bershad, N.J.; Costa, M.H. A statistical analysis of the affine projection algorithm for unity step size and autoregressive inputs. *IEEE Trans. Circ. Syst.* **2005**, *52*, 1394–1405. [[CrossRef](#)]
15. Mayyas, K. Performance analysis of deficient length LMS adaptive algorithm. *IEEE Trans. Signal Process.* **2005**, *53*, 2727–2734. [[CrossRef](#)]
16. Shin, H.C.; Sayed, A.H. Mean-square performance of a family of affine projection algorithms. *IEEE Trans. Signal Process.* **2004**, *52*, 90–102. [[CrossRef](#)]

