Review and Comparison of Variable Step-Size LMS Algorithms

Dariusz Bismor, Krzysztof Czyz and Zbigniew Ogonowski

Institute of Automatic Control, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland

(Received 4 December 2013; accepted 11 February 2015)

The inherent feature of the Least Mean Squares (LMS) algorithm is the step size, and it requires careful adjustment. Small step size, required for small excess mean square error, results in slow convergence. Large step size, needed for fast adaptation, may result in loss of stability. Therefore, many modifications of the LMS algorithm, where the step size changes during the adaptation process depending on some particular characteristics, were and are still being developed.

The paper reviews seventeen of the best known variable step-size LMS (VS-LMS) algorithms to the degree of detail that allows to implement them. The performance of the algorithms is compared in three typical applications: parametric identification, line enhancement, and adaptive noise cancellation. The paper suggests also one general modification that can simplify the choice of the upper bound for the step size, which is a crucial parameter for many VS-LMS algorithms.

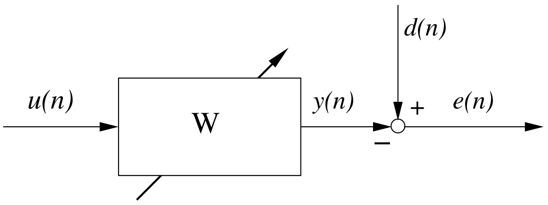

Figure 1. Adaptive filtering problem.

1. INTRODUCTION

In applications where adaptation is needed, the LMS algorithm is probably the most frequently used algorithm. It is simple, fast, and surprisingly robust. Despite its simplicity, the complete mathematical analysis of the LMS algorithm as well as exact rules for the step size adjustment are not currently known, which is probably due to its highly nonlinear character.[1] Therefore, new VS-LMS algorithms appear in the literature every few years with the aim to be useful in practical applications.

The basic block diagram illustrating the LMS algorithm operation is shown on Fig. 1.[1] The adaptive filter **W** is fed with the input sequence $u(n)$. The output of the filter, $y(n)$, is compared with the desired signal, $d(n)$, to produce the error signal, $e(n)$. The algorithm adjusts the filter to minimize the error.

If the adaptive filter is of finite impulse response (FIR) type, with the taps stored in a row vector:

$$\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \ldots\ w_{L-1}(n)]^T\ ; \tag{1}$$

where T denotes transpose, the LMS algorithm updates the filter taps according to the well-known formula:[2]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{u}(n)e(n); \tag{2}$$

where μ is the step size parameter and $\mathbf{u}(n)$ is a row vector containing the input signal samples. The latter may be, depending on the application, of spatial type:

$$\mathbf{u}(n) = [u_0(n)\ u_1(n)\ \ldots u_{L-1}(n)]^T\ ; \tag{3}$$

or of temporal type, with regressive samples of the same input signal:

$$\mathbf{u}(n) = [u(n)\ u(n-1)\ u(n-2)\ \ldots u(n-L+1)]^T\ . \tag{4}$$

The problem with the step size choice can be summarized as follows. Large step size allows for fast adaptation, but also gives large excess mean square error (EMSE, see Section 4.1 for definition). Too large step size may lead to the loss of stability of the system using the LMS algorithm. On the other hand, too small step size gives slow convergence, and even if it results in small excess MSE, it cannot be accepted in many practical applications.

At this point a very important remark should be made about theoretical convergence of the LMS algorithm. First of all, there are different types of convergence,[3] e.g. convergence of the mean (the poorest), convergence in the mean, convergence in the mean square sense, etc. However, if convergence in the mean square sense of the LMS algorithm (2) is desired, and the algorithm operates in real conditions (not noise-free environment), such convergence can only be proved for the vanishing step size, i.e. for $\mu \xrightarrow{n\to\infty} 0$.[3,4] In other words, no constant step-size LMS algorithm can result in convergence in the mean square sense, or stronger. On the other hand, it is possible to bound the EMSE within certain limits, depending on the step size.

The idea of variable step-size is not new. Actually, the Normalized LMS (NLMS) algorithm may be considered as the first variable step-size modification of the LMS, and NLMS was proposed in 1967 independently by Nagumo *et al.*[5] and Albert *et al.*[6] Next VS-LMSes were proposed in 1986 by Harris *et al.*,[7] and by Mikhael *et al.*[8] Many VS-LMS algorithms were developed since then: the search for 'variable step LMS' in article titles only on Scopus or IEEEXplore returns more than 130 publications. The research in this field is by no means finished, new results are still being published.[9,10]

The VS-LMS algorithms may be grouped by the techniques they use to adjust the step size. One of these techniques is the adjustment of the step size based solely on the input signal $u(n)$. Historically, it is the oldest idea, because the NLMS acts in this way.[5] Due to popularity of the NLMS, it may also be considered the most popular technique. Another algorithm using this concept is the one proposed by Mikhael[8] (if FIR filter is considered).

The next technique relates the step size to the error signal. This technique is based on the observation that the step size should be small when the error is small in order to provide small EMSE. On the other hand, when the error is large, it is desirable to operate with large step size to adapt the filter taps fast. This technique was used by Kwong,[11] Aboulnasr,[12] Pazaitis[13], and Zou.[14]

Finally, there is a technique combining the use of the input signal and the error signal. It is a technique used by majority of the algorithms presented in this paper, namely by Harris,[7] Shan,[15] Karni,[16] Benveniste,[17] Evans,[18] Mathews,[19] Ang,[20] Benesty,[21] Wahab,[22] Hwang[23] and Wang.[24] In many solutions this technique is based on the *orthogonality principle*, which states that (under some assumptions) the necessary and sufficient condition for the mean-square error to attain its minimum value is that the error signal $e(n)$ and the input signal $u(n)$ are orthogonal.[1]

The VS-LMS algorithms may also be grouped based on the number of additional parameters which have to be adjusted before running each of the algorithms. Surprisingly, the group of algorithms needing no parameters, i.e. algorithms running fully autonomously, has only two members: Mikhael's algorithm[8] and Wang's algorithm.[24] Next follows the group containing only one parameter to adjust, and this group contains only the NLMS algorithm.[5] Very large is the group of the algorithms needing an upper bound for the step size in addition to one or two parameters; this group includes Shan's algorithms[15] (one of them known as the correlation-LMS), Karni's algorithm,[16] Benveniste's algorithms,[17] and Mathews' algorithms.[19] Three parameters, but without an upper bound for the step size, are also required by Benesty's algorithm.[21] The remaining algorithms are parametrized by more than three values, with the maximum of eight in case of Zou's algorithm.[14]

The goal of this paper is to review the VS-LMS techniques developed by different authors for different uses and to compare them in three typical applications: identification, line enhancement, and adaptive noise cancellation. The paper also suggests how to reduce by one the number of parameters required to run those algorithms which use the upper bound for the step size. The paper is organized as follows. In the next section, all the algorithms considered in this paper are detailed to a degree which allow them to be implemented. However, to make the paper of reasonable length, we will omit all the derivations leading to the final formula as well as all the derivations of MSE, learning curves, etc. Section 3 describes a possible way of reduction of the number of parameters in case of some of the algorithms. Section 4.1 describes the results of simulation of the system identification case. Section 4.2 describes the results of simulation of the line enhancer. Section 4.3 describes the results of simulation of the adaptive noise cancellation system. Finally, some concluding remarks are given in Section 5.

2. VS-LMS ALGORITHMS

In this section, VS-LMS algorithms are described in chronological order. The majority of the algorithms described below use common (scalar) step size, and fall under the equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)\mathbf{u}(n)e(n); \tag{5}$$

where $\mu(n)$ is the (variable) step size calculated with an appropriate formula.

Some of the algorithms allow for individual step sizes for each of the filter taps; such VS-LMS algorithms are described by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{M}(n)\mathbf{u}(n)e(n); \tag{6}$$

where $\mathbf{M}(n)$ is a diagonal matrix of the size equal to the adaptive filter length.

2.1. Normalized LMS Algorithm

The first modification of the LMS algorithm which can be considered a variable step-size modification, was developed by Nagumo and Noda,[5] and independently by Albert and Gardner.[6] However, the name "Normalized LMS" was not proposed by the authors, but appears in the literature much later.[1]

The NLMS algorithm uses the following equation for the step size:

$$\mu(n) = \frac{\bar{\mu}}{\mathbf{u}^T(n)\mathbf{u}(n)} = \frac{\bar{\mu}}{\sum_{i=0}^{L-1} u^2(n-i)}; \tag{7}$$

where $\bar{\mu} > 0$ is a scalar which allows for a change of the adaptation speed. The upper limit for $\bar{\mu} > 0$ to provide stability in some cases is equal to two, but lower values must be used in many practical applications.[1]

The NLMS algorithm may be considered as a standard algorithm for the majority of the adaptive signal processing applications. The normalization of the step size with the input signal power estimate makes the algorithm invulnerable to the input signal power changes; therefore, the algorithm does not require the step size readjustment when such changes occur. On the other hand, the division operation required by the normalization may be time-consuming and required to be avoided in some time-critical applications. Nevertheless, the time overhead introduced by the step-size normalization is the smallest when compared to the overhead introduced by other VS-LMS algorithms discussed below.

2.2. VS-LMS Algorithm by Harris *et al.*

The algorithm proposed by Harris *et al.* uses individual step sizes, and the step-size matrix $\mathbf{M}(n)$ is constructed as:

$$\begin{gathered}\mathbf{M}(n) = 2 \cdot \mathrm{diag}\left[\mu_0(n),\ \mu_1(n),\ \ldots \mu_{L-1}(n)\right]; \\ \mu_{min} < \mu_i(n) < \mu_{max}, \quad i = 0 \ldots L-1.\end{gathered} \tag{8}$$

The authors suggest that $\mu_{max} < 1/\lambda_{max}$, where λ_{max} is the maximum eigenvalue of the autocorrelation matrix $\mathbf{R}_{uu}$ of the input signal. μ_{min}, on the other hand, should be chosen to provide the desired steady-state misadjustment.

With the above restrictions, the individual step sizes are changed as follows:

$$\mu_i(n{+}1) = \begin{cases} \mu_i(n)/\alpha & \text{if } e(n)u(n-i) \text{ alternates sign on } m_0 \text{ successive samples;} \\ \alpha\mu_i(n) & \text{if } e(n)u(n-i) \text{ has the same sign on } m_1 \text{ successive samples;} \\ \mu_i(n) & \text{otherwise.} \end{cases} \tag{9}$$

The authors suggest that the typical value for α is 2.

The authors distinguish the following useful cases:

- $m_0 = 1, m_1 = \infty$ — the algorithm decreases the step sizes with every sign change,
- $m_0 > 1, m_1 = \infty$ — the algorithm decreases the step sizes if m_0 consecutive sign changes occur,
- $m_1 > m_0 > 0$ — such algorithm results in general decrease of the step sizes; the authors claim this is useful for stationary input signal processing,
- $m_0 = m_1 \geq 1$ — the symmetric increase or decrease algorithm; the authors claim this is useful for nonstationary input signal processing.

This algorithm is parametrized by five parameters, including the upper bound for the step size.

The algorithm was developed in times when the hardware was many times less powerful than nowadays. The authors admit that they tried to achieve performance comparable with the Recoursive Leas Squares (RLS) algorithm, which was too hardware-demanding to use. The authors claim that the VS-LMS algorithm given above can “provide faster convergence and less misadjustment than the constant μ or LMS algorithm,” but that the RLS-type algorithms are still faster to converge. It must be noted, however, that this goal was achieved at the price of four additional parameters, which makes the algorithm difficult to adjust.

This algorithm is also one of the two algorithms that proposed for the first time the individual step sizes for each of the filter taps.

2.3. VS-LMS Algorithm by Mikhael *et al.*

The algorithm proposed by Mikhael *et al.*[8] can be used for FIR filters as well as for infinite impulse response (IIR) filters. It also comes in two flavors: with scalar step size (which the authors call *homogeneous adaptation*), and with individual step sizes. To allow comparison with other algorithms discussed in this paper, we will introduce the FIR version only. Moreover, only individual step sizes will be used. Thus, the algorithm is given by:

$$\mathbf{M}(n) = \mathrm{diag}\left[\mu_0(n),\ \mu_1(n),\ \ldots \mu_{L-1}(n)\right];$$

$$\mu_i(n) = \frac{0.5\,|u(n-i)|}{\sum\limits_{l=0}^{L-1} |u(n-l)|^3}, \quad i = 0 \ldots L-1. \tag{10}$$

As mentioned in the introduction, this algorithm is one of the two that *do not require any parameters to run*, and it is its main advantage. The algorithm is also the second of the two algorithms that proposed the individual step sizes (both the algorithms appeared in the same year). Although the authors allow also for scalar step sizes, they insist that the individual step sizes give better results.

The authors applied the algorithm to the classical adaptive noise cancellation problem (see Section 4.3), with the stationary, white Gaussian noise input. The algorithm resulted in faster convergence rate than the constant step size LMS algorithm.

2.4. VS-LMS Algorithm by Shan *et al.*

The original algorithm by Shan *et al.*[15] is given by:

$$\mu(n+1) = \frac{\alpha|\rho(n)|}{\mathbf{u}^T(n)\mathbf{u}(n)};$$

$$\rho(n) = \lambda\rho(n) + (1-\lambda)\bar{u}(n)e(n);$$
$$\bar{u}(n) = \frac{1}{L}\sum_{l=0}^{L-1} u(n-l); \tag{11}$$

where α is a scaling factor, $\rho(n)$ is an estimate of the correlation between the input and the error at time instance n, λ is a forgetting factor used in calculation of the correlation, and $\bar{u}(n)$ is the mean value of the elements in the input vector. Thus, the algorithm uses the orthogonality principle.[1] Typically, the forgetting factor λ is in range from 0.9 to 1. The value of the scaling factor α, on the other hand, should be chosen experimentally with the knowledge that it affects both the tracking capability and the speed of the convergence of the algorithm.

In its first published form, the original algorithm did not include the absolute value in the numerator of Eq. (11). After comments on their paper[25] the authors added the absolute value and agreed that μ should be bounded ($\mu(n) < \mu_{max}$).[26]

The algorithm by Shan *et al.* was simplified afterwards to the form:

$$\mu(n+1) = \frac{\alpha|\rho(n)|}{\mathbf{u}^T(n)\mathbf{u}(n)};$$

$$\rho(n) = \lambda\rho(n) + (1-\lambda)u(n)e(n); \tag{12}$$

which we currently know under the name *correlation LMS*. The only difference is that the mean value of the elements in the input vector is substituted with the instantaneous input value.

Both the original Shan’s algorithm and the correlation LMS are members of the group of algorithms requiring no more than two parameters to adjust in addition to the upper bound for the step size, which is still a reasonable number. The authors claim that their algorithm was developed to address the issue of the robustness of the LMS algorithm to disturbances present in real adaptive systems and to the sudden changes in the noise level. As a response, the algorithm with adaptive gain control similar to the gain control of the RLS algorithm was proposed. Thus, the algorithm is “insensitive to disturbances but sensitive to system changes”, i.e. with tracking capabilities.[15]

2.5. VS-LMS Algorithm by Karni *et al.*

The algorithm by Karni *et al.*[16] is given by:

$$0 \le \mu(n) \le \mu_{max};$$
$$\mu(n) = \mu_{max}\left(1 - e^{-\alpha\|\mathbf{u}(n)e(n)\|^2}\right); \qquad (13)$$

where $\alpha > 0$ is the dumping factor. The authors suggest that the $\alpha > 1$ and observe that for $\alpha \to \infty$ the algorithm degenerates to the conventional LMS, with constant step size equal to μ_{max}.

Karni's algorithm requires only one parameter and the upper bound for the step size to run.

The algorithm was developed to address the issue of high misadjustment that is a side effect of fast convergence, or more precisely, of using large step sizes. The authors compare their algorithm with a two-stage method, which uses large values of μ during the initial stage of adaptation to speed up the convergence, and smaller values afterwards to minimize the misadjustment. The authors claim that their algorithm gives better results (both the convergence speed and the misadjustment), especially under nonstationary environment conditions.

2.6. VS-LMS Algorithm by Benveniste *et al.*

Benveniste *et al.* proposed a VS-LMS algorithm in the book.[17] The author of this work failed to find this book; however, as the algorithm was cited by many authors, it is reproduced here after reference[20] and.[27] The algorithm may be used with scalar step size, and in this case it is defined as:

$$\mu(n) = \mu(n-1) + \rho e(n)\mathbf{u}^T(n)\boldsymbol{\phi}(n); \quad \mu(n) < \mu_{max}, \qquad (14)$$
$$\boldsymbol{\phi}(n+1) = \boldsymbol{\phi}(n) - \mu(n)\mathbf{u}(n)\mathbf{u}^T(n)\boldsymbol{\phi}(n) + \mathbf{u}(n)e(n); \qquad (15)$$

where ρ is a small positive value used to control the convergence and MSE, and $\boldsymbol{\phi}(n)$ is an estimate of the gradient vector — the vector of derivatives of the filter coefficients with the respect to the step size. After reference[20] we observe that a larger value of ρ results in faster convergence and may result in lower overall MSE, but may also cause a large fluctuations of MSE in short range of time; we also observe that the behavior of the algorithm is not influenced very much if the value of ρ varies in range $[10^{-4} \dots 10^{-2}]$.

Benveniste's algorithm can also be used in individual step-sizes form. In this case, the step-size matrix is constructed as:

$$\mathbf{M}(n) = \mathrm{diag}\left[\mu_0(n),\ \mu_1(n),\ \dots \mu_{L-1}(n)\right]; \qquad (16)$$
$$0 < \mu_i(n) < \mu_{max}, \quad i = 0 \dots L-1;$$

where:

$$\mu_i(n) = \mu_i(n-1) + \rho e(n)u(n-i)\phi_i(n); \qquad (17)$$
$$\phi_i(n+1) = \phi_i(n) - \mu_i(n)|u(n-i)|^2\phi_i(n) + u(n-i)e(n). \qquad (18)$$

In both its forms, Benveniste's algorithm requires one parameter and the upper bound for the step size to run.

2.7. VS-LMS Algorithm by Kwong *et al.*

The algorithm given by Kwong *et al.*[11] is given by:

$$\mu'(n+1) = \alpha\mu(n) + \gamma e^2(n);$$
$$\mu(n+1) = \begin{cases} \mu_{max} & \text{if } \mu'(n+1) > \mu_{max}; \\ \mu_{min} & \text{if } \mu'(n+1) < \mu_{min}; \\ \mu'(n) & \text{otherwise}; \end{cases} \qquad (19)$$

where $0 < \alpha < 1$ and $\gamma > 0$ are tuning parameters, and $0 < \mu_{min} < \mu_{max}$ are chosen to provide tracking capability and stability, respectively. The authors claim that typical value for α, working well in many simulations they performed, is 0.97. The parameter γ influences both the speed of convergence and the EMSE, and it should be small — $4.8 \cdot 10^{-4}$ was used in the original paper.

The algorithm was developed with the goal to have the step size dependent on the square of the error. The authors' motivation was that the error is frequently large when fast adaptation is required, while the step size may be lowered when the error becomes low. Moreover, simple relation of the step size and the error made it possible to provide theoretical analysis of this VS-LMS algorithm, even in case of the nonstationary environment. The authors compared their results with the results obtained with the algorithms by Harris (Sec. 2.2), and obtained comparable performence.

Kwong's algorithm requires four parameters to run, including the upper bound for the step size. Two of them are the maximum and minimum step size, which usually do not interact with other parameters; therefore, the number of the parameters to adjust is still reasonable.

2.8. VS-LMS Algorithm by Evans *et al.*

The paper by Evans *et al.*[18] discusses two similar algorithms. The first one is identical with Harris's algorithm described in Section 2.2, the second differs only in the step size update equation (9), which is additive rather than multiplicative:

$$\mu_i(n{+}1) = \begin{cases} \mu_i(n) - \alpha & \text{if } e(n)u(n-i) \text{ alternates sign} \\ & \text{on } m_0 \text{ successive samples;} \\ \mu_i(n) + \alpha & \text{if } e(n)u(n-i) \text{ has the same sign} \\ & \text{on } m_1 \text{ successive samples;} \\ \mu_i(n) & \text{otherwise.} \end{cases} \qquad (20)$$

Naturally, the value α can no longer be equal to 2, but should be chosen as some small value (2^{-10} was used for the experiments presented by the authors). The authors claim that this algorithm gives slightly better performance that the algorithm defined by (9). The number of parameters parametrizing the algorithm is identical as in the case of Harris's algorithm.

2.9. VS-LMS Algorithm by Mathews *et al.*

Mathews's algorithm[19] is another algorithm that can use both the individual and common step sizes. In the first case the step-size matrix is again a diagonal matrix with the upper bound on each of the diagonal elements, as in Eq. (16). The individual step sizes are calculated as:

$$\mu_i(n) = \mu_i(n-1) + \rho e(n)e(n-1)u(n-i)u(n-i-1); \qquad (21)$$

where ρ is a small positive value allowing to control the convergence process. The authors used the values in range $10^{-4} \leq \rho \leq 6 \cdot 10^{-3}$ and claim that the choice does not influence the steady-state performance.

In the second case, common step size (also subject to the upper bound μ_{max}) is updated as:

$$\mu(n) = \mu(n-1) + \rho e(n)e(n-1)\mathbf{u}^T(n-1)\mathbf{u}(n). \qquad (22)$$

For some hypothetical input sequences, the algorithm may fail to start when $\mu(0) = 0$; therefore, the authors suggest to initialize the step size with some small value. The choice of $\mu(0) > 0$ may also improve the initial speed of convergence of the algorithm.

The concept the algorithm is based on is to change the step size in a manner proportional to the negative of the gradient of the squared estimation error, with respect to the step size. An algorithm offering very good converge speed and small misadjustment resulted. Moreover, the authors claim that this algorithm offers "close-to-the-best-possible performance" when applied in nonstationary conditions.[19]

This algorithm requires two parameters and the upper bound for the step size to run, but one of them is the minimum step size, and the authors claim that the choice the other (ρ) is not critical.

2.10. VS-LMS Algorithm by Aboulnasr *et al.*

The algorithm by Aboulnasr *et al.*[12] is based on the algorithm by Kwong (see Eq. (19)) and the observation that the instantaneous energy of the error signal it uses may lead to unexpected behavior in the presence of measurement noise. Therefore, Aboulnasr proposes to calculate an estimate of the autocorrelation between $e(n)$ and $e(n-1)$, and use this estimate to control the step size instead of $e^2(n)$. The estimate may be calculated as:

$$p(n) = \beta p(n-1) + (1-\beta)\, e(n)e(n-1); \qquad (23)$$

where $0 \ll \beta < 1$ is an exponential weighing factor controlling the averaging process. Then, the estimate should be used in calculation of $\mu'(n+1)$ as:

$$\mu'(n+1) = \alpha\mu(n) + \gamma p^2(n). \qquad (24)$$

Finally, the same limits on the step size should be applied as in Eq. (19). Also, the meaning of the parameters α and γ is the same as in the case of Kwong's algorithm.

As stated previously, the proposed modification introduced by this algorithm makes it immune to the uncorrelated measurement noise. The authors claim that this raises the performance of the algorithm in case of stationary environments, and offers performance comparable with the algorithms by Kwong and Mathews in case of nonstationary environments. Unfortunately, the modification rises the total number of parameters parametrizing the algorithm up to five, which is a number unsuitable for many practical applications.[28]

2.11. VS-LMS Algorithm by Pazaitis *et al.*

A very original algorithm by Pazaitis *et al.*[13] uses the kurtosis of the error signal to control the step size. The authors propose two different equations for the step size update:

$$\mu(n) = \alpha|\text{Kurt}\,(e(n))\,|; \qquad (25)$$

and

$$\mu(n) = \mu_{max}\left(1 - e^{-\alpha|\text{Kurt}(e(n))|}\right); \qquad (26)$$

where α is a positive constant controlling both the speed of convergence and the excess MSE. The authors suggest that the value of α should be in range $[1, 10]$ for the update according to Eq. (26), whereas it should be approximately an order smaller for the update according to Eq. (25). The authors also observe that the simpler-to-use update by Eq. (25) may require additional upper bound to ensure convergence; therefore, only the algorithm defined by Eq. (26) will be considered in this paper.

To apply Eqs. (25)–(26) in practice, an estimate of the kurtosis must be calculated. The authors suggest to calculate this estimate as:

$$C_4^e(n) = \hat{E}\left\{e^4(n)\right\} - \rho(n)\hat{E}^2\left\{e^2(n)\right\}; \qquad (27)$$

where $C_4^e(n)$ is the estimate, $\hat{E}$ is an estimate of the expected value, and $\rho(n)$ is used to trace the changes in the characteristics of the noise. The latter factor should be calculated recursively as:

$$\rho(n+1) = \rho(n) +$$
$$\delta \text{ sgn}\left(\hat{E}\left\{e^4(n)\right\} - \rho(n)\hat{E}^2\left\{e^2(n)\right\}\right)\hat{E}^2\left\{e^2(n)\right\}; \qquad (28)$$

where δ is a small positive constant (the authors used $\delta = 0.01$ in their simulations).

The authors suggest to calculate the estimates of the expected values using rectangular window of some size, or to calculate them recursively as:

$$\hat{E}\left\{e^{2i}(n)\right\} = \beta\hat{E}\left\{e^{2i}(n-1)\right\} + (1-\beta)e^{2i}(n), \quad i = 1, 2 \qquad (29)$$

(this paper uses the recursive calculation of the estimates).

Both the rectangular window size and the forgetting factor β are additional parameters that need to be selected before application of the algorithm; therefore, the number of parameters to adjust is equal to four, including the upper bound for the step size. This number is rather high and may be unsuitable for some practical applications.

The main advantage of the algorithms lies in the originality of using higher order statistics to adjust the step size. In theory, this should result in the algorithm immune to the disturbance noise, if the noise is Gaussian. However, the main advantage is also a main disadvantage, as in practice higher order statistics are hard to estimate.

2.12. VS-LMS Algorithm by Ang *et al.*

Ang *et al.*[20] proposed a simplification of Benveniste's algorithm given by Eqs. (14)–(15). The simplification applies to the gradient vector Eq. (15), which is now calculated as:

$$\boldsymbol{\phi}(n+1) = \alpha\boldsymbol{\phi}(n) + \mathbf{u}(n)e(n); \qquad (30)$$

where α is a constant smaller than but close to one. The authors explain that the algorithm may be viewed as a filtering of the noisy gradient of $\mathbf{u}(n)e(n)$ with a first-order low-pass filter with transfer function $\frac{1}{1-\alpha z^{-1}}$. This should result in a more stable adaptation of the step size. Moreover, for $\alpha = 0$, this algorithm is reduced to Mathews's algorithm (Eq. (22)).

Ang's algorithm may also be used with the individual step sizes, in which case Eq. (18) is simplified to:

$$\phi_i(n+1) = \alpha\phi_i(n) + u(n-i)e(n). \tag{31}$$

Unfortunately, the modification by Ang *et al.* adds another parameter when comparing with Benveniste's algorithm, so the total number of parameters is equal to three, including the upper bound for the step size.

2.13. VS-LMS Algorithm by Benesty *et al.*

The algorithm by Benesty *et al.*[21] is given by:

$$\mu'(n) = \frac{1}{\delta + \mathbf{u}^T(n)\mathbf{u}(n)}\left(1 - \frac{\sigma_v}{\epsilon + \hat{\sigma}_e(n)}\right);$$
$$\mu(n) = \begin{cases} \mu'(n) & \text{if } \hat{\sigma}_e(n) \geq \sigma_v, \\ 0 & \text{otherwise,} \end{cases}; \tag{32}$$

where σ_v is standard deviation of the system noise (see below), $\hat{\sigma}_e(n)$ is an estimate of standard deviation of the error, and δ and ϵ are small positive constants to avoid division by zero.

The authors suggest to estimate the power of the error signal using the recursion:

$$\hat{\sigma}_e^2(n) = \lambda\hat{\sigma}_e^2(n-1) + (1-\lambda)e^2(n); \tag{33}$$

where the exponential parameter λ can be calculated as $\lambda = 1 - \frac{1}{KL}$, with $K \geq 2$.

The concept of the system noise $v(n)$ comes from an assumption that the desired signal $d(n)$ is a sum of the response of some optimal filter W_o and the zero-mean noise $v(n)$:

$$d(n) = \mathbf{w}_o^T\mathbf{u}(n) + v(n); \tag{34}$$

in this case, $\sigma_v^2 = E[v^2(n)]$. In some applications, the standard deviation of the system noise may be known before running the algorithm, but usually it must also be estimated, e.g. using an a posteriori error:

$$\varepsilon(n) = y(n) - \mathbf{w}^T(n+1)\mathbf{u}(n). \tag{35}$$

The estimation of $\hat{\sigma}_v^2 = E[\varepsilon^2(n)]$ may be performed using the recursion in Eq. (33).

The authors showed that the proposed algorithm performs better in acoustic echo cancellation simulations than the NLMS, giving faster convergence and lower excess MSE. However, the authors assumed the knowledge of the variance of the system noise σ_v^2 for these experimets.

The total number of parameters required by this algorithm may be different in different implementations. In the optimistic case, when we fix the δ and ϵ at some small value, and when we estimate both the variances with the same forgetting factor, only one parameter remains to be adjusted (the forgetting factor or the K multiplier). However, if we assume the knowledge of σ_v^2, this variance becomes the second parameter. The second parameter also appears when we decide to estimate both the variances with different forgetting factors. Thus, the author's claim that the algorithm is "nonparametric" should be treated as an exaggeration.

2.14. VS-LMS Algorithm by Wahab *et al.*

The algorithm by Wahab *et al.*[22] was developed by calculation of the optimal step size that minimizes the cost function defined as a quadratic norm of the difference between the new filter weights $w(n+1)$ and the optimal ones. Practical implementation of their algorithm is given by:

$$\mu(n) = \frac{e^2(n) - e(n)\hat{v}(n)}{e^2(n)\|\mathbf{u}(n)\|^2}; \tag{36}$$

where $\hat{v}(n)$ is an estimate of the noise defined as in Eq. (34). The authors propose to estimate the noise as:

$$\hat{v}(n) = g(n)e(n); \tag{37}$$

where

$$g(n) = 1 - \exp\left(\frac{\alpha\hat{\sigma}_v^2}{\hat{\sigma}_u^2(n)(\rho\hat{\sigma}_e^2(n) - \hat{\sigma}_v^2)}\right); \tag{38}$$

where $\hat{\sigma}_v^2$ is the system noise power, $\sigma_u^2(n)$ is the input signal power, $\sigma_e^2(n)$ is the error signal power, $\rho \geq 1$ and α are positive constants used to make a compromise between tracking capabilities and the EMSE (for calculation of $\hat{\sigma}_v^2$ and $\hat{\sigma}_e^2$ see Section 2.13). The authors suggest to estimate the input signal power using rectangular window: $\hat{\sigma}_u^2(n) = \|\mathbf{u}(n)\|^2/L$.

The authors claim that the derivation of their algorithm takes the disturbance noise into account; therefore their algorithm outperforms the algorithms by Ang or Aboulnasr in noisy environments. Moreover, the authors claim that their algorithm is less influenced by inproper choice of the adjustment parameters.

Please note that the algorithm by Wahab is equivalent to the NLMS in case of noise-free environment ($v(n) \equiv 0$). Note also the possible problems in practical implementation of the algorithm introduced by two subtractions, which may make the step size negative. Another disadvantage is that this algorithm is parametrized by as many as four parameters.

2.15. VS-LMS Algorithm by Hwang *et al.*

The algorithm by Hwang *et al.*[23] is based on the algorithm by Kwong given by Eq. (19). The authors observed that the parameter γ from Eq. (19) should be large when fast convergence is required (e.g. at the start of an operation), but should be small during steady-state operation and when the system noise is high. The authors suggest to adjust this parameter using an estimate of the correlation between the input and the error in the vector form:

$$\hat{\mathbf{p}}(n) = \beta\hat{\mathbf{p}}(n-1) + (1-\beta)\mathbf{u}(n)e(n); \tag{39}$$

where $0 < \beta < 1$ is an exponential parameter. Then, the step size should be adjusted as:

$$\mu(n+1) = \alpha\mu(n) + \gamma_s\|\hat{\mathbf{p}}(n)\|^2e^2(n); \tag{40}$$

with $\gamma_s > 0$.

The authors demonstrate the advantages of the algorithm using adaptive channel estimation application: the algorithm converges fast despite of large ranges of γ_s and different levels of noise.

Unfortunately, the modification raises the number of parameters required to run the algorithm up to five, including the upper bound for the step size. This may create a serious disadvantage for many practical applications.[28]

2.16. VS-LMS Algorithm by Wang *et al.*

Algorithm by Wang *et al.*[24] is given by:

$$\mu'(n) = \frac{\sum_{i=1}^{n} \varepsilon(i)e(i-1)\mathbf{u}^T(i-1)\mathbf{u}(i)}{\sum_{i=1}^{n} e^2(i-1)\mathbf{u}^T(i-1)\mathbf{u}(i)}; \qquad (41)$$

$$\mu_{min} < \mu'(n) < \mu_{max};$$

$$\mu(n) = \frac{\mu'(n)}{\mathbf{u}^T(n)\mathbf{u}(n)}; \qquad (42)$$

where:

$$\varepsilon(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{u}(n). \qquad (43)$$

The authors suggest that the lower bound on $\mu'(n)$ may be set to any small positive value, e.g. 10^{-10}, and the upper bound may be set to $\mu_{max} = \mathbf{u}^T(n)\mathbf{u}(n)$. If such values are used, the algorithm is the other algorithm of the two mentioned in Section 1 that *do not require any parameters to run*, which is its greatest advantage. Judging by the simulation results the authors present, this algorithm offers performance comparable with other VS-LMS algorithms, e.g. Aboulnasr's algorithm.

2.17. VS-LMS Algorithm by Zou *et al.*

The algorithm by Zou *et al.*[14] combines the algorithms by Kwong (Eq. (19)) and by Aboulnasr (Eq. (24)) and is composed of three steps. In the first step, the new step size is calculated as:

$$\mu(n+1) = \alpha\mu(n) + \gamma p^2(n);$$
$$\mu_{min} < \mu(n+1) < \mu_{max}; \qquad (44)$$

where $0 < \alpha < 1$ and $\gamma > 0$ are tuning parameters (as in Aboulnasr's algorithm). In the next step, the time-averaged correlation between two successive error signal samples $p(n)$ is updated as:

$$p(n+1) = (1-\beta(n))\,p(n) + \beta(n)e(n)e(n-1). \qquad (45)$$

Finally, the new time-averaged error signal power $\beta(n)$ is calculated as:

$$\beta(n+1) = \eta\beta(n) + \lambda e^2(n);$$
$$\beta_{min} < \beta(n+1) < \beta_{max}; \qquad (46)$$

where $0 < \eta < 1$ and $\lambda > 0$ are tuning parameters (as in Kwong's algorithm).

The authors observe that $0 < \beta(n) < 1$; therefore, one should choose $\beta_{max} < 1$. The authors claim that the algorithm combines the advantages of the algorithms by Kwong and Aboulnasr: good ability to cope with the noise and good tracking capability. Moreover, the authors claim that very good convergence speed and low misadjustment are achieved, too. However, the algorithm requires *eight parameters* to be adjusted—this is definitely too many to use the algorithm in practical applications.[28]

3. UPPER BOUND FOR THE STEP SIZE

The majority of the algorithms described above require the upper bound for the step size μ_{max}. In fact, the only algorithms that do not require the choice of μ_{max} are NLMS and algorithms by Mikhael (Sec. 2.3), by Benesty (Sec. 2.13) and by Wahab (Sec. 2.14). Please note that μ_{max} must be chosen very carefully in order to guarantee convergence as well as not impose too much restrictions on the step size. Moreover, the upper bound for the step size may depend on the experiment conditions (see Section 4.3 for an example). This is quite opposite of the minimum step size, which can be chosen very roughly.

It is possible to simplify the choice of the upper bound for the step size by assuming it to be similar to the LMS algorithm stability sufficient condition:[29]

$$\mu_{\text{max}}(n) = \varepsilon\frac{2}{\mathbf{u}^T(n)\mathbf{u}(n)}; \qquad (47)$$

where $0 < \varepsilon < 1$ is a scaling factor. Although it may seem as substitution of one upper bound by another, there are substantial differences. Namely, contrary to the (constant) μ_{max}, the ε factor may be roughly chosen from a well-defined range $(0\ldots 1)$ (theoretically) or $[0.5\ldots 0.9]$ (in practice). For example, a value 0.8 (assumed for the experiments presented below) or 0.9 will work well in many typical applications. The choice of ε is so easy that we may assume that this technique allows us to lower the number of parameters for each algorithm using μ_{max} by one. On the other hand, calculation of the $\mu_{\text{max}}(n)$ in each iteration requires additional processor power.

To check how the proposed modification influences stability and performance of the VS-LMS algorithms, in the next section we will show simulations with constant upper bound for the step size as well as with modification defined by Eq. (47).

4. SIMULATIONS

The VS-LMS algorithms described in the previous section have been tested in three typical applications: system identification, adaptive line enhancement (ALE) and adaptive noise cancellation (ANC).[33] The main problem that occurred during the simulations and concerned especially algorithms with more than two parameters was that it was not easy to adjust the parameters because of parameter interactions. This remark does not apply to the maximum and minimum step sizes, which are relatively easy to choose.

In this sense, the most difficult algorithms to parametrize were those by Aboulnasr, Hwang, Zou, and Wahab. Algorithms presenting moderate complication (two parameters to adjust, except for the minimum and maximum step sizes) are those by Kwong, Shan, Ang and Benesty. This group also includes algorithms by Harris and Evans, which have three parameters, but two of them are integers in the range from 1 to 3.

Another problem was the need to readjust algorithms for different applications. Moreover, the need to readjust the algorithm for one application (ANC) after changing the noise level has also arisen during the experiments.

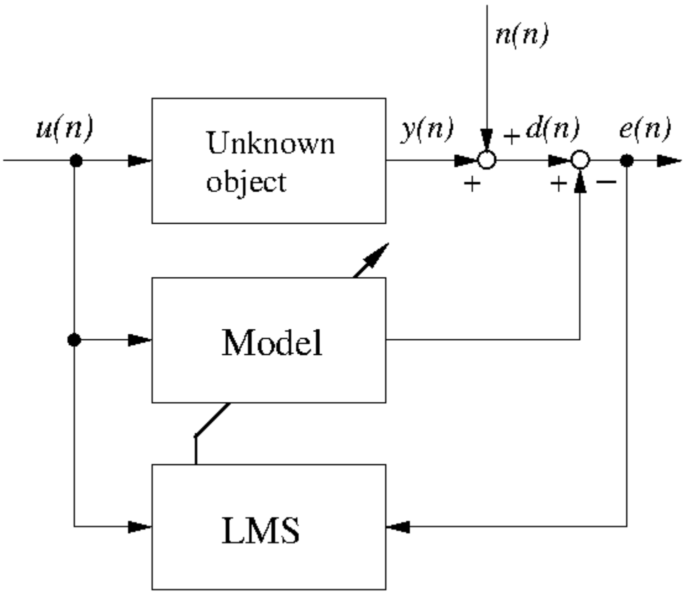

Figure 2. System Identification block diagram.

Despite these difficulties, an effort was taken to adjust the algorithms in a way that allows them to obtain the best possible results with regards to the quantities discussed below. One more remark must be observed: the adjustment of the parameters was performed with the aim of obtaining the behavior which really exhibits the step-size variation. This remark is important because it was very easy to adjust the parameters in the way that the VS-LMS algorithm operated with the (almost) maximum allowed step size. Moreover, this behavior sometimes minimized the sum of MSE — mainly due to maximization of the speed of convergence. However, it did not allow it to minimize the misadjustment and variation of the parameters of an adaptive filter during the steady state. Therefore, such adjustment was considered wrong.

4.1. Simulation of System Identification

The first application the VS-LMS algorithms were tested against is system identification — see Fig. 2. In this application, the LMS algorithm input signal $u(n)$ is the signal used to excite the unknown object, while the LMS algorithm desired signal $d(n)$ is a sum of the object output signal $y(n)$ and the output noise $n(n)$ (the disturbance). Observe that in this case the output noise is the same as the system noise $v(n)$ in Eq. (34). This desired signal is compared to the output of the identified model to produce the error signal $e(n)$.

System identification presents many challenges,[30] but for the purpose of this paper, a relatively simple case of identification of white-noise excited FIR filter was chosen. Moreover, the exact match in the structure of the object and the model was assumed (no modeling error). This, together with the control over the output noise power, allowed us to observe some important differences in the behavior of the VS-LMS algorithms.

It should be noted that in case of an identification of a stationary FIR object, the LMS filter should converge to the Wiener solution.[1] Therefore, it is possible to perform the zero-order analysis of the steady-state phase of the LMS adaptation, with the assumption of the small step size. If we define the instantaneous MSE as:

$$J(n) = E(|e(n)|^2); \tag{48}$$

we may express this MSE as:

$$J(n) = J_{\text{min}} + J_{\text{ex}}(n); \tag{49}$$

where J_{min} is the minimum achievable error, and $J_{\text{ex}}(n)$ is called excess MSE (EMSE).

Assuming that the identified model converges to the original filter, the minimum achievable MSE is equal to the variance of the system noise $n(n)$:

$$J_{\text{min}} = E(n^2(n)). \tag{50}$$

Furthermore, form the small step-size theory it follows that in the steady state the MSE can be expressed as:

$$J(\infty) = J_{\text{min}} + \mu J_{\text{min}} \sum_{k=1}^{L} \frac{\lambda_k}{2 - \mu\lambda_k} \approx J_{\text{min}} + \frac{\mu J_{\text{min}}}{2} \sum_{k=1}^{L} \lambda_k; \tag{51}$$

where λ_k are the eigenvalues of the input signal autocorrelation matrix.[1] Thus, the EMSE is approximately equal to:

$$J_{\text{ex}}(\infty) \approx \frac{\mu J_{\text{min}}}{2} \sum_{k=1}^{L} \lambda_k; \tag{52}$$

which means that it depends on the variance of the system noise. For this reason it is convenient to define additional quantity, called *misadjustment*, as:

$$\mathcal{M} = \frac{J_{\text{ex}}(\infty)}{J_{\text{min}}} \approx \frac{\mu}{2} \sum_{k=1}^{L} \lambda_k. \tag{53}$$

Please note that the misadjustment depends only on the step size and the properties of the input signal $u(n)$ (in the form of the eigenvalues of the autocorrelation matrix), and *does not* depend on the system noise $n(n)$.

In the experiments described below EMSE was used to evaluate each of the algorithms in a quantitative manner in the form of the sum of the EMSE for the whole experiment:

$$\sum \text{EMSE} = \sum_{n=0}^{N} J_{\text{ex}}(n); \tag{54}$$

where N is the number of iterations in each experiment ($N = 4096$ in case of identification experiments). This parameter shows overall performance of the algorithm, combining the speed of convergence and the resulting EMSE. However, to observe the last two quantities separately, the estimate of the misadjustment was also calculated, as the mean value of 100 samples of EMSE, divided by the variance of the system noise. Time of convergence was calculated as a number of samples after which the EMSE falls below the value 10^{-2}:

$$\tau = \min_{n} \left[J_{\text{ex}}(n) < 10^{-2} \right]. \tag{55}$$

The last quantity calculated for evaluation of the identification performance was the variance of two of the identified

Table 1. Results of identification experiments averaged over 250 runs for SNR = 20 dB. A dash in the 'Modif' column means that the particular algorithm does not use the upper bound for the step size; and thus the modification proposed in Section 3 does not apply.

Algorithm	Excess MSE		End EMSE		Misadjustment		Time		var(w_1)		var(w_2)	
	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.
LMS	357.9	—	4.93e-04	—	4.93e-02	—	329	—	5.02e-05	—	1.91e-05	—
NLMS	356.4	—	6.27e-04	—	6.27e-02	—	324	—	3.71e-05	—	3.89e-05	—
Harris	612.6	344.8	1.14e-03	2.60e-03	1.14e-01	2.60e-01	1115	337	7.04e-05	3.05e-04	1.47e-05	2.62e-05
Mikhael	107.7	—	4.43e-03	—	4.43e-01	—	96	—	5.86e-04	—	4.91e-04	—
Kwong	145.2	107.0	1.48e-04	6.56e-05	1.48e-02	6.56e-03	144	87	5.23e-06	1.80e-06	1.94e-06	8.95e-07
Evans	201.8	117.9	9.20e-04	2.60e-03	9.20e-02	2.60e-01	149	92	1.63e-05	2.00e-04	6.95e-05	2.51e-04
Shan_correlation	174.9	145.4	2.15e-04	8.61e-05	2.15e-02	8.61e-03	131	88	1.03e-05	1.52e-05	3.85e-06	1.52e-05
Shan	239.2	203.0	1.44e-03	2.26e-03	1.44e-01	2.26e-01	201	110	7.11e-07	1.82e-07	4.88e-06	1.40e-05
Benveniste	162.5	112.8	2.23e-04	5.08e-04	2.23e-02	5.08e-02	147	100	6.83e-06	5.27e-05	1.63e-05	6.64e-05
Benveniste_indiv	156.2	∞	4.25e-04	∞	4.25e-02	∞	199	0	8.12e-06	2.07e-04	2.06e-05	5.67e-05
Karni	146.2	116.7	4.19e-04	2.51e-03	4.19e-02	2.51e-01	194	106	7.69e-05	1.91e-04	1.88e-05	2.48e-04
Mathews	223.3	149.8	6.62e-04	7.46e-04	6.62e-02	7.46e-02	204	152	9.11e-06	1.25e-05	6.39e-05	3.15e-06
Mathews_indiv	178.8	126.6	1.11e-03	2.58e-03	1.11e-01	2.58e-01	187	165	1.71e-05	2.87e-04	6.60e-05	3.09e-04
Aboulnasr	164.4	145.7	1.49e-04	1.39e-04	1.49e-02	1.39e-02	134	149	2.13e-09	4.41e-09	8.37e-09	1.10e-09
Ang	160.5	121.7	1.01e-04	2.85e-04	1.01e-02	2.85e-02	145	128	1.25e-06	4.72e-05	1.72e-06	4.29e-05
Ang_indiv	152.3	∞	3.75e-04	∞	3.75e-02	∞	174	0	2.27e-06	1.53e-04	1.15e-05	1.61e-05
Hwang	143.0	128.5	1.05e-04	1.19e-05	1.05e-02	1.19e-03	133	115	3.13e-07	8.70e-08	1.11e-07	3.39e-07
Wang	559.9	—	7.34e-04	—	7.34e-02	—	579	—	7.13e-05	—	7.72e-05	—
Zou	153.3	139.8	9.97e-05	1.20e-04	9.97e-03	1.20e-02	134	113	3.97e-10	1.19e-09	4.69e-10	1.41e-09
Pazaitis	143.3	110.0	3.86e-04	2.35e-04	3.86e-02	2.35e-02	136	151	2.19e-09	2.74e-10	3.21e-09	1.17e-09
Pazaitis	140.1	139.8	1.00e-04	5.21e-05	1.00e-02	5.21e-03	128	486	1.16e-08	2.30e-07	1.78e-08	1.26e-06
Benesty	71.7	—	9.78e-04	—	9.78e-02	—	79	—	1.49e-05	—	6.44e-05	—
Wahab_optimal	95.5	—	1.02e-03	—	1.02e-01	—	103	—	1.28e-04	—	6.27e-05	—

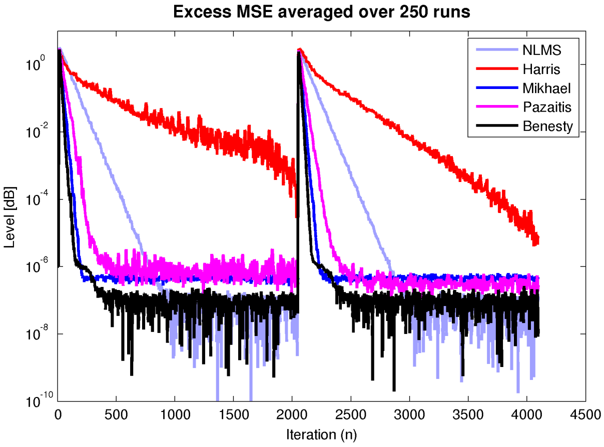

Figure 3. Excess MSE in identification experiments for selected algorithms.

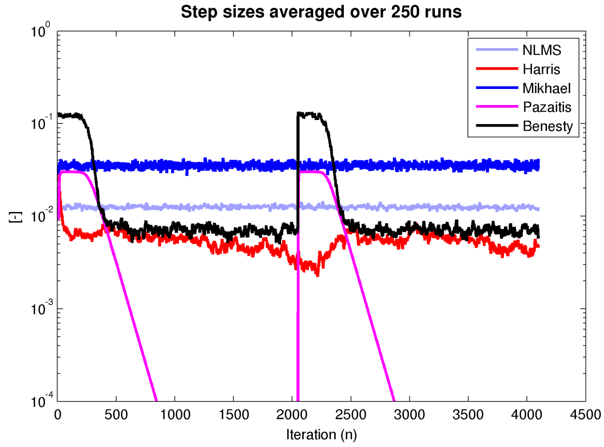

Figure 4. Step sizes in identification experiments for selected algorithms.

coefficients calculated during the steady state at the end of the experiment.

The results of the identification experiments are presented in Table 1. This results were obtained after averaging 250 individual runs, with different excitation sequences and system noise sequences. The identified object was in a form of FIR filter with initial coefficients 0.5, 1.1, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1. In the middle of the experiment, that is after 2048 iterations, the coefficients were changed to 0.3, -0.5, 0.2, 0.8, 0.5, 0.3, 0.1, 0.1, 0.0, 0.0. The signal to noise ratio was equal to 20 dB, and the assumed minimum and maximum step sizes were equal to 10^{-7} and 0.03. The table presents results for the original algorithm, i.e. the algorithm operating with $\mu_{\max} = 0.03$ as well as for the algorithm operating with variable $\mu_{\max}$, as described in Section 3.

By observing the sum of EMSE for the original algorithms we may conclude that the best algorithm is the algorithm by Benesty, the second is the algorithm by Wahab, and the third the algorithm by Mikhael. By consulting the time of convergence we notice that the algorithm by Benesty is the best due to its speed of convergence (the second by the same criterion is the algorithm by Mikhael, the third by Wahab). On the other hand, the algorithm by Benesty exhibits poor steady-state performance, poorer even than LMS and NLMS — the conclusion that arises after examination of the misadjustment. So, if speed of convergence is not critical, but accuracy is in value, the best algorithm is the one by Zou, beating only slightly the one by Pazaitis (second form, Eq. (26)) and by Hwang. Finally, the variance of the estimates allows us to award the algorithms by Zou, Pazaitis (both forms), and Aboulnasr.

With the analysis of the same quantities for the algorithms using variable maximum step size, we may conclude that this modification usually influences the algorithms positively. Additionally, the speed of convergence can be significantly improved, as can be seen in case of the algorithms by Harris, by Shan, or by Karni. Sometimes, even if the speed of convergence is worse, the overall performance is comparable due to better steady-state performance (e.g. the algorithm by Aboulnasr and by Pazaitis, second form).

Figure 3 presents the excess MSE in the identification experiments for selected algorithms. Considering the NLMS as a reference, we observe that the algorithm by Harris (included here for historical reasons, as the next VS-LMS algorithm after the NLMS) is dramatically slower to converge. On the other hand, the algorithms by Mikhael and Pazaitis, although faster to converge, do not achieve the excess MSE as low as the NLMS. Only the algorithm by Benesty maintains both the fast convergence speed and low excess MSE.

Figure 4 shows the step sizes ($\mu(n)$) used by the algorithms discussed above during the experiments. We observe that the algorithms result in different values of the step sizes, compared to the NLMS. The algorithm by Harris decreases the step size rapidly at the begining of the simulation; therefore, it is very slow to converge. The algorithm by Mikhael uses approximately three times higher value of the step size than the NLMS; thus it converges fast, but results in slightly higher excess MSE in the steady state. The algorithms by Pazaitis and Benesty use what we may consider to be a reasonable approach: they start with higher step sizes, and start to decrease them once the initial convergence is finished (compare with Fig. 3). However, the algorithm by Benesty exploits this technique much more efficiently that the algorithm by Pazaitis, and therefore it achieves both the fast convergence speed and the low excess MSE.

4.1.1. Further observations

More conclusions can be drawn if we analyze the system identification results obtained for different signal to noise ratios (the results of experiments, including the plots of the MSE, the EMSE, the misadjustment and the step sizes, are organized in the form of a web page, and available at http://zpss.aei.polsl.pl/dbismor/vslms/; the results can also be e-mailed as a compressed archive by emailing a request to Dariusz.Bismor@polsl.pl). From Eq. (52), it follows that different EMSE levels should be observed for different SNRs: the lower the system noise, the lower the EMSE should be observed. On the other hand, the misadjustment should not depend on the system noise variance and SNR (Eq. (10)). Such behavior was *not* observed for the algorithms by Harris, by Kwong, by Shan, by Karni, and by Wahab. For those algorithms the EMSE tended to reach similar, high values for different SNRs. This means that the algorithms mentioned above may give higher than expected EMSE values.

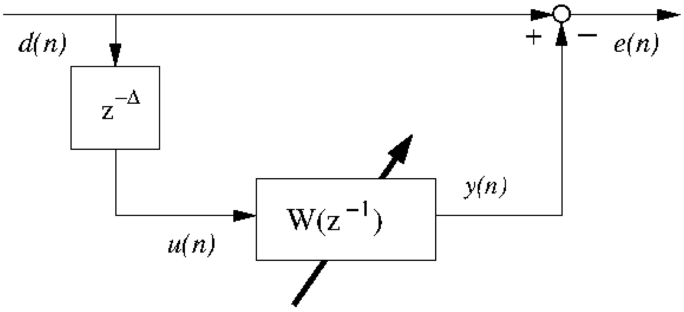

Figure 5. Adaptive line enhancer block diagram.

4.2. Simulation of Adaptive Line Enhancement

The Adaptive Line Enhancement (ALE), introduced by Widrow in 1975, is a technique used to detect highly correlated signals (mainly sinusoids) buried in a wideband noise.[2] Nowadays, ALE is used in many applications, e.g. instantaneous frequency estimation, spectral analysis, and speech enhancement.[31,32] The block diagram of the adaptive line enhancer (ALE) is presented in Fig. 5. The input signal to the ALE, $d(n)$, is usually a mixture of highly correlated signals (e.g. sines) and uncorrelated signals (e.g. white noise or speech recording). This signal is the desired signal for the adaptive LMS filter, while its delayed version constitutes the LMS input signal: $u(n) = d(n - \Delta)$. The delay, Δ, is also called decorrelation delay.[1]

In the experiments presented below, the input signal to the ALE was a short (1.8 s) recording of speech contaminated with four sine signals or with chirp signal. This application tests completely different aspects of the VS-LMS algorithms, as the speech recording can be considered to be a non-stationary signal. Thus, the ALE filter coefficients must be constantly adjusted, except for the short periods between syllables, and only in case of the contamination with the four sines. In case of the chirp signal, its frequency changes during the whole experiment, and no steady state occurs at all.

The performance of the VS-LMS algorithms in the ALE application was evaluated by the means of two quantities. The first one was the sum of MSE during the whole experiment. Please note that excess MSE could not have been used due to the lack of the Wiener filter model, which implies impossibility to calculate the system noise. Moreover, even the calculation of MSE was difficult due to the fact that ALEs may have different amplifications (gains). Therefore, some scaling was necessary before comparing the original speech signal with the ALE output $e(n)$. Scaling by the maximum absolute value was chosen in this case, with the omission of the first 1000 samples to account for the zero initial conditions.

The second quantity used to compare the algorithms performance was the final signal to noise ratio (SNR). This quantity was calculated by the means of the spectrum of the error signal $e(n)$. It was observed that the spectra of the speech recording and the contaminating signals did not overlap during the last 400 ms of the recording. This allowed us to calculate the power of the speech recording and the contaminating signal (after filtration) separately, in different frequency bands.

The speech signal was recorded with the sampling frequently 8 kHz. The ALE had 10 coefficients, and the decorrelation delay was equal to 10. In the case of the four sines, the frequencies were 1.5, 1.8, 2.5 and 3.7 kHz. In case of the chirp signal, the frequency range was 100–3500 Hz. The minimum and maximum step sizes were 10^{-5} and 0.25. The results of the experiments obtained after averaging 250 individual runs, are presented in Table 2.

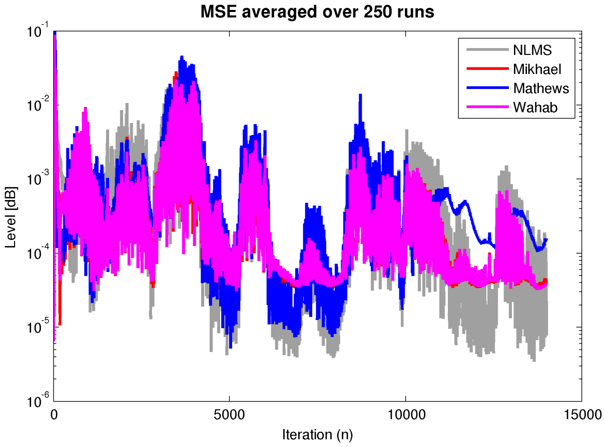

Figure 6. MSE in ALE experiments for selected algorithms.

For the chirp signal, the algorithm by Wahab produced the lowest sum of MSE, while the algorithm by Mikhael was only slightly worse, and the algorithm by Wang was third. However, if the SNR is considered, none of these algorithms perform as well as algorithms by Mathews (with individual step sizes), by Ang (both versions) and even the NLMS.

Figure 6 presents the MSE in the ALE experiments with the chirp signal for the selected algorithms. Treating the NLMS as a reference, we observe that the algorithm by Mathews results in a very similar MSE curve during most of the simulation. Near the end of the experiment, the MSE for this algorithm is noticably higher. The algorithms by Mikhael and Wahab, on the other hand, result in different MSE curves, but very similar to each other. The level of the MSE they produce is comparable with the level produced by the NLMS algorithm.

Figure 7 presents the value of the first filter coefficient during the experiments with the chirp signal. From the figure it is clear that the coefficient does not reach the steady state. However, the values it takes are very similar for all the presented algorithms.

For the four sines signal, and judging by the sum of MSE, the best algorithms are those by Shan, Benveniste, and Benesty. But judging by the SNR, the best algorithms are the one by Evans, Ang, and Benesty.

In the authors' opinion, the SNR better represents the ability of an algorithm to attenuate unwanted contamination — this observation was confirmed by listening to the played error signal. If, for example, the SNR was poor (close to zero or negative), the comprehension of the filtered speech was difficult. Following this observation, we can distinguish VS-LMS algorithms unsuitable for ALE, which certainly are algorithms by Harris, Kwong, Shan, Aboulnasr, Wang, Zou, and Pazaitis. Also, algorithms by Benveniste, Mathews (version with the common step size), and Hwang may not work as well as the others.

Table 2. Results of line enhancer experiments averaged over 250 runs, ALE length: 10. A dash in the 'Modif' column means that the particular algorithm does not use the upper bound for the step size; and thus the modification proposed in Section 3 does not apply.

Algorithm	Chirp				Four sines			
	$\sum$MSE		Final S/N		$\sum$MSE		Final S/N	
	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.
LMS	14.1	—	26.3	—	20.4	—	18.2	—
NLMS	15.8	—	27.0	—	20.0	—	16.9	—
Harris	22.4	28.4	5.3	5.6	31.5	31.0	-15.7	-18.5
Mikhael	10.5	—	17.7	—	20.1	—	17.3	—
Kwong	43.3	25.1	1.3	1.3	20.9	20.8	8.2	6.4
Evans	24.0	16.6	12.5	13.7	20.6	20.0	18.8	18.0
Shan_correlation	26.2	12.5	-2.1	-2.2	19.4	19.8	12.6	11.4
Shan	22.0	15.0	-3.5	-3.4	19.5	20.0	8.4	10.7
Benveniste	13.1	11.0	20.4	17.5	19.5	19.3	6.7	5.9
Benveniste_indiv	14.9	11.6	22.3	19.7	19.7	19.6	8.2	8.5
Karni	25.5	19.6	12.3	18.1	22.2	22.5	17.6	17.2
Mathews	20.8	19.9	5.6	5.6	24.1	24.8	3.1	7.0
Mathews_indiv	15.8	13.8	27.3	25.8	20.6	20.1	16.5	17.0
Aboulnasr	31.3	28.5	-2.0	-1.7	21.3	21.2	5.0	4.1
Ang	16.4	11.9	27.2	21.8	20.1	19.9	18.1	18.9
Ang_indiv	18.9	12.4	27.2	22.9	20.3	20.0	17.2	17.0
Hwang	48.0	22.4	5.0	5.0	21.4	20.7	6.6	7.1
Wang	12.5	—	7.3	—	113.6	—	-29.9	—
Zou	46.4	35.5	0.8	1.1	21.7	21.4	13.2	13.8
Pazaitis	46.7	26.2	-0.7	-0.0	21.8	22.4	15.7	17.2
Pazaitis	30.5	22.7	-0.6	0.2	21.6	21.9	8.8	18.1
Benesty	12.9	—	26.6	—	19.6	—	18.1	—
Wahab_optimal	9.8	—	16.9	—	20.1	—	17.2	—

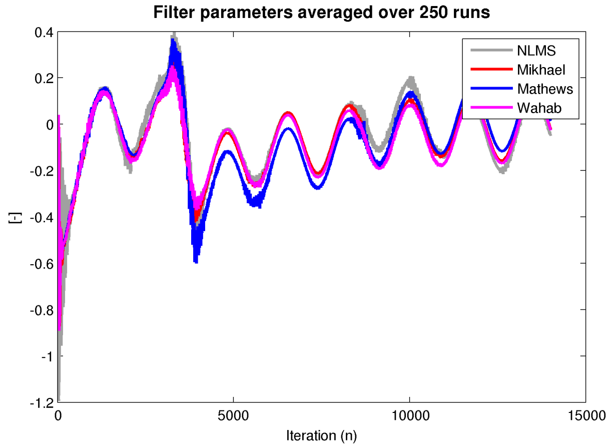

Figure 7. First filter coefficient in ALE experiments for selected algorithms.

It must also be noted that introduction of the variable maximum step-size modification does not always positively influences the performance, especially if we judge the algorithms by the sum of MSE. However, if we consider the SNR only, the set of algorithms for which the performance is significantly reduced is limited to the algorithms by Karni and Ang.

4.2.1. Further observations

The best way to evaluate the ALE performance is by analyzing the spectrograms (available at http://zpss.aei.polsl.pl/dbismor/vslms/). By comparing the spectrograms for the four sines signal, we immediately observe complete failure of the algorithms by Harris and Wang. We also notice very good results obtained with the NLMS and the algorithms by Mikhael, by Evans, by Karni and by Benesty — no trace of the four sines can be observed in the spectrograms for those algorithms. This group may be extended over the algorithms by Pazaitis, if we introduce the variable maximum step size modification discussed in Section 3.

By analyzing the spectrograms for the chirp signal, we observe that it is not possible to remove the constantly changing chirp signal completely with this setup. However, the NLMS and the algorithms by Mathews (individual step sizes), Ang, and Benesty are doing a very good job to weaken the power of the chirp considerably. By analyzing the step sizes, we observe that the most efficient strategy here is to keep the step size as high as possible during the whole experiment.

4.3. Simulation of Adaptive Noise Cancellation

The Adaptive Noise Cancellation (ANC) block diagram is presented in Fig. 8. The diagram is similar to the system identification, except for the presence of the output noise $n(n)$. However, the conditions and goals of operation of the two systems are different. One of the differences is in the unknown system dynamic ($P(z^{-1})$ in Fig. 8). In case of the system identification, we seek for the model of this dynamic, while

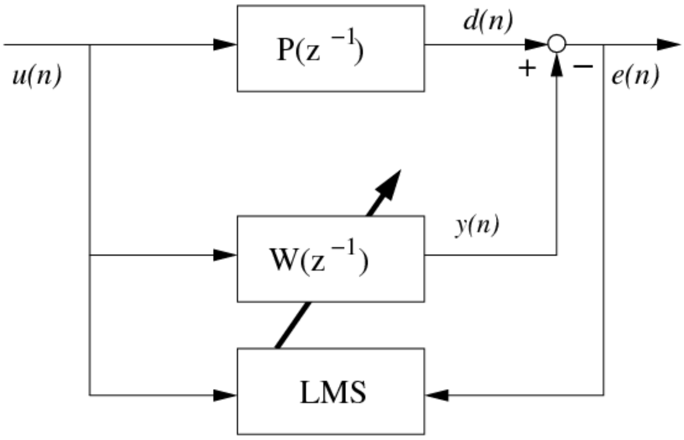

Figure 8. Adaptive noise cancellation block diagram.

in case of the ANC, the dynamic is assumed to be too difficult to be modeled in the whole frequency range. For example, in the experiments described below, the dynamic $P(z^{-1})$ was nonstationary and was implemented as a bank of 300-th order FIR filters switched each 3 seconds, while the ANC filter had only 16 or 128 coefficients.

Another difference is the input signal $u(n)$. In case of the system identification, we usually employ persistently exciting signals of sufficient degree (white noise, if possible). In case of the ANC, we usually have no chance to choose the input signal, which often contains a mixture of narrow-band signals (e.g. single tones) and band-limited noise. In the experiments described below, the input signal consisted of two sines, one with frequency 112 Hz and amplitude 3 and the second with frequency 200 Hz and amplitude 1. The sines were embedded in an amount of white noise with different variances.

The performance of the VS-LMS algorithms in ANC application was evaluated by the means of three quantities. The first one was again the sum of MSE during the whole experiment. The second was the MSE obtained during the steady state of operation, and it will be referred to as 'Final MSE'. Please note that misadjustment cannot be used because there was no additional noise, which means that $J_{\text{min}} = 0$ in Eq. (53). The third quantity used to evaluate the performance of the algorithms was the variance of the ANC filter coefficient w_0 during the steady-state operation.

The results of the experiments, with the ANC filter length 16, obtained after averaging 250 individual runs are presented in Table 3. By analyzing the results for the variance of the wideband noise 10^{-2} we come to a surprising conclusion that none of the VS-LMS algorithms performed better than the NLMS algorithm — this is clear from both the sum of MSE and the Final MSE. Slightly worse results are obtained with the algorithms by Mathews (both versions), Karni, Ang (with individual step sizes), and Evans. Spectacular failure of the algorithm by Wang should also be noted, as it is one of the only two algorithms that does not require any parameters to run. Another conclusion is that very small variance of the coefficients during the steady state, as e.g. in case of the algorithm by Pazaitis, is not correlated with high overall performance.

In case of the wideband noise variance 10^{-4} the advantage of the NLMS algorithm is no more very clear. Although it still produces the smallest final MSE, the sum of MSE is smaller for other algorithms, such as those by Evans and Karni. Other algorithms with similar performance are correlation LMS and those by Pazaitis, Ang (with individual step sizes), Shan, Benveniste (with individual step sizes), Kwong, and Hwang. Again, the algorithm by Wang failed to operate satisfactorily.

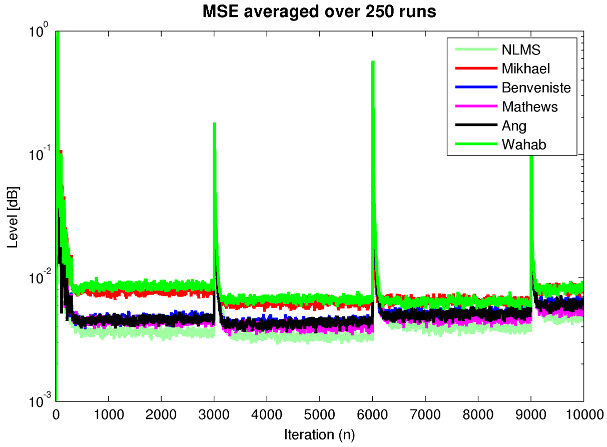

Figure 9. MSE in ANC experiments for selected algorithms.

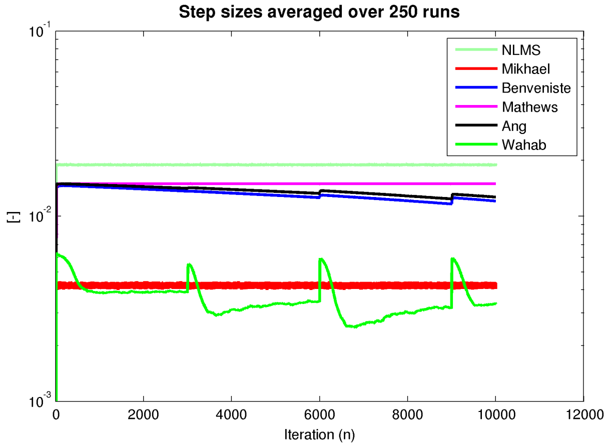

Figure 10. Step sizes in ANC experiments for selected algorithms.

It must be noted that the majority of the algorithms had to be readjusted when switching from the wideband noise variance 10^{-4} to 10^{-2} — because otherwise they failed to maintain convergence. This only does not apply the NLMS and the algorithms by Harris, Mikhael, Evans, Shan, and by Wang. The fact that the NLMS did not need to be readjusted emphasizes its usefulness in ANC applications.

By analyzing the algorithms with variable maximum step-size modification, we may conclude that this modification usually does not degrade the algorithm's performance. On the contrary, the modification usually increased the performance slightly, especially in case of higher wideband noise variance. The algorithms by Karni and Benveniste are the examples.

Figure 9 presents the MSE curves for the wideband noise 10^{-2} for the selected algorithms, and Fig. 10 shows the step sizes for those experiments. From the figures it is clear that, despite different step size strategies, the algorithms show very similar speed of convergence. The only noticable difference is in the final MSE during the stead-state phases, which is slightly higher for the algorithms by Mikhael and Wahab. The step

Table 3. Results of adaptive noise cancellation experiments averaged over 250 runs, ANC filter length: 16. A dash in the 'Modif' column means that the particular algorithm does not use the upper bound for the step size; and thus the modification proposed in Section 3 does not apply.

	Noise variance 1e-2						Noise variance 1e-4					
Algorithm	$\sum$MSE		Final MSE		var(w_1)		$\sum$MSE		Final MSE		var(w_1)	
	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.	Orig.	Modif.
LMS	49.1	—	3.9e-03	—	2.2e-05	—	12.0	—	3.9e-05	—	3.5e-07	—
NLMS	49.4	—	4.0e-03	—	2.7e-05	—	12.2	—	4.0e-05	—	3.2e-07	—
Harris	87.7	86.0	7.0e-03	7.2e-03	2.1e-05	2.1e-05	18.4	18.9	7.4e-05	7.4e-05	3.0e-07	3.2e-07
Mikhael	101.8	—	6.4e-03	—	1.1e-05	—	33.2	—	6.5e-05	—	1.0e-07	—
Kwong	113.7	112.5	5.9e-03	6.0e-03	2.4e-07	2.2e-07	14.0	14.4	6.3e-05	6.3e-05	1.4e-08	1.4e-08
Evans	59.6	54.6	4.9e-03	4.5e-03	2.4e-05	3.4e-05	11.9	12.3	4.2e-05	4.5e-05	1.6e-07	2.4e-07
Shan_correlation	88.2	86.6	6.7e-03	6.6e-03	2.9e-06	3.6e-06	12.5	13.1	6.4e-05	6.4e-05	2.1e-08	1.7e-08
Shan	80.8	89.0	7.1e-03	8.3e-03	1.4e-05	2.6e-05	13.3	13.7	6.7e-05	6.7e-05	1.9e-08	1.2e-08
Benveniste	71.4	65.5	6.4e-03	6.2e-03	5.2e-06	1.3e-05	33.9	36.1	4.6e-05	4.7e-05	2.6e-07	2.5e-07
Benveniste_indiv	62.0	56.1	5.3e-03	4.7e-03	1.7e-05	2.5e-05	13.3	13.1	4.4e-05	4.6e-05	2.3e-07	1.8e-07
Karni	57.8	49.4	4.7e-03	3.9e-03	2.3e-05	9.0e-05	12.0	12.8	6.3e-05	6.2e-05	1.5e-07	1.5e-07
Mathews	57.5	53.1	4.7e-03	4.3e-03	2.3e-05	3.1e-05	20.1	23.1	4.1e-05	4.3e-05	5.7e-07	2.6e-07
Mathews_indiv	57.8	53.3	4.7e-03	4.4e-03	1.6e-05	3.1e-05	13.7	13.5	4.2e-05	4.5e-05	2.4e-07	2.4e-07
Aboulnasr	96.4	96.9	6.0e-03	5.9e-03	4.3e-07	6.8e-07	28.5	30.5	5.7e-05	5.7e-05	3.9e-15	4.5e-15
Ang	61.0	57.7	5.2e-03	5.0e-03	1.7e-05	1.8e-05	24.6	22.3	4.3e-05	4.3e-05	1.9e-07	2.6e-07
Ang_indiv	58.2	54.9	4.7e-03	4.5e-03	2.0e-05	3.1e-05	12.9	12.7	4.3e-05	4.5e-05	2.3e-07	2.7e-07
Hwang	87.8	86.6	6.1e-03	6.2e-03	1.3e-06	8.8e-07	14.3	15.2	5.8e-05	5.8e-05	1.5e-10	2.1e-10
Wang	1475.3	—	1.1e-02	—	6.9e-07	—	1386.2	—	5.4e-03	—	6.4e-07	—
Zou	111.5	112.6	6.2e-03	6.1e-03	5.5e-06	2.6e-06	16.7	16.8	5.5e-05	5.4e-05	2.3e-07	1.0e-07
Pazaitis	92.5	92.1	5.7e-03	5.8e-03	1.9e-09	7.3e-09	12.6	13.0	5.8e-05	5.7e-05	1.5e-12	1.0e-12
Pazaitis	116.7	110.0	5.8e-03	5.8e-03	4.1e-10	2.7e-10	14.1	13.8	5.7e-05	5.7e-05	2.2e-16	3.7e-16
Benesty	67.1	—	5.4e-03	—	2.0e-05	—	16.0	—	5.3e-05	—	1.4e-07	—
Wahab_optimal	107.5	—	6.4e-03	—	1.8e-06	—	35.5	—	6.5e-05	—	4.6e-08	—

sizes have similar values for the NLMS and the algorithm by Mikhael, but all the other algorithms presented on the figure use higher values of the step sizes.

5. CONCLUSIONS

The very first conclusion which arises from the above analysis of the seventeen most popular variable step-size modifications of the LMS algorithm presented in this paper is that there is no VS-LMS algorithm which is the best for all the applications. Different applications present different challenges for the adaptation algorithm, thus requiring different step-size update algorithms for the most efficient operation. This is especially apparent with stationary data processing (e.g. the system identification) as opposed to nonstationary data processing (e.g. speech signal processing). However, some general guidelines for the choice of the proper VS-LMS algorithm are as follows:

First of all, the more parameters the algorithm requires to be adjusted prior to operation, the more problematic the use of this algorithm is. Even in simulations, the choice of more than two or three parameters requires many time-consuming trials. This is the effect of parameter interactions: the optimal choice of one parameter is no longer optimal if we change other parameters. Moreover, the optimal set of parameters is likely to be different if experiment conditions (e.g. noise level) change. Thus, the algorithms with more than three parameters should be considered only if very special properties need to be obtained, and probably only for simulations.

The above remark does not apply to the lower and upper bound on the step size, required by many of the algorithms discussed in this paper. The former is very easy to choose and, for properly parametrized VS-LMS algorithm, influences only steady-state error. The latter may easily be substituted with stability sufficient condition bound given by Eq. (47), with low computational cost and almost no influence on the performance.

With the regard to the ease of use, the algorithm by Mikhael (Section 2.3) should be mentioned. This algorithm requires no parameters to run, and performs reasonably well in both the stationary and nonstationary data processing. However, this algorithm resulted in higher error levels in the ANC simulations and may be not very well-suited for this application.

Second, very interesting and original is the algorithm by Pazaitis (Section 2.11). This algorithm performed very well in stationary data processing (the system identification). However, the kurtosis estimation involved in this algorithm requires substantial number of operations; therefore, this algorithm may not be applicable for real-time processing.

Finally, for nonstationary data processing, considering the ease of use, numerical complexity, and performance, the NLMS algorithm still appears to be unrivaled. However, if another algorithm is desired for some reasons, the algorithm by Mathews (Section 2.9) should be considered. This algorithm, based on the orthogonality principle, requires two parameters to be adjusted (excluding the upper bound for the step size), presents moderate degree of numerical complexity, and performs very well in both the ALE and the ANC applications.

To conclude, we will repeat this very important finding that comes from the research reported in this paper: no VS-LMS algorithm appears to be as versatile, easy to use, and well-suited for real-time applications as the NLMS. In our opinion, despite the constant effort to develop new VS-LMS algorithms, the 49-years-old NLMS algorithm is going to dominate the adaptive solutions for many years on, if not forever.

Acknowledgment

The research is within a project financed by the National Science Centre, Poland, based on decision no. DEC-2012/07/B/ST7/01408.

REFERENCES

[1] Haykin, S. *Adaptive Filter Theory, Fourth Edition*. Prentice Hall, New York, (2002). ISBN 0-13-048434-2.

[2] Widrow, B., Glover, J. R., Jr., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., Zeidler, J. R., Eugene Dong, Jr., and Goodlin, R. C. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716, (1975). ISSN 0018-9219. http://dx.doi.org/10.1109/PROC.1975.10036.

[3] Macchi, O. *Adaptive Processing. The Least Mean Squares Approach with Applications in Transmission.* John Wiley & Sons, Chichester, 1995. ISBN 0 471 93403 8.

[4] Landau, I. D., Lozano, R., M'Saad, M., and Karimi, A. *Adaptive Control. Algorithms, Analysis and Applications*. Springer-Verlag, London, (2011). ISBN 987-0-85729-663-4.

[5] Nagumo, J., and Noda, A. A learning method for system identification. *IEEE Transactions on Automatic Control*, **12**(3):282–287, (1967). ISSN 0018-9286. http://dx.doi.org/10.1109/TAC.1967.1098599.

[6] Albert, A. E., and Gardner,L. S. *Stochastic Approximation and Nonlinear Regression*. MIT Press, Cambridge, MA., (1967).

[7] Harris, R. W., Chabries, D. M., and Bishop, F. A. A variable step (VS) adaptive filter algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **ASSP-34**(2):309–316, (1986). http://dx.doi.org/10.1109/TASSP.1986.1164814.

[8] Mikhael, W., Wu, F., Kazovsky, L., Kang, G., and Fransen, L. Adaptive filters with individual adaptation of parameters. *IEEE Transactions on Circuits and Systems*, **33**(7):677–686, (1986). ISSN 0098-4094d. http://dx.doi.org/10.1109/TCS.1986.1085982.

[9] Ao Wei, Xiang Wan-Qin, Zhang You-Peng, Wang Lei, Lv Chun-Ying, and Wang Zheng-Hua. A new variable step size LMS adaptive filtering algorithm. In 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE), volume 2, pages 265–268, (2012). http://dx.doi.org/10.1109/ICCSEE.2012.115.

[10] Tian, F., and Luo, R. *A novel variable step size LMS algorithm based on modified hyperbolic tangent and its simulation*, volume 490–495 of *Advanced Materials Research*, pages 1426–1430. (2012). http://dx.doi.org/10.4028/www.scientific.net/AMR.490-495.1426.

[11] Kwong, R. H., and Johnston, E. W. Variable step size LMS algorithm. *IEEE Transactions on Signal Processing*, **40**(7): 1633–1642, (1992). http://dx.doi.org/10.1109/78.143435.

[12] Aboulnasr, T., and Mayyas, K. A robust variable step-size LMS-type algorithm: Analysis and simulations. *IEEE Transactions on Signal Processing*, **45**(3):631–639, (1997). http://dx.doi.org/10.1109/78.558478.

[13] Pazaitis, D. I., and Constantinides, A. G.. A novel kurtosis driven variable step-size adaptive algorithm. *IEEE Transactions on Signal Processing*, **47**(3):864–872, (1999). ISSN 1053-587X. http://dx.doi.org/10.1109/78.747793.

[14] Zou, K., and Zhao, X. A new modified robust variable step size LMS algorithm. In The 5th IEEE Conference on Industrial Electronics and Applications, pages 2699–2703. Xi'an, China, (2009). http://dx.doi.org/10.1109/ICIEA.2009.5138654.

[15] Shan, T. J., and Kailath, T. Adaptive algorithms with an automatic gain control feature. *IEEE Transactions on Circuits and Systems*, **35**(1):122–127, (1988). ISSN 0098-4094. http://dx.doi.org/10.1109/31.1709.

[16] Karni, S., and Zeng, G. A new convergence factor for adaptive filters. *IEEE Transactions on Circuits and Systems*, **36**(7):1011–1012, (1989). ISSN 0098-4094. http://dx.doi.org/10.1109/31.31337.

[17] Benveniste, A., Métivier, M., and Priouret, P. *Adaptive algorithms and stochastic approximations*. Applications of mathematics. Springer-Verlag, (1990). ISBN 9783540528944. LCCN 91127124.

[18] Evans, J. B., Xue, P., and Liu, B. Analysis and implementation of variable step size adaptive algorithms. *IEEE Transactions on Signal Processing*, **41**(8):2517–2235, (1993). http://dx.doi.org/10.1109/78.229885.

[19] Mathews, V. J., and Xie, Z. A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing*, **41**(6):2075–2087, (1993). http://dx.doi.org/10.1109/78.218137.

[20] Ang, W., and Farhang-Boroujeny, B. A new class of gradient adaptive step-size LMS algorithms. *IEEE Transactions on Signal Processing*, **49**(4):805–810, (2001). http://dx.doi.org/10.1109/78.912925.

[21] Benesty, J., Rey, H., Vega, L. R., and Tressens, S. A non-parametric VSS NLMS algorithm. *IEEE Signal Processing Letters*, **13**(10):581–584, oct. (2006). ISSN 1070-9908. http://dx.doi.org/10.1109/LSP.2006.876323.

[22] Wahab, M. A., Uzzaman, M. A., Hai, M. S., Haque, M. A., and Kasan, M. K. Least-squares optimal variable step-size LMS for nonblind system identification with noise. In 5th International Conference on Electrical and Computer Engineering, pages 428–433. Bangladesh, (2008). http://dx.doi.org/10.1109/ICECE.2008.4769245.

[23] Hwang, J., and Li, Y. Variable step-size LMS algorithm with a gradient-based weighted average. *IEEE Signal Processing Letters*, **16**(12):1043–1046, (2009). http://dx.doi.org/10.1109/LSP.2009.2027653.

[24] Wang, P., Kam, P. Y., and Chia, M. W. A novel automatic step-size adjustment approach in the LMS algorithm. In The First International Conference on Wireless VITAE, pages 867–871. Aalborg, Denmark, (2009). http://dx.doi.org/10.1109/WIRELESSVITAE.2009.5172563.

[25] Karni, S., and Zeng, G. Comments on "Adaptive algorithms with an automatic gain control feature". *IEEE Transactions on Circuits and Systems*, **37**(7):974–975, (1990). ISSN 0098-4094. http://dx.doi.org/10.1109/31.55079.

[26] Karni, S., and Zeng, G. Comments, with reply, on "Adaptive algorithms with an automatic gain control feature" by t.j. shan and t. kailath. *IEEE Transactions on Circuits and Systems*, **37**(7):974–975, (1990). ISSN 0098-4094. http://dx.doi.org/10.1109/31.55079.

[27] Kushner,H. J., and Yang, J. Analysis of adaptive step size SA algorithms for parameter tracking. In *Proceedings of 33rd Conference on Decision and Control*, pages 730–737. Lake Buena Vista, Fl, USA, (1994). http://dx.doi.org/10.1109/CDC.1994.410867.

[28] Mazur, K., and Pawelczyk. M. Hammerstein nonlinear active noise control with the Filtered-Error LMS algorithm. *Archives of Acoustics*, **38**(2):197–203, (2013). http://dx.doi.org/10.2478/aoa-2013-0023.

[29] Bismor, D. Extension of LMS stability condition over a wide set of signals. *International Journal of Adaptive Control and Signal Processing*, **29**:653–670, (2015). ISSN 1099-1115. http://dx.doi.org/10.1002/acs.2500.

[30] Söderström, T., and Stoica, P. *System Identification*. Prentice Hall International, Inc., New York, (1989).

[31] Zeidler, J. R. Performance analysis of LMS adaptive prediction filters. *Proceedings of the IEEE*, **78**(12):1781–1806, (1990). ISSN 0018-9219. http://dx.doi.org/10.1109/5.60921.

[32] Latos, M., and Pawelczyk, M. Adaptive algorithms for enhancement of speech subject to a high-level noise. *Archives of Acoustics*, **35**(2):203–212, (2010). http://dx.doi.org/10.2478/v10168-010-0019-z.

[33] Ławryńczuk, M. Nonlinear state–space predictive control with on—line linearisation and state estimation. *International Journal of Applied Mathematics and Computer Science*, **25**(4): 833–847, (2015). http://dx.doi.org/10.1515/amcs-2015-0060.