A Variable Step Size Normalized Least-Mean-Square Algorithm Based on Data Reuse

Alexandru-George Rusu¹,², Constantin Paleologu¹,* Jacob Benesty³ and Silviu Ciochină¹

¹ Department of Telecommunications, University Politehnica of Bucharest, 060042 Bucharest, Romania; alexandru.rusu@rohde-schwarz.com (A.-G.R.); silviu@comm.pub.ro (S.C.)
² Department of Research and Development, Rohde & Schwarz Topex, 020335 Bucharest, Romania
³ INRS-EMT, University of Quebec, Montreal, QC H5A 1K6, Canada; jacob.benesty@inrs.ca

* Correspondence: pale@comm.pub.ro

Abstract: The principal issue in acoustic echo cancellation (AEC) is to estimate the impulse response between the loudspeaker and microphone of a hands-free communication device. This application can be addressed as a system identification problem, which can be solved by using an adaptive filter. The most common one for AEC is the normalized least-mean-square (NLMS) algorithm. It is known that the overall performance of this algorithm is controlled by the value of its normalized step size parameter. In order to obtain a proper compromise between the main performance criteria (e.g., convergence rate/tracking versus accuracy/robustness), this specific term of the NLMS algorithm can be further controlled and designed as a variable parameter. This represents the main motivation behind the development of variable step size algorithms. In this paper, we propose a variable step size NLMS (VSS-NLMS) algorithm that exploits the data reuse mechanism, which aims to improve the convergence rate/tracking of the algorithm by reusing the same set of data (i.e., the input and reference signals) several times. Nevertheless, we involved an equivalent version of the data reuse NLMS, which provides the convergence modes of the algorithm. Based on this approach, a sequence of normalized step sizes can be a priori scheduled, which is advantageous in terms of the computational complexity. The simulation results in the context of AEC supported the good performance features of the proposed VSS-NLMS algorithm.

Keywords: acoustic echo cancellation (AEC); adaptive filters; data reuse; normalized least-mean-square (NLMS) algorithm; variable step size (VSS)

1. Introduction

The normalized least-mean-square (NLMS) algorithm is the workhorse in many practical applications of adaptive filtering [1–3]. The main reasons behind its popularity consist of its simple and efficient implementation, together with fairly reasonable performance, in terms of the convergence rate, tracking, and accuracy of the solution. Nevertheless, there is always a need to improve the performance features of the NLMS algorithm, especially in challenging scenarios, e.g., when using highly correlated input signals and/or dealing with long-length adaptive filters. In this framework, a very important application is acoustic echo cancellation (AEC) [4–6], where the input signal is mainly speech (i.e., a nonstationary and highly correlated sequence), while the impulse response of the acoustic echo path is usually of the order of hundreds or even thousands of coefficients.

There are several strategies to improve the performance of the NLMS algorithm in AEC scenarios. Among them, proportionate NLMS algorithms exploit the sparseness nature of the echo paths to increase the convergence rate and tracking of the adaptive filter [7–13]. Besides, variable step size NLMS (VSS-NLMS) algorithms are designed to achieve a proper compromise between the convergence rate and misadjustment (i.e., accuracy of the estimate) [14–18]. Moreover, the data reuse approach consists of performing...
more than one filter update for the same set of data (i.e., the input and reference signals), thus targeting a higher convergence rate [19–25].

In this short communication, we focused on the data reuse NLMS algorithm from a different perspective, showing how it can be exploited in order to develop an improved NLMS version with variable step sizes. As compared to other previous works related to the data reuse approach, the proposed VSS-NLMS algorithm uses a sequence of scheduled step size parameters. These values are a priori computed by exploiting the convergence modes of the data reuse NLMS algorithm. On the contrary, in [24], the variable step size (VSS) from [14] was used in conjunction with the data reuse NLMS algorithm from [21], thus resulting in a new version that was similar to an affine projection algorithm with VSS parameter. Furthermore, in [25], the data reuse approach was applied to an optimized least-mean-square algorithm, which behaves similar to a VSS version. In these previous works, the VSS parameter was evaluated within each iteration of the algorithm, which is fundamentally different as compared to the current proposed approach, where the step size parameters were a priori computed and scheduled within the algorithm.

Our main application framework was AEC [4], where the main goal was to estimate the impulse response between the loudspeaker and microphone of a hands-free communication device. Basically, we dealt with a system identification problem, which can be solved by using an adaptive filter. Currently, hands-free communication devices are involved in many popular applications, such as mobile telephony and teleconferencing systems. Due to their specific features, they can be used in a wide range of environments with different acoustic characteristics. In this context, an important issue that has to be addressed when dealing with such devices is the acoustic coupling between the loudspeaker and microphone. In other words, besides the voice of the near-end speaker and the background noise, the microphone of the hands-free equipment captures another signal (coming from its own loudspeaker), known as the acoustic echo. Depending on the environments’ characteristics, this phenomenon can be very disturbing for the far-end speaker, which hears a replica of her/his own voice. From this point of view, there is a need to enhance the quality of the microphone signal by canceling the unwanted acoustic echo. 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 [4–6]. Consequently, the adaptive filter has to model an unknown system, i.e., the acoustic echo path between the loudspeaker and microphone.

Even if the problem formulation is straightforward, the specific features of AEC 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; therefore, long-length adaptive filters are required (hundreds or even thousands of coefficients), influencing the convergence rate of the algorithm. Second, the acoustic echo paths are time-variant systems (depending on the temperature, pressure, humidity, and movement of objects or bodies), requiring good tracking capabilities for the echo canceler. Third, the input of the adaptive filter (i.e., the far-end signal) is mainly speech, which is a nonstationary and highly correlated signal that can influence the overall performance of the adaptive algorithm.

Since the conventional NLMS algorithm uses a constant step size parameter to control its performances, a compromise should be made when choosing this value. It is known that a large value implies a fast convergence rate and tracking, while a small value leads to low misadjustment and good robustness features. Since in AEC, there is a need for all these performance criteria, the step size should be controlled; therefore, a VSS version represents a more reliable choice. In addition, for real-world AEC applications, it is highly desirable to use nonparametric algorithms, in the sense that no information about the acoustic environment is required. Nevertheless, we should note that the applications of the VSS-NLMS algorithms are not limited only to the framework of AEC. Different other application areas can be explored in the field of communication systems, such as the carrier phase recovery, e.g., see [26,27] and the references therein.
Following this Introduction, in Section 2, we present the data reuse NLMS algorithm, but using a computationally efficient method to exploit the data reuse process, thus keeping almost the same complexity as the conventional version. Then, in Section 3, we develop a VSS-NLMS algorithm based on the data reuse approach. Simulation results in the context of AEC are provided in Section 4 in order to support the performance of the proposed VSS-NLMS algorithm. Finally, several conclusions are summarized in Section 5. The acronyms used in this paper are provided in Table A1 in Appendix A, while the notation and symbols are summarized in Table A2 from the same section.

2. Data Reuse NLMS Algorithm

The conventional NLMS algorithm [1,3] is defined by two main equations, which provide the error signal and the filter update, respectively, as follows:

$$e(n) = d(n) - x^T(n)\hat{h}(n-1),$$  \hspace{1cm} \text{(1)}

$$\hat{h}(n) = \hat{h}(n-1) + \frac{\alpha x(n)e(n)}{x^T(n)x(n)},$$  \hspace{1cm} \text{(2)}

where $e(n)$ is the a priori error signal at the discrete-time index $n$, $d(n)$ is the desired (or reference) signal, $x(n) = [x(n) x(n-1) \cdots x(n-L+1)]^T$ is a vector containing the most recent $L$ time samples of the zero-mean input signal $x(n)$, the superscript $^T$ denotes the transpose operator, $\hat{h}(n) = [\hat{h}_0(n) \hat{h}_1(n) \cdots \hat{h}_{L-1}(n)]^T$ is the adaptive filter (of length $L$) at the discrete-time index $n$, which is an estimate of the unknown impulse response, and $\alpha$ is the normalized step size parameter of the NLMS algorithm. Theoretically, the range for this parameter is $0 < \alpha < 2$, but practically, the recommended choice is $0 < \alpha \leq 1$ [2]. For the sake of simplicity, we omitted in the developments the regularization parameter $\delta > 0$ [28], which is usually added to the denominator in (2).

Based on the adaptive filter coefficients at the discrete-time index $n$, we may also define the a posteriori error signal as:

$$e_1(n) = d(n) - x^T(n)\hat{h}(n).$$  \hspace{1cm} \text{(3)}

Using (1) and (2) in (3), we can further develop the previous expression of the a posteriori error, which results in:

$$e_1(n) = (1 - \alpha)e(n).$$  \hspace{1cm} \text{(4)}

It can be noticed that the a posteriori error is canceled for $\alpha = 1$ (assuming that $e(n) \neq 0$).

Now, let us consider that we perform multiple iterations for the NLMS algorithm using the same input data $x(n)$ and reference signal $d(n)$, i.e., performing the data reuse process [19–25]. Hence, in the first step (using (1), (2), and (4)), we have:

$$\hat{h}_1(n) = \hat{h}(n) + \frac{\alpha x(n)e_1(n)}{x^T(n)x(n)}$$

$$= \hat{h}(n-1) + \frac{(1 - (1 - \alpha)^2)x(n)e(n)}{x^T(n)x(n)}.$$  \hspace{1cm} \text{(5)}
Then, by repeating the process, we obtain:

\[ e_2(n) = d(n) - \mathbf{x}^T(n) \hat{\mathbf{h}}_1(n) = (1 - \alpha)^2 e(n), \quad (6) \]

\[ \hat{\mathbf{h}}_2(n) = \hat{\mathbf{h}}_1(n) + \frac{\alpha \mathbf{x}(n) e_2(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} = \hat{\mathbf{h}}(n - 1) + \frac{[1 - (1 - \alpha)^2] \mathbf{x}(n) e(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}, \quad (7) \]

and so on.

Using mathematical induction, it results that the \( k \)-th order a posteriori error is:

\[ e_k(n) = (1 - \alpha)^k e(n) \quad (8) \]

and the \( k \)-th order update is given by:

\[ \hat{\mathbf{h}}_k(n) = \hat{\mathbf{h}}(n - 1) + \frac{[1 - (1 - \alpha)k] \mathbf{x}(n) e(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}, \quad (9) \]

which is similar to the original update (2), but using the normalized step size:

\[ \alpha_k = 1 - (1 - \alpha)^{k+1}. \quad (10) \]

In other words, performing multiple iterations for the same set of data is equivalent to increasing the value of the normalized step size, thus leading to a faster convergence mode [1–3].

A few remarks can be outlined based on the previous results, as follows:

(1) If \( \alpha = 1 \), then \( e_k(n) = 0 \) and \( \alpha_k = 1 \) for any value of \( k \). This supports the well-known fact that the value of the normalized step size \( \alpha = 1 \) provides the fastest convergence mode;

(2) For any \( 0 < \alpha < 1 \), \( e_k(n) \mid_{k \to \infty} = 0 \);

(3) For any \( 0 < \alpha < 1 \), \( \alpha_k \mid_{k \to \infty} = 1 \). This is shown in Figure 1, where the parameter \( \alpha_k \) is plotted for different values of \( \alpha \). These curves can be interpreted as the data reuse convergence modes of the NLMS algorithm, for a certain value of \( \alpha \).

![Figure 1. Evolution of the parameter \( \alpha_k \) for different values of the normalized step size \( \alpha \).](image-url)
In addition, based on (10), we can easily evaluate the number of data reuse iterations, \( k_{\text{max}} \), required to attain a certain (maximum) value of the normalized step size, \( \alpha_{\text{max}} \), starting from a given value of \( \alpha \). Thus, we can write:

\[
1 - (1 - \alpha)^{k_{\text{max}} + 1} = \alpha_{\text{max}},
\]

which results in:

\[
k_{\text{max}} = \left\lfloor \log(1 - \alpha_{\text{max}}) \right\rfloor - 1,
\]

where \( \left\lfloor \cdot \right\rfloor \) denotes the ceiling function and \( \log(\cdot) \) stands for the natural logarithm.

### 3. Variable Step Size NLMS Algorithm

Based on the previous considerations, we can design a simple VSS-NLMS algorithm, by implementing the previous approach in a “reverse” manner, using scheduled normalized step sizes. Of course, more complex/performant detectors can be used in Step 3 [15], but this is beyond the scope of this paper.

For tracking purposes, in order to not stall the algorithm in Step 2 (i.e., when the lower bound of the normalized step size parameter \( \alpha \) is used), the condition from Step 3 is introduced. The power estimates required in Step 3 can be recursively evaluated as:

\[
\hat{\sigma}_\ell^2(n) = \lambda \hat{\sigma}_\ell^2(n - 1) + (1 - \lambda) e_\ell^2(n),
\]

\[
\hat{\sigma}_v^2(n) = \gamma \hat{\sigma}_v^2(n - 1) + (1 - \gamma) e_v^2(n),
\]

where \( \lambda = 1 - 1/(KL) \) and \( \gamma = 1 - 1/(QL) \), with \( K > 1 \) and \( Q > K \) [29]. The power estimate of the system noise can be roughly estimated from the error signal by using a larger smoothing parameter \( \gamma \). By checking the condition from Step 3, we can identify the change of the system (in a system identification scenario) and reset the step size of the algorithm. Of course, more complex/performant detectors can be used in Step 3 [15], but this is beyond the scope of this paper.

The main parameters to be set are \( k_{\text{max}} \) and \( l \). The first parameter, \( k_{\text{max}} \), is related to the maximum value of the normalized step size, \( \alpha_{\text{max}} \), as shown in (12). For example, if we consider \( \alpha_{\text{max}} = 0.99 \) (which is high enough and very close to the fastest convergence mode) and the lower bound \( \alpha \ll 1 \) (so that \( \log(1 - \alpha) \approx -\alpha \)), according to (12) we obtain:

\[
k_{\text{max}} \approx \left\lfloor \frac{\log(0.01)}{-\alpha} \right\rfloor - 1 \approx \left\lfloor \frac{5}{\alpha} \right\rfloor.
\]
The second parameter, \( l \) (with \( 0 < l \leq 1 \)), determines how many iterations (i.e., \( \lceil IL \rceil \)) the algorithm will use a certain value of the step size (see Step 1). As we can notice from Step 2, the algorithm will reach the lower bound of the normalized step size \( \alpha \) after \( k_{\text{max}} \lceil IL \rceil \) iterations; from this point of view, it is not recommended to set a very low value of \( \alpha \) (e.g., \( \alpha < 0.01 \)), which leads to a high value of \( k_{\text{max}} \) according to (16). Since the convergence rate of the NLMS-based algorithms depends on the character of the input signal [1–3] (in terms of the condition number of the correlation matrix), the value of \( l \) should be chosen accordingly. For example, for white Gaussian inputs, we can choose a smaller value of \( l \), e.g., \( l < 0.3 \). For more correlated inputs (such as autoregressive processes), the value of \( l \) should be increased, e.g., \( 0.3 \leq l < 0.5 \). Furthermore, for more challenging signals such as speech (i.e., highly correlated and nonstationary), the value of \( l \) could be even higher, e.g., \( 0.5 \leq l \leq 1 \).

The computational complexity of the proposed VSS-NLMS algorithm is similar to the complexity of the conventional NLMS benchmark, since the values of the step size parameters of the VSS-NLMS algorithm are a priori computed and scheduled based on (10) and (12); thus, these values can be used directly in Steps 1 and 2 of the algorithm, as describe before. The only extra computational amount is required in Step 3 of the VSS-NLMS algorithm, which is mainly related to the power estimates from (14) and (15). Nevertheless, this step requires only 5 multiplications, 2 additions, and 1 comparison, which represents a negligible computational effort as compared to the overall complexity of the conventional NLMS algorithm that is proportional to \( O(L) \), especially for very large values of \( L \) as in the context of AEC.

In addition, the proposed VSS-NLMS does not require additional parameters from the environment, so it can be considered a nonparametric algorithm [14]. Of course, it would be very useful to find more practical ways to set (or adjust) the parameter \( l \) within the algorithm, based on the convergence status of the adaptive filter, but this will be the subject for future works.

4. Simulation Results

Simulations were performed in the context of AEC, in a single-talk scenario [4–6]. In this case, the microphone (reference/desired) signal results in:

\[
\mathbf{d}(n) = \mathbf{h}^T \mathbf{x}(n) + \mathbf{v}(n) = y(n) + \mathbf{v}(n),
\]

where \( \mathbf{h} \) contains the coefficients of the acoustic impulse response (of length \( L \)), \( \mathbf{x}(n) \) contains the last \( L \) time samples of the far-end signal (i.e., the input signal), \( \mathbf{v}(n) \) is the background noise, and \( y(n) = \mathbf{h}^T \mathbf{x}(n) \) represents the echo signal. In this framework, the main goal is to identify the acoustic echo path, \( \mathbf{h} \), with an adaptive filter, \( \hat{\mathbf{h}}(n) \).

In our experiments, it was considered that \( \mathbf{v}(n) \) is a white Gaussian noise, so that the echo-to-noise ratio (ENR) was 20 dB. The ENR is defined as \( \frac{\sigma_y^2}{\sigma_v^2} \), where \( \sigma_y^2 \) denotes the variance of the echo signal. The acoustic echo path \( \mathbf{h} \) was obtained from a measured acoustic impulse response using \( L = 512 \) coefficients (depicted in Figure 2a and available at www.comm.pub.ro/plant, accessed on 24 February 2022), and the same length was used for the adaptive filter, using a sampling rate of 8 kHz. Three types of input signals were considered: (i) a white Gaussian noise, (ii) a first-order autoregressive process, referred to as AR(1), which was obtained by filtering a white Gaussian noise through a first-order autoregressive model with a pole at 0.8, and (iii) a recorded speech sequence (depicted in Figure 2b); for a longer simulation time (as needed in some experiments), this voice signal can be simply repeated several times. The experimental framework is summarized in Table A3 in Appendix A. In the middle of some simulations, an echo path change scenario was simulated by shifting the acoustic impulse response to the right by 12 samples, in order
to assess the tracking capabilities of the algorithms. The performance measure used in all the experiments was the normalized misalignment (in dB), which is evaluated as:

$$\text{NM}(n) = 20\log_{10} \frac{\| h - \hat{h}(n) \|}{\| h \|},$$

(18)

where $\| \cdot \|$ denotes the Euclidean norm.

![Figure 2](image-url) Experimental framework for acoustic echo cancellation: (a) measured acoustic impulse response used in the simulations, with $L = 512$; (b) recorded speech signal used in some of the experiments.

The parameters of the proposed VSS-NLMS algorithm were set as follows. Based on (16), we set $k_{\text{max}} = \lceil 5/\alpha \rceil$, using different values for the lower bound of the normalized step size parameter $\alpha$. The power estimates from (14) and (15) were computed using $K = 2$ for the white Gaussian input and $K = 6$ for the AR(1) process or speech sequence [14]; also, we used $Q = 3K$ and $\xi = 1.5$ [29]. The main parameters of the algorithms used in comparisons are provided in Table A4 in Appendix A, while their main values are summarized in Table A5 from the same section.

First, we evaluated the influence of the parameter $l$ on the performance of the proposed algorithm (using the lower bound $\alpha = 0.05$), for different types of input signals. The results are given in Figures 3–5, using white Gaussian noise, the AR(1) process, and the speech signal, respectively. These results supported the discussion from Section 3, concerning the influence of the parameter $l$ (and the recommended choices). Consequently, in the following simulations, we used $l = 0.1$ for white Gaussian inputs, $l = 0.3$ for the AR(1) process, and $l = 0.5$ for the speech signal.

Next, the performance of the VSS-NLMS algorithm was evaluated for different values of the lower bound of the normalized step size parameter, i.e., $\alpha = 0.1, 0.05, 0.03$, and 0.01. The results are given in Figure 6, using a white Gaussian noise as the input. It can be noticed that the algorithm was efficient for different values of $\alpha$. However, as was explained in Section 3, it is not recommended to choose a very small value of this lower bound.
Figure 3. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.05$ and misalignment of the VSS-NLMS algorithm using different values of the parameter $l$ and the lower bound $\alpha = 0.05$. The input signal is white and Gaussian, $L = 512$, and ENR = 20 dB.

Figure 4. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.05$ and misalignment of the VSS-NLMS algorithm using different values of the parameter $l$ and the lower bound $\alpha = 0.05$. The input signal is an AR(1) process, $L = 512$, and ENR = 20 dB.
Figure 5. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.05$ and misalignment of the VSS-NLMS algorithm using different values of the parameter $l$ and the lower bound $\alpha = 0.05$. The input signal is speech, $L = 512$, and ENR $= 20$ dB.

Figure 6. Misalignment of the NLMS algorithm with different values of $\alpha$ and misalignment of the VSS-NLMS algorithm using $l = 0.1$ and different values of the lower bound for the normalized step size parameter. The input signal is white and Gaussian, $L = 512$, and ENR $= 20$ dB.
In the following experiments, the nonparametric VSS-NLMS (NPVSS-NLMS) algorithm proposed in [14] is introduced for comparison, assuming that the true power of the system noise, $\sigma_v^2$, is available for this algorithm. The update of the NPVSS-NLMS algorithm is:

$$\hat{h}(n) = \hat{h}(n-1) + \frac{\bar{\mu}(n)x(n)e(n)}{x^T(n)x(n)},$$

(19)

where:

$$\bar{\mu}(n) = \begin{cases} 
1 - \frac{\hat{\sigma}_e^2}{\hat{\sigma}_e^2(n)}, & \text{if } \hat{\sigma}_e^2(n) \geq \sigma_v^2 \\
0, & \text{otherwise}
\end{cases}$$

(20)

and $\hat{\sigma}_e^2(n)$ was evaluated similar to (14). Furthermore, a very small positive constant should be added to the denominator of $\bar{\mu}(n)$, in order to avoid any potential division by zero related to the estimation of $\hat{\sigma}_e^2(n)$. It should be outlined that the NPVSS-NLMS algorithm is usually considered as a benchmark (for many VSS-NLMS algorithms), due to its good performance and practical features. In the following experiments, the proposed VSS-NLMS algorithm uses the lower bound value $\alpha = 0.03$. In order to test the tracking capabilities of the algorithms, an echo path change scenario was considered. All three types of input signals were used, i.e., white Gaussian noise, the AR(1) process, and speech; the results are provided in Figures 7–9, respectively. It can be noticed that the NPVSS-NLMS algorithm outperformed the proposed VSS-NLMS algorithm in the case of the white Gaussian input. However, the proposed algorithm performed better for the AR(1) process and speech inputs.
Figure 8. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.03$, misalignment of the NPVSS-NLMS algorithm, and misalignment of the VSS-NLMS algorithm using $l = 0.3$ and the lower bound $\alpha = 0.03$. The input signal is an AR(1) process, $L = 512$, and ENR = 20 dB, and the echo path changes at time 60 s.

Figure 9. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.03$, misalignment of the NPVSS-NLMS algorithm, and misalignment of the VSS-NLMS algorithm using $l = 0.5$ and the lower bound $\alpha = 0.03$. The input signal is speech, $L = 512$, and ENR = 20 dB, and the echo path changes at time 135 s.
Finally, a variation of the ENR was considered, in order to assess the robustness of the algorithms. In Figure 10, the ENR decreased from 20 dB to 10 dB after 120 s. In this experiment, the input signal was speech. Here, we considered that the initial value of $\sigma^2_v$ was available for the NPVSS-NLMS algorithm, but not the new one (i.e., after the decrease of the ENR). As expected, the proposed VSS-NLMS algorithm reacted similar to the NLMS algorithm with the smaller normalized step size, outperforming the NPVSS-NLMS algorithm.

![Figure 10](image)

Figure 10. Misalignment of the NLMS algorithm with $\alpha = 1$ and $\alpha = 0.03$, misalignment of the NPVSS-NLMS algorithm, and misalignment of the VSS-NLMS algorithm using $l = 0.5$ and the lower bound $\alpha = 0.03$. The input signal is speech, $L = 512$, and the ENR decreases from 20 dB to 10 dB after 120 s.

5. Conclusions

In this short communication, we developed a simple VSS-NLMS algorithm from a data reuse perspective. The analysis of the data reuse process in conjunction with the NLMS algorithm revealed the convergence modes of this algorithm. As a consequence, a sequence of normalized step sizes can be a priori scheduled to further design a VSS mechanism. Therefore, the resulting VSS-NLMS algorithm based on this approach had almost the same computational complexity as the conventional NLMS algorithm. Simulations were performed in the context of acoustic echo cancellation, and the results indicated the good performance of the proposed algorithm, in terms of its convergence rate, tracking capability, and accuracy of the estimate. In future works, we plan to implement and assess the performance of the proposed VSS-NLMS algorithm in a real-world experimental framework of an AEC application. In addition, we aim to extend the solution toward the affine projection algorithm, targeting further performance improvements, especially in terms of the convergence rate and tracking.

Author Contributions: Conceptualization, A.-G.R.; software, C.P.; methodology, J.B.; formal analysis, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS—UEFISCDI, project number: PN-III-P1-1.1-TE-2019-0420, within PNCDI III.
Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Table A1. List of acronyms.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEC</td>
<td>acoustic echo cancellation</td>
</tr>
<tr>
<td>AR</td>
<td>autoregressive</td>
</tr>
<tr>
<td>AR(1)</td>
<td>first-order autoregressive process</td>
</tr>
<tr>
<td>ENR</td>
<td>echo-to-noise ratio</td>
</tr>
<tr>
<td>NLMS</td>
<td>normalized least-mean-square</td>
</tr>
<tr>
<td>NPVSS-NLMS</td>
<td>nonparametric variable step size normalized least-mean-square</td>
</tr>
<tr>
<td>VSS</td>
<td>variable step size</td>
</tr>
<tr>
<td>VSS-NLMS</td>
<td>variable step size normalized least-mean-square</td>
</tr>
</tbody>
</table>

Table A2. Notation, symbols, and parameters.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>transpose of a vector or a matrix</td>
</tr>
<tr>
<td>$||_2$</td>
<td>$\ell_2$ (Euclidean) norm</td>
</tr>
<tr>
<td>$E[\cdot]$</td>
<td>mathematical expectation</td>
</tr>
<tr>
<td>$\log(\cdot)$</td>
<td>natural logarithm</td>
</tr>
<tr>
<td>$\log_{10}(\cdot)$</td>
<td>decimal logarithm</td>
</tr>
<tr>
<td>$\lceil \cdot \rceil$</td>
<td>ceiling function</td>
</tr>
<tr>
<td>$n$</td>
<td>discrete-time index</td>
</tr>
<tr>
<td>$L$</td>
<td>length of the filter</td>
</tr>
<tr>
<td>$\lambda = 1 - 1/(KL)$</td>
<td>weighting parameter, with $K \geq 1$</td>
</tr>
<tr>
<td>$\gamma = 1 - 1/(QL)$</td>
<td>weighting parameter, with $Q = 3K$</td>
</tr>
<tr>
<td>$\xi$</td>
<td>specific threshold, with $\xi = 1.5$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>normalized step size parameter of the NLMS algorithm (lower bound of the VSS-NLMS algorithm)</td>
</tr>
<tr>
<td>$\mu(n)$</td>
<td>variable step size parameter of the VSS-NLMS algorithm</td>
</tr>
<tr>
<td>$\tilde{\mu}(n)$</td>
<td>variable step size parameter of the NPVSS-NLMS algorithm</td>
</tr>
<tr>
<td>$h$</td>
<td>impulse response of the unknown system (echo path)</td>
</tr>
<tr>
<td>$\hat{h}(n)$</td>
<td>impulse response of the adaptive filter</td>
</tr>
<tr>
<td>$\hat{h}_k(n)$</td>
<td>impulse response of the adaptive filter in the $k$th data reuse iteration</td>
</tr>
<tr>
<td>$k_{\text{max}}$</td>
<td>maximum number of data reuse iterations</td>
</tr>
<tr>
<td>$\alpha_{\text{max}}$</td>
<td>maximum value of the normalized step size</td>
</tr>
<tr>
<td>$q$</td>
<td>specific step of the data reuse iteration, with $q = 1, 2, \ldots, k_{\text{max}}$</td>
</tr>
<tr>
<td>$l$</td>
<td>specific parameter of the VSS-NLMS algorithm, with $0 &lt; l \leq 1$</td>
</tr>
<tr>
<td>$x(n)$</td>
<td>input signal (far-end signal)</td>
</tr>
<tr>
<td>$x(n)$</td>
<td>vector containing the most recent $L$ time samples of the input signal</td>
</tr>
<tr>
<td>$d(n)$</td>
<td>reference or desired signal (microphone signal)</td>
</tr>
<tr>
<td>$y(n)$</td>
<td>output signal of the unknown system (echo signal)</td>
</tr>
<tr>
<td>$e(n)$</td>
<td>a priori error signal</td>
</tr>
<tr>
<td>$e_1(n)$</td>
<td>a posteriori error signal</td>
</tr>
<tr>
<td>$e_k(n)$</td>
<td>$k$th-order a posteriori error signal</td>
</tr>
<tr>
<td>$n(n)$</td>
<td>additive noise signal (background noise)</td>
</tr>
<tr>
<td>$\sigma^2_n$</td>
<td>variance of the noise signal</td>
</tr>
<tr>
<td>$\sigma^2_e$</td>
<td>variance of the output (echo) signal</td>
</tr>
<tr>
<td>$\hat{\sigma}^2_e(n)$</td>
<td>power estimate of the error signal</td>
</tr>
<tr>
<td>$\hat{\sigma}^2_e(n)$</td>
<td>power estimate of the noise signal</td>
</tr>
</tbody>
</table>
Table A3. Experimental framework.

<table>
<thead>
<tr>
<th>Filter Length</th>
<th>Echo-to-Noise Ratio</th>
<th>Input Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 512$</td>
<td>ENR = 20 dB</td>
<td>white Gaussian noise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AR(1) process speech</td>
</tr>
</tbody>
</table>

Table A4. Main parameters of the algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLMS</td>
<td>$\alpha$ (normalized step size)</td>
</tr>
<tr>
<td>NPVSS-NLMS</td>
<td>$\sigma_v^2$ (known or estimated) $K$</td>
</tr>
<tr>
<td>VSS-NLMS</td>
<td>$\alpha$ (lower bound) $K$ $l$</td>
</tr>
</tbody>
</table>

Table A5. Settings of the main parameters.

<table>
<thead>
<tr>
<th>↓Parameters/Inputs</th>
<th>White Gaussian Noise</th>
<th>AR(1) Process</th>
<th>Speech Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l$</td>
<td>0.1</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>$K$</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

References


