

# A Fast Converging Biased NLMS Algorithm

YanLing Hao, Ying Cui, YongGang Zhang

Department of Automation, Harbin Engineering University Harbin, Heilongjiang Province, China  
haoyanling@hrbeu.edu.cn, yingcui1987@163.com, zhangyg@hrbeu.edu.cn

**Abstract** - It's well known, biased estimation can effectively reduce the steady-state mean-square error (MSE), as for system identification with a low signal-to-noise ratio (SNR). A new biased normalized least-mean-squares (NLMS) algorithm is proposed in this paper. In the proposed algorithm the optimal solution is deduced on the basis of maximizing the decrease of the mean square deviation (MSD). To facilitate practical application approximation and parameter choice guidelines are provided for the new algorithm. The proposed algorithm is confirmed by simulations to obtain both a small steady-state excess MSE (EMSE) and a fast convergence rate, and to outperform the existing convex combination of two NLMS filters (CNLMS) algorithm. Steady state performance and transient performance analyses are provided to interpret the simulation results.

**Index Terms** - biased NLMS algorithm, low signal-to-noise ratio, system identification

## 1. Introduction

As is well known, the normalized least mean squares algorithm (NLMS) is widespread used in engineering because of its simpleness and stable performance<sup>[1]</sup>. In system identification, it provides an unbiased estimation for impulse response of the unknown system, and it has good transient performance and steady-state performance in Medium SNR and High SNR situations<sup>[1]</sup>. Fig. 1 shows a general model of system identification, the input signal  $u$  goes through the unknown system to get a desired output signal  $d$ , at the same time, it goes through a adaptive filter to get an estimate  $y$  of the desired output signal, and then the subtraction of the two signals is being utilized to adjust the filter parameter until the algorithm enter the steady state.

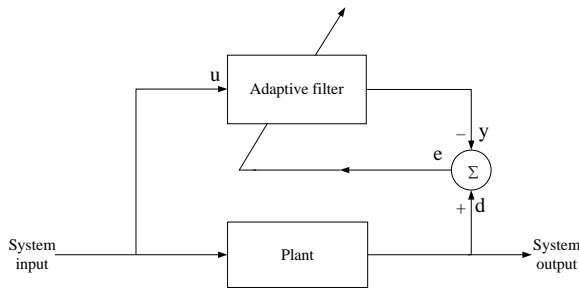


Fig1. System Identification Model

However, when the SNR gets lower, the adaptive noise of unbiased estimation algorithm gets greater, leading to large steady-state error, performance deteriorates. The biased estimation algorithm provides a new thought for filtering problems under a low SNR, its main idea is establishing balance between estimate bias and estimate noise by adjusting shrinkage coefficient to reduce the steady-state error [2~4]. A biased least squares algorithm is proposed in [2], iteration of

the shrinkage coefficient is provided based on the idea of least mean square algorithm. However, it suffers from too much computation, thus limits its practical application. A simple biased adaptive filter is proposed in [3], in which the idea of convex combination of two NLMS filters, namely the CNLMS algorithm, is used to deduce the iteration of the shrinkage coefficient, and it performs well in low SNR region of (-25dB~15dB). An improved version of the CNLMS algorithm is proposed in [4] for sparse systems, in which the impulse response is divided into several blocks, the shrinkage coefficient is computed independently in each block, yet as a whole, the two algorithms are basically consistent, we just take the CNLMS algorithm into further consideration.

Following the CNLMS algorithm, its basic thought about biased estimation is taking the output of the biased estimation as a combination of an NLMS algorithm with a random step and a filter with output of 0 (namely the NLMS filter's output is zero) [3]. In this way, the transient performance of the CNLMS algorithm is affected both by the NLMS algorithm and the shrinkage coefficient. Moreover, as is shown in our simulations, it converges slower as the SNR gets lower. Besides, from the aspect of steady state performance, it is not enough to reflect the advantage of the biased estimation over the NLMS algorithm, since the NLMS algorithm can get a low steady state error by adjusting the step properly for the system whose noise is slightly higher than the signal.

In this paper, we are aimed to provide a biased NLMS algorithm that has a fast convergence rate as well as a low steady-state error, especially for the SNR region of (-40dB~-25dB). The optimal shrinkage coefficient is deduced for each step, by maximizing the mean square deviation descent velocity. To facilitate practical application, approximation and parameter choice guidelines are provided for the new algorithm. The subsequent simulation results prove the effectiveness of the algorithm.

## 2. CNLMS Algorithm

### A. A brief introduction of the CNLM Algorithm

Fig.2 shows the biased estimation model in [3], where  $\mathbf{u}(n)=[u(n), u(n-1), \dots, u(n-L+1)]^T$  is the system input vector, with  $L$  being the filter length;  $y(n)$  is the output that is obtained from the unbiased adaptive filter;  $y_a(n)=\alpha y(n)$  is the biased output, where  $\alpha$  is the shrinkage coefficient; The biased estimation error  $e_a(n)$  is the difference between the system desired signal  $d(n)$  and the biased output  $y_a(n)$ .

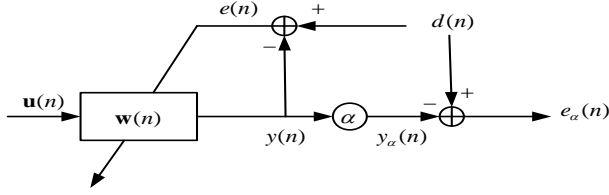


Fig.2 Biased Estimation Model in [3]

The biased NLMS Algorithm in [3] is summarized as follows:

Firstly, an unbiased estimation is obtained for the impulse response of the system following the NLMS algorithm:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{\mathbf{u}(n)e(n)}{\|\mathbf{u}(n)\|^2} \quad (1)$$

where  $\mu$  is step size.

and the estimation error is:

$$e(n) = d(n) - y(n) \quad (2)$$

in which the output of the filter is:

$$y(n) = \mathbf{w}^T(n) \mathbf{u}(n) \quad (3)$$

with the desired output being:

$$d(n) = \mathbf{w}_o^T \mathbf{u}(n) + t(n) \quad (4)$$

in which  $t(n)$  is system noise and is assumed to be zero-mean with variance of  $\sigma_t^2$ .  $\mathbf{w}_o$  is the optimal solution of filter weight vectors with L weights.

Then the shrinkage coefficient is calculated

$$\alpha = f[\bullet] \quad (5)$$

which ranges from 0 to 1, and is defined by a certain function  $f[\bullet]$ . And the output of the biased filter is obtained:

$$y_\alpha(n) = \alpha y(n) \quad (6)$$

namely the biased estimation of the impulse response is

$$\mathbf{w}_\alpha(n) = \alpha \mathbf{w}(n)$$

Finally the biased estimation error is denoted as:

$$e_\alpha(n) = d(n) - y_\alpha(n) \quad (7)$$

The shrinkage coefficient is defined as follows in the CNLMS algorithm [3].

$$\alpha(n) = \frac{\text{sgm}[a(n)] - \text{sgm}[-a^+]}{\text{sgm}[a^+] - \text{sgm}[-a^+]} \quad (8)$$

In order to show the performance of the CNLMS algorithm and conquer the influence of gradient noise, we only introduce the indirect method of calculating the shrinkage coefficient (Just as the equation (22) in [3]), where

$\text{sgm}[a(n)] = \frac{1}{1 + \exp[-a(n)]}$ ,  $a(n) \in [-a^+, a^+]$ , and is calculated as the following equation:

$$a(n+1) = a(n) + \frac{\mu_a}{p(n)} e_\alpha(n) y(n) \frac{\partial \alpha(n)}{\partial a(n)} \quad (9)$$

where  $\mu_a$  is the step size of the shrinkage coefficient;

$p(n) \approx E\{y^2(n)\}$  and is specifically:

$$p(n) = \gamma p(n-1) + (1-\gamma) y^2(n) \quad (10)$$

in which,  $\gamma=0.9$  is the smoothing factor in exponential window.

Besides, for convenience of description, several variables are defined as follows:

the autocorrelation matrix of the input vector is

$$\mathbf{R} = E\{\mathbf{u}(n)\mathbf{u}^T(n)\}$$

the SNR of system is

$$SNR = \frac{E\{(\mathbf{w}_o^T \mathbf{u}(n))^2\}}{E\{t^2(n)\}} = \frac{\mathbf{w}_o^T \mathbf{R} \mathbf{w}_o}{\sigma_t^2} \quad (11)$$

extra estimation error is:

$$\begin{aligned} \varepsilon(n) &= e(n) - t(n) \\ &= (\mathbf{w}_o - \mathbf{w}(n))^T \mathbf{u}(n) \\ &= \tilde{\mathbf{w}}^T(n) \mathbf{u}(n) \end{aligned} \quad (12)$$

where the estimation bias  $\tilde{\mathbf{w}}(n)$  is defined as above.

As for a biased estimation, (12) is rewritten as:

$$\begin{aligned} \varepsilon_\alpha(n) &= e_\alpha(n) - t(n) \\ &= (\mathbf{w}_o - \mathbf{w}_\alpha(n))^T \mathbf{u}(n) \\ &= \tilde{\mathbf{w}}_\alpha^T(n) \mathbf{u}(n) \end{aligned} \quad (13)$$

with the excess mean square error (EMSE) being

$$\mathbf{J}_{e_\alpha}(n) = E\{\varepsilon^2(n)\} \quad (14)$$

and

$$\mathbf{J}_{e_\alpha, \alpha}(n) = E\{\varepsilon_\alpha^2(n)\} \quad (15)$$

### B. CNLMS algorithm convergence performance simulation

We study the convergence performance of CNLMS algorithm under a low SNR of -35dB in this section, specifically the simulation settings are as follows: The system input vectors come from white noise sequence and the filter length  $L = 200$ , and then serial variance is adjusted to make the trace of the autocorrelation matrix of the input vector  $\text{tr}(\mathbf{R}) = 0.1$ . The system noise is white noise, and the variance is 0.1. The impulse response is randomly generated and the variance is scaled to make  $\text{SNR} = -35\text{dB}$ . Note that we only test the situation where the signal power is low in this section, while the case of high noise power is studied in subsequent sections. Moreover, the step size is set as 0.1, 0.5, 1.0, 1.5 respectively in NLMS algorithm, and the step size of shrinkage coefficient is 0.1 in CNLMS algorithm.

$a(n)$  is limited in the interval of  $[-4, 4]$ , and the initial value of shrinkage coefficient is set as 1, namely  $\alpha(0)=1, a(0)=4$ . The EMSE is obtained by smoothing the square of the excess error with a moving window of 50, Fig.3 is the average over 50 Monte Carlo simulations.

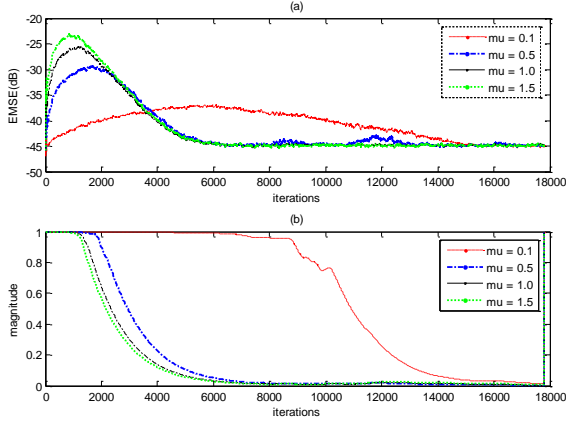


Fig.3 The Curves of CNLMS Algorithm Under Low SNR

Fig. 3(a) shows the EMSE curves of CNLMS algorithm with different step-sizes, while Fig.3(b) shows the curves of the shrinkage coefficient. As is shown in Fig. 3, the CNLMS algorithm suffers from a low convergence rate, moreover, its convergence rate is highly dependent on that of the NLMS algorithm. Specifically, the smaller the step size is in NLMS algorithm, the slower the shrinkage coefficient converges, thus the slower the convergence rate is in CNLMS algorithm. Besides, we can find in Fig.3(b) that the influence of gradient noise can be avoided by using indirect methods to calculate shrinkage coefficient, yet the gradient at bounding values (namely 0 and 1) lowers on the other hand, which results in a low convergence rate of the CNLMS algorithm.

As a result, we need to consider another method to compute the shrinkage coefficient, which can satisfy the requirement of a fast convergence rate.

### 3. A Fast Converging biased NLMS Algorithm

#### A. The optimal shrinkage coefficient

The original intention of the algorithm is improving the rate of convergence, so we calculate the optimal shrinkage coefficient in the foundation of maximizing the descent velocity of the MSD<sup>[5-6]</sup>. This algorithm is named BNLS. The specific derivation process is as follows:

Firstly, equation (1) is multiplied by  $\alpha(n+1)$  on both sides:

$$\mathbf{w}_\alpha(n+1) = \mathbf{w}_\alpha(n) + \alpha(n+1)\mu \frac{\mathbf{u}(n)e(n)}{\|\mathbf{u}(n)\|^2} \quad (16)$$

where we assume  $\alpha(n)$  is slowly varying, then we get  $\alpha(n+1) \approx \alpha(n)$ , namely  $\mathbf{w}_\alpha(n) \approx \alpha(n+1)\mathbf{w}(n)$ . Thus the iteration of the estimate bias  $\tilde{\mathbf{w}}_\alpha(n)$  is

$$\tilde{\mathbf{w}}_\alpha(n+1) = \tilde{\mathbf{w}}_\alpha(n) - \alpha(n+1)\mu \frac{\mathbf{u}(n)e(n)}{\|\mathbf{u}(n)\|^2} \quad (17)$$

We further take the square of the Euclid 2 norm on both sides of the equation (17), then take the expectations:

$$E\{\|\tilde{\mathbf{w}}_\alpha(n+1)\|^2\} - E\{\|\tilde{\mathbf{w}}_\alpha(n)\|^2\} = -2\alpha(n+1)\mu E\left\{\frac{\varepsilon_\alpha(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\} + \alpha^2(n+1)\mu^2 E\left\{\frac{e^2(n)}{\|\mathbf{u}(n)\|^2}\right\} \quad (18)$$

Note that in the above equation, the shrinkage coefficient is regarded as a constant, since we are aiming to deduce the optimal solution of shrinkage coefficient in every iteration.

The cost function is defined as

$J(\alpha(n+1)) = E\{\|\tilde{\mathbf{w}}_\alpha(n+1)\|^2\} - E\{\|\tilde{\mathbf{w}}_\alpha(n)\|^2\}$  and we can see from equation (18) that  $J(\alpha(n+1))$  has a least value. It's not difficult to find that the MSD has the fastest rate of descent when  $J(\alpha(n+1))$  is the least value. Then we take a derivative of the cost function with respect to  $\alpha(n+1)$ , and the following is obtained:

$$\frac{\partial J(\alpha(n+1))}{\partial \alpha(n+1)} = -2\mu E\left\{\frac{\varepsilon_\alpha(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\} + 2\alpha(n+1)\mu E\left\{\frac{y(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\} + 2\alpha(n+1)\mu^2 E\left\{\frac{e^2(n)}{\|\mathbf{u}(n)\|^2}\right\} \quad (19)$$

Note that the first item on the right side of equation (19) can be simplified into

$$\frac{\partial \varepsilon_\alpha(n)}{\partial \alpha(n+1)} = \frac{\partial}{\partial \alpha(n+1)} \{\mathbf{w}_o^T \mathbf{u}(n) - \alpha(n+1)y(n)\} = -y(n) \quad (20)$$

The optimal shrinkage coefficient  $\alpha_{opt}(n+1)$  is obtained as we set (19) equal to zero

$$\alpha_{opt}(n+1) = \frac{E\left\{\frac{\varepsilon_\alpha(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\}}{\mu E\left\{\frac{e^2(n)}{\|\mathbf{u}(n)\|^2}\right\} + E\left\{\frac{y(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\}} \quad (21)$$

#### B. The approximate optimal shrinkage coefficient

In practical application, because of unknown system noise, the excess error  $\varepsilon_\alpha(n)$  is unknown, an alternative way is to approximate it as estimate error, and we get the approximate optimal shrinkage coefficient

$$\alpha_{opt}(n+1) \approx \frac{E\left\{\frac{e_\alpha(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\}}{\mu E\left\{\frac{e^2(n)}{\|\mathbf{u}(n)\|^2}\right\} + E\left\{\frac{y(n)e(n)}{\|\mathbf{u}(n)\|^2}\right\}} \quad (22)$$

The approximate algorithm in equation (22) brings in the influence of noise, thus affects the accuracy. As for white system noise, namely  $E\{t(n)t(n-1)\} = 0$ , we can bring in noise in adjacent moment and smooth it to weaken the influence of noise on the algorithm [7]. As a result, we redefine the biased estimate error as follows

$$\begin{aligned} e_\alpha(n) &= d(n-1) - \alpha(n) \mathbf{w}^T(n) \mathbf{u}(n-1) \\ &= \tilde{\mathbf{w}}_\alpha^T(n) \mathbf{u}(n-1) + t(n-1) \end{aligned} \quad (23)$$

while definition of  $e(n)$  is the same as before.

Besides, we can get the real-time estimation of the three expectations by using three exponential windows.

$$\begin{aligned} \hat{p}(n) &= \beta \hat{p}(n-1) + (1-\beta) \frac{e_\alpha(n) e(n)}{\|\mathbf{u}(n)\|^2} \\ \hat{q}(n) &= \beta \hat{q}(n-1) + (1-\beta) \frac{e^2(n)}{\|\mathbf{u}(n)\|^2} \\ \hat{s}(n) &= \beta \hat{s}(n-1) + (1-\beta) \frac{y(n) e(n)}{\|\mathbf{u}(n)\|^2} \end{aligned} \quad (24)$$

For convenience of description, the following are defined

$$p(n) = E\left\{\frac{e_\alpha(n) e(n)}{\|\mathbf{u}(n)\|^2}\right\}, q(n) = E\left\{\frac{e^2(n)}{\|\mathbf{u}(n)\|^2}\right\}, s(n) = E\left\{\frac{y(n) e(n)}{\|\mathbf{u}(n)\|^2}\right\}$$

and  $\hat{p}(n), \hat{q}(n), \hat{s}(n)$  in (24) are their estimation separately.

Finally the shrinkage coefficient is approximated as follows in practice:

$$\alpha(n+1) = \frac{\hat{p}(n)}{\mu \hat{q}(n) + \hat{s}(n)} \quad (25)$$

Note that there is a bias between the estimation of exponential window and the actual expectation, however, if we take the same smoothing factor in the three exponential windows, the bias that affects the algorithm can be ignored [5]. Besides, the smoothing factor is usually set as  $\beta \in [0.9, 0.999]$ , and we set  $\beta = 0.999$  in this paper to ensure the smoothness.

#### 4. Simulation

In the first subsection, we provide the simulation results of the NLMS, CNLMS and BNLMS algorithms under a low SNR of  $-35\text{dB}$ , both the signal power and the noise power are scaled to make  $\text{SNR} = -35\text{dB}$ . In the second subsection, the tracking performance of the corresponding algorithms are tested. Besides, the simulation curves in this section are all the average of 50 Monte Carlo simulations.

##### A. The Simulation of Convergence Speed

###### A.1 The simulation under low signal

The simulation in Section 2.B proves that the transient performance of CNLMS algorithm is not ideal. In order to compare the convergence rate of two biased NLMS algorithms, the simulation settings are similar to those in above simulation, and the initial value of BNLMS algorithm is set as 1. Fig. 4 shows the simulation curves of BNLMS algorithm.

Fig. 4(a) shows EMSE curves of BNLMS algorithm with different step-sizes, and Fig. 4(b) shows the convergence curves of shrinkage coefficient in BNLMS algorithm. It can be seen from Fig. 4 that the convergence rate of the BNLMS algorithm is fast compared with those of the CNLMS

algorithm in Fig. 3, and it gets less influenced by the convergence rate of NLMS algorithm. For any step-size that we set, the BNLMS algorithm can enter the steady state in 4000 steps, in addition, the shrinkage coefficient converges faster (as compared with Fig. 3(b)), that is to say, the transient performance of the BNLMS algorithm is better than that of the CNLMS algorithm in this case. It deserves to be mentioned that, in Fig. 4(b) when the step-size is set as  $\mu = 0.1$ , the shrinkage coefficient  $\alpha$  converges to a larger value, which ensures a fast descent rate of the estimate bias, and provides an acceptable transient performance at the same time.

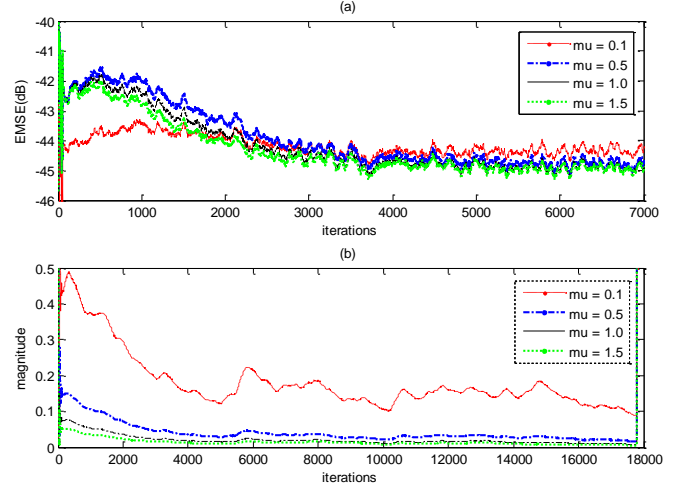


Fig. 4 Simulation Curve of BNLMS Algorithm under Low Signal

###### A.2 The simulation under high noise

In this simulation, we test the convergence performance of the NLMS, CNLMS and BNLMS algorithms under a high noise power condition, in which the signal power is fixed at 0.1, while the noise power is scaled to make  $\text{SNR} = -35\text{dB}$ , with the other settings similar as above. Fig. 5 and Fig. 6 show the simulation results in this case.

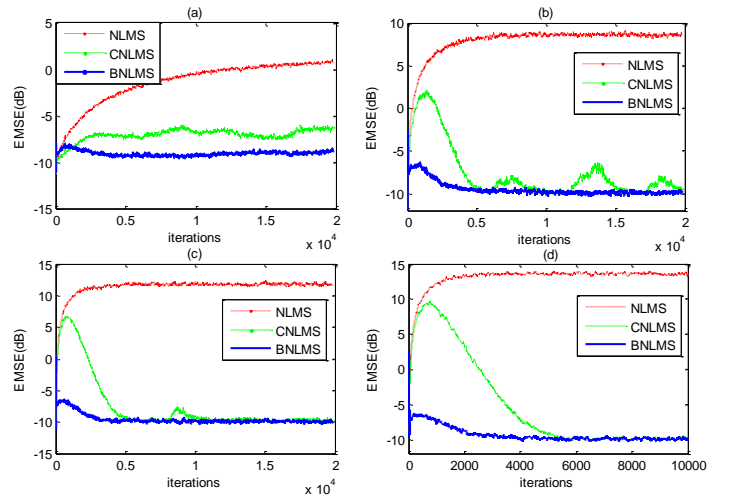


Fig. 5 The Simulation Curves under High Noise Condition

Fig.5(a)~Fig.5(d) are simulation curves when the step-size is set as 0.1,0.5,1.0,1.5 respectively. Similar conclusions are obtained in this case that the BNLMS algorithm has good transient and steady-state performance, outperforming the CNLMS algorithm under high noise power condition. Fig. 6 shows the convergence curves of the shrinkage coefficients accordingly, which takes a close insight into the convergence processes of the two biased NLMS algorithms. As is shown in Fig. 6, the shrinkage coefficient of the BNLMS algorithm always converges faster than that of the CNLMS algorithm, which ensures a better transient performance of the BNLMS algorithm.

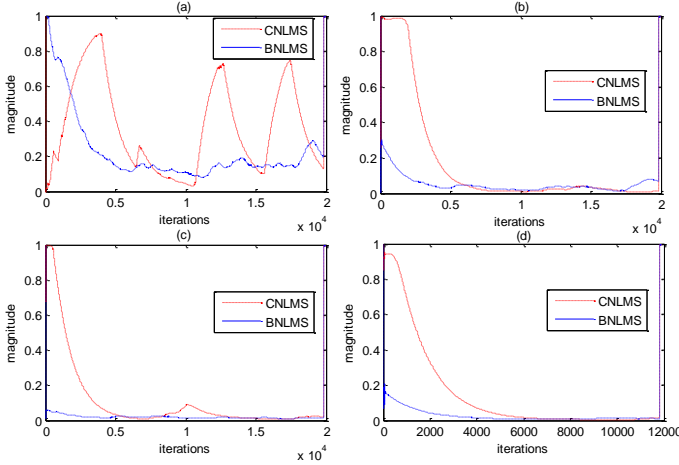


Fig.6 The Simulation Curves of Shrinkage Coefficient under High Noise Condition

### B. Tracking Performance Simulation

We are aimed to study the tracking performance of CNLMS and BNLMS algorithm under the situation when the SNR is unknown or variable in the region of  $(-40, -25)dB$ . In this section the influence of variable signal power and variable noise power on the algorithms are tested respectively. At first, we study the influence of variable signal power. The settings of the simulation is as follows: the noise power is set as 0.1, the variance of the impulse response is scaled to make  $SNR = -40dB$  initially, then adjusted to make  $SNR = -25dB$  at the  $20000^{th}$  iteration, and  $SNR = -35dB$  at the  $30000^{th}$  iteration; The step-size of NLMS, CNLMS and BNLMS algorithms is set as 0.5, 0.5, 0.8 respectively to achieve a similar steady-state error; The other settings remain the same as above. Fig. 7(a) shows the test results in this case. Fig. 7(b) shows the simulation curves under a variable noise power, in which the signal power is fixed at 0.1, while the noise power is scaled to make and adjust the variance of noise to make  $SNR = -40dB$  initially, and  $SNR = -25dB$  at the  $15000^{th}$  iteration, then  $SNR = -35dB$  at the  $35000^{th}$  iteration, with the step-size set as 0.5 for the testing algorithms.

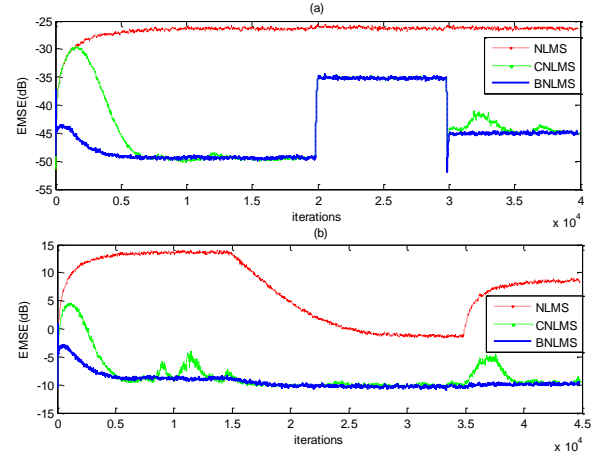


Fig.7 The Related Tracking Performance Comparison Of The Algorithms

It can be seen from Fig. 7 that, in low SNR region of  $(-40, -25)dB$ , regardless of variable signal power or variable noise power, the BNLMS algorithm can always track the impulse response very quickly, outperforming the CNLMS algorithm. Besides, as for the situation where the signal power is fixed, while the noise power varies, the BNLMS algorithm keeps at a low steady-state error, slightly changing as the noise power varies, while the rest performs the other way round, as is shown in Fig. 7(b), which also suggests that the BNLMS algorithm has definite robust performance to changing noise.

### 5. Performance Analysis

The BNLMS algorithm is analysed from the aspects of transient and steady-state performance in this section. First of all, we study the steady-state performance. For convenience of deduce, we rewrite the biased excess error:

$$\begin{aligned}\varepsilon_\alpha(n) &= (\mathbf{w}_o - \mathbf{w}_\alpha(n))^T \mathbf{u}(n) \\ &= (1-\alpha) \mathbf{w}_o^T \mathbf{u}(n) + \alpha \varepsilon(n)\end{aligned}\quad (26)$$

and the EMSE

$$\begin{aligned}J_{e,\alpha}(n) &= (1-\alpha)^2 \mathbf{w}_o^T \mathbf{R} \mathbf{w}_o + \alpha^2 J_{ex}(n) \\ &\quad + 2E\{\alpha(1-\alpha) \mathbf{w}_o^T \mathbf{u}(n) \varepsilon(n)\}\end{aligned}\quad (27)$$

At steady state, since  $\mathbf{w}(n)$  is an unbiased estimate of the impulse response,  $E\{\tilde{\mathbf{w}}(\infty)\} = \lim_{n \rightarrow \infty} E\{\tilde{\mathbf{w}}(n)\} = 0$ , and  $\mathbf{u}(n)$  and  $\tilde{\mathbf{w}}(n)$  are mutually independent, thus  $E\{\varepsilon(\infty)\} = 0$ , as a result, the steady-state error of biased NLMS algorithm is <sup>[3]</sup>

$$J_{e,\alpha}(\infty) = \alpha^2 J_{ex}(\infty) + (1-\alpha)^2 \mathbf{w}_o^T \mathbf{R} \mathbf{w}_o \quad (28)$$

That is to say, the steady-state error of biased NLMS algorithm is a complex of the steady-state error of NLMS algorithm and the signal power. When the SNR is high, the effect of unbiased estimate is better than that of the biased one, thus  $\alpha \rightarrow 1$ , its steady-state performance is similar to that of NLMS algorithm; When the SNR is low, the biased estimate

can weaken the influence of estimation error, thus  $\alpha \rightarrow 0$ , its steady state error is mainly determined by signal power.

Note that under low SNR scenario, the shrinkage coefficient of BNLMS algorithm will quickly converge to its steady-state value, which is much less than 1, so the steady state EMSE is

$$J_{e,x,BNLMS}(\infty) \approx \mathbf{w}_o^T \mathbf{R} \mathbf{w}_o \quad (29)$$

Namely, under low SNR situation, the steady-state error of BNLMS algorithm is only determined by the signal power. It is verified in the tracking performance simulation above that when the noise power is fixed at 0.1, while the signal power is scaled to make the SNR is -40dB, -25dB, -35dB respectively, the BNLMS algorithm converges to -50dB, -35dB, -45dB accordingly, as is shown in Fig.7(a); However the EMSE of the BNLMS algorithm keeps at -10dB when the signal power is fixed at 0.1, as is shown in Fig.7(b).

Moreover, under low SNR situation, the steady-state value of shrinkage coefficient can be seen as a unary function of the step-size, unaffected by variable SNR. When the step-size of BNLMS algorithm is fixed, the steady-state shrinkage coefficient is fixed. In other words, the steady-state performance of BNLMS algorithm is improved by adjusting the step-size, specifically a large step-size can decrease the steady-state value of the shrinkage coefficient, thus lower the steady-state error.

From the aspect of transient performance, it's not difficult to find, from the derivation process in section 2, that the BNLMS algorithm is coming out from the idea of variable step-size NLMS (VSSNLMS) algorithm. The shrinkage coefficient is derived following the optimal rule of maximizing descent velocity of MSD, thus it is actually a correction factor of step-size, and that's why the BNLMS algorithm has so fast rate of convergence. However, the BNLMS algorithm is different from the VSSNLMS algorithm in nature. The VSSNLMS algorithm provides an unbiased estimate, using the self-adaptation of the step-size to change the update process of the weight vector, in order to decrease the influence of the gradient noise on the steady-state performance of the algorithm, but when the proportion of the system noise is high in the desired signal, the VSSNLMS algorithm can't provide a good estimate; While in the BNLMS algorithm, the shrinkage coefficient doesn't affect the iterative process of the weight vector, instead it corrects the estimate after each iteration. The BNLMS algorithm provides a biased estimate to decrease the estimate noise so as to lower the steady-state error, hence it performs well under low SNR condition.

## 6. Conclusion

As for low SNR condition, the biased estimation can decrease the estimation noise effectively to improve the steady-state performance of filter. Among the existing biased NLMS algorithms, the CNLMS algorithm is one of the most simple and efficient version, yet it suffers from a low convergence rate. A new biased NLMS algorithm, namely the BNLMS algorithm, is proposed in this paper, following the optimal rule of maximizing descent velocity of MSD, which greatly improves the transient performance of biased NLMS algorithm. Simulations show that as for low SNR in the region of  $(-40, -25)$ dB, the BNLMS algorithm ensures a good steady-state performance as well as an excellent transient performance, outperforming the CNLMS algorithm. In the performance analysis, we analyse the BNLMS algorithm from the aspects of transient and steady-state performance. The analysis results indicate: under low SNR condition, the steady-state error of the BNLMS algorithm is only related to signal power, unaffected by the noise power; The steady-state value of shrinkage coefficient in BNLMS algorithm is a unary function of step-size, and the steady-state performance of the BNLMS algorithm can be improved by adjusting the step-size; The introduction of the idea of VSSNLMS algorithm ensures the fast convergence rate of the BNLMS algorithm, but the two algorithms are distinctive in that the VSSNLMS algorithm can decrease the influence of the gradient noise on the steady-state performance, while the BNLMS can weaken the influence of large system noise on the steady-state performance.

## References

- [1] S. Haykin, *Adaptive Filter Theory*. 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [2] S. Li, R. C. de Lamare, et al. Adaptive frequency-domain biased estimation algorithms with automatic adjustment of shrinkage factors. *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011: 4268-4271.
- [3] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, et al. Adaptively biasing the weights of adaptive filters. *IEEE Trans.*, 2010, 58(7): 3890-3895.
- [4] L. A. Azpicueta-Ruiz, M. Lázaro-Gredilla, et al. A block-based approach to adaptively bias the weights of adaptive filters. *IEEE Int. Workshop Machine Learning for Signal Process*, 2011.
- [5] A. Mader, H. Puder, et al. Step-size control for acoustic echo cancellation filters- an overview. *Signal Processing*, 2000, 80(9): 1697-1719.
- [6] N. Li, Y. Zhang, et al. A new variable step-size NLMS algorithm designed for applications with exponential decay impulse responses. *Signal Processing*, 2008, 88(9): 2346-2349.
- [7] T. Aboulnasr, K. Mayyas. A robust variable step-size LMS-type algorithm: analysis and simulations, *IEEE Trans.*, 1997, 45(3): 631-639.