A PRACTICAL DATA-REUSE ADAPTIVE ALGORITHM FOR ACOUSTIC ECHO CANCELLATION

Constantin Paleologu[†] *and Jacob Benesty*[‡]

[†] University Politehnica of Bucharest, Romania, e-mail: pale@comm.pub.ro
[‡] INRS-EMT, University of Quebec, Montreal, Canada, e-mail: benesty@emt.inrs.ca

ABSTRACT

There are many strategies to improve the overall performance of the classical adaptive filters. Among these strategies, the data-reuse algorithms aim to improve the convergence rate by reusing the same set of data (i.e., the input and reference signals) several times. Another possibility is to use a variable step size (VSS) to achieve a proper compromise between the convergence rate and misadjustment of the adaptive filter. Nevertheless, both approaches increase the computational complexity. In this paper, we present an efficient data-reuse algorithm, which is the result of a combination between 1) a low-complexity implementation of the data-reuse process and 2) a simple and practical mechanism for controlling the step size. Simulations performed in the context of acoustic echo cancellation indicate the good performance of the proposed approach.

Index Terms— Adaptive filters, acoustic echo cancellation, data-reuse, variable step size (VSS).

1. INTRODUCTION

Adaptive filters are widely used in many signal processing applications. Maybe the most popular adaptive algorithm is the least-mean-square (LMS) [1], mainly because it is simple and easy to implement. However, its convergence rate is reduced when dealing with high length filters or correlated inputs. Motivated by these limitations, many interesting strategies were developed in order to improve the convergence features of the LMS-based algorithms.

It is known that the classical LMS algorithm works in a sample-by-sample manner and performs a single filter update for each set of data (i.e., the input signal vector and the reference signal). Consequently, when targeting a higher convergence rate, a natural approach is to perform more than one filter update for the same set of data. This is the straightforward way to obtain the so-called data-reuse LMS (DR-LMS) algorithm [2], [3]. Obviously, this is not an attractive approach from the complexity point of view. Moreover, it was shown that the convergence rate of the DR-LMS algorithm lies between the LMS and the normalized LMS (NLMS) algorithms [4]. Following the data-reuse idea, many interesting approaches have been developed in order to address the compromise between convergence rate and complexity, e.g., [5], [6], [7], and the data-reuse algorithms have been involved in the context of different applications, e.g., [8], [9], [10]. An insightful analysis of the data-reuse algorithms in connection with different approaches can be found in [9].

Another strategy to improve the overall performance of LMS-based algorithms is to control the adaptation step size. The choice of this parameter leads to a compromise between the convergence rate and misadjustment. In this context, variable step-size (VSS) adaptive filters are designed to achieve both fast convergence/tracking and low misadjustment. Many different VSS schemes have been proposed, e.g., [11], [12], [13], and references therein. However, most of these algorithms depend on several parameters that are not easy to tune in practice.

In this paper, we propose an efficient data-reuse algorithm, which combines two practical features. First, the data-reuse process is performed in a low-complexity manner, which is also simple to implement. Second, the VSS of the algorithm is designed to be easy to control in practice. The rest of the paper is organized as follows. Section 2 briefly reviews the basic data-reuse principle and algorithms. The proposed algorithm is developed in Section 3. Experimental results performed in the context of acoustic echo cancellation are provided in Section 4. Finally, Section 5 concludes this work.

2. DATA-REUSE ADAPTIVE ALGORITHMS

The data-reuse approach was first developed in combination with the LMS algorithm. Let us consider the system identification setup, having the reference (or desired) signal obtained as

$$\begin{aligned} d(n) &= \mathbf{h}^T \mathbf{x}(n) + w(n) \\ &= y(n) + w(n), \end{aligned} \quad (1)$$

where n represents the discrete-time index,

$$\mathbf{h} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{L-1} \end{bmatrix}^T$$

This work was supported under the Grant POSDRU/89/1.5/S/62557 and Grant UEFISCDI PN-II-RU-TE no. 7/5.08.2010.

is the impulse response (of length L) of the system that we need to identify, the superscript ^{T} denotes transpose of a vector or a matrix,

$$\mathbf{x}(n) = \left[\begin{array}{cccc} x(n) & x(n-1) & \cdots & x(n-L+1) \end{array}\right]^T$$

is a vector containing the most recent L samples of the zero-mean input signal $x(n)$, and $w(n)$ is a zero-mean additive noise signal, which is independent of $x(n)$. In this context, the main objective is to identify $\mathbf{h}$ with an adaptive filter,

$$\hat{\mathbf{h}}(n) = \left[\begin{array}{cccc} \hat{h}_0(n) & \hat{h}_1(n) & \cdots & \hat{h}_{L-1}(n) \end{array}\right]^T.$$

The well-known LMS algorithm [1] is defined by the following equations:

$$\begin{aligned} e(n) &= d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n), && (2) \\ \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + \mu\mathbf{x}(n)e(n), && (3) \end{aligned}$$

where μ is the step-size parameter ($0 < \mu < 2/\lambda_{\text{max}}$, where λ_{max} is the maximum eigenvalue of the input signal's correlation matrix). It can be noticed that the LMS filter update is performed only once for each set of data, i.e., $\mathbf{x}(n)$ and $d(n)$. The basic idea of the DR-LMS algorithm [2], [3] is to repeat this process for the same time index n, i.e., to reuse the same set of data N times. Consequently, the DR-LMS algorithm can be summarized as follows:

$$\begin{aligned} \text{Initialization: } \hat{\mathbf{h}}_0(n) &= \hat{\mathbf{h}}(n-1) \\ \text{Data-reuse: } i &= 1, 2, \ldots, N \\ e_i(n) &= d(n) - \hat{\mathbf{h}}_{i-1}^T(n)\mathbf{x}(n) && (4) \\ \hat{\mathbf{h}}_i(n) &= \hat{\mathbf{h}}_{i-1}(n) + \mu\mathbf{x}(n)e_i(n) && (5) \\ \text{Update: } \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}_N(n). \end{aligned}$$

For $N = 1$, the DR-LMS is equivalent to the LMS algorithm. Also, as it was shown in [4], the filter update of the DR-LMS algorithm is equivalent to

$$\begin{aligned} \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) && (6) \\ &+ \frac{\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} e(n) \left\{1 - \left[1 - \mu\mathbf{x}^T(n)\mathbf{x}(n)\right]^N\right\}. \end{aligned}$$

Consequently, for $N \to \infty$ we get

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}, \qquad (7)$$

which is equivalent to the update of the NLMS algorithm. In practice, a positive adaptation constant $0 < \alpha < 2$ (i.e., the normalized step size) multiplies the numerator of the second term in the right-hand side of (7) to achieve a proper compromise between the convergence rate and the misadjustment. Also, a positive constant δ (i.e., the regularization parameter) is added to the denominator of the same term, in order to avoid division by small numbers. For the sake of clarity, we do not take into account these two parameters for the moment.

Concluding, the DR-LMS algorithm is not very practical in its classical form [i.e., (4) and (5)], since it requires an infinite complexity to attain the performance of the NLMS. In order to address this issue, the algorithm proposed in [5] instead of iterating with an LMS on the same present data, it iterates with data from the past and present with an NLMS. Therefore, this data-reuse NLMS (DR-NLMS) algorithm can be described as follows:

$$\begin{aligned} \text{Initialization: } \hat{\mathbf{h}}_0(n) &= \hat{\mathbf{h}}(n-1) \\ \text{Data-reuse: } i &= 1, 2, \ldots, N \\ e_i(n) &= d(n-i+1) && (8) \\ &\quad - \hat{\mathbf{h}}_{i-1}^T(n)\mathbf{x}(n-i+1) \\ \hat{\mathbf{h}}_i(n) &= \hat{\mathbf{h}}_{i-1}(n) && (9) \\ &\quad + \frac{\mathbf{x}(n-i+1)e_i(n)}{\mathbf{x}^T(n-i+1)\mathbf{x}(n-i+1)} \\ \text{Update: } \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}_N(n). \end{aligned}$$

According to (8) and (9) [as compared to (4) and (5), respectively], the only two differences between the DR-NLMS [5] and the DR-LMS [2], [3] algorithms are: 1) at iteration i, we use $x(n-i+1)$ [resp. $d(n-i+1)$] instead of $x(n)$ [resp. $d(n)$] and 2) at iteration i, we use $1/[\mathbf{x}^T(n-i+1)\mathbf{x}(n-i+1)]$ instead of μ. Due to its specific features, this DR-NLMS algorithm converges to the Wiener solution faster than the NLMS [5].

3. VSS DATA-REUSE ALGORITHM

Let us start by rewriting the previous DR-NLMS algorithm in a different manner. In the first iteration [see (8)–(9)], we have

$$\begin{aligned} e_1(n) &= d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n), && (10) \\ \hat{\mathbf{h}}_1(n) &= \hat{\mathbf{h}}(n-1) + \frac{\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} e_1(n). && (11) \end{aligned}$$

Further, in the second iteration, by taking (10) and (11) into account, we get

$$\begin{aligned} e_2(n) &= d(n-1) - \hat{\mathbf{h}}_1^T(n)\mathbf{x}(n-1) \\ &= d(n-1) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n-1) && (12) \\ &\quad - \frac{\mathbf{x}^T(n-1)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} e_1(n), \\ \hat{\mathbf{h}}_2(n) &= \hat{\mathbf{h}}_1(n) + \frac{\mathbf{x}(n-1)}{\mathbf{x}^T(n-1)\mathbf{x}(n-1)} e_2(n) \\ &= \hat{\mathbf{h}}(n-1) + \frac{\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} e_1(n) && (13) \\ &\quad + \frac{\mathbf{x}(n-1)}{\mathbf{x}^T(n-1)\mathbf{x}(n-1)} e_2(n). \end{aligned}$$

Continuing the same process until iteration N, we obtain an equivalent compact form of the DR-NLMS algorithm, i.e.,

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1), \quad (14)$$
$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{X}(n)\mathbf{M}^{-1}(n)\mathbf{e}(n), \quad (15)$$

where

$$\mathbf{d}(n) = \begin{bmatrix} d(n) & d(n-1) & \cdots & d(n-N+1) \end{bmatrix}^T,$$
$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{x}(n) & \mathbf{x}(n-1) & \cdots & \mathbf{x}(n-N+1) \end{bmatrix},$$

and $\mathbf{M}(n)$ is a $N \times N$ lower triangular matrix (i.e., all the elements above the main diagonal are zeroes) with the non-zero elements defined as

$$\mathbf{M}_{l,k}(n) = \mathbf{x}^T(n-l+1)\mathbf{x}(n-k+1), \quad (16)$$

where $l, k = 1, 2, \ldots, N$ and $l \geq k$.

It can noticed that the lower triangular parts of the matrices $\mathbf{M}(n)$ and $\mathbf{X}^T(n)\mathbf{X}(n)$ are identical. Consequently, it is natural to make a connection with the affine projection algorithm (APA) [14], which is based on the update:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{X}(n)\right]^{-1}\mathbf{e}(n). \quad (17)$$

As we did it in the case of the NLMS algorithm [see (7)], the normalized step size α and the regularization matrix $\delta\mathbf{I}_N(n)$ [where $\mathbf{I}_N(n)$ denotes the $N \times N$ identity matrix] are also omitted here for the sake of clarity. Note that N plays the role of a multi-dimensional projection (or projection order) in the case of the APA.

The regular form of the DR-NLMS algorithm given in (8) and (9) is quite inefficient from a complexity point of view (as compared to the NLMS), because it requires $2NL$ operations per time sample. However, the DR-NLMS as shown in (14) and (15) can be computed more efficiently using $2L + O(N^2)$ operations by borrowing some of the ideas used in the fast affine projection algorithm [15] or dichotomous coordinate descent (DCD) method [16]; also, it can be noticed that $\mathbf{M}(n)$ is a triangular matrix and all its elements can be computed recursively [see (16)]. Therefore, when $N \ll L$, which is the case with long adaptive filters (e.g., like in echo cancellation), the complexity of the DR-NLMS from (14) and (15) is comparable to the NLMS (and lower than the APA) but with a faster convergence rate.

Next, we can take advantage of the similarity between DR-NLMS and APA [see (15) and (17)], in order to obtain a VSS-DR-NLMS algorithm. In this sense, it is expected that a VSS approach designed for APA should be also reliable for the DR-NLMS algorithm. A simple but efficient VSS-APA was proposed in [17]; the update of this algorithm is defined as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{X}(n)\right]^{-1}\mathbf{D}_\alpha(n)\mathbf{e}(n), \quad (18)$$

where

$$\mathbf{D}_\alpha(n) = \mathrm{diag}\begin{bmatrix} \alpha_1(n) & \alpha_1(n) & \cdots & \alpha_N(n) \end{bmatrix} \quad (19)$$

is an $N \times N$ diagonal matrix. It is clear that (17) is obtained when $\alpha_1(n) = \alpha_2(n) = \cdots = \alpha_N(n) = 1$. The VSS-APA presented in [17] aims to recover the system noise [see (1)] from the error signal of the adaptive filter, which is the basic goal in system identification. Following this condition, the elements of the step-size matrix $\mathbf{D}_\alpha(n)$ result as [17]

$$\alpha_p(n) = 1 - \frac{\sigma_w(n)}{\sigma_{e_p}(n)}, \quad (20)$$

where $\sigma_w^2(n) = E[w^2(n)]$ is the variance of the system noise [with $E(\cdot)$ denoting mathematical expectation], $e_p(n)$ denotes the pth element of the vector $\mathbf{e}(n)$, with $p = 1, 2, \ldots, N$, and $\sigma_{e_p}^2(n) = E[e_p^2(n)]$. The variable in the denominator can be computed in a recursive manner, i.e.,

$$\sigma_{e_p}^2(n) = \lambda\sigma_{e_p}^2(n-1) + (1-\lambda)e_p^2(n), \quad (21)$$

where $\lambda = 1 - 1/(KL)$, with $K > 1$. Also, the power of the system noise can be evaluated in different practical ways [17], [18]. Based on these findings, the update of the VSS-DR-NLMS algorithm results as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{X}(n)\mathbf{M}^{-1}(n)\mathbf{D}_\alpha(n)\mathbf{e}(n). \quad (22)$$

Since the VSS-APA presented in [17] is a generalization of the non-parametric VSS-NLMS (NPVSS-NLMS) algorithm proposed in [13], we could interpret the previous VSS-DR-NLMS as a data-reuse version of the same NPVSS-NLMS algorithm. Finally, it should be mentioned that for $N = 1$ all these three algorithms (i.e., NPVSS-NLMS, VSS-APA, and VSS-DR-NLMS) are equivalent.

4. SIMULATION RESULTS

Simulations are performed in the context of acoustic echo cancellation [19]. The acoustic impulse response has 512 coefficients and the sampling rate is 8 kHz; the same length is used for all adaptive filters, i.e., $L = 512$. The far-end signal (i.e., the input signal) is either an AR(1) process generated by filtering a white Gaussian noise through a first-order system $1/(1 - 0.8z^{-1})$, or a speech sequence. An independent white Gaussian noise signal $w(n)$ is added to the echo signal $y(n)$, with 30-dB echo-to-noise ratio (ENR).

For practical reasons, the regularization term is included within the update of the algorithms, i.e., the constant δ for the NLMS and NPVSS-NLMS algorithms, or the matrix $\delta\mathbf{I}_N(n)$ for the DR-NLMS, APA, VSS-APA, and VSS-DR-NLMS. For all the adaptive filters we set $\delta = 20\sigma_x^2$, where $\sigma_x^2 = E[x^2(n)]$ is the variance of $x(n)$. In practice, the value of δ should be selected taking into account the level of the ENR [20], [21], [22]. In order to evaluate the tracking capabilities of the algorithms, an abrupt change in the acoustic environment is introduced by shifting the impulse response to the right by 12 samples, in the middle of each experiment.

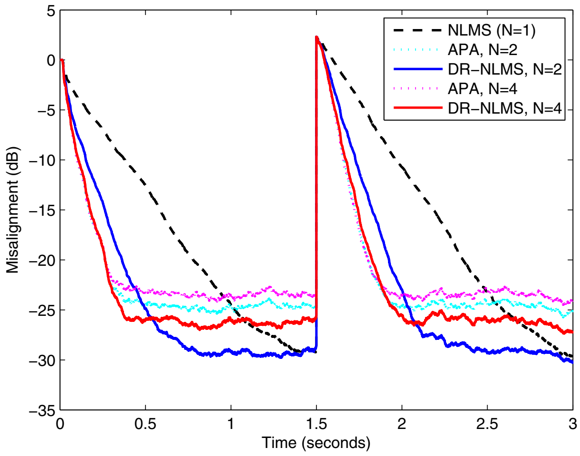

Fig. 1. Misalignment of the APA and DR-NLMS algorithm (using $\alpha = 1$), for different values of N. The input signal is an AR(1) process and echo path changes at time 1.5 seconds.

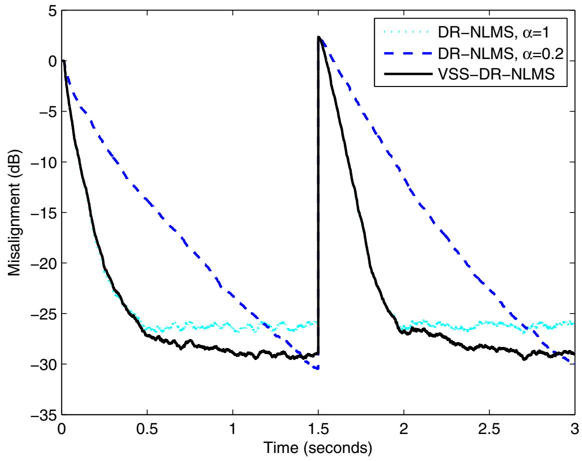

Fig. 2. Misalignment of the DR-NLMS algorithms using $\alpha = 1$ and $\alpha = 0.2$, and misalignment of the VSS-DR-NLMS; $N = 4$ for all the algorithms. The input signal is an AR(1) process and echo path changes at time 1.5 seconds.

The measure of performance is the normalized misalignment (in dB), defined as $20 \log_{10} \left\| \mathbf{h} - \hat{\mathbf{h}}(n) \right\|_2 / \left\| \mathbf{h} \right\|_2$ (with $\|\cdot\|_2$ denoting the ℓ_2 norm).

In the first experiment we compare the DR-NLMS algorithm and APA, for different values of N (i.e., 1, 2, and 4). Clearly, for $N = 1$ both algorithms are equivalent to the NLMS. The normalized step-size for all the algorithms is set to $\alpha = 1$, thus providing the fastest convergence mode. The results are presented in Fig. 1, where the input signal is an AR(1) process. First, it can be noticed that the convergence rate of both DR-NLMS and APA increases when the value of N increases; however, the steady-state misalignment also increases with N. Also, it is important to notice that for the same value of N, the convergence rate of the APA is better as compared to the DR-NLMS. Nevertheless, the differences (in terms of convergence rate and tracking) between DR-NLMS and APA become smaller when the value of N increases. For $N = 4$ both algorithms achieve very similar convergence rate and tracking, but the DR-NLMS outperforms APA in terms of the final misalignment level. This is an important result, since the DR-NLMS algorithm is also less computationally expensive as compared to the APA, especially when the value of N increases.

The second experiment compares the DR-NLMS algorithm using two normalized step size ($\alpha = 1$ and $\alpha = 0.2$) with the VSS-DR-NLMS. The results are presented in Fig. 2, where the input signal is an AR(1) process. It can be noticed that the proposed VSS-DR-NLMS algorithm has an initial convergence rate (and tracking) similar to the DR-NLMS with $\alpha = 1$, but it achieves a lower misalignment, which is close to the one obtained by the DR-NLMS with $\alpha = 0.2$.

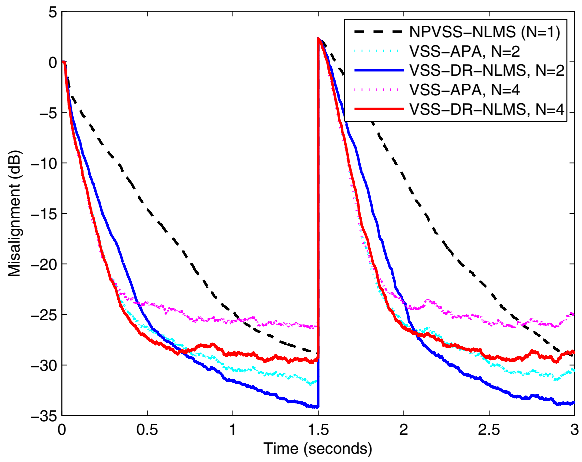

Fig. 3. Misalignment of the VSS-APA and VSS-DR-NLMS algorithm, for different values of N. The input signal is an AR(1) process and echo path changes at time 1.5 seconds.

Next, the proposed VSS-DR-NLMS algorithm is compared with the VSS-APA [17], for different values of N (i.e., 1, 2, and 4). For $N = 1$ both algorithms are equivalent to the NPVSS-NLMS developed in [9]. The results are depicted in Fig. 3; the input signal is an AR(1) process. As expected, both VSS-APA and VSS-DR-NLMS outperform the NPVSS-NLMS algorithm in terms of convergence rate and tracking. Also, for $N = 4$ the proposed VSS-DR-NLMS algorithm performs similarly to the VSS-APA taking into account the previous performance criteria. However, the VSS-DR-NLMS achieves a lower steady-state misalignment as compared to the VSS-APA.

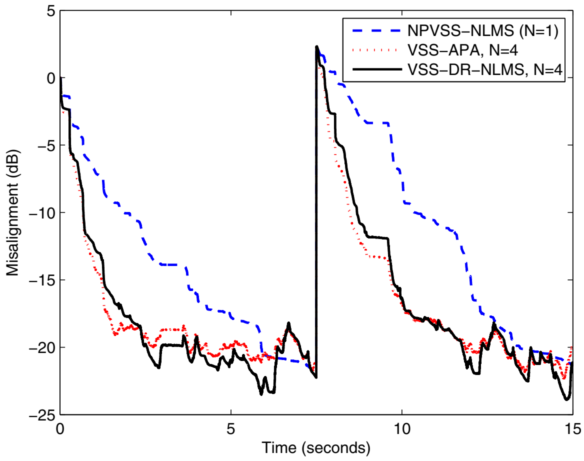

Fig. 4. Misalignment of the NPVSS-NLMS algorithm, VSS-APA, and VSS-DR-NLMS, for $N = 4$. The input signal is speech and echo path changes at time 7.5 seconds.

Finally, a speech sequence is considered as input in the last simulation. The NPVSS-NLMS algorithm is compared with the VSS-APA and VSS-DR-NLMS, both using $N = 4$. The results are given in Fig. 4. Clearly, the NPVSS-NLMS is outperformed by the other algorithms, while the proposed VSS-DR-NLMS performs similarly to the VSS-APA.

5. CONCLUSIONS

An efficient data-reuse algorithm has been proposed in this paper. The algorithm results as a combination between the DR-NLMS algorithm (implemented in a computationally efficient form) and a VSS approach originally developed for the APA. The overall complexity of the resulted algorithm is lower as compared with its VSS-APA counterpart. Simulation results performed in the context of acoustic echo cancellation indicate that the proposed VSS-DR-NLMS algorithm can achieve both fast convergence and tracking, but also low misalignment, inheriting the specific features of the VSS approach. Also, due to its nonparametric nature and simplicity, the proposed algorithm could be very suitable in real-world applications.

6. REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1985.

[2] S. Shaffer and C. S. Williams, "Comparison of LMS, alpha LMS, and data reusing LMS algorithms," in *Conference Record of the Seventeenth Asilomar Conference on Circuits, Systems and Computers*, Nov. 1983, pp. 260–264.

[3] R. Nitzberg, "Application of the normalized LMS algorithm to MSLC," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 21, pp. 79–91, Jan. 1985.

[4] S. Roy and J. J. Shynk, "Analysis of the data-reusing LMS algorithm," in *Proc. of the 32nd Midwest Symposium on Circuits and Systems*, 1990, pp. 1127–1130.

[5] B. A. Schnaufer and W. K. Jenkins, "New data-reusing LMS algorithms for improved convergence," in *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 1584–1588.

[6] J. A. Apolinário, Jr., M. L. R. de Campos, and P. S. R. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," *IEEE Trans. Signal Process.*, vol. 48, pp. 3235–3242, Nov. 2000.

[7] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Trans. Signal Process.*, vol. 51, pp. 124–134, Jan. 2003.

[8] S. Tantikovit and M. Z. Wang,"An adaptive data-reuse interference cancellation scheme for DS-CDMA systems," in *Proc. Int. Conf. Information, Communications and Signal Processing*, 1993, vol. 2, pp. 854–857.

[9] J. Benesty and T. Gänsler, "On data-reuse adaptive algorithms," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, 2003, pp. 31–34.

[10] F. Tong, B. Benson, Y. Li, and R. Kastner, "Channel equalization based on data reuse LMS algorithm for shallow water acoustic communication," in *Proc. Int. Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2010, pp. 95–98.

[11] A. Mader, H. Puder, and G. U. Schmidt, "Step-size control for acoustic echo cancellation filters – An overview," *Signal Process.*, vol. 80, pp. 1697–1719, Sept. 2000.

[12] H.-C. Shin, A. H. Sayed, and W.-J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, pp. 132–135, Feb. 2004.

[13] J. Benesty, H. Rey, L. Rey Vega, and S. Tressens, "A nonparametric VSS NLMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, pp. 581–584, Oct. 2006.

[14] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Jpn.*, vol. 67-A, pp. 19–27, May 1984.

[15] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. IEEE ICASSP*, 1995, pp. 3023–3026.

[16] Y. V. Zakharov, "Low complexity implementation of the affine projection algorithm," *IEEE Signal Process. Lett.*, vol. 15, pp. 557–560, 2008.

[17] C. Paleologu, J. Benesty, and S. Ciochină, "A variable step-size affine projection algorithm designed for acoustic echo cancellation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, pp. 1466–1478, Nov. 2008.

[18] M. A. Iqbal and S. L. Grant, "Novel variable step size NLMS algorithm for echo cancellation," in *Proc. IEEE ICASSP*, 2008, pp. 241–244.

[19] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Berlin, Germany: Springer-Verlag, 2001.

[20] J. Benesty, C. Paleologu, and S. Ciochină, "On regularization in adaptive filtering," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, pp. 1734–1742, Aug. 2011.

[21] C. Paleologu, J. Benesty, and S. Ciochină, "Regularization of the affine projection algorithm," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 58, pp. 366–370, June 2011.

[22] C. Paleologu, J. Benesty, and S. Ciochină, "Practical regularization of the affine projection algorithm," in *Proc. EUSIPCO*, 2011, pp. 704–708.