

REVIEW

Open Access



# An overview on optimized NLMS algorithms for acoustic echo cancellation

Constantin Paleologu<sup>1\*</sup>, Silviu Ciochină<sup>1</sup>, Jacob Benesty<sup>2</sup> and Steven L. Grant<sup>3</sup>

## Abstract

Acoustic echo cancellation represents one of the most challenging system identification problems. The most used adaptive filter in this application is the popular normalized least mean square (NLMS) algorithm, which has to address the classical compromise between fast convergence/tracking and low misadjustment. In order to meet these conflicting requirements, the step-size of this algorithm needs to be controlled. Inspired by the pioneering work of Prof. E. Hänsler and his collaborators on this fundamental topic, we present in this paper several solutions to control the adaptation of the NLMS adaptive filter. The developed algorithms are “non-parametric” in nature, i.e., they do not require any additional features to control their behavior. Simulation results indicate the good performance of the proposed solutions and support the practical applicability of these algorithms.

**Keywords:** Acoustic echo cancellation, Adaptive filters, Normalized least mean square (NLMS) algorithm, Step-size control, Variable step-size NLMS, Variable regularized NLMS

## 1 Review

### 1.1 Introduction

Hands-free audio terminals are required in many popular applications, such as mobile telephony and teleconferencing systems. An important issue that has to be addressed when dealing with such devices is the acoustic coupling between the loudspeaker and the microphone. Due to this coupling, the microphone of the device captures a signal coming from its own loudspeaker, known as the acoustic echo. This phenomenon is influenced by the environment’s characteristics, and it can be very disturbing for the users. For example, in a telephone conversation, the user could hear a replica of her/his own voice. Consequently, in order to enhance the overall quality of the communication, there is a need to cancel the unwanted acoustic echo.

In this context, acoustic echo cancellation (AEC) provides one of the best solutions to the control of acoustic echoes generated by hands-free audio terminals. The basic issue in AEC is then to estimate the impulse response between the loudspeaker and the microphone of the device. The most reliable solution to this problem is

the use of an adaptive filter that generates at its output a replica of the echo, which is further subtracted from the microphone signal [1–9]. In other words, the adaptive filter has to model an unknown system (i.e., the acoustic echo path between the loudspeaker and the microphone), like in a “system identification” problem [10–12].

Despite the straightforward formulation of the problem, there are several specific features of AEC, which represent a challenge for any adaptive algorithm. First, the acoustic echo paths have excessive lengths in time (up to hundreds of milliseconds), due to the slow speed of sound in the air, together with multiple reflections caused by the environment; consequently, long length adaptive filters are required (hundreds or even thousands of coefficients), thus influencing the convergence rate of the algorithm. Also, the acoustic echo paths are time-variant systems, depending on temperature, pressure, humidity, and movement of objects or bodies; hence, good tracking capabilities are required for the echo canceller. Second, the echo signal is combined with the near-end signal; ideally, the adaptive filter should separate this mixture and provide an estimate of the echo at its output and an estimate of the near-end from the error signal (from this point of view, the adaptive filter works as in an “interference cancelling” configuration [10–12]). This is not an easy

\*Correspondence: pale@comm.pub.ro

<sup>1</sup>Department of Telecommunications, University Politehnica of Bucharest, 1-3 Iuliu Maniu Blvd., 061071 Bucharest, Romania

Full list of author information is available at the end of the article

task, since the near-end signal can contain both the background noise and the near-end speech; the background noise can be non-stationary and strong (it is also amplified by the microphone of the hands-free device), while the near-end speech acts like a large level disturbance. Moreover, the input of the adaptive filter (i.e., the far-end signal) is mainly speech, which is a non-stationary and highly correlated signal that can influence the overall performance of the adaptive algorithm.

In addition, the double-talk case (i.e., the talkers on both sides speak simultaneously) is perhaps the most challenging situation in AEC. The behavior of the adaptive filter can be seriously affected in this case, up to divergence. For this reason, the echo canceller is usually equipped with a double-talk detector (DTD), in order to slow down or completely halt the adaptation process during double-talk periods [6, 7]. Nevertheless, there is some inherent delay in the decision of any DTD; during this small period, a few undetected large amplitude samples can perturb the echo path estimate considerably. Consequently, it is highly desirable to improve the robustness of the adaptive algorithm in order to handle a certain amount of double-talk without diverging.

Many adaptive algorithms were proposed in the context of AEC [1–9, 13], but the workhorse remains the normalized least mean square (NLMS) algorithm [10–12]. The main reasons behind this popularity are its moderate computational complexity, together with its good numerical stability. The performance of the NLMS algorithm is influenced by two important parameters, i.e., the normalized step-size and regularization terms [1, 8, 11]. The first one reflects a trade-off between convergence rate and misadjustment of the algorithm. The second parameter is essential in all ill-posed and ill-conditioned problems such as in adaptive filters; it depends on the signal-to-noise ratio (SNR) of the system [14]. Both these parameters can be controlled (i.e., making them time dependent) in order to address the conflicting requirement of fast convergence and low misadjustment. This was the main motivation behind the development of variable step-size (VSS) and variable regularized (VR) versions of the NLMS algorithm, e.g., [13, 15–25]. Even if they focus on the optimization of different parameters, the VSS-NLMS and VR-NLMS algorithms are basically equivalent in terms of their purpose [1, 19]. In general, most of them require the tuning of some additional parameters that are difficult to control in practice. For real-world AEC applications, it is highly desirable to design “non-parametric” algorithms, which can operate without requiring additional features related to the acoustic environment (e.g., system change detector).

In this context, the contributions of Prof. E. Hänsler and his collaborators represent real milestones in the field. For example, in [1], Hänsler and Schmidt present a

comprehensive and insightful review of the methods and algorithms used for acoustic echo and noise control. In their work, a special interest is given to the performance analysis of the NLMS algorithm (e.g., see Chapters 7 and 13 from [1]), in terms of developing optimal expressions for its control parameters, i.e., the normalized step-size and regularization term. In Section 1.2 of this paper, we summarize their main findings related to the control of the NLMS algorithm. Also, in Section 1.3, we present another benchmark solution, i.e., the non-parametric variable step-size NLMS (NPVSS-NLMS) algorithm [19]. Motivated and inspired by the work of Hänsler and Schmidt [1] (summarized in Section 1.2), we extend their findings in the framework of a state variable model (similar to Kalman filtering) [26]. The joint-optimized NLMS (JO-NLMS) algorithm developed in Section 1.4 brings together three main elements: a time-variant system model, an optimization criterion based on the minimization of the system misalignment, and an iterative procedure for adjusting the system model parameter. Consequently, it achieves a proper compromise between the performance criteria, i.e., fast convergence/tracking and low misadjustment, without requiring any additional features to control its behavior (like stability thresholds or system change detector). Simulations performed in Section 1.5 support the theoretical findings and indicate the good performance of the presented algorithms. Finally, Section 2 concludes this work and outlines several perspectives.

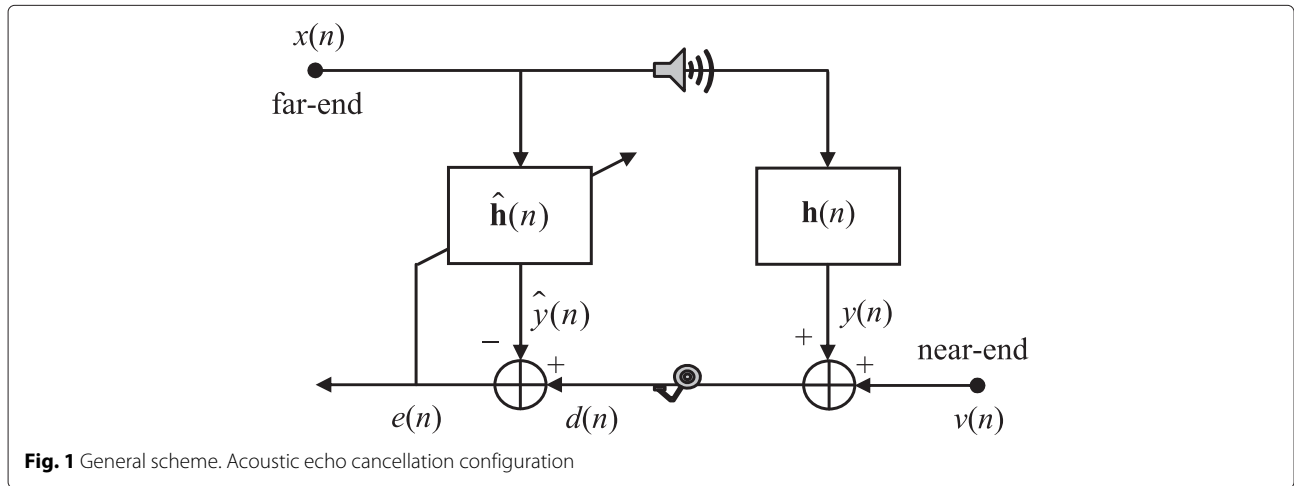
## 1.2 Control of the NLMS algorithm

Let us consider the framework of a system identification problem (as shown in Fig. 1), like in AEC [1–9]. The far-end (or loudspeaker) signal,  $x(n)$ , goes through the echo path,  $\mathbf{h}(n)$ , providing the echo signal,  $y(n)$ , where  $n$  is the time index. This signal is added to the near-end signal,  $v(n)$  (which can contain both the background noise and the near-end speech), resulting the microphone signal,  $d(n)$ . The adaptive filter, defined by the vector  $\hat{\mathbf{h}}(n)$ , aims to produce at its output an estimate of the echo,  $\hat{y}(n)$ , while the error signal,  $e(n)$ , should contain an estimate of the near-end signal.

Summarizing, the main goal of this application is to model an unknown system using an adaptive filter, both driven by the same zero-mean input signal,  $x(n)$ . These two systems are assumed to be finite impulse response (FIR) filters of length  $L$ , defined by the real-valued vectors:

$$\begin{aligned}\mathbf{h}(n) &= [h_0(n) \ h_1(n) \ \cdots \ h_{L-1}(n)]^T, \\ \hat{\mathbf{h}}(n) &= [\hat{h}_0(n) \ \hat{h}_1(n) \ \cdots \ \hat{h}_{L-1}(n)]^T,\end{aligned}$$

where superscript  $T$  denotes transposition. The desired (or microphone) signal for the adaptive filter is



$$d(n) = \mathbf{x}^T(n)\mathbf{h}(n) + v(n) \tag{1}$$

$$= y(n) + v(n),$$

where

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]^T$$

is a real-valued vector containing the  $L$  most recent time samples of the input signal,  $x(n)$ , and  $v(n)$  (i.e., the near-end signal) plays the role of the system noise (assumed to be quasi-stationary, zero mean, and independent of  $x(n)$ ) that corrupts the output of the unknown system.

Using the previous notation, we may define the a priori and a posteriori error signals as

$$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n-1) \tag{2}$$

$$= \mathbf{x}^T(n) [\mathbf{h}(n) - \hat{\mathbf{h}}(n-1)] + v(n),$$

$$\varepsilon(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n) \tag{3}$$

$$= \mathbf{x}^T(n) [\mathbf{h}(n) - \hat{\mathbf{h}}(n)] + v(n),$$

where the vectors  $\hat{\mathbf{h}}(n-1)$  and  $\hat{\mathbf{h}}(n)$  contain the adaptive filter coefficients at time  $n-1$  and  $n$ , respectively. The update equation for NLMS-type algorithms is

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu(n)\mathbf{x}(n)e(n), \tag{4}$$

where  $\mu(n)$  is a positive factor known as the step-size, which governs the stability, the convergence rate, and the misadjustment of the algorithm. A reasonable way to derive  $\mu(n)$ , taking into account the stability conditions, is to cancel the a posteriori error signal [27]. Replacing (4) in (3) with the requirement  $\varepsilon(n) = 0$ , it results that

$$\varepsilon(n) = e(n) [1 - \mu(n)\mathbf{x}^T(n)\mathbf{x}(n)] = 0 \tag{5}$$

and assuming that  $e(n) \neq 0$ , we find

$$\mu(n) = \frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)}. \tag{6}$$

We should note that the above procedure makes sense in the absence of noise [i.e.,  $v(n) = 0$ ], where the condition  $\varepsilon(n) = 0$  implies that  $\mathbf{x}^T(n) [\mathbf{h}(n) - \hat{\mathbf{h}}(n)] = 0$ . Finding the parameter  $\mu(n)$  in the presence of noise will introduce noise in  $\hat{\mathbf{h}}(n)$ , since the condition  $\varepsilon(n) = 0$  leads to  $\mathbf{x}^T(n) [\mathbf{h}(n) - \hat{\mathbf{h}}(n)] = -v(n) \neq 0$ . In fact, we would like to have  $\mathbf{x}^T(n) [\mathbf{h}(n) - \hat{\mathbf{h}}(n)] = 0$ , which implies that  $\varepsilon(n) = v(n)$ .

In practice, a positive constant  $\alpha$  (with  $0 < \alpha < 2$ ), known as the normalized step-size, multiplies (6) to achieve a proper compromise between the convergence rate and the misadjustment [10–12]; also, a positive constant  $\delta$ , known as the regularization parameter, is added to the denominator of (6) in order to make the adaptive filter work well in the presence of noise. Consequently, the well-known update equation of the NLMS algorithm becomes

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\alpha\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta}. \tag{7}$$

### 1.2.1 Performance analysis

Both the control parameters, i.e.,  $\alpha$  and  $\delta$ , highly influence the overall performance of the NLMS algorithm. An insightful analysis of their influence was developed by Hänsler and Schmidt in [1]. To begin, let us define the a posteriori misalignment (also known as the system mismatch [1]) as

$$\mathbf{m}(n) = \mathbf{h}(n) - \hat{\mathbf{h}}(n). \tag{8}$$

Assuming that the unknown system is time-invariant, i.e.,

$$\mathbf{h}(n) = \mathbf{h}(n-1), \tag{9}$$

the updated equation (7) of the NLMS algorithm can be used, together with (8), in order to derive an update in

terms of the a posteriori misalignment. Consequently, it results

$$\mathbf{m}(n) = \mathbf{m}(n-1) - \frac{\alpha \mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta}. \quad (10)$$

Taking the  $\ell_2$  norm in (10), we obtain

$$\begin{aligned} \|\mathbf{m}(n)\|_2^2 &= \|\mathbf{m}(n-1)\|_2^2 - 2 \frac{\alpha \mathbf{m}^T(n-1)\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta} \\ &\quad + \frac{\alpha^2 \|\mathbf{x}(n)\|_2^2 e^2(n)}{[\|\mathbf{x}(n)\|_2^2 + \delta]^2}. \end{aligned} \quad (11)$$

Based on (2) and (9), the a priori error signal can be expressed as

$$e(n) = \mathbf{x}^T(n)\mathbf{m}(n-1) + v(n), \quad (12)$$

so that, using (12) in (11), it results

$$\begin{aligned} \|\mathbf{m}(n)\|_2^2 &= \|\mathbf{m}(n-1)\|_2^2 \\ &\quad - 2 \frac{\alpha \mathbf{m}^T(n-1)\mathbf{x}(n) [\mathbf{x}^T(n)\mathbf{m}(n-1) + v(n)]}{\|\mathbf{x}(n)\|_2^2 + \delta} \\ &\quad + \frac{\alpha^2 \|\mathbf{x}(n)\|_2^2 [\mathbf{x}^T(n)\mathbf{m}(n-1) + v(n)]^2}{[\|\mathbf{x}(n)\|_2^2 + \delta]^2}. \end{aligned} \quad (13)$$

Next, taking mathematical expectation on both sides of (13) and removing the uncorrelated products (since  $x(n)$  and  $v(n)$  are assumed to be independent and zero mean), we obtain

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] &= E[\|\mathbf{m}(n-1)\|_2^2] \\ &\quad - 2\alpha E\left[\frac{\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1)}{\|\mathbf{x}(n)\|_2^2 + \delta}\right] \\ &\quad + \alpha^2 E\left\{\frac{\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1) \|\mathbf{x}(n)\|_2^2}{[\|\mathbf{x}(n)\|_2^2 + \delta]^2}\right\} \\ &\quad + \alpha^2 E\left\{\frac{v^2(n) \|\mathbf{x}(n)\|_2^2}{[\|\mathbf{x}(n)\|_2^2 + \delta]^2}\right\}. \end{aligned} \quad (14)$$

It is clear that  $E[\|\mathbf{x}(n)\|_2^2] = L\sigma_x^2$ , where  $\sigma_x^2 = E[x^2(n)]$  is the variance of the input signal. For large values of  $L$  (i.e.,  $L \gg 1$ ), it holds that  $\|\mathbf{x}(n)\|_2^2 \approx L\sigma_x^2$  [1, 19]. Consequently,

$$\frac{1}{\|\mathbf{x}(n)\|_2^2 + \delta} \approx \frac{1}{L\sigma_x^2 + \delta}, \quad (15)$$

so that, for a large value of  $L$  and a certain stationarity degree of the input signal, we can treat this term as

a deterministic quantity at this point [1]. Under these circumstances, (14) becomes

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] &= E[\|\mathbf{m}(n-1)\|_2^2] \\ &\quad - \frac{2\alpha}{L\sigma_x^2 + \delta} E\left[\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1)\right] \\ &\quad + \frac{\alpha^2 L\sigma_x^2}{(L\sigma_x^2 + \delta)^2} E\left[\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1)\right] \\ &\quad + \frac{\alpha^2 L\sigma_x^2}{(L\sigma_x^2 + \delta)^2} \sigma_v^2, \end{aligned} \quad (16)$$

where  $\sigma_v^2 = E[v^2(n)]$  is the variance of the system noise. Next, it can be assumed that the input vector,  $\mathbf{x}(n)$ , and the a posteriori misalignment vector,  $\mathbf{m}(n-1)$ , are statistically independent, and  $x(n)$  is white. In this case,

$$E\left[\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1)\right] = \sigma_x^2 E[\|\mathbf{m}(n-1)\|_2^2]. \quad (17)$$

Summarizing, (16) can be rewritten as

$$E[\|\mathbf{m}(n)\|_2^2] = A(\alpha, \delta, L, \sigma_x^2) E[\|\mathbf{m}(n-1)\|_2^2] + B(\alpha, \delta, L, \sigma_x^2) \sigma_v^2, \quad (18)$$

where

$$A(\alpha, \delta, L, \sigma_x^2) = 1 - \frac{2\alpha\sigma_x^2}{L\sigma_x^2 + \delta} + \frac{\alpha^2 L\sigma_x^4}{(L\sigma_x^2 + \delta)^2}, \quad (19)$$

$$B(\alpha, \delta, L, \sigma_x^2) = \frac{\alpha^2 L\sigma_x^2}{(L\sigma_x^2 + \delta)^2}, \quad (20)$$

represent the so-called *contraction* and *expansion* parameters, respectively [1].

Clearly, the contraction parameter,  $A(\alpha, \delta, L, \sigma_x^2)$ , should be always smaller than 1, which is certainly fulfilled for  $0 < \alpha < 2$  and  $\delta \geq 0$ . The expansion parameter,  $B(\alpha, \delta, L, \sigma_x^2)$ , is related to the influence of the system noise, since it multiplies  $\sigma_v^2$ . Both terms depend on the control parameters,  $\alpha$  and  $\delta$ , as well as on the filter length,  $L$ , and the input signal power,  $\sigma_x^2$ . However, a compromise should be made when setting the values of the control parameters. For example, if the influence of the system noise should be eliminated completely, i.e.,  $B(\alpha, \delta, L, \sigma_x^2) = 0$ , we should set  $\alpha = 0$  or  $\delta = \infty$ , which, on the other hand, leads to  $A(\alpha, \delta, L, \sigma_x^2) = 1$ , i.e., the filter will not be updated. The fastest convergence (FC) mode is achieved when  $A(\alpha, \delta, L, \sigma_x^2)$  reaches its minimum (e.g., for  $\alpha = 1$  and  $\delta = 0$ ), which, unfortunately, increases the misadjustment (in terms of the influence



of the system noise). For example, taking the normalized step-size as the reference parameter and evaluating

$$\left. \frac{\partial A(\alpha, \delta, L, \sigma_x^2)}{\partial \alpha} \right|_{\alpha=\alpha_{FC}} = 0, \quad (21)$$

it results

$$\alpha_{FC} = 1 + \frac{\delta}{L\sigma_x^2}. \quad (22)$$

Neglecting the regularization constant (i.e.,  $\delta \approx 0$ ), the fastest convergence mode is achieved for  $\alpha \approx 1$ , which is a well-known result [1, 11, 12]. Also, the stability condition can be found by imposing  $|A(\alpha, \delta, L, \sigma_x^2)| < 1$ , which leads to

$$0 < \alpha_{\text{stable}} < 2 \left( 1 + \frac{\delta}{L\sigma_x^2} \right) = 2\alpha_{FC}. \quad (23)$$

Again, taking  $\delta = 0$  in (23), the standard stability condition of the NLMS algorithm results, i.e.,  $0 < \alpha < 2$ . On the other hand, the lowest misadjustment (LM) is obtained when the term from (20) reaches its minimum. Also, taking the normalized step-size as the reference parameter and evaluating

$$\left. \frac{\partial B(\alpha, \delta, L, \sigma_x^2)}{\partial \alpha} \right|_{\alpha=\alpha_{LM}} = 0, \quad (24)$$

the lowest misadjustment mode requires

$$\alpha_{LM} = 0, \quad (25)$$

which is also a well-known result [1, 11, 12]; unfortunately, the filter is not updated in this case.

Summarizing, the convergence rate of the algorithm is not influenced by the level of the system noise, but the misadjustment increases when the system noise increases. More importantly, it can be noticed that the expansion term from (20) always increases when  $\alpha$  increases; this concludes the fact that a higher value of the normalized step-size increases the misadjustment. Nevertheless, the ideal requirements of the algorithm are for both fast convergence and low misadjustment. It is clear that (22) and (25) “push” the normalized step-size in opposite directions. This aspect represents the motivation behind the VSS approaches, i.e., the normalized step-size needs to be controlled in order to meet these conflicting requirements. The regularization constant also influences the performance of the algorithm, but in a “milder” way. It can be noticed that the contraction term from (19) always decreases when the regularization constant increases, while the expansion term from (20) always increases when the regularization constant decreases.

### 1.2.2 Optimal choice of the control parameters

Motivated by these findings, Hänsler and Schmidt proposed in [1] (Chapter 13) an optimal choice for the control parameters of the NLMS algorithm. First, the

non-regularized version of the NLMS algorithm is considered (also imposing that the normalized step-size is time dependent), with the update

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\alpha(n)\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2}. \quad (26)$$

Next, developing in (12), it results

$$\mathbf{x}^T(n)\mathbf{m}(n-1) = e(n) - v(n) \triangleq e_u(n), \quad (27)$$

where  $e_u(n)$  denotes the so-called undistorted error signal [1], i.e., the part of the error that is not affected by the system noise. Using this notation, (11) can be rewritten as

$$\|\mathbf{m}(n)\|_2^2 = \|\mathbf{m}(n-1)\|_2^2 - 2 \frac{\alpha(n)e_u(n)e(n)}{\|\mathbf{x}(n)\|_2^2} + \frac{\alpha^2(n)e^2(n)}{\|\mathbf{x}(n)\|_2^2}, \quad (28)$$

which implies that

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] &= E[\|\mathbf{m}(n-1)\|_2^2] - 2\alpha(n)E\left[\frac{e_u(n)e(n)}{\|\mathbf{x}(n)\|_2^2}\right] \\ &\quad + \alpha^2(n)E\left[\frac{e^2(n)}{\|\mathbf{x}(n)\|_2^2}\right]. \end{aligned} \quad (29)$$

A natural optimization criterion to follow in any system identification problem is the minimization of system misalignment. Consequently, imposing the condition:

$$\left. \frac{\partial E[\|\mathbf{m}(n)\|_2^2]}{\partial \alpha(n)} \right|_{\alpha(n)=\alpha_{\text{opt}}(n)} = 0 \quad (30)$$

and assuming that the normalized step-sizes at different time instants are uncorrelated, the optimal normalized step-size results as

$$\alpha_{\text{opt}}(n) = \frac{E\left[\frac{e_u(n)e(n)}{\|\mathbf{x}(n)\|_2^2}\right]}{E\left[\frac{e^2(n)}{\|\mathbf{x}(n)\|_2^2}\right]}. \quad (31)$$

For large values of  $L$  (i.e.,  $L \gg 1$ ), the assumption  $\|\mathbf{x}(n)\|_2^2 \approx L\sigma_x^2$  is valid [1, 19]. Also, since the input signal,  $\mathbf{x}(n)$ , and the system noise,  $v(n)$ , are uncorrelated, the undistorted error signal,  $e_u(n)$ , is also uncorrelated with the system noise. Therefore, (31) simplifies to

$$\alpha_{\text{opt}}(n) = \frac{E[e_u^2(n)]}{E[e^2(n)]}. \quad (32)$$

In the absence of the system noise [i.e.,  $v(n) = 0$ ], the a priori error signal,  $e(n)$ , equals the undistorted error signal,  $e_u(n)$ , so that and the optimal normalized step-size is equal to 1, which justifies the discussion related to (6). In the presence of the system noise, when the adaptive filter starts to converge, the power of the undistorted error

signal,  $e_u(n)$ , decreases and, consequently, the normalized step-size decreases, thus leading to low misadjustment.

Unfortunately, the undistorted error signal,  $e_u(n)$ , is not available in practice. In order to overcome this issue, several solutions were proposed in [1]. For example, assuming that the excitation,  $x(n)$ , is white and considering that the input vector,  $\mathbf{x}(n)$ , and the a posteriori misalignment vector,  $\mathbf{m}(n - 1)$ , are statistically independent, (32) can be developed based on (17) as

$$\begin{aligned} \alpha_{\text{opt}}(n) &= \frac{E[e_u^2(n)]}{E[e^2(n)]} \\ &= \frac{E[\mathbf{m}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n-1)]}{E[e^2(n)]} \\ &= \frac{\sigma_x^2 E[\|\mathbf{m}(n-1)\|_2^2]}{E[e^2(n)]}. \end{aligned} \quad (33)$$

Now the problem is reduced to the estimation of  $E[\|\mathbf{m}(n-1)\|_2^2]$ . A solution to estimate this term is based on the “delay and extrapolation” approach [1]. In other words, if an additional artificial delay is introduced into the unknown system, this delay is also modeled by the adaptive filter. Thus, utilizing the property of adaptive algorithms to spread the filter misalignment evenly over all coefficients, the known part of the (true) system misalignment vector can be extrapolated, thus resulting

$$\|\mathbf{m}(n-1)\|_2^2 \approx \frac{L}{L_D} \sum_{l=0}^{L_D-1} \widehat{h}_l^2(n), \quad (34)$$

where  $L_D$  denotes the number of coefficients corresponding to the artificial delay. However, this method may freeze the adaptation when the unknown system changes, which would require an additional system change detector [1].

The second control parameter of the NLMS algorithm is the regularization term,  $\delta$ . Using a similar approach as before, the only-regularized version of the NLMS algorithm is considered (also imposing that the regularization parameter is time dependent), with the update

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta(n)}. \quad (35)$$

In this case, (11) can be rewritten as

$$\begin{aligned} \|\mathbf{m}(n)\|_2^2 &= \|\mathbf{m}(n-1)\|_2^2 - 2 \frac{e_u(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta(n)} \\ &\quad + \frac{e^2(n) \|\mathbf{x}(n)\|_2^2}{[\|\mathbf{x}(n)\|_2^2 + \delta(n)]^2}, \end{aligned} \quad (36)$$

so that,

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] &= E[\|\mathbf{m}(n-1)\|_2^2] - 2E\left[\frac{e_u(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta(n)}\right] \\ &\quad + E\left\{\frac{e^2(n) \|\mathbf{x}(n)\|_2^2}{[\|\mathbf{x}(n)\|_2^2 + \delta(n)]^2}\right\}. \end{aligned} \quad (37)$$

Following the same criterion, i.e., minimization of system misalignment, it can be imposed that the condition

$$\left. \frac{\partial E[\|\mathbf{m}(n)\|_2^2]}{\partial \delta(n)} \right|_{\delta(n)=\delta_{\text{opt}}(n)} = 0. \quad (38)$$

Under similar considerations and assumptions as in the case of  $\alpha_{\text{opt}}(n)$ , (38) leads to the optimal regularization parameter, which can be further developed as

$$\begin{aligned} \delta_{\text{opt}}(n) &= \frac{\{E[e^2(n)] - E[e_u^2(n)]\} L \sigma_x^2}{E[e_u^2(n)]} \\ &= \frac{E[v^2(n)] L \sigma_x^2}{E[e_u^2(n)]} \\ &= \frac{L \sigma_v^2}{E[\|\mathbf{m}(n-1)\|_2^2]}. \end{aligned} \quad (39)$$

The denominator of (39) can be evaluated based on (34). Also, another important parameter to be found is the noise power,  $\sigma_v^2$ . There are different methods for estimating this parameter; for example, in echo cancellation, it can be estimated during silences of the near-end talker [19]. Also, other practical methods to estimate  $\sigma_v^2$  in AEC can be found in [28, 29] (which are briefly detailed in the end of Section 1.3). However, we should note that different other estimators can be used for the noise power; the analysis of their influence on the algorithms’ performance is beyond the scope of this paper.

Concluding, both control methods proposed by Hänsler and Schmidt in [1] [i.e.,  $\alpha_{\text{opt}}(n)$  and  $\delta_{\text{opt}}(n)$ ] are theoretically equivalent and represent valuable benchmarks in the field of VSS/VR-NLMS algorithms. However, in practice, their implementations are usually different. In most cases, the control of the normalized step-size is preferred, mainly due to the limited dynamic range of its values; on the other hand, the regularization control usually requires an upper bound (to avoid overflow in case of very large values).

### 1.3 NPVSS-NLMS algorithm

In the previous section, the optimization criterion used for adjusting the control parameters was the minimization of the system misalignment. However, in a system identification setup like AEC (as shown in Fig. 1), this is equivalent to recover the system noise from the error of the adaptive filter [1].

Consequently, getting back to the discussion related to (2)–(6), the step-size parameter,  $\mu(n)$  (which is deterministic in nature), can be found by imposing the condition [19]

$$E[\varepsilon^2(n)] = \sigma_v^2. \quad (40)$$

Following this requirement, we rewrite (5) as

$$\varepsilon(n) = e(n) \left[ 1 - \mu(n)\mathbf{x}^T(n)\mathbf{x}(n) \right] = v(n). \quad (41)$$

Squaring the previous equation, then taking mathematical expectation on both sides, and using the approximation  $\mathbf{x}^T(n)\mathbf{x}(n) \approx LE[x^2(n)] = L\sigma_x^2$  (which is valid for  $L \gg 1$ ), it results

$$\left[ 1 - \mu(n)L\sigma_x^2 \right]^2 \sigma_e^2(n) = \sigma_v^2, \quad (42)$$

where  $\sigma_e^2(n) = E[e^2(n)]$  is the power of the error signal. Thus, developing (42), we obtain the quadratic equation

$$\mu^2(n) - \frac{2}{L\sigma_x^2}\mu(n) + \frac{1}{(L\sigma_x^2)^2} \left[ 1 - \frac{\sigma_v^2}{\sigma_e^2(n)} \right] = 0, \quad (43)$$

for which the obvious solution is (also using  $L\sigma_x^2 \approx \mathbf{x}^T(n)\mathbf{x}(n)$ )

$$\begin{aligned} \mu_{\text{NPVSS}}(n) &= \frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)} \left[ 1 - \frac{\sigma_v}{\sigma_e(n)} \right] \\ &= \frac{\alpha_{\text{NPVSS}}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}, \end{aligned} \quad (44)$$

where

$$\alpha_{\text{NPVSS}}(n) = 1 - \frac{\sigma_v}{\sigma_e(n)} \quad (45)$$

is the variable normalized step-size. Therefore, the update of the non-parametric variable step-size NLMS (NPVSS-NLMS) algorithm [19] is

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{\text{NPVSS}}(n)\mathbf{x}(n)e(n). \quad (46)$$

Let us examine the behavior of the algorithm in terms of its normalized step-size. Looking at (44), it is obvious that before the algorithm converges,  $\sigma_e(n)$  is large compared to  $\sigma_v$  and, consequently,  $\alpha_{\text{NPVSS}}(n) \approx 1$ . When the algorithm has converged to the true solution,  $\sigma_e(n) \approx \sigma_v$  and  $\alpha_{\text{NPVSS}}(n) \approx 0$ . This is the desired behavior for the adaptive algorithm, leading to both fast convergence and low misadjustment.

We can compare (45) to the optimal step-size parameter from (32), which results in

$$\alpha_{\text{opt}}(n) = \alpha_{\text{NPVSS}}(n) \left[ 1 + \frac{\sigma_v}{\sigma_e(n)} \right]. \quad (47)$$

It is clear that  $\alpha_{\text{opt}}(n)$  is larger than  $\alpha_{\text{NPVSS}}(n)$  by a factor between 1 and 2, but the two variable step-sizes have the same effect for good convergence and low misadjustment.

In order to analyze the convergence of the misalignment for the NPVSS-NLMS algorithm, we suppose that the system is stationary (as in (9)). Using the a posteriori

misalignment vector defined in (8), the update equation of the algorithm (46) can be rewritten in terms of the misalignment as

$$\mathbf{m}(n) = \mathbf{m}(n-1) - \mu_{\text{NPVSS}}(n)\mathbf{x}(n)e(n). \quad (48)$$

Taking the  $l_2$  norm in (48), then mathematical expectation on both sides, and assuming that

$$E[v(n)\mathbf{x}^T(n)\mathbf{m}(n-1)] = 0, \quad (49)$$

which is true if  $v(n)$  is white, we obtain

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] - E[\|\mathbf{m}(n-1)\|_2^2] &= -\mu_{\text{NPVSS}}(n) [\sigma_e(n) \\ &\quad - \sigma_v] [\sigma_e(n) + 2\sigma_v] \leq 0. \end{aligned} \quad (50)$$

The previous expression proves that the length of the misalignment vector for the NPVSS-NLMS algorithm is non-increasing, which implies that

$$\lim_{n \rightarrow \infty} \sigma_e^2(n) = \sigma_v^2. \quad (51)$$

It should be noticed that the previous relation does not imply that  $E[\|\mathbf{m}(\infty)\|_2^2] = 0$ . However, under the independence assumption, we can show the equivalence. Indeed, from (12), it can be shown that

$$E[e^2(n)] = \sigma_v^2 + \text{tr}[\mathbf{R}\mathbf{K}(n-1)] \quad (52)$$

if  $\mathbf{x}(n)$  are independent (i.e., the white input assumption), where  $\text{tr}(\cdot)$  is the trace of a matrix,  $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ , and  $\mathbf{K}(n-1) = E[\mathbf{m}(n-1)\mathbf{m}^T(n-1)]$ . Taking (51) into account, (52) becomes

$$\text{tr}[\mathbf{R}\mathbf{K}(\infty)] = 0. \quad (53)$$

Assuming that  $\mathbf{R} > 0$  (i.e.,  $\mathbf{R}$  is a positive definite matrix), it results that  $\mathbf{K}(\infty) = \mathbf{0}$  and, consequently,

$$E[\|\mathbf{m}(\infty)\|_2^2] = 0. \quad (54)$$

Finally, some practical considerations have to be stated. First, in order that the algorithm behaves properly, a regularization constant,  $\delta$ , should be added to the denominator of  $\mu_{\text{NPVSS}}(n)$ . A second consideration is related to the estimation of the parameter  $\sigma_e(n)$ . In practice, the power of the error signal is estimated as follows:

$$\hat{\sigma}_e^2(n) = \lambda \hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n), \quad (55)$$

where  $\lambda$  is a weighting factor. Its value is chosen as  $\lambda = 1 - 1/(KL)$ , where  $K > 1$ . The initial value for (55) is  $\hat{\sigma}_e^2(0) = 0$ . Theoretically, it is clear that  $\sigma_e^2(n) \geq \sigma_v^2$ , which implies that  $\mu_{\text{NPVSS}}(n) \geq 0$ . Nevertheless, the estimation from (55) could result in a lower magnitude than the noise power estimate, which would make  $\mu_{\text{NPVSS}}(n)$  negative. In this situation, the problem is solved by setting  $\mu_{\text{NPVSS}}(n) = 0$ .

The NPVSS-NLMS algorithm is summarized in Table 1. The only parameter that is needed in the step-size formula

**Table 1** NPVSS-NLMS algorithm

Initialization:

$$\hat{\mathbf{h}}(0) = \mathbf{0}_{L \times 1}$$

$$\hat{\sigma}_e^2(0) = 0$$

Parameters:

$$\lambda = 1 - \frac{1}{KL}, \text{ weighting factor with } K > 1$$

$\sigma_v^2$ , noise power known or estimated

$\delta > 0$ , regularization

$\zeta > 0$ , very small number to avoid division by zero

For time index  $n = 1, 2, \dots$ :

$$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n-1)$$

$$\hat{\sigma}_e^2(n) = \lambda\hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n)$$

$$\alpha_{\text{NPVSS}}(n) = 1 - \frac{\sigma_v}{\zeta + \hat{\sigma}_e(n)}$$

$$\mu_{\text{NPVSS}}(n) = \begin{cases} \alpha_{\text{NPVSS}}(n) [\delta + \mathbf{x}^T(n)\mathbf{x}(n)]^{-1}, & \text{if } \alpha_{\text{NPVSS}}(n) > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu_{\text{NPVSS}}(n)\mathbf{x}(n)e(n)$$

of the NPVSS-NLMS algorithm is the power estimate of the system noise. In the case of AEC, this system noise is represented by the near-end signal. Nevertheless, the estimation of the near-end signal power is not always straightforward in real-world AEC applications. Some practical solutions to this problem can be found in [28, 29].

For example, it was demonstrated in [28] that the power estimate of the near-end signal can be evaluated as

$$\hat{\sigma}_v^2(n) = \hat{\sigma}_e^2(n) - \frac{1}{\hat{\sigma}_x^2(n)} \hat{\mathbf{r}}_{\text{ex}}^T(n) \hat{\mathbf{r}}_{\text{ex}}(n), \quad (56)$$

where the variance of  $e(n)$  is estimated based on (55) and the other terms are evaluated in a similar manner, i.e.,

$$\hat{\sigma}_x^2(n) = \lambda\hat{\sigma}_x^2(n-1) + (1-\lambda)x^2(n), \quad (57)$$

$$\hat{\mathbf{r}}_{\text{ex}}(n) = \lambda\hat{\mathbf{r}}_{\text{ex}}(n-1) + (1-\lambda)\mathbf{x}(n)e(n). \quad (58)$$

A more practical solution was proposed in [29]. It is known that the desired signal of the adaptive filter is expressed as  $d(n) = y(n) + v(n)$ . Since the echo signal and the near-end signal can be considered uncorrelated, the previous relation can be rewritten in terms of variances as

$$E[d^2(n)] = E[y^2(n)] + E[v^2(n)]. \quad (59)$$

Assuming that the adaptive filter has converged to a certain degree, we can use the approximation

$$E[y^2(n)] \approx E[\hat{y}^2(n)]. \quad (60)$$

Consequently, using power estimates, we may compute

$$\hat{\sigma}_v^2(n) = \left| \hat{\sigma}_d^2(n) - \hat{\sigma}_y^2(n) \right|, \quad (61)$$

where  $\hat{\sigma}_d^2(n)$  and  $\hat{\sigma}_y^2(n)$  are the power estimates of  $d(n)$  and  $\hat{y}(n)$ , respectively. These parameters can be recursively evaluated similar to (55), i.e.,

$$\hat{\sigma}_d^2(n) = \lambda\hat{\sigma}_d^2(n-1) + (1-\lambda)d^2(n), \quad (62)$$

$$\hat{\sigma}_y^2(n) = \lambda\hat{\sigma}_y^2(n-1) + (1-\lambda)\hat{y}^2(n). \quad (63)$$

The absolute values in (61) prevent any minor deviations (due to the use of power estimates) from the true values, which can make the normalized step-size negative or complex.

When only the background noise is present, an estimate of its power is obtained using the right-hand term in (61). This expression holds even if the level of the background noise changes, so that there is no need for the estimation of this parameter during silences of the near-end talker. In case of double-talk, when the near-end speech is also present (assuming that it is uncorrelated with the background noise), the right-hand term in (61) still provides a power estimate of the near-end signal. Most importantly, this term depends only on the signals that are available within the AEC application, i.e., the microphone signal,  $d(n)$ , and the output of the adaptive filter,  $\hat{y}(n)$ . Moreover, as it was demonstrated in [29], the estimation from (61) is also suitable for the under-modeling case, i.e., when the length of  $\hat{\mathbf{h}}(n)$  is smaller than the length of  $\mathbf{h}(n)$ , so that an under-modeling noise appears (i.e., the residual echo caused by the part of the echo path that is not modeled by the adaptive filter; it can be interpreted as an additional noise that corrupts the near-end signal).

The main drawback of (61) is due to the approximation in (60). This assumption will be biased in the initial convergence phase or when there is a change of the echo path. Concerning the first problem, we can use a regular NLMS algorithm in the first steps (e.g., in the first  $L$  iterations).

#### 1.4 JO-NLMS algorithm

In both previous sections, the assumption from (9) was used when evaluating the a posteriori misalignment, i.e., the unknown system is time-invariant. However, in AEC and also in many other system identification problems, this assumption is quite strong. In practice, the system to be identified could be variable in time. For example, in AEC, it can be assumed that the impulse response of the echo path is modeled by a time-varying system following a first-order Markov model [9]. Therefore, a more reliable approach could be based on the Kalman filter, since the state variable model fits better in this context [26, 30, 31].

Motivated by the work of Hänsler and Schmidt [1] (summarized in Section 1.2), we extend here their analysis by

assuming that  $\mathbf{h}(n)$  is a zero-mean random vector, which follows a simplified first-order Markov model, i.e.,

$$\mathbf{h}(n) = \mathbf{h}(n - 1) + \mathbf{w}(n), \quad (64)$$

where  $\mathbf{w}(n)$  is a zero-mean white Gaussian noise signal vector, which is uncorrelated with  $\mathbf{h}(n - 1)$ . The correlation matrix of  $\mathbf{w}(n)$  is assumed to be  $\mathbf{R}_w = \sigma_w^2 \mathbf{I}_L$ , where  $\mathbf{I}_L$  is the  $L \times L$  identity matrix. The variance,  $\sigma_w^2$ , captures the uncertainties in  $\mathbf{h}(n)$ . Equations (1) and (64) define now a state variable model, similar to Kalman filtering setup.

### 1.4.1 Convergence analysis

In the context of the previously defined model, let us consider the update of the NLMS algorithm from (7). Next, developing (7) in terms of the a posteriori misalignment from (8), also taking (64) into account, we obtain

$$\mathbf{m}(n) = \mathbf{m}(n - 1) + \mathbf{w}(n) - \frac{\alpha \mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta}. \quad (65)$$

For large values of  $L$  (i.e.,  $L \gg 1$ ), it holds that  $\mathbf{x}^T(n)\mathbf{x}(n) \approx L\sigma_x^2$  [1, 19]. Consequently,

$$\frac{\alpha}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta} \approx \frac{\alpha}{L\sigma_x^2 + \delta}. \quad (66)$$

This term contains both the control parameters, i.e.,  $\alpha$  and  $\delta$ , and also the statistical information on the input signal. However, for a large value of  $L$  and a certain stationarity degree of the input signal, we can treat this term as a deterministic quantity [1].

Under these circumstances, taking the  $\ell_2$  norm in (65), then mathematical expectation on both sides (also using (66)), and removing the uncorrelated products, we obtain

$$\begin{aligned} E[\|\mathbf{m}(n)\|_2^2] &= E[\|\mathbf{m}(n - 1)\|_2^2] \\ &+ L\sigma_w^2 - \frac{2\alpha}{L\sigma_x^2 + \delta} E[\mathbf{x}^T(n)\mathbf{m}(n - 1)e(n)] \\ &- \frac{2\alpha}{L\sigma_x^2 + \delta} E[\mathbf{x}^T(n)\mathbf{w}(n)e(n)] \\ &+ \frac{\alpha^2}{(L\sigma_x^2 + \delta)^2} E[e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)]. \end{aligned} \quad (67)$$

In order to further process (67), let us focus on its last three cross-correlation terms. Based on (1), (8), and (64), the a priori error signal from (2) can be rewritten as

$$e(n) = \mathbf{x}^T(n)\mathbf{m}(n - 1) + \mathbf{x}^T(n)\mathbf{w}(n) + v(n). \quad (68)$$

Therefore, taking (68) into account within the first cross-correlation term from (67) (also removing the uncorrelated products), it results in

$$\begin{aligned} E[\mathbf{m}^T(n - 1)\mathbf{x}(n)e(n)] &\approx E[\mathbf{m}^T(n - 1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{m}(n - 1)] \\ &= E\left\{\text{tr}\left[\mathbf{m}(n - 1)\mathbf{m}^T(n - 1)\mathbf{x}(n)\mathbf{x}^T(n)\right]\right\}. \end{aligned} \quad (69)$$

Next, the following assumptions can be considered: (i) the a posteriori misalignment at time index  $n - 1$  is uncorrelated with the input vector at time index  $n$  and (ii) the correlation matrix of the input is close to a diagonal one, i.e.,  $E[\mathbf{x}(n)\mathbf{x}^T(n)] \approx \sigma_x^2 \mathbf{I}_L$  (this is a fairly restrictive assumption, however, it has been widely used to simplify the analysis [16]). Consequently, (69) becomes

$$\begin{aligned} E[\mathbf{m}^T(n - 1)\mathbf{x}(n)e(n)] &\approx \text{tr}\left\{E\left[\mathbf{m}(n - 1)\mathbf{m}^T(n - 1)\right]\right. \\ &\quad \left.E[\mathbf{x}(n)\mathbf{x}^T(n)]\right\} \\ &= \sigma_x^2 E[\|\mathbf{m}(n - 1)\|_2^2]. \end{aligned} \quad (70)$$

The second cross-correlation term from (67) can be also developed based on (68). Taking into account that the correlation matrix of  $\mathbf{w}(n)$  is assumed to be diagonal and removing the uncorrelated products, it results in

$$\begin{aligned} E[\mathbf{w}^T(n)\mathbf{x}(n)e(n)] &= E[\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)] \\ &= E\left\{\text{tr}\left[\mathbf{w}(n)\mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\right]\right\} \\ &\approx \text{tr}\left\{E[\mathbf{w}(n)\mathbf{w}^T(n)]E[\mathbf{x}(n)\mathbf{x}^T(n)]\right\} \\ &= L\sigma_x^2\sigma_w^2. \end{aligned} \quad (71)$$

The last expectation term from (67) can be also expressed taking (68) into account. Using a similar approach, it results in

$$\begin{aligned} E[e^2(n)\mathbf{x}^T(n)\mathbf{x}(n)] &= \text{tr}\left\{E\left[e^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\right]\right\} \\ &= \text{tr}\left\{E\left[v^2(n)\mathbf{x}(n)\mathbf{x}^T(n)\right]\right\} \\ &\quad + \text{tr}\left\{E\left\{\left[\mathbf{r}^T(n)\mathbf{x}(n)\right]^2\mathbf{x}(n)\mathbf{x}^T(n)\right\}\right\}, \end{aligned} \quad (72)$$

with the notation

$$\begin{aligned} \mathbf{r}(n) &= \mathbf{m}(n - 1) + \mathbf{w}(n) \\ &= [r_0(n) \ r_1(n) \ \dots \ r_{L-1}(n)]^T. \end{aligned}$$

Next, let us focus on the last expectation term in (72); since the correlation matrix of the input was assumed to be diagonal, this term can be developed based on the

Gaussian moment factoring theorem [32] (also known as the Isserlis' theorem) and results in

$$\begin{aligned}
 & E \left[ \mathbf{r}^T(n) \mathbf{x}(n) \mathbf{r}^T(n) \mathbf{x}(n) \mathbf{x}(n) \mathbf{x}^T(n) \right] \\
 &= \sigma_x^4 \mathbf{I}_L \sum_{i=0}^{L-1} E \left[ r_i^2(n) \right] + 2\sigma_x^4 E \left[ \mathbf{r}(n) \mathbf{r}^T(n) \right] \\
 &= \sigma_x^4 \mathbf{I}_L \left\{ L\sigma_w^2 + E \left[ \|\mathbf{m}(n-1)\|_2^2 \right] \right\} + 2\sigma_x^4 \left\{ \sigma_w^2 \mathbf{I}_L \right. \\
 &\quad \left. + E \left[ \mathbf{m}(n-1) \mathbf{m}^T(n-1) \right] \right\}.
 \end{aligned} \tag{73}$$

Therefore, using the result from (73), (72) becomes

$$\begin{aligned}
 & E \left[ e^2(n) \mathbf{x}^T(n) \mathbf{x}(n) \right] \\
 &= L\sigma_x^2 \sigma_v^2 + \text{tr} \left\{ \sigma_x^4 \left\{ L\sigma_w^2 + E \left[ \|\mathbf{m}(n-1)\|_2^2 \right] \right\} \mathbf{I}_L \right\} \\
 &\quad + \text{tr} \left\{ 2\sigma_x^4 \left\{ \sigma_w^2 \mathbf{I}_L + E \left[ \mathbf{m}(n-1) \mathbf{m}^T(n-1) \right] \right\} \right\} \\
 &= L\sigma_x^2 \sigma_v^2 + (L+2)\sigma_x^4 \left\{ E \left[ \|\mathbf{m}(n-1)\|_2^2 \right] + L\sigma_w^2 \right\}.
 \end{aligned} \tag{74}$$

Having all these terms, we can introduce (70), (71), and (74) in (67), also denoting  $m(n) = E \left[ \|\mathbf{m}(n)\|_2^2 \right]$ , to obtain

$$m(n) = \tilde{A}(\alpha, \delta, L, \sigma_x^2) m(n-1) + \tilde{B}(\alpha, \delta, L, \sigma_x^2, \sigma_v^2, \sigma_w^2), \tag{75}$$

where

$$\tilde{A}(\alpha, \delta, L, \sigma_x^2) = 1 - \frac{2\sigma_x^2}{L\sigma_x^2 + \delta} \alpha + \frac{(L+2)\sigma_x^4}{(L\sigma_x^2 + \delta)^2} \alpha^2, \tag{76}$$

$$\begin{aligned}
 \tilde{B}(\alpha, \delta, L, \sigma_x^2, \sigma_v^2, \sigma_w^2) &= \frac{\alpha^2 L \sigma_x^2 \left[ \sigma_v^2 + (L+2)\sigma_x^2 \sigma_w^2 \right]}{(L\sigma_x^2 + \delta)^2} \\
 &\quad - \frac{2\alpha L \sigma_x^2 \sigma_w^2}{L\sigma_x^2 + \delta} + L\sigma_w^2.
 \end{aligned} \tag{77}$$

The result from (75) illustrates a “separation” between the convergence and misadjustment components, similar to (18)–(20) from Section 1.2. Therefore, the term  $\tilde{A}(\alpha, \delta, L, \sigma_x^2)$  influences the convergence rate of the algorithm. As expected, it depends on the normalized step-size value, the regularization constant, the filter length, and the input signal power. It is interesting to notice that it does not depend on the system noise power,  $\sigma_v^2$ , or the uncertainties,  $\sigma_w^2$ ; in other words, the convergence rate should not be influenced by these two terms. In fact, the term from (76) is very similar to the contraction

parameter,  $A(\alpha, \delta, L, \sigma_x^2)$ , from (19). Similarly, it can be noticed that the fastest convergence mode is obtained when the function from (76) reaches its minimum. Taking the normalized step-size as the reference parameter, we obtain

$$\tilde{\alpha}_{\text{FC}} = \frac{\delta + L\sigma_x^2}{(L+2)\sigma_x^2}, \tag{78}$$

which is similar to the result obtained in (22). For example, neglecting the regularization constant (i.e.,  $\delta \approx 0$ ) and assuming that  $L \gg 2$ , the fastest convergence mode is achieved for  $\alpha \approx 1$ , which is the same conclusion related to (22). Also, similar to (23), the stability condition can be found by imposing  $|\tilde{A}(\alpha, \delta, L, \sigma_x^2)| < 1$ , which leads to

$$0 < \tilde{\alpha}_{\text{stable}} < 2 \frac{\delta + L\sigma_x^2}{(L+2)\sigma_x^2} = 2\tilde{\alpha}_{\text{FC}}. \tag{79}$$

For example, taking  $\delta = 0$  and  $L \gg 2$  in (79), the standard stability condition of the NLMS algorithm results, i.e.,  $0 < \alpha < 2$ .

The term  $\tilde{B}(\alpha, \delta, L, \sigma_x^2, \sigma_v^2, \sigma_w^2)$  influences the misadjustment of the algorithm and it depends on both  $\sigma_v^2$  and  $\sigma_w^2$  (clearly, the misadjustment increases when these two factors increase). As we can notice, it is very similar to the expansion parameter,  $B(\alpha, \delta, L, \sigma_x^2)$ , from (20), except the fact that the term from (77) includes now the contributions of  $\sigma_v^2$  and  $\sigma_w^2$ . However, the lowest misadjustment is obtained in a similar way, i.e., when the function from (77) reaches its minimum. Thus, taking the normalized step-size as the reference parameter, the lowest misadjustment is achieved for

$$\tilde{\alpha}_{\text{LM}} = \frac{\sigma_w^2 (L\sigma_x^2 + \delta)}{\sigma_v^2 + (L+2)\sigma_x^2 \sigma_w^2}. \tag{80}$$

In order to compare this result with (25), let us assume that the system is time-invariant, i.e.,  $\sigma_w^2 \approx 0$ . Consequently, (80) leads to  $\alpha \approx 0$  (i.e., the lowest misadjustment is obtained for a normalized step-size close to zero), which is the same result obtained in (25).

#### 1.4.2 Derivation of the algorithm

It is known that the ideal requirements of any adaptive algorithm are for both fast convergence and low misadjustment. In our framework, there are two important issues to be considered: (1) we have two main parameters to control,  $\alpha$  and  $\delta$ , which influence the overall performance of the NLMS algorithm and (2) in the context of system identification, it is reasonable to follow a minimization problem in terms of the system misalignment, as outlined by Hänsler and Schmidt in [1].

Thus, following (75) and considering that the two important parameters are time dependent, it can be imposed

$$\frac{\partial m(n)}{\partial \alpha(n)} = 0, \tag{81}$$

$$\frac{\partial m(n)}{\partial \delta(n)} = 0. \tag{82}$$

After straightforward computations, both equations lead to the same result, i.e.,

$$\frac{\alpha(n)}{L\sigma_x^2 + \delta(n)} = \frac{m(n-1) + L\sigma_w^2}{L\sigma_v^2 + (L+2)\sigma_x^2[m(n-1) + L\sigma_w^2]}, \tag{83}$$

which suggests a joint optimization process. With a proper estimation of its parameters (as will be discussed in the end of this section), the term from the right-hand side of (83) acts like a variable step-size. At this point, we can introduce (83) in (7), thus obtaining

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{[m(n-1) + L\sigma_w^2] \mathbf{x}(n)e(n)}{L\sigma_v^2 + (L+2)\sigma_x^2[m(n-1) + L\sigma_w^2]}. \tag{84}$$

Next, there is a need to update the parameter  $m(n)$  in (84). Using (83) in (75), followed by several straightforward computations, it results in

$$m(n) = \left\{ 1 - \frac{\sigma_x^2 [m(n-1) + L\sigma_w^2]}{L\sigma_v^2 + (L+2)\sigma_x^2 [m(n-1) + L\sigma_w^2]} \right\} \times [m(n-1) + L\sigma_w^2]. \tag{85}$$

Consequently, the resulting JO-NLMS algorithm is defined by the relations (2), (84), and (85).

Finally, there are some practical considerations that should be outlined. The JO-NLMS algorithm requires the estimation of three main parameters, i.e.,  $\sigma_x^2$ ,  $\sigma_v^2$ , and  $\sigma_w^2$ . The first one can be easily evaluated as in the NLMS algorithm, i.e.,  $\hat{\sigma}_x^2(n) = \frac{1}{L} \mathbf{x}^T(n) \mathbf{x}(n)$ . The second parameter (i.e.,  $\sigma_v^2$ ) appears in many VSS and VR versions of the NLMS algorithm. Different methods can be used to estimate it, e.g., [19, 28, 29], as mentioned at the end of Sections 1.2 and 1.3. Maybe the most important parameter to be found is  $\sigma_w^2$ . Small values of  $\sigma_w^2$  (i.e., the system is assumed to be time-invariant) imply a good misadjustment but a poor tracking; on the other hand, large values of  $\sigma_w^2$  (i.e., assuming that the uncertainties in the system are high) imply a good tracking but a high misadjustment. In practice, we propose to estimate  $\sigma_w^2$  by taking the  $\ell_2$  norm on both sides of (64) and replacing  $\mathbf{h}(n)$  by its estimate  $\hat{\mathbf{h}}(n)$ , thus resulting in

$$\hat{\sigma}_w^2(n) = \frac{1}{L} \left\| \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1) \right\|_2^2. \tag{86}$$

According to (86), the parameter  $\hat{\sigma}_w^2(n)$  takes large values in the beginning of adaptation (or when there is an abrupt change of the system), thus providing fast convergence and tracking. On the other hand, when the algorithm starts to converge, the value of  $\hat{\sigma}_w^2(n)$  reduces, which leads to low misadjustment. In this way, the algorithm achieves a proper compromise between the performance criteria. In finite precision implementations, in order to avoid any risk of freezing in (86), it is recommended to set a lower bound for  $\hat{\sigma}_w^2(n)$  (e.g., the smallest positive number available).

The JO-NLMS algorithm is summarized in Table 2, in such a way that its implementation is facilitated. This algorithm is similar to the simplified Kalman filter presented in [31]. However, contrary to this one (which was obtained as an approximation of the general Kalman filter), the JO-NLMS algorithm was derived in a different manner following a specific optimization criterion. In fact, this is an alternative way to obtain the same results as with the Kalman filter.

### 1.5 Simulation results

Simulations were performed in an AEC configuration, as shown in Fig. 1. The measured acoustic impulse response was truncated to 512 coefficients (Fig. 2), and the same length was used for the adaptive filter, i.e.,  $L = 512$ ; the sampling rate is 8 kHz. We should note that in many real-world AEC scenarios, the adaptive filter works most likely in an under-modeling situation, i.e., its length is

**Table 2** JO-NLMS algorithm

Initialization:

$$\hat{\mathbf{h}}(0) = \mathbf{0}_{L \times 1}$$

$$m(0) = \varepsilon > 0$$

$$\hat{\sigma}_w^2(0) = 0$$

Parameters:

$\sigma_v^2$ , noise power known or estimated

For time index  $n = 1, 2, \dots$ :

$$\hat{\sigma}_x^2(n) = \frac{1}{L} \mathbf{x}^T(n) \mathbf{x}(n)$$

$$e(n) = d(n) - \mathbf{x}^T(n) \hat{\mathbf{h}}(n-1)$$

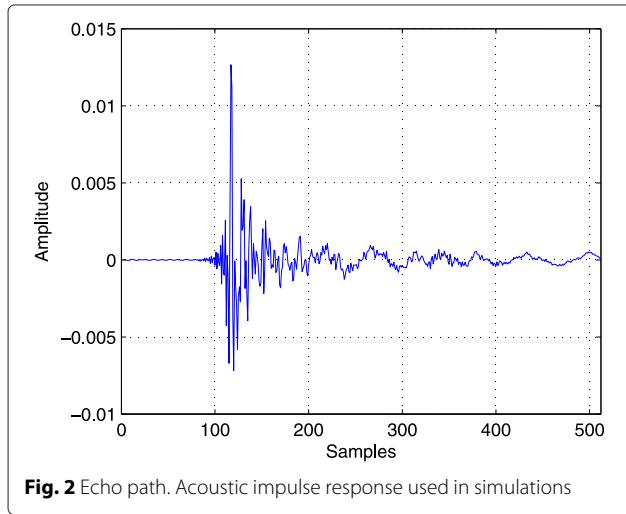
$$p(n) = m(n-1) + L\hat{\sigma}_w^2(n-1)$$

$$q(n) = \frac{p(n)}{L\sigma_v^2 + (L+2)p(n)\hat{\sigma}_x^2(n)}$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + q(n)\mathbf{x}(n)e(n)$$

$$m(n) = [1 - q(n)\hat{\sigma}_x^2(n)]p(n)$$

$$\hat{\sigma}_w^2(n) = \frac{1}{L} \left\| \hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1) \right\|_2^2$$



smaller than the length of the acoustic impulse response. Hence, the residual echo caused by the part of the system that cannot be modeled acts like an additional noise (that corrupts the near-end signal) and disturbs the overall performance. However, for experimental purposes, we set the same length for both the unknown system (i.e., the acoustic echo path) and the adaptive filter.

The input signal,  $x(n)$ , is either a white Gaussian noise, an AR(1) process generated by filtering a white Gaussian noise through a first-order system  $1/(1 - 0.8z^{-1})$  or a speech sequence. An independent white Gaussian noise  $v(n)$  is added to the echo signal  $y(n)$ , with SNR = 20 dB (except in the last experiment where the SNR is variable and the near-end speech is also present). In most of the experiments (except in the last one), we assume that  $\sigma_v^2$  is known; in practice, this variance can be estimated like in [19, 28, 29] (as presented in the end of Section 1.3). The tracking capability of the algorithm is an important issue in AEC, where the acoustic impulse response may rapidly change at any time during the connection. Consequently, an echo path change scenario is simulated in most of the experiments, by shifting the impulse response to the right by 12 samples, in the middle of the experiment. The measure of performance is the normalized misalignment (in dB), defined as

$$\bar{m}(n) = 20 \log_{10} \left[ \frac{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2}{\|\mathbf{h}(n)\|_2} \right]. \quad (87)$$

In the first set of experiments, we evaluate the performance of the optimal control parameters proposed by Hänslér and Schmidt in [1] (also summarized in Section 1.2), in order to set the benchmark for further comparisons. In this context, we consider the ideal estimation of these parameters [i.e.,  $\alpha_{\text{opt}}(n)$  and  $\delta_{\text{opt}}(n)$  from Section 1.2], assuming that the undistorted error signal

$e_u(n)$  from (27) is available and its power,  $E[e_u^2(n)] = \sigma_{e_u}^2(n)$ , can be evaluated similar to (55), i.e.,

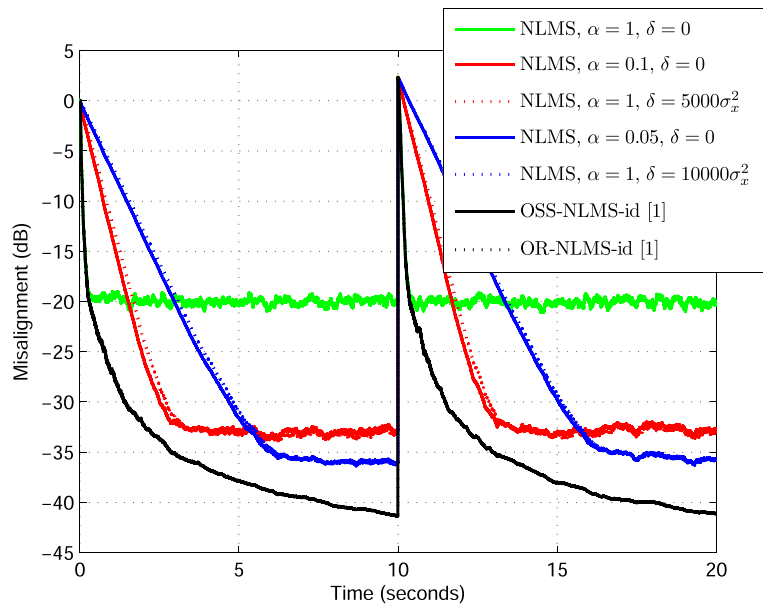
$$\begin{aligned} \hat{\sigma}_{e_u}^2(n) &= \lambda \hat{\sigma}_{e_u}^2(n-1) + (1-\lambda)e_u^2(n) \\ &= \lambda \hat{\sigma}_{e_u}^2(n-1) + (1-\lambda)[e(n) - v(n)]^2, \end{aligned} \quad (88)$$

where  $\lambda$  is a weighting factor [ $\lambda = 1 - 1/(KL)$ , with  $K > 1$ ]. Of course, in practice, the near-end signal  $v(n)$  is not available; however, for comparison purpose, we consider that it is available in (88).

In the first simulation, we evaluate the performance of the NLMS algorithm using  $\alpha_{\text{opt}}(n)$  and  $\delta_{\text{opt}}(n)$ , respectively. Since the estimation from (88) is used for both these parameters, we deal with the ideal behavior of the algorithms. Consequently, we will refer to these algorithms as the ideal optimal step-size NLMS (OSS-NLMS-id) and the ideal optimal regularized NLMS (OR-NLMS-id), respectively. In Fig. 3, these ideal benchmarks are compared to the NLMS algorithm using different constant values of the normalized step-size,  $\alpha$ , and regularization parameter,  $\delta$ ; the input signal is a white Gaussian noise. First, it can be noticed that the performance of the regular NLMS algorithm can be controlled in terms of both parameters,  $\alpha$  and  $\delta$ , either by setting the fastest convergence mode (i.e.,  $\alpha = 1$ ) and adjusting the value of  $\delta$ , or by neglecting the regularization constant (i.e.,  $\delta = 0$ ) and tuning the value of  $\alpha$ . On the other hand, in case of the optimal control parameters, the OSS-NLMS-id and OR-NLMS-id algorithms achieve both fast convergence/tracking and low misalignment, outperforming the NLMS algorithms that use constant values for  $\alpha$  and  $\delta$ . Besides, it should be noted that the OSS-NLMS-id and OR-NLMS-id algorithms are equivalent in terms of their performance (their misalignment curves are overlapped), which justifies the findings from Section 1.2. For this experiment, the evolution of  $\alpha_{\text{opt}}(n)$  and  $\delta_{\text{opt}}(n)$  is depicted in Fig. 4, also supporting the expected behavior of these parameters.

Next, the same experiment is repeated using an AR(1) process as input; the results are presented in Figs. 5 and 6. As expected, the convergence rate of the algorithms is affected in this case, due to correlated inputs. Also, as we can notice from Fig. 5, the OSS-NLMS-id and OR-NLMS-id algorithms (which behave the same) still outperform their classical counterparts. Based on the evolution of  $\alpha_{\text{opt}}(n)$  and  $\delta_{\text{opt}}(n)$  depicted in Fig. 6, we can outline again the discussion from the end of Section 1.2, related to the dynamic range of these parameters. In practice, it is usually more convenient to control the performance of the algorithm in terms of the normalized step-size, since its values are limited in a specific interval. On the other hand, it could be more



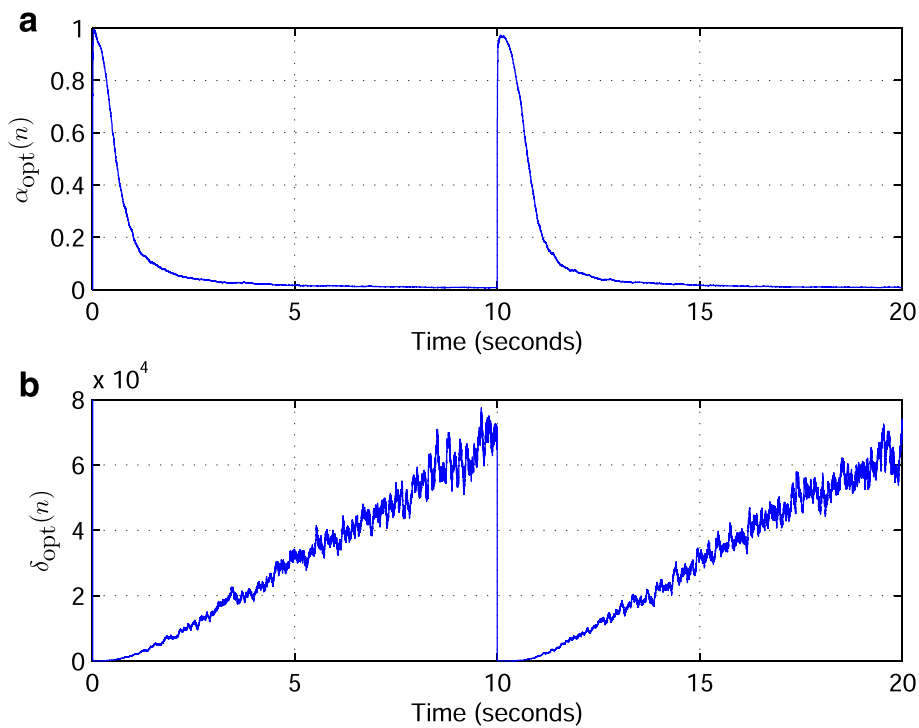


**Fig. 3** Performance of the optimal algorithms for white Gaussian input. Misalignment of the NLMS (for different values of  $\alpha$  and  $\delta$ ), OSS-NLMS-id, and OR-NLMS-id algorithms. The input signal is white Gaussian, echo path changes at time 10 s,  $L = 512$ , and SNR = 20 dB

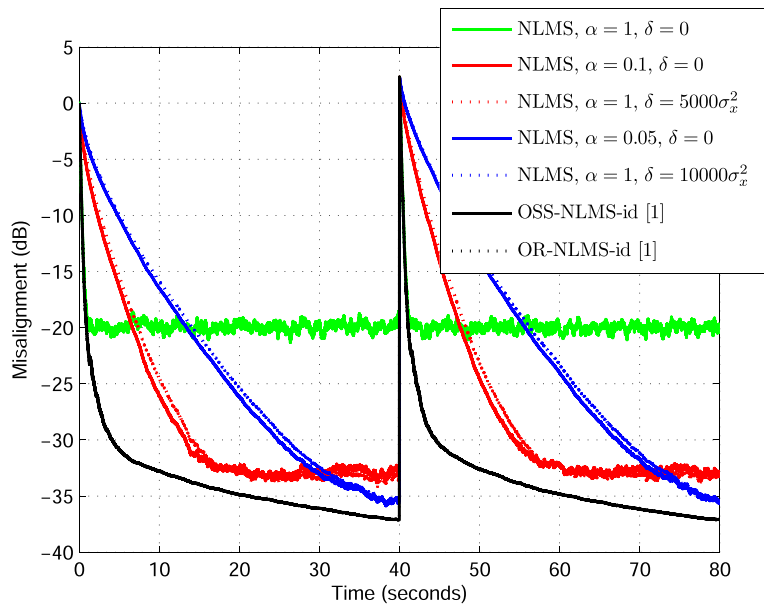
difficult to control the adaptation in terms of the regularization term, since its values are increasing and could lead to overflows. Usually, an upper bound on the regularization parameter could be imposed, but this would introduce an extra tuning parameter in the algorithm.

Due to these aspects, only the OSS-NLMS-id algorithm will be considered as a benchmark in the following experiments.

Nevertheless, the OSS-NLMS-id algorithm still requires a constant regularization parameter, especially in case of



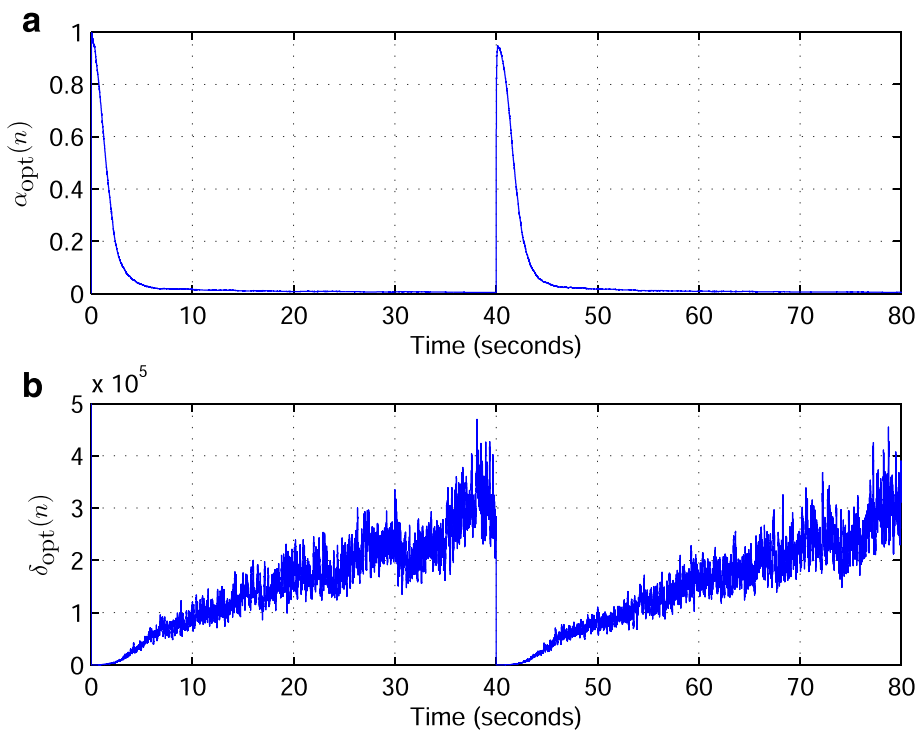
**Fig. 4** Optimal control parameters for white Gaussian input. Evolution of the optimal control parameters: **a**  $\alpha_{opt}(n)$  of the OSS-NLMS-id algorithm and **b**  $\delta_{opt}(n)$  of the OR-NLMS-id algorithm. Other conditions are the same as in Fig. 3



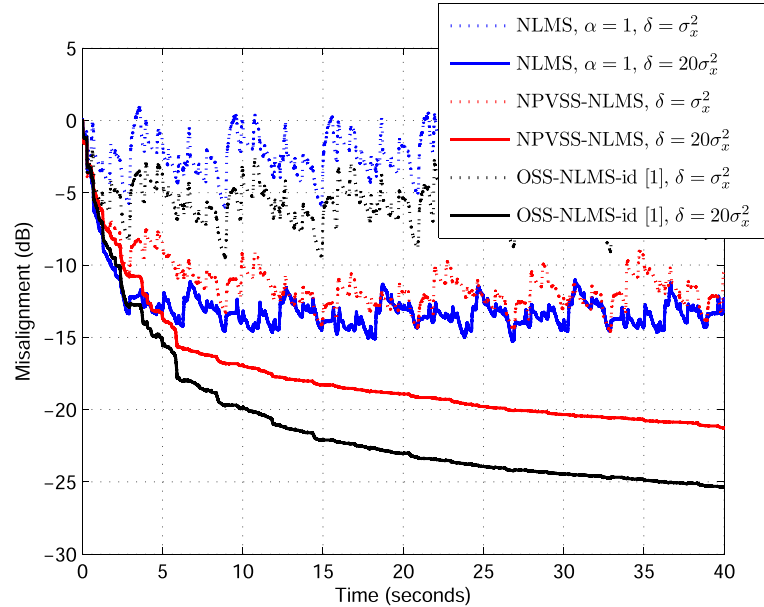
**Fig. 5** Performance of the optimal algorithms for AR(1) input. Misalignment of the NLMS (for different values of  $\alpha$  and  $\delta$ ), OSS-NLMS-id, and OR-NLMS-id algorithms. The input signal is an AR(1) process, echo path changes at time 40 s,  $L = 512$ , and SNR = 20 dB

non-stationary inputs like speech. This is also the case of the NPVSS-NLMS algorithm presented in Section 1.3. While in the previous experiments, this regularization constant was neglected (due to the stationary nature of the input signals), and the next simulation shows the

importance of this parameter in practice. For this purpose, in Fig. 7, a speech sequence is considered at the far-end. The NLMS, NPVSS-NLMS, and OSS-NLMS-id algorithms are compared when using two different values of the regularization constant, i.e.,  $\delta = \sigma_x^2$  and  $\delta =$



**Fig. 6** Optimal control parameters for AR(1) input. Evolution of the optimal control parameters: **a**  $\alpha_{opt}(n)$  of the OSS-NLMS-id algorithm and **b**  $\delta_{opt}(n)$  of the OR-NLMS-id algorithm. Other conditions are the same as in Fig. 5



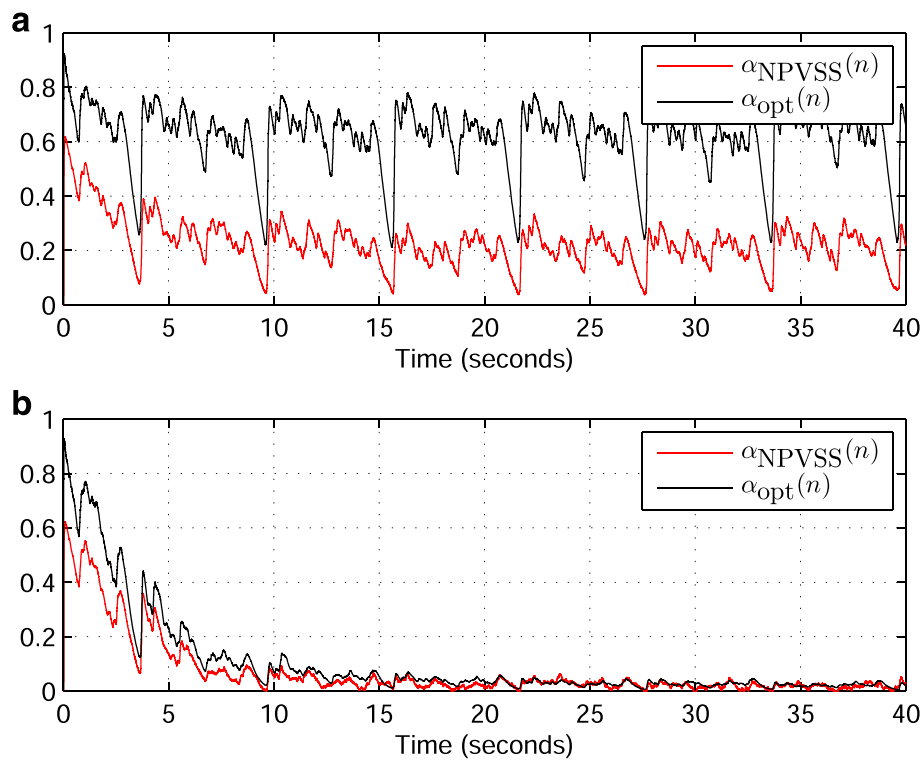
**Fig. 7** Regularization influence on the algorithms' performance. Misalignment of the NLMS (with  $\alpha = 1$ ), NPVSS-NLMS, and OSS-NLMS-id algorithms for different values of  $\delta$ . The input signal is speech,  $L = 512$ , and SNR = 20 dB

$20\sigma_x^2$ . As expected, the small regularization is not suitable in this case, leading to large misalignment. On the other hand, the rule of thumb  $\delta = 20\sigma_x^2$  (used in many echo cancellation scenarios [6–8]) is more appropriate here. Thus, the regularization parameter is a must in this case. In fact, the regularization parameter is required in all ill-posed and ill-conditioned problems such as in AEC; some insights for choosing this parameter in practice can be found in [14]. However, in all the following experiments, we will consider a constant regularization  $\delta = 20\sigma_x^2$  for the NLMS, NPVSS-NLMS, and OSS-NLMS-id algorithms. As shown in [14], the regularization parameter of the NLMS algorithm is related to the value of SNR and the filter's length  $L$ . For our experimental setup, i.e.,  $L = 512$  and SNR = 20 dB, the value  $\delta = 20\sigma_x^2$  fits well. However, this value should be increased for larger values of  $L$  or lower SNRs [14]. To conclude this experiment, the influence of the regularization parameter can be also noticed in Fig. 8, where the control parameters of the NPVSS-NLMS and OSS-NLMS-id algorithms are depicted, i.e.,  $\alpha_{\text{NPVSS}}(n)$  and  $\alpha_{\text{opt}}(n)$ , respectively. Clearly, their behavior is strongly biased in case of the small regularization parameter, while they perform similarly in case of a proper regularization.

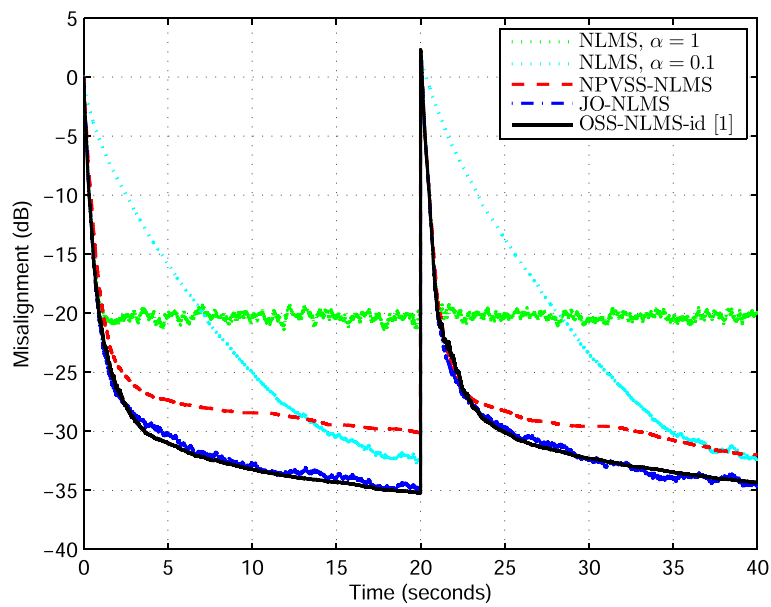
Next, the JO-NLMS algorithm (presented in Section 1.4) is also involved in the rest of experiments. As compared to its counterparts, this algorithm does not require an explicit regularization term. Its global step-size from (83) resulted based on the joint-optimization on both the normalized step-size and regularization

parameter. In Figs. 9 and 10, the NLMS algorithm (for different values of  $\alpha$ ) is compared with the NPVSS-NLMS, JO-NLMS, and OSS-NLMS-id algorithms, when the far-end signal is an AR(1) process or a speech sequence, respectively. According to these results, it can be noticed that the NLMS algorithm is clearly outperformed by the other algorithms, in terms of convergence rate, tracking, and misalignment. Also, the NPVSS-NLMS and JO-NLMS algorithms perform in a similar manner (with a slight advantage for the JO-NLMS algorithm); besides, they are close to the performance of the OSS-NLMS-id algorithm, which represents the ideal benchmark.

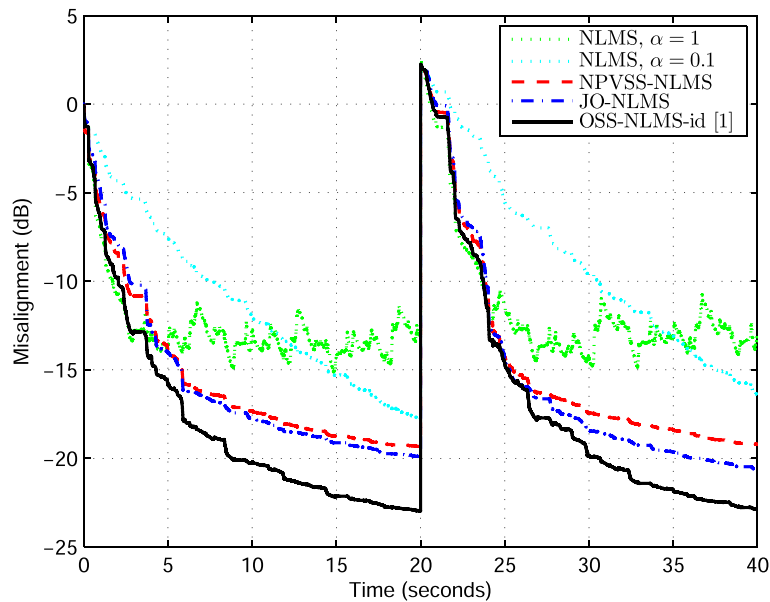
In all the previous experiments involving the NPVSS-NLMS and JO-NLMS algorithms, it was assumed that the power of the system noise,  $\sigma_v^2$ , is available. However, in practice, it should be also estimated. Moreover, in AEC, the signal  $v(n)$  represents the near-end signal, which can contain both the background noise and the near-end speech; since both these signals could be non-stationary, the estimation of  $\sigma_v^2$  becomes more difficult. There are different methods for estimating this parameter; for example, in a single-talk scenario, it can be estimated during silences of the near-end talker [19]. Also, other practical methods to estimate  $\sigma_v^2$  can be found in [28, 29], as shown in the end of Section 1.3. In the last experiment, the estimation from (61) is used within the NPVSS-NLMS and JO-NLMS algorithms. Two challenging scenarios are considered in Fig. 11, where the far-end signal is a speech sequence. First, a variation of the background noise is



**Fig. 8** Regularization influence on the control parameters. Evolution of the normalized step-sizes of the NPVSS-NLMS algorithm [ $\alpha_{NPVSS}(n)$ ] and OSS-NLMS-id algorithm [ $\alpha_{opt}(n)$ ] for different values of  $\delta$ : **a**  $\delta = \sigma_x^2$  and **b**  $\delta = 20\sigma_x^2$ . Other conditions are the same as in Fig. 7



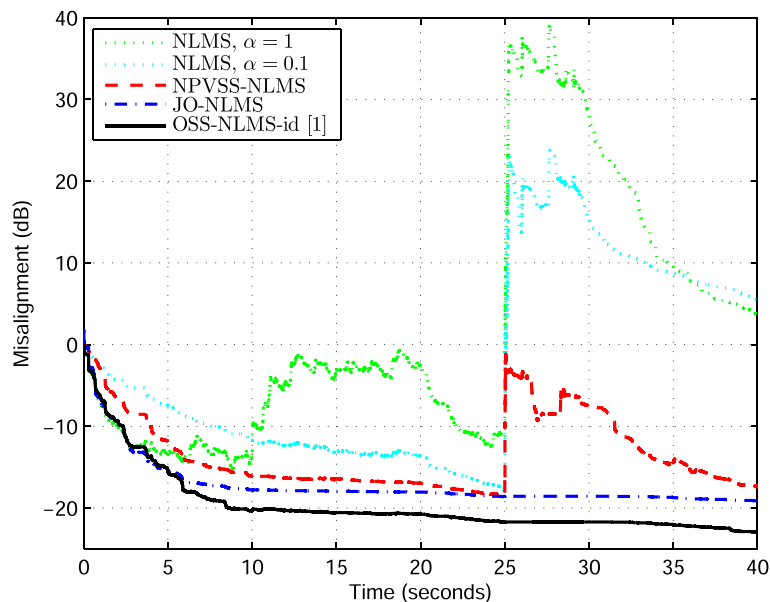
**Fig. 9** Performance of the algorithms for AR(1) input. Misalignment of the NLMS (for different values of  $\alpha$ ), NPVSS-NLMS, JO-NLMS, and OSS-NLMS-id algorithms. The input signal is an AR(1) process; echo path changes at time 20 s,  $L = 512$ , and SNR = 20 dB



**Fig. 10** Performance of the algorithms for speech input. Misalignment of the NLMS (for different values of  $\alpha$ ), NPVSS-NLMS, JO-NLMS, and OSS-NLMS-id algorithms. The input signal is speech; echo path changes at time 20 s,  $L = 512$ , and SNR = 20 dB

simulated, by decreasing the SNR from 20 to 10 dB between times 10 and 20 s; second, the near-end speech appears between times 25 and 30 s (i.e., double-talk case), without using any DTD. The results from Fig. 11 indicate that the NLMS algorithm fails in this case, especially during double-talk. The NPVSS-NLMS and JO-NLMS

algorithms show good robustness features in both situations (with an advantage for the JO-NLMS algorithm during double-talk). In terms of robustness, the JO-NLMS algorithm performs similar to the ideal case represented by the OSS-NLMS-id algorithm. Finally, it should be noted that both the NPVSS-NLMS and JO-NLMS



**Fig. 11** Performance of the algorithms during near-end variations. Misalignment of the NLMS (for different values of  $\alpha$ ), NPVSS-NLMS, JO-NLMS, and OSS-NLMS-id algorithms. The NPVSS-NLMS and JO-NLMS algorithms use the estimated  $\hat{\sigma}_v^2(n)$  from (61). The input signal is speech,  $L = 512$ , and SNR = 20 dB. The SNR decreases from 20 to 10 dB between times 10 and 20 s and the near-end speech appears between times 25 and 30 s (without using a DTD)

algorithms do not require any additional features to control their behavior, thus being reliable candidates for AEC applications.

## 2 Conclusions

In this paper, we have presented several NLMS-based algorithms suitable for AEC applications. These algorithms are based on different control strategies for adjusting their main parameters, i.e., the normalized step-size and regularization term, in order to achieve a proper compromise between the performance criteria (i.e., fast convergence/tracking and low misadjustment). The main motivation behind this approach was the reference work of Hänsler and Schmidt from [1]. Following their ideas, we presented here two related solutions, i.e., the NPVSS-NLMS and JO-NLMS algorithms. The first one (originally proposed in [19]) represents a simple and efficient method to control the normalized step-size. Due to its non-parametric nature, it is a reliable choice in many practical applications. The second one is developed in the context of a state-variable model and follows an optimization criterion based on the minimization of the system misalignment. It is also a non-parametric algorithm, which does not require any additional control features (e.g., system change detector, stability thresholds, etc.). It also gives good robustness against double-talk, which is one of the most challenging situations in AEC. Consequently, it could be an appealing candidate for real-world applications.

There are several perspectives that could follow the ideas presented in this paper. First, the extension to the affine projection algorithm represents a straightforward approach. Second, it would be highly interesting to further develop these solutions in the context of proportionate-type algorithms, which are also attractive choices for sparse system identification.

Concluding, despite the fact that the NLMS algorithm was the workhorse in AEC and also in many other applications, it is still highly studied and very often represents the algorithm of choice in practice. Therefore, let us end this paper with a neat remark of Hänsler and Schmidt from [4], which fits best in this context: “The NLMS algorithm has often been declared to be dead. According to a popular saying, this is an infallible sign of a very long life.”

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

This work was supported by the UEFISCDI Romania under Grant PN-II-RU-TE-2014-4-1880.

### Author details

<sup>1</sup>Department of Telecommunications, University Politehnica of Bucharest, 1-3 Iuliu Maniu Blvd., 061071 Bucharest, Romania. <sup>2</sup>INRS-EMT, University of Quebec, 800 de la Gauchetière Ouest, QC H5A 1K6 Montreal, Canada.

<sup>3</sup>Department of Electrical and Computer Engineering, Missouri University of Science and Technology, 301 W. 16th St., 65409-0040 Rolla, USA.

Received: 30 June 2015 Accepted: 6 November 2015

Published online: 19 November 2015

### References

1. E Hänsler, G Schmidt, *Acoustic Echo and Noise Control—A Practical Approach*. (Wiley, Hoboken, NJ, 2004)
2. E Hänsler, The hands-free telephone problem—an annotated bibliography. *Signal Process.* **27**(3), 259–271 (1992)
3. E Hänsler, G Schmidt, Hands-free telephones—joint control of echo cancellation and post filtering. *Signal Process.* **80**(11), 2295–2305 (2000)
4. E Hänsler, G Schmidt, in *Least-Mean-Square Adaptive Filters*, ed. by S Haykin, B Widrow. Control of LMS-type adaptive filters (Wiley New York, NY, 2003), pp. 175–240
5. E Hänsler, in *Encyclopedia of Telecommunications*, ed. by J Proakis. Acoustic echo cancellation (Wiley New York, NY, 2003), pp. 1–15
6. SL Gay, J Benesty (eds.), *Acoustic Signal Processing for Telecommunication* (Kluwer Academic Publisher, Boston, MA, 2000)
7. J Benesty, T Gänsler, DR Morgan, MM Sondhi, SL Gay, *Advances in Network and Acoustic Echo Cancellation*. (Springer-Verlag, Berlin, Germany, 2001)
8. J Benesty, Y Huang (eds.), *Adaptive Signal Processing—Applications to Real-World Problems* (Springer-Verlag, Berlin, Germany, 2003)
9. G Enzner, H Buchner, A Favrot, F Kuech, in *Academic Press, Library in Signal Processing*, ed. by R Chellappa, S Theodoridis. Acoustic echo control, vol. 4 (Academic Press Chennai, 2014), pp. 807–877
10. B Widrow, SD Stearns, *Adaptive Signal Processing*. (Prentice Hall, Englewood Cliffs, NJ, 1985)
11. S Haykin, *Adaptive Filter Theory*, 4th edn. (Upper Saddle River, NJ, Prentice-Hall, 2002)
12. AH Sayed, *Adaptive Filters*. (Wiley, New York, NY, 2008)
13. C Breining, P Dreiseitel, E Hänsler, A Mader, B Nitsch, H Puder, T Schertler, G Schmidt, J Tilp, Acoustic echo control—an application of very-high-order adaptive filters. *IEEE Signal Processing Mag.* **16**(4), 42–69 (1999)
14. J Benesty, C Paleologu, S Ciochină, On regularization in adaptive filtering. *IEEE Trans. Audio, Speech, Language Processing.* **19**(6), 1734–1742 (2011)
15. A Mader, H Puder, GU Schmidt, Step-size control for acoustic echo cancellation filters—an overview. *Signal Process.* **80**(9), 1697–1719 (2000)
16. AI Sulyman, A Zerguine, Convergence and steady-state analysis of a variable step-size NLMS algorithm. *Signal Process.* **83**(6), 1255–1273 (2003)
17. H-C Shin, AH Sayed, W-J Song, Variable step-size NLMS and affine projection algorithms. *IEEE Signal Processing Lett.* **11**(2), 132–135 (2004)
18. DP Mandic, A generalized normalized gradient descent algorithm. *IEEE Signal Processing Lett.* **11**(2), 115–118 (2004)
19. J Benesty, H Rey, L Rey Vega, S Tressens, A nonparametric VSS-NLMS algorithm. *IEEE Signal Processing Lett.* **13**(10), 581–584 (2006)
20. Y-S Choi, H-C Shin, W-J Song, Robust regularization for normalized LMS algorithms. *IEEE Trans. Circuits and Systems II: Express Briefs.* **53**(8), 627–631 (2006)
21. H Rey, L Rey Vega, S Tressens, J Benesty, Variable explicit regularization in affine projection algorithm: robustness issues and optimal choice. *IEEE Trans. Signal Process.* **55**(5), 2096–2108 (2007)
22. P Park, M Chang, N Kong, Scheduled-step-size NLMS algorithm. *IEEE Signal Processing Lett.* **16**(12), 1055–1058 (2009)
23. H-C Huang, J Lee, A new variable step-size NLMS algorithm and its performance analysis. *IEEE Trans. Signal Process.* **60**(4), 2055–2060 (2012)
24. H-C Huang, J Lee, in *Proc. IEEE Asilomar*. A variable regularization control method for NLMS algorithm (Pacific Grove CA, 2012), pp. 396–400
25. I Song, P Park, A normalized least-mean-square algorithm based on variable-step-size recursion with innovative input data. *IEEE Signal Processing Lett.* **19**(12), 817–820 (2012)
26. S Ciochină, C Paleologu, J Benesty, An optimized NLMS algorithm for system identification. *Signal Process.* **118**(1), 115–121 (2016)
27. DR Morgan, SG Kratzer, On a class of computationally efficient, rapidly converging, generalized NLMS algorithms. *IEEE Signal Processing Lett.* **3**(8), 245–247 (1996)
28. MA Iqbal, SL Grant, in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Novel variable step size NLMS algorithms for echo cancellation (Las Vegas, NV, 2008), pp. 241–244
29. C Paleologu, S Ciochină, J Benesty, Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation. *IEEE Signal Processing Lett.* **15**, 5–8 (2008)

30. G Enzner, P Vary, Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones. *Signal Process.* **86**(6), 1140–1156 (2006)
31. C Paleologu, J Benesty, S Ciochină, Study of the general Kalman filter for echo cancellation. *IEEE Trans. Audio, Speech, Language Processing.* **21**(8), 1539–1549 (2013)
32. L Isserlis, On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika.* **12**(1/2), 134–139 (1918)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---