

Article

Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation

Ionuț-Dorinel Fîciu ¹, Cristian-Lucian Stanciu ^{1,*}, Constantin Paleologu ^{1,*} and Jacob Benesty ²

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

² INRS-EMT, University of Quebec, 800 de la Gauchetiere Ouest, Suite 6900, Montreal, QC H5A 1K6, Canada

* Correspondence: cristian@comm.pub.ro (C.-L.S.); pale@comm.pub.ro (C.P.)

Abstract: Stereophonic audio devices employ two loudspeakers and two microphones in order to create a bidirectional sound effect. In this context, the stereophonic acoustic echo cancellation (SAEC) setup requires the estimation of four echo paths, each one corresponding to a loudspeaker-to-microphone pair. The widely linear (WL) model was proposed in recent literature in order to simplify the handling of the SAEC mathematical model. Moreover, low complexity recursive least-squares (RLS) adaptive algorithms were developed within the WL framework and successfully tested for SAEC scenarios. This paper proposes to apply a data-reuse (DR) approach for the combination between the RLS algorithm and the dichotomous coordinate descent (DCD) iterative method. The resulting WL-DR-RLS-DCD algorithm inherits the convergence properties of the RLS family and requires an amount of mathematical operations attractive for practical implementations. Simulation results show that the DR approach improves the tracking capabilities of the RLS-DCD algorithm, with an acceptable surplus in terms of computational workload.

Keywords: adaptive filters; data-reuse (DR); dichotomous coordinate descent (DCD); recursive least-squares (RLS) algorithm; system identification; stereophonic acoustic echo cancellation (SAEC); widely linear (WL)



Citation: Fîciu, I.-D.; Stanciu, C.-L.; Paleologu, C.; Benesty, J. Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation. *Appl. Sci.* **2023**, *13*, 2227. <https://doi.org/10.3390/app13042227>

Academic Editors: Pyung Soo Kim and Bokyu Kwon

Received: 30 December 2022

Revised: 2 February 2023

Accepted: 7 February 2023

Published: 9 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The standard acoustic echo cancellation (AEC) setup is used by communication terminals comprised of one loudspeaker and one microphone. The AEC process employs an adaptive filter to estimate an unknown impulse response, which generates undesired replicas of the loudspeaker signal [1–3]. The goal is to eliminate the unknown system's output (or the echo) contribution from the mix of signals captured by the microphone, which is situated in the same room [1,4–7]. The corresponding unknown system identification configuration uses as input the loudspeaker signal, with the microphone's input as a reference. From the microphone contributions, an echo estimate is subtracted, and the resulting signal is sent to the interlocutor, ideally without any unwanted acoustic replicas.

When the communication terminal consists of N loudspeakers with M microphones, the number of possible echo paths is given by the value $N \times M$ [4,8–10]. The stereophonic communication terminal represents the minimal configuration which can provide the impression of audio directionality, having $N = M = 2$. In order to perform the stereophonic AEC (SAEC), we must take into consideration that each loudspeaker-to-microphone pair can generate unwanted acoustic replicas. Correspondingly, two input (loudspeaker) signals with two microphone signals, are used in four AEC-type system identification setups in order to estimate four corresponding unknown impulse responses [4,8,11–13]. Additionally, compared to the single channel scenarios, the SAEC process is also hindered by the existing correlation between the two acoustic channels, which might not generate unique solutions when using adaptive algorithms [8,9,14–17]. In the recent literature, the solution to the problem represented by the coherence between the acoustic channels has been approached

by employing a pre-distortion block, which mitigates the correlation with some cost in terms of signal quality [4,5,9,18].

In [5,18], the widely linear (WL) model was introduced in order to improve the handling of the SAEC framework when employing several types of adaptive filters. It groups the four adaptive filters and multiple real valued variables into a single adaptive filter and fewer complex valued variables (CRVs). Moreover, the WL approach also opened the way for other various developments proposed for SAEC systems, such as improved behaviour when working in low signal-to-noise ratio (SNR) conditions [19]. Although considerable effort was invested in enhancing robustness features, the SAEC setup interpreted using the WL framework leaves more room for applying developments which has proved to be successful for other configurations employing adaptive systems.

Among the essential performance indicators for any adaptive algorithm in AEC/SAEC scenarios (i.e., *system identification* setups) [1,4,20], the tracking speed represents the capacity of following any changes that might occur in the unknown echo paths. The current adaptive systems in the industry employ algorithms from the least-mean-square (LMS) family for most practical applications [1]. The LMS methods have the advantage of delivering satisfactory performance with a relatively limited arithmetic effort with respect to other potential solutions. However, they provide poor results when working with highly correlated input signals (such as speech) in terms of tracking speed and accuracy at steady state (i.e., when convergence is achieved).

Consequently, due to their capabilities of reducing the correlation of input signals and generating superior tracking performances, the recursive least-squares (RLS) family of algorithms is the subject to considerable research. They are also known for their excessively costly arithmetic workloads, which make them prohibitive for practical applications. However, a successful attempt at mitigating this problem is the combination between the exponentially weighted RLS and the dichotomous coordinate descent (DCD) iterations [5,21,22]. The RLS-DCD algorithm generates performance similar to classical RLS versions, such as the one employing the matrix inversion lemma [1]. The RLS-DCD replaces the classical matrix inversion problem with an auxiliary system of equations and exploits the time-shift nature of the input data in order to generate a solution using only additions and bit shifts. The overall complexity corresponding to the RLS-DCD algorithm is proportional to the adaptive filter's length multiplied by a small factor (in terms of multiplications and additions).

In this paper, we improve the tracking capabilities of the RLS-DCD algorithm working within the WL framework by employing the recently introduced data-reuse (DR) approach for adaptive algorithms from the RLS family [23,24]. The DR methodology is based on performing multiple filter updates using the same input data (i.e., in the same adaptive filter iteration). In the recent literature, it proved to generate superior performance for other RLS versions, with the cost of some extra arithmetic effort when used for real valued adaptive filters in system identification scenarios [23]. Simulation results will demonstrate that the namely WL-DR-RLS-DCD has increased tracking capabilities even when identifying long impulse responses and using highly correlated input signals, such as speech sequences. The complexity analysis will also reveal that the proposed method requires acceptable hardware resources, and it is a suitable candidate for practical applications in SAEC scenarios with long acoustic paths.

The remainder of this paper is organized as follows. In Section 2, the SAEC system configuration based on the WL model is introduced. Section 3 describes the complex valued WL-RLS-DCD algorithm for SAEC scenarios, and Section 4 introduces a new version of the presented adaptive method based on the DR approach with improved tracking capabilities. Simulation results are discussed in Section 5. Several relevant aspects are discussed in Section 6, and a few conclusions are drawn in Section 7 based on the theoretical contributions in the presented simulations.

2. System Model

For the SAEC configuration (see Figure 1), we refer to the acoustic channels as the *left* (L_c) and the *right* one (R_c). We employ the discrete time index n to denote the input (loudspeaker) signals as $x_{L_c}(n)$ and $x_{R_c}(n)$. They can be used to express two $L \times 1$ vectors comprising the most recent L input samples for each channel as [4]:

$$\mathbf{x}_{L_c}(n) = [x_{L_c}(n) \quad x_{L_c}(n-1) \quad \cdots \quad x_{L_c}(n-L+1)]^T, \quad (1)$$

$$\mathbf{x}_{R_c}(n) = [x_{R_c}(n) \quad x_{R_c}(n-1) \quad \cdots \quad x_{R_c}(n-L+1)]^T. \quad (2)$$

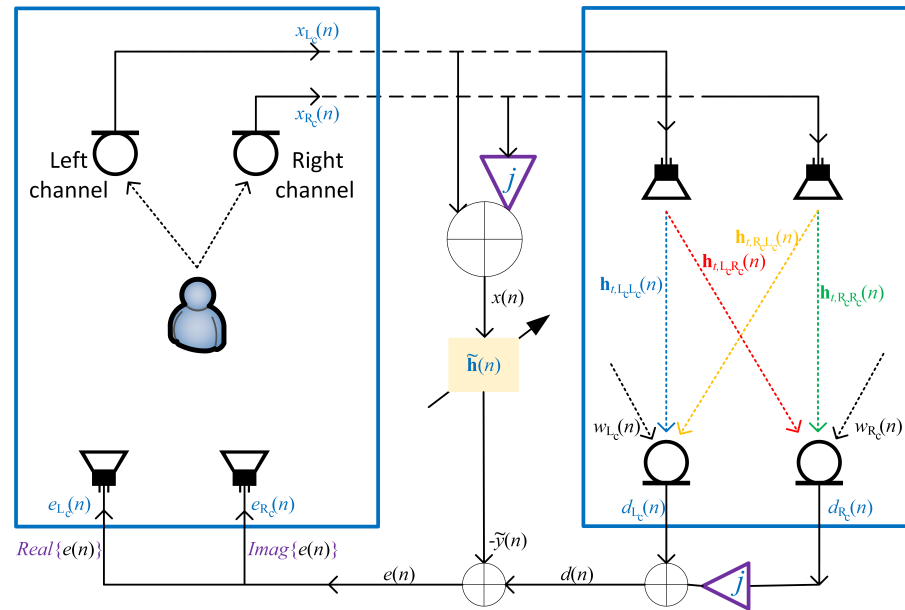


Figure 1. The WL model for SAEC systems.

Moreover, we combine the four loudspeaker-to-microphone echo path impulse responses of length L (denoted by the $L \times 1$ vectors $\mathbf{h}_{t,L_c L_c}$, $\mathbf{h}_{t,L_c R_c}$, $\mathbf{h}_{t,R_c L_c}$, and $\mathbf{h}_{t,R_c R_c}$) to write the echo contributions corresponding to the stereophonic channels [4]:

$$y_{L_c}(n) = \mathbf{h}_{t,L_c L_c}^T \mathbf{x}_{L_c}(n) + \mathbf{h}_{t,R_c L_c}^T \mathbf{x}_{R_c}(n), \quad (3)$$

$$y_{R_c}(n) = \mathbf{h}_{t,L_c R_c}^T \mathbf{x}_{L_c}(n) + \mathbf{h}_{t,R_c R_c}^T \mathbf{x}_{R_c}(n), \quad (4)$$

where superscript T denotes the transpose operator. Consequently, the microphone signals can be expressed as [4]:

$$d_{L_c}(n) = y_{L_c}(n) + w_{L_c}(n), \quad (5)$$

$$d_{R_c}(n) = y_{R_c}(n) + w_{R_c}(n), \quad (6)$$

where $w_{L_c}(n)$ and $w_{R_c}(n)$ represent environmental noise signals, which are uncorrelated with $y_{L_c}(n)$ and $y_{R_c}(n)$.

The WL model combines the previously introduced real random variables into fewer CRVs in order to improve the handling of the SAEC setup. The signals associated with the inputs and the outputs of the unknown echo paths can be paired using the notation $j = \sqrt{-1}$ into:

$$x(n) = x_{L_c}(n) + jx_{R_c}(n), \quad (7)$$

$$y(n) = y_{L_c}(n) + jy_{R_c}(n). \quad (8)$$

Moreover, the four unknown impulse responses can be combined into

$$\mathbf{h}_{\alpha,1} = \frac{\mathbf{h}_{t,L_c L_c} + \mathbf{h}_{t,R_c R_c}}{2}, \quad \mathbf{h}_{\alpha,2} = \frac{\mathbf{h}_{t,R_c L_c} - \mathbf{h}_{t,R_c R_c}}{2}, \quad (9)$$

$$\mathbf{h}_{\beta,1} = \frac{\mathbf{h}_{t,L_c L_c} - \mathbf{h}_{t,R_c R_c}}{2}, \quad \mathbf{h}_{\beta,2} = -\frac{\mathbf{h}_{t,R_c L_c} + \mathbf{h}_{t,L_c R_c}}{2}, \quad (10)$$

with the purpose of obtaining the complex valued forms [4]

$$\mathbf{h}_{\alpha} = \mathbf{h}_{\alpha,1} + j\mathbf{h}_{\alpha,2}, \quad (11)$$

$$\mathbf{h}_{\beta} = \mathbf{h}_{\beta,1} + j\mathbf{h}_{\beta,2}, \quad (12)$$

and express the output complex signal as

$$y(n) = \mathbf{h}_{\alpha}^H \mathbf{x}(n) + \mathbf{h}_{\beta}^H \mathbf{x}^*(n), \quad (13)$$

where the superscripts H and $*$ denote the transpose-conjugate and conjugate operators, respectively, and

$$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T \quad (14)$$

$$= \mathbf{x}_{L_c}(n) + j\mathbf{x}_{R_c}(n). \quad (15)$$

Consequently, the complex vectors from (11) and (12) can be concatenated to obtain the $2L \times 1$ complex valued vector comprising the information corresponding to the coefficients of the four unknown impulse responses

$$\tilde{\mathbf{h}}_t = [\mathbf{h}_{\alpha}^T \mathbf{h}_{\beta}^T]^T, \quad (16)$$

and a corresponding $2L \times 1$ input vector can be defined as

$$\tilde{\mathbf{x}}(n) = [\mathbf{x}^T(n) \mathbf{x}^H(n)]^T. \quad (17)$$

Using (16) and (17), a more compact expression can be written for the complex echo signal as

$$y(n) = \tilde{\mathbf{h}}_t^H \tilde{\mathbf{x}}(n). \quad (18)$$

Finally, by also expressing the complex valued noise $w(n)$ as

$$w(n) = w_{L_c}(n) + jw_{R_c}(n), \quad (19)$$

the microphone signals corresponding to the *left* and *right* channels can be written as:

$$d(n) = d_{L_c}(n) + jd_{R_c}(n) = y(n) + w(n). \quad (20)$$

The error signal can now be expressed as

$$e(n) = d(n) - \tilde{y}(n) = d(n) - \tilde{\mathbf{h}}^H(n) \tilde{\mathbf{x}}(n), \quad (21)$$

where $\tilde{y}(n)$ is the estimate of the complex valued system's output computed using the adaptive filter's set of coefficients $\tilde{\mathbf{h}}(n)$ (which approximates $\tilde{\mathbf{h}}_t$).

The described framework improves the handling of SAEC scenarios using adaptive algorithms. Moreover, it favors the development of corresponding new features by recasting four unknown system identification problems into a single one. The two-input/two-output system with real random variables is viewed as a single-input/single-output system with CRVs.

3. WL-RLS-DCD

For the RLS family of adaptive algorithms employed in SAEC scenarios with the WL framework, the vectors from (16) and (17) are interleaved instead of concatenated (as it was presented in the previous section). However, the new formulations do not affect the definition of the complex reference signal $d(n)$ [4,5].

Firstly, the vector comprising the input samples from (14) and (17) is redefined to reflect the new positioning of the complex valued samples and their transpose-conjugate counterparts. Consequently, we write:

$$\tilde{\mathbf{x}}(n) = [x(n) \ x^*(n) \ \dots \ x(n-L+1) \ x^*(n-L+1)]^T. \quad (22)$$

In a similar manner, the elements of (11) and (12) are also interleaved in order to express the new complex valued impulse response to be estimated [4,5]:

$$\tilde{\mathbf{h}}_t = [h_{\alpha,0} \ h_{\beta,0} \ \dots \ h_{\alpha,L-1} \ h_{\beta,L-1}]^T, \quad (23)$$

where $h_{\alpha,l}$ and $h_{\beta,l}$, with $l = 0, 1, \dots, L-1$, are the elements of the vectors $\mathbf{h}_\alpha$ and $\mathbf{h}_\beta$, respectively.

When employing the least-squares approach to generate the complex valued $2L \times 1$ estimate vector $\tilde{\mathbf{h}}(n)$ of $\tilde{\mathbf{h}}_t$, the corresponding cost function is written using (22), (23), and the complex observation $d(n)$ as:

$$J[\tilde{\mathbf{h}}(n)] = \sum_{i=1}^n \lambda^{n-i} [d(i) - \tilde{\mathbf{h}}^H(n) \tilde{\mathbf{x}}(i)]^2, \quad (24)$$

where $0 < \lambda \leq 1$ denotes the forgetting factor, which controls the memory of the algorithm (i.e., values closer to one increase the adaptive filter's performance in steady state, while low values for λ improve the tracking capabilities of the system) [1,4]. When performing the minimization of the error criterion, two new notations are introduced, i.e., the $2L \times 2L$ estimate of the correlation matrix

$$\mathbf{R}(n) = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{x}}(i) \tilde{\mathbf{x}}^H(i) = \lambda \mathbf{R}(n-1) + \tilde{\mathbf{x}}(n) \tilde{\mathbf{x}}^H(n), \quad (25)$$

and the $2L \times 1$ cross-correlation vector between the complex input signal and the desired signal

$$\mathbf{p}(n) = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{x}}(i) d(i) = \lambda \mathbf{p}(n-1) + \tilde{\mathbf{x}}(n) d(n). \quad (26)$$

Thus, the solution for the normal set of equations

$$\mathbf{R}(n) \tilde{\mathbf{h}}(n) = \mathbf{p}(n), \quad (27)$$

leads to the straight-forward expression for the adaptive filter's coefficients:

$$\tilde{\mathbf{h}}(n) = \mathbf{R}^{-1}(n) \mathbf{p}(n). \quad (28)$$

The direct approach for computing $\tilde{\mathbf{h}}(n)$ has a complexity of $\mathcal{O}(8L^3)$, which is too costly for practical implementations, such as the ones for SAEC configurations. In acoustic applications, the number of filter coefficients $2L$ can easily be in the range of hundreds or even thousands. Other classical solutions, such as the one based on Woodbury's identity (also known as the matrix inversion lemma [1,4]), provide an alternative to (28) with a complexity $\mathcal{O}(4L^2)$. However, the required amount of arithmetic operations is still prohibitive for most computer chips and would considerably increase device prices, power consumption, etc.

In this context, the combination between the exponentially weighted RLS adaptive algorithm and the DCD iterations was proposed as a computationally efficient method, with a complexity proportional to the adaptive filter's length multiplied by a small number (the actual details will be discussed in the next paragraphs). The WL-RLS-DCD is accepted as a reliable algorithm in multiple types of configurations for acoustic echo cancellation, including for stereophonic communication setups [5,18,21].

The steps of the WL-RLS-DCD are described in Table 1, where we denoted by $\mathbf{R}^{(1)}(n)$ the first column of the matrix $\mathbf{R}(n)$ and by ϵ a small positive constant employed to initialize the correlation matrix in a non-singular form. The updates in Steps 1 and 2 can be performed by relying on the time-shift property of the input vector $\tilde{\mathbf{x}}(n)$, which is also translated to $\mathbf{R}(n)$. The upper-left $(2L - 2) \times (2L - 2)$ sub-matrix of $\mathbf{R}(n)$ can be copied to the lower-right $(2L - 2) \times (2L - 2)$ sub-matrix, and only the first two columns and first two rows must be computed. If we take into consideration that the matrix is Hermitian, the first two rows can be obtained from the first two columns. Moreover, for every pair of consecutive columns, starting with an odd position (e.g., column 1, column 3, etc.), the sets of values comprising one of the columns can also be found in the other column in a conjugate form, at a distance of ± 1 row. Consequently, the only arithmetic operations necessary for updating $\mathbf{R}(n)$ can be employed for a single column (i.e., a set of $2L$ values). The described methodology shows how the arithmetic complexity corresponding to Steps 1 and 2 from Table 1, which is apparently $\mathcal{O}(4L^2)$, can be kept to value proportional to $2L$ (i.e., the filter's length). Further possible optimizations, such as the option of storing only a part of the matrix, are up to the developers performing the hardware implementations and deciding on the associated compromises. The latter depend considerably on the value of L (i.e., the associated memory requirements versus the necessary extra logic elements required to exploit the time-shift properties).

Steps 3 and 4 of the WL-RLS-DCD represent the standard filtering operation and the trivial computation of the error signal. In Step 5, the *residual component* $\mathbf{p}_0(n)$ is updated for the current time index by applying the forgetting factor to the so-called *residual vector* $\mathbf{r}(n - 1)$. The residual component is then employed in Step 6 as part of an auxiliary system of equations, which can be solved by exploiting the properties of the correlation matrix $\mathbf{R}(n)$ with the DCD iterations. Step 6 produces two sets of results. Firstly, the *solution vector*, denoted by $\tilde{\Delta \mathbf{h}}(n)$, is used in Step 7 by adding it to the filter coefficients $\tilde{\mathbf{h}}(n - 1)$ determined at the previous time index, thus generating the updated estimate $\tilde{\mathbf{h}}(n)$ of the unknown complex valued system. Secondly, the residual vector $\mathbf{r}(n)$ is computed to reflect the changes performed in the filter update process through $\Delta \mathbf{h}(n)$. As can be noticed in Steps 5 and 6, the vector $\mathbf{r}(n)$ will be relevant at the next time index as part of the next DCD operation.

Table 2 shows a description of the DCD iterations with a *leading element*. It can be noted that $\tilde{\Delta \mathbf{h}}(n)$ is initialized with $2L \times 1$ zeros values, and at the end of the entire DCD procedure, it represents the solution vector to a system which also comprises the matrix $\mathbf{R}(n)$ and the $2L \times 1$ residual component $\mathbf{p}_0(n)$.

The DCD algorithm works under the presumption that $\mathbf{R}(n)$ is comprised of complex values on all positions except the main diagonal, where all values are real positive numbers. Moreover, on the main diagonal, all values are statistically considerably higher than all other real or imaginary values in the rest of the matrix (in terms of absolute values). By exploiting this property, Step 1 of the algorithm chooses the position with the greatest absolute value in $\mathbf{r}_{aux}$ (all real and imaginary parts) and performs a comparison with its counterpart (i.e., same position) on the main diagonal of $\mathbf{R}(n)$ in order to determine if an update should be made in the solution vector. This approach is considered a *greedy* behaviour because it searches for the most probable position where an update could be triggered for $\Delta \mathbf{h}$.

The parameters of the DCD iterations are the maximum expected amplitude H , the step size α , the number of bits M_b used for the numerical representation of the real and imaginary values comprising $\Delta \mathbf{h}$, and the maximum number of allowed updates N_u . The value

of H is usually chosen as a power of two (e.g., $H = 1$), and it is used to initialize the value of the step size. Considering how the latter is updated in Step 2 of Table 2, this choice ensures that the multiplication operations performed with α are equivalent to bit shifts.

In Step 1 of the DCD iterations, a search is performed amongst all real and imaginary parts comprising the vector $\mathbf{r}_{aux}$ in order to determine the maximum absolute value and its corresponding position. The target value r_{temp} is employed in Step 4 to decide if an update is triggered, by comparing it to the (real and positive) value $R_{p,p}$ situated on the corresponding position of the main diagonal of $\mathbf{R}(n)$, which is scaled using the step size. If an update is decided, then the same position in the solution vector (i.e., $\Delta\tilde{\mathbf{h}}_p$) is updated using α and the sign of r_{temp} . The operation is equivalent to setting a bit in the binary representation of one of the real and imaginary values comprising the solution vector $\Delta\tilde{\mathbf{h}}$. Moreover, the coefficient update is always followed by another change in the vector $\mathbf{r}_{aux}$ on the same position p (Step 6). It is important to note that $\mathbf{r}_{aux}$ is initialized at the beginning of the DCD with the values comprising the residual component $\mathbf{p}_0(n)$, and after the changes performed by the algorithm, it will represent the residual vector $\mathbf{r}(n)$.

If the condition in Step 4 does not trigger an update, then a jump is made to Step 2, and the value of the step size is halved, making α point to the next lesser significant bit possible. Such an operation can be performed until the algorithm reaches the least significant bit it can process (i.e., until $m = M_b$). Consequently, the algorithm can perform a total number of updates upper limited by N_u (see the *for* loop), or it can exit when it would need to perform updates in the solution vector equivalent to binary values smaller than the least significant bit used to represent the real and imaginary values comprising $\Delta\tilde{\mathbf{h}}(n)$ (Step 3). Due to the nature of the auxiliary system of equations, it is expected the WL-RLS-DCD will generate satisfactory performance with a small value of N_u (less than 10, or comparable to it, for larger systems, with thousands of coefficients). In addition, it is natural to assume at this point that the value M_b also denotes the number of bits used to represent the real and imaginary parts of the coefficients comprising the adaptive filter $\tilde{\mathbf{h}}(n)$.

The overall complexity of the employed DCD algorithm (at any time index n) takes into account that bit shifts replace all potential multiplications, and only additions are employed. Moreover, a maximum number of N_u update operations can be performed on the solution vector. The final number of modified $\Delta\tilde{\mathbf{h}}(n)$ coefficients (which are all initialized with 0) is expected to be much smaller than the filter's length (i.e., $N_u \ll 2L$). In [5,21], it was demonstrated that the DCD iterations can generate performance similar to other classical RLS methods for values of N_u smaller than 10. Consequently, the upper limit for the amount of additions is noted in Table 2.

Information regarding the arithmetic complexity of the WL-RLS-DCD is also displayed in Table 1, in terms of real valued multiplications and additions. For the DCD stage of the adaptive algorithm, we evaluated the most complex scenarios when the maximum number of allowed updates N_u is employed. It can be noted that the amount of necessary multiplications is a small multiple of the complex valued adaptive filter's length. The same conclusion can be drawn for the number of additions, which are combined with bit shifts in order to solve the auxiliary system of the equation in Step 6.

Table 1. WL-RLS-DCD Algorithm.

Step	Actions	×	+
	Initialization :		
0	Set: $\tilde{\mathbf{h}}(0) = \mathbf{0}_{2L \times 1}$; $\mathbf{r}(0) = \mathbf{0}_{2L \times 1}$ $\mathbf{R}(0) = \epsilon \mathbf{I}_{2L}$, $\epsilon > 0$ For $n = 1, 2, \dots$, number of iterations :		
1	Update $\tilde{\mathbf{x}}(n)$ using (22)		
2	Update $\mathbf{R}(n)$ using time-shift $\mathbf{R}^{(1)}(n) = \lambda \mathbf{R}^{(1)}(n-1) + \mathbf{x}^*(n) \tilde{\mathbf{x}}(n)$	12L	8L
3	$\tilde{\mathbf{y}}(n) = \tilde{\mathbf{h}}^H(n-1) \tilde{\mathbf{x}}(n)$	8L	6L - 2
4	$e(n) = d(n) - \tilde{\mathbf{y}}(n)$		2
5	$\mathbf{p}_0(n) = \lambda \mathbf{r}(n-1) + e^*(n) \tilde{\mathbf{x}}(n)$	12L	6L
6	$\mathbf{R}(n) \Delta \mathbf{h}(n) = \mathbf{p}_0(n) \xrightarrow{\text{DCD}} \Delta \tilde{\mathbf{h}}(n), \mathbf{r}(n)$		$\leq (8L+1)N_u + M_b$
7	$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \Delta \tilde{\mathbf{h}}(n)$		$\leq N_u$

Table 2. Complex valued DCD iterations with a leading element at time index n .

Step	Actions	+
	Initialization :	
	Set: $\Delta \tilde{\mathbf{h}} = \mathbf{0}_{2L \times 1}$; $\mathbf{r}_{aux} = \mathbf{p}_0(n)$ $\alpha = H$, $m = 0$ For $k = 1, 2, \dots, N_u$:	
1	$(val_1; pos_1) = \max\{ \Re\{\mathbf{r}_{aux}\} \}$ $(val_2; pos_2) = \max\{ \Im\{\mathbf{r}_{aux}\} \}$ $(p; \eta) = (val_1 > val_2) ? (pos_1; 1) : (pos_2; j)$ Jump to Step 4	2L - 1 2L - 1 1
2	$\alpha = \alpha/2$; $m = m + 1$	
3	If $m > M_b \xrightarrow{\text{return}} \{\Delta \tilde{\mathbf{h}}; \mathbf{r}_{aux}\}$ as $\{\Delta \tilde{\mathbf{h}}(n); \mathbf{r}(n)\}$	
4	$r_{temp} = (\eta == 1) ? \Re\{r_{aux,p}\} : \Im\{r_{aux,p}\}$ If $ r_{temp} \leq \frac{\alpha}{2} R_{p,p} \xrightarrow{\text{jump}} \text{Step 2}$	1
5	$\Delta h_p = \Delta \tilde{h}_p + \text{sign}\{r_{temp}\} \eta \alpha$	1
6	$\mathbf{r}_{aux} = \mathbf{r}_{aux} - \text{sign}\{r_{temp}\} \eta \alpha \mathbf{R}_p$ Return $\{\Delta \tilde{\mathbf{h}}; \mathbf{r}_{aux}\}$ as $\{\Delta \tilde{\mathbf{h}}(n); \mathbf{r}(n)\}$	4L
$\leq (8L+1)N_u + M_b$		

4. WL-DR-RLS-DCD

In the recent literature, for the process of determining $\tilde{\mathbf{h}}(n)$ from $\tilde{\mathbf{h}}(n-1)$ when employing RLS adaptive algorithms, the DR approach was introduced in order to improve their corresponding tracking speeds [23,24]. Correspondingly, in order to update the filter's coefficients multiple times at the same time index n (i.e., for the same input data), Steps 3–7 from Table 1 will be run in a loop for a number of N_{it} iterations (including the DCD method). We call the proposed algorithm the WL-DR-RLS-DCD, which is equivalent to the WL-RLS-DCD for $N_{it} = 1$.

For each of the N_{it} DR iterations, the algorithm will determine a new solution vector $\Delta \tilde{\mathbf{h}}_q(n)$, $q = 0, \dots, N_{it} - 1$ based on the same input data [i.e., $\tilde{\mathbf{x}}(n)$ and $\mathbf{R}(n)$] and add its contribution to the filter estimate $\tilde{\mathbf{h}}(n-1)$ accumulated with all previous DR iterations from time index n . The complex reference $d(n)$ will remain the same for all the N_{it} steps, as no updated value will be made available from outside the adaptive system. However, the error signal and the residual contribution will be updated accordingly and will go through N_{it} stages. We will distinguish them using the notations $e_q(n)$ and $\mathbf{p}_{0,q}(n)$, with $q = 0, \dots, N_{it} - 1$.

At the beginning of the DR process (i.e., for $q = 0$), the form of the complex output signal remains the same as (18), corresponding to the estimate $\tilde{\mathbf{h}}(n-1)$. For the next iterations ($q = 1, \dots, N_{it} - 1$), the output is adjusted according to the new updates performed by the DCD, which leads to the analytical expression:

$$\tilde{y}_q(n) = \begin{cases} \tilde{\mathbf{h}}^H(n-1)\tilde{\mathbf{x}}(n), & q = 0, \\ \left[\tilde{\mathbf{h}}(n-1) + \sum_{k=0}^{q-1} \Delta \tilde{\mathbf{h}}_k(n) \right]^H \mathbf{x}(n), & q = 1, \dots, N_{it} - 1. \end{cases} \quad (29)$$

We denote the first complex output sample for time index n as

$$\tilde{y}_0(n) = \tilde{\mathbf{h}}^H(n-1)\tilde{\mathbf{x}}(n), \quad (30)$$

and we can recursively express the output for iterations $q > 0$ as

$$\tilde{y}_q(n) = \tilde{y}_0(n) + \sum_{k=0}^{q-1} \Delta \tilde{\mathbf{h}}_k^H(n)\tilde{\mathbf{x}}(n) = \tilde{y}_{q-1}(n) + \Delta \tilde{\mathbf{h}}_{q-1}^H(n)\tilde{\mathbf{x}}(n). \quad (31)$$

Consequently, by using (21) and (31), the form of the complex valued error can also be recursively written as

$$e_q(n) = \begin{cases} d(n) - \tilde{\mathbf{h}}^H(n-1)\tilde{\mathbf{x}}(n) \triangleq e_0(n), & q = 0 \\ e_{q-1}(n) + \Delta \tilde{\mathbf{h}}_{q-1}^H(n)\tilde{\mathbf{x}}(n), & q = 1, \dots, N_{it} - 1. \end{cases} \quad (32)$$

The expression in (32) is relevant from an implementation point of view because when dealing with values of L specific to acoustic environments, most of the arithmetic costs for updating the complex error correspond to the computation of $e_0(n)$ (the workload associated with the classical WL-RLS-DCD method). For the iterations corresponding to $q = 1, \dots, N_{it}$, we must take into account in (32) that a maximum amount of N_u values are non-zero at each iteration in the corresponding solution vector $\Delta \tilde{\mathbf{h}}_{q-1}(n)$. Therefore, all the rest of $N_{it} - 1$ iterations require no more than $(N_{it} - 1)N_u$ complex multiplications, and the same number of complex additions, in order to accumulate them to the error signal.

For the residual elements associated with the DCD method, at each iteration of the DR process we consider that a *residual component* $\mathbf{p}_{0,q}(n)$ is used as input for each DCD run, and an associated *residual vector* $\mathbf{r}_q(n)$ is generated [besides $\Delta \tilde{\mathbf{h}}(n)$]. The DR approach creates a cycle, which comprises, for each iteration, updates performed for the error signal $e_q(n)$, then for the residual component $\mathbf{p}_{0,q}(n)$, in order to prepare for the DCD run. The latter generates a solution vector $\Delta \tilde{\mathbf{h}}_q(n)$ and a residual vector $\mathbf{r}_q(n)$, which will be used for updating the filter coefficients, and also in the next DR iteration, in order to repeat the same steps.

In a manner similar to $e_q(n)$, the update of $\mathbf{p}_{0,q}(n)$ must be split into two possible types of operations. Firstly, for $q = 0$ (i.e., the first DR iteration at time index n), the update will employ the forgetting factor in order to perform the transition from $n - 1$. Secondly, for $q = 1, \dots, N_{it} - 1$, $\mathbf{p}_{0,q}(n)$ is changed in order to reflect the error updates from (32) and the latest residual vector generated by the DCD algorithm. Consequently, we can write:

$$\mathbf{p}_{0,q}(n) = \begin{cases} \lambda \mathbf{r}_{N_{it}-1}(n-1) + e_0^*(n)\tilde{\mathbf{x}}(n), & q = 0 \\ \mathbf{r}_{q-1}(n) + e_0^*(n)\tilde{\mathbf{x}}(n), & q = 1, \dots, N_{it} - 1. \end{cases} \quad (33)$$

where $\mathbf{r}_{N_{it}-1}(n-1)$ is the residual vector determined by the last run of the DCD at time index $n - 1$.

Furthermore, we can also express the update process for the filter estimate as

$$\tilde{\mathbf{h}}_q(n) = \begin{cases} \tilde{\mathbf{h}}_{N_{it}-1}(n-1) + \Delta\tilde{\mathbf{h}}_0(n), & q = 0 \\ \tilde{\mathbf{h}}_{q-1}(n) + \Delta\tilde{\mathbf{h}}_q(n), & q = 1, \dots, N_{it} - 1, \end{cases} \quad (34)$$

where $\tilde{\mathbf{h}}_{N_{it}-1}(n-1)$ represents the last set of filter coefficients computed at the previous iteration of the adaptive algorithm. It can be noted that for a number of iterations $N_{it} = 1$, the updates described in (32)–(34) are each performed only for the first branch, which makes the WL-DR-RLS-DCD behave exactly as the WL-RLS-DCD algorithm.

The proposed adaptive algorithm is summarized in Table 3. It also comprises the corresponding information related to arithmetic workloads (in terms of real valued multiplications and additions). Just as in the case of the WL-RLS-DCD, we considered that counters with small values can be implemented as delay lines (i.e., bit shifts performed for binary arrays). The extra arithmetic effort determined by the DR approach (with respect to Table 1) is directly influenced by the number of iterations N_{it} , which is expected to be a number lower than 10. As was already discussed in [23,24], the expected gain in terms of tracking speed becomes limited when N_{it} increases beyond a certain value. Consequently, the overall complexity of the WL-DR-RLS-DCD remains proportional to the length of the complex valued filter multiplied by a small integer.

Table 3. WL-DR-RLS-DCD algorithm.

Step	Actions	×	+
0	Initialization : Set: $\tilde{\mathbf{h}}(0) = \mathbf{0}_{2L \times 1}$; $\mathbf{r}(0) = \mathbf{0}_{2L \times 1}$ $\mathbf{R}(0) = \epsilon \mathbf{I}_{2L}$, $\epsilon > 0$, $q = 0$ For $n = 1, 2, \dots$, number of iterations :		
1	Update $\tilde{\mathbf{x}}(n)$ using (22)		
2	Update $\mathbf{R}(n)$ using time-shift $\mathbf{R}^{(1)}(n) = \lambda \mathbf{R}^{(1)}(n-1) + x^*(n)\tilde{\mathbf{x}}(n)$	12L	8L
3	$q = q + 1$		
4	Determine $e_q(n)$ using (32)	$\leq 8L + 4(N_{it} - 1)N_u$	$\leq 6L - 2 + 2N_{it} + 4N_u(N_{it} - 1)$
5	Determine $\mathbf{p}_{0,q}(n)$ using (33)	$4L(2N_{it} + 1)$	$4LN_{it}$
6	$\mathbf{R}(n)\Delta\mathbf{h}_q(n) = \mathbf{p}_{0,q}(n) \xrightarrow{\text{DCD}} \Delta\tilde{\mathbf{h}}_q(n), \mathbf{r}_q(n)$		$\leq M_b + N_{it}[(8L + 1)N_u]$
7	Determine $\tilde{\mathbf{h}}_q(n)$ using (34) If $q < N_{it} \xrightarrow{\text{jump}} \text{Step 3}$		$\leq N_{it}N_u$

5. Simulation Results

Simulations were performed in the context of SAEC using the WL framework. The input signal was filtered through two distinct real acoustic impulse responses in order to simulate the effect of the *left* and *right* channels and generate the samples $x_{Lc}(n)$ and $x_{Rc}(n)$. The input signal source was chosen as a Gaussian noise filtered through an auto-regressive (AR) system with a single pole and a speech sequence [25].

In order to overcome the problem of the correlation between the sequences $x_{Lc}(n)$ and $x_{Rc}(n)$, we employed the nonlinear pre-distortion approach presented in [4,18]. The method implies the addition of a specific level of distortion to the input signals, $x_{Lc}(n)$ and $x_{Rc}(n)$, respectively, in order to obtain a unique solution. The namely *half-wave rectifier* generates two new signals using the expressions [4,9,18]

$$\mathbf{x}'_{Lc}(n) = \mathbf{x}_{Lc}(n) + \alpha_r \frac{\mathbf{x}_{Lc}(n) + |\mathbf{x}_{Lc}(n)|}{2} \quad (35)$$

$$\mathbf{x}'_{R_c}(n) = \mathbf{x}_{R_c}(n) + \alpha_r \frac{\mathbf{x}_{R_c}(n) - |\mathbf{x}_{R_c}(n)|}{2}, \quad (36)$$

where the parameter $\alpha_r < 1$ controls the amount of nonlinearity. In practice, the stereo effect is considered not to be drastically affected for the range $0 < \alpha_r \leq 0.5$.

The performance of the proposed WL-DR-RLS-DCD algorithm has been validated in the context of the identification of the interleaved impulse response $\tilde{\mathbf{h}}(n)$, with respect to the results provided by the version with $N_{it} = 1$ DR iterations (i.e., the WL-RLS-DCD). The performance has been measured using the normalized misalignment, having the following expression in the WL context:

$$\text{Mis}(n) = 20 \log_{10} \frac{\|\tilde{\mathbf{h}}_t - \tilde{\mathbf{h}}(n)\|_2}{\|\tilde{\mathbf{h}}_t\|_2} \quad [\text{dB}], \quad (37)$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm [4].

The forgetting factor was chosen with the form $\lambda = 1 - 1/(KL)$, and the parameters specific to the DCD iterations were set to $H = 1$, $N_u = 4$, and $M_b = 16$. For the simulations using an AR(1) sequence as input, the pole was set to 0.95. The four unknown echo paths are real measured acoustic impulse responses, which have been decimated in order to have the desired lengths (indicated in the caption of each figure). Finally, the SNR was experimentally set to 25 dB for all scenarios [4,5].

In Figure 2, we studied the performance of the proposed algorithm for different values of the DR parameter N_{it} and for two values of the pre-distortion parameter: (A) $\alpha_r = 0$ (no pre-distortion is applied) and (B) $\alpha_r = 0.33$ (pre-distortion used for the complex input signal). It can be noted that for every value of N_{it} the performances corresponding to (B) are considerably superior to their counterparts from case (A). The mitigation of correlation between the real valued input signals associated with the two channels leads to better performance for the adaptive filter. For every value of $N_{it} > 1$, the gain in terms of normalized misalignment is at least 5 dB at steady state. Moreover, as the value of the parameter N_{it} increases, so does the initial convergence speed of the WL-DR-RLS-DCD. The corresponding compromise is made with the associated extra arithmetic workload and a reduction in accuracy of the adaptive algorithm when steady state is reached. Finally, it can also be observed that with higher values of N_{it} , the progress in terms of convergence speed becomes smaller, such that the corresponding gain tends to become capped.

In Figure 3, we compare the performance of the WL-RLS-DCD for different values of α_r with the WL-DR-RLS-DCD having $N_{it} > 1$. We increased the pre-distortion parameter above the recommended limit in order to analyze the possibility of compensating a high number of DR iterations and save extra costs in terms of arithmetic operations required by the WL-DR-RLS-DCD with higher values for N_{it} . It can be noted that even the convergence speeds associated with values $\alpha_r > 0.5$ (i.e., when the stereo effect is altered) cannot match the misalignment performance of using $\alpha_r = 0.33$ and $N_{it} > 1$. All normalized misalignment curves corresponding to $N_{it} = 1$ require more than 10^6 iterations to reach steady state (i.e., around -25 dB), which exceeds the boundaries of the presented simulation interval.

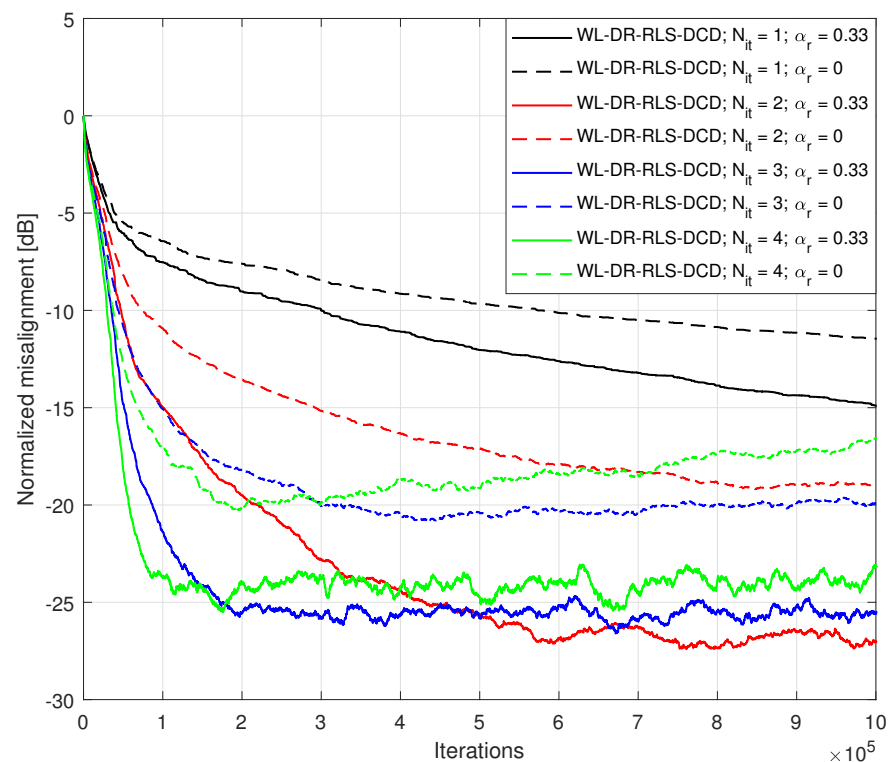


Figure 2. Misalignment of the WL-DR-RLS-DCD algorithm for different values of N_{it} . The input signal is an AR(1) sequence with the pole 0.95, and it is pre-distorted ($\alpha_r = 0.33$) or not pre-distorted at all ($\alpha_r = 0$). The length of the four unknown impulse responses is $L = 128$ and $\lambda = 1 - 1/(64L)$.

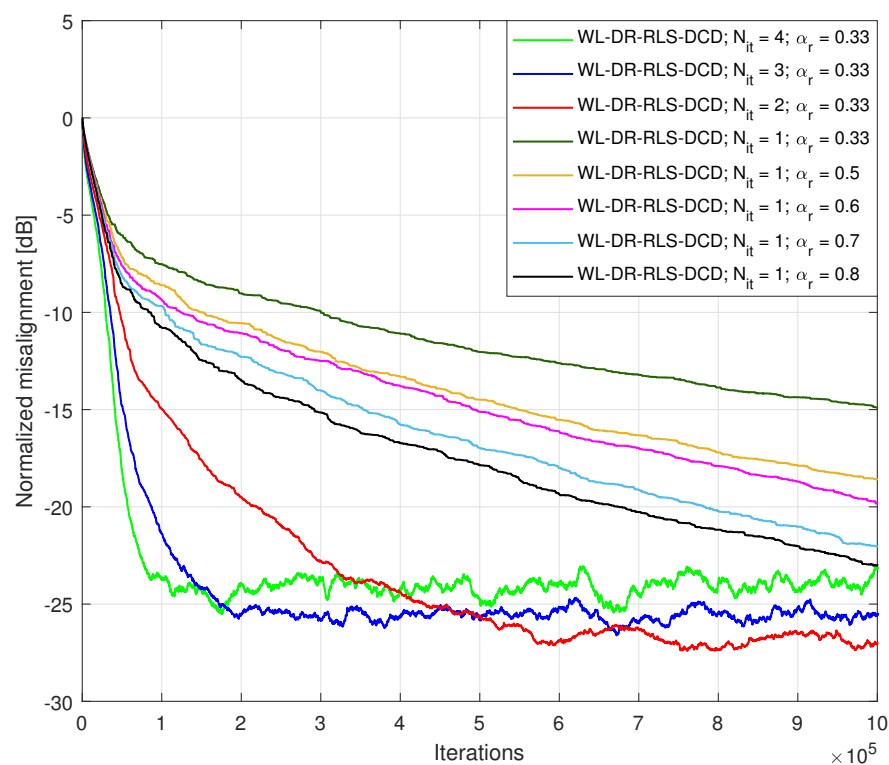


Figure 3. Misalignment of the WL-DR-RLS-DCD algorithm for different values of N_{it} and different values of the pre-distortion parameter α_r . The input signal is an AR(1) sequence with the pole 0.95, and the length of the four unknown impulse responses is $L = 128$ and $\lambda = 1 - 1/(64L)$.

The last experiment is presented in Figure 4. The input signal is a speech sequence. The unknown impulse responses are longer than the ones employed in the previous scenarios and are changed after half of the simulation is completed by shifting them 25 positions to the right (tracking scenario). We compared the performances of the WL-DR-RLS-DCD for the same value of α_r and different number N_{it} of DR iterations. As a reference, we also added the corresponding results for the lowest and highest value of N_{it} when the input signal is not pre-distorted. It can be noted that as the number of DR iterations increases, the tracking capabilities of the WL-DR-RLS-DCD improve, and the corresponding steady-state accuracy is diminished. Moreover, the versions of WL-DR-RLS-DCD which work without a pre-distorted input signal cannot match the performance of the algorithm employed with pre-processed input data. The curves corresponding to $\alpha_r = 0$ do not have normalized misalignment values lower than -10 dB, while the least accurate WL-DR-RLS-DCD version having $\alpha_r = 0.33$ generates performance levels with more than 5 dB better values. Similar to the previous cases, we can note the compromise between tracking speeds on one side, respectively performance at steady state and arithmetic complexity, on the other side. In addition, values of the parameter N_{it} situated above a certain threshold might generate limited progress with respect to the tracking capabilities, which would not justify some of the corresponding additional arithmetic costs. For example, the curve corresponding to $N_{it} = 2$, $\alpha_r = 0.33$ has similar tracking speed when compared to $N_{it} = 3$, $\alpha_r = 0.33$ and more than 3 dB better accuracy at steady state.

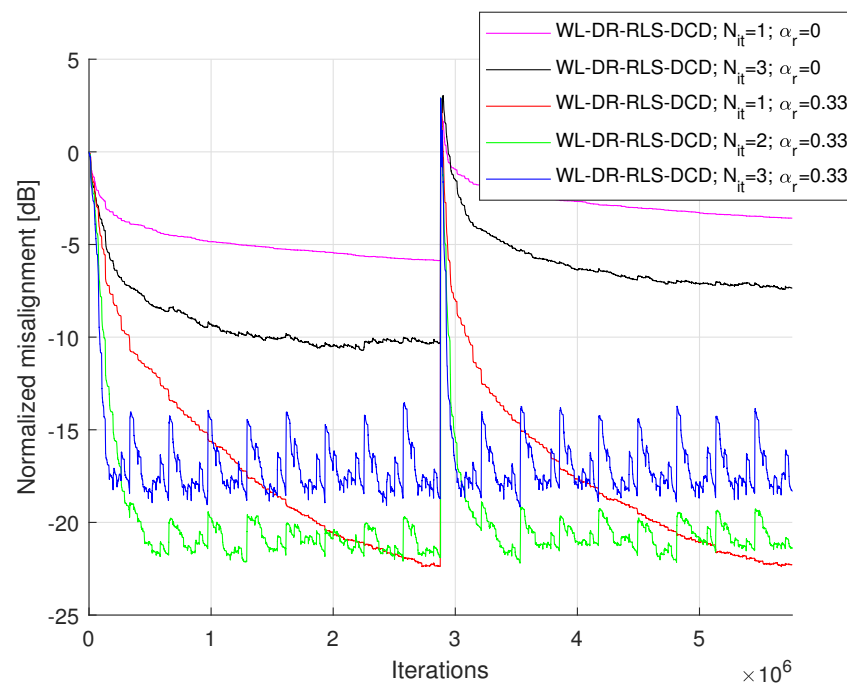


Figure 4. Misalignment of the WL-DR-RLS-DCD algorithm for different values of N_{it} and α_r . The input signal is a speech sequence, and the length of the four unknown impulse responses is $L = 256$ and $\lambda = 1 - 1/(64L)$. The four echo paths change in the middle of the simulation.

6. Discussion

The proposed WL-DR-RLS-DCD algorithm has several advantages compared to previous work. First, as compared to the DR-RLS method presented in [23], the proposed solution has two specific features: (i) it can be applied in the framework of SAEC thanks to the WL model, as shown in Section 2, and (ii) it achieves a lower computational complexity due to the DCD iterations, as presented in Sections 3 and 4. Second, as compared with the WL-RLS-DCD from [5], the proposed version leads to better tracking capabilities due to the DR approach, as also supported by the simulation results presented in Section 5.

As mentioned in Sections 4 and 5, the WL-DR-RLS-DCD algorithm also comes with some compromises. When increasing the number of DR iterations (i.e., the value of N_{it}), the algorithm runs the DCD iterations for a number of N_{it} times. The arithmetic complexity needed for solving the associated auxiliary systems of equations becomes proportional to the parameter N_{it} (besides the length of the complex valued adaptive filter and N_u). However, in practice, the values of N_{it} and N_u , are small, and the overall costs associated with the WL-DR-RLS-DCD, in terms of necessary chip areas, remain attractive for practical implementations on hardware devices.

Another part of the compromise corresponding to the DR approach is the loss of some accuracy at steady state for the WL-DR-RLS-DCD with respect to the WL-RLS-DCD. In order to mitigate this downside, we can employ several adjustments to the WL-DR-RLS-DCD, which add minimal extra arithmetic costs. Based on past experience with implementation of variable parameters for adaptive algorithms, a new version of the WL-DR-RLS-DCD with a variable number of DR iterations for SAEC scenarios can be developed. The new algorithm will adjust the value of N_{it} with respect to the convergence state of the algorithm by gradually increasing it when approaching steady state and by decreasing it when tracking situations occur.

Furthermore, we will explore the option of mitigating the DR accuracy reduction at steady state by employing a variable forgetting factor approach, which was studied by our team in [26,27]. The goal is to implement a system with variable memory features in order to counteract the negative aspects of the compromise between convergence speed and accuracy.

A third option is to apply decomposition techniques based on the nearest Kronecker product (NKP), in conjunction with the DR version [28–30]. The DR-NKP approach applied on the WL-RLS-DCD would replace a single long adaptive filter with a combination of two shorter filters. Consequently, the advantage of such decomposition is twofold. Gains can be obtained in terms of both performance and complexity.

7. Conclusions

The paper proposed a low-complexity RLS algorithm for SAEC scenarios with enhanced tracking capabilities. The new WL-DR-RLS-DCD is derived from the already established WL-RLS-DCD method, and it employs an acceptable arithmetic workload in order to adapt faster to changes in the environment. The information presented in Tables 1 and 3, respectively, shows that the overall arithmetic complexity remains proportional to the length of the complex valued filter multiplied by a small value, making the WL-DR-RLS-DCD an attractive choice for hardware implementations. Furthermore, simulation results demonstrated that the DR approach can improve the tracking performance of the RLS-DCD-type algorithms when working with highly correlated input signals, such as AR sequences and speech.

The simulations also revealed that increasing the number of DR iterations (i.e., the value of N_{it}) leads to a reduction in accuracy at steady state. Consequently, as explained in Section 6, our future efforts will be concentrated on developing low-complexity versions of the RLS algorithm for SAEC scenarios based on the DCD method, with variable parameters, such as the number of DR iterations or the forgetting factor λ . Moreover, we will invest research effort into applying decomposition techniques of impulse responses by exploiting associated low-rank approximations.

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

Funding: This work was supported by two grants of the Ministry of Research, Innovation and Digitization, CNCS-UEFISCDI, projects PN-III-P1-1.1-TE-2021-0393 and PN-III-P4-PCE-2021-0438, within PNCDI III.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

- Haykin, S. *Adaptive Filter Theory*; Prentice-Hall Information and System Sciences Series; Prentice Hall: Upper Saddle River, NJ, USA, 2002.
- Sayed, A. *Adaptive Filters*; Wiley: New York, NY, USA, 2008.
- Diniz, P.S.R. *Adaptive Filtering: Algorithms and Practical Implementation*, 4th ed.; Springer: New York, NY, USA, 2013.
- Benesty, J.; Paleologu, C.; Gaensler, T.; Ciochina, S. *A Perspective on Stereophonic Acoustic Echo Cancellation*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 4. [\[CrossRef\]](#)
- Stanciu, C.; Benesty, J.; Paleologu, C.; Gänsler, T.; Ciochină, S. A widely linear model for stereophonic acoustic echo cancellation. *Signal Process.* **2013**, *93*, 511–516. [\[CrossRef\]](#)
- Hänsler, E.; Schmidt, G. *Acoustic Echo and Noise Control—A Practical Approach*; Wiley: Hoboken, NJ, USA, 2004.
- Enzner, G.; Buchner, H.; Favrot, A.; Kuech, F. Acoustic echo control. In *Academic Press Library in Signal Processing*; Elsevier: Amsterdam, The Netherlands, 2014; Volume 4, pp. 807–877. [\[CrossRef\]](#)
- Sondhi, M.; Morgan, D.; Hall, J. Stereophonic acoustic echo cancellation—an overview of the fundamental problem. *IEEE Signal Process. Lett.* **1995**, *2*, 148–151. [\[CrossRef\]](#)
- Benesty, J.; Morgan, D.; Sondhi, M. A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. *IEEE Trans. Speech Audio Process.* **1998**, *6*, 156–165. [\[CrossRef\]](#)
- Malik, S.; Wung, J.; Atkins, J.; Naik, D. Double-Talk Robust Multichannel Acoustic Echo Cancellation Using Least-Squares MIMO Adaptive Filtering: Transversal, Array, and Lattice Forms. *IEEE Trans. Signal Process.* **2020**, *68*, 4887–4902. [\[CrossRef\]](#)
- Hong, J. Stereophonic Acoustic Echo Suppression for Speech Interfaces for Intelligent TV Applications. *IEEE Trans. Consum. Electron.* **2018**, *64*, 153–161. [\[CrossRef\]](#)
- Cho, B.J.; Park, H.M. Stereo Acoustic Echo Cancellation Based on Maximum Likelihood Estimation with Inter-Channel-Correlated Echo Compensation. *IEEE Trans. Signal Process.* **2020**, *68*, 5188–5203. [\[CrossRef\]](#)
- Zhang, H.; Wang, D. Neural Cascade Architecture for Multi-Channel Acoustic Echo Suppression. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2326–2336. [\[CrossRef\]](#)
- Romoli, L.; Cecchi, S.; Peretti, P.; Piazza, F. A Mixed Decorrelation Approach for Stereo Acoustic Echo Cancellation Based on the Estimation of the Fundamental Frequency. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 690–698. [\[CrossRef\]](#)
- Schneider, M.; Huemmer, C.; Kellermann, W. Wave-domain loudspeaker signal decorrelation for system identification in multichannel audio reproduction scenarios. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, USA, 26–31 May 2013; pp. 605–609. [\[CrossRef\]](#)
- Romoli, L.; Cecchi, S.; Piazza, F. A novel decorrelation approach for multichannel system identification. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 6652–6656. [\[CrossRef\]](#)
- Schneider, M.; Kellermann, W. Multichannel Acoustic Echo Cancellation in the Wave Domain With Increased Robustness to Nonuniqueness. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 518–529. [\[CrossRef\]](#)
- Stanciu, C.; Benesty, J.; Paleologu, C.; Gaensler, T.; Ciochină, S. A novel perspective on stereophonic acoustic echo cancellation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 25–28. [\[CrossRef\]](#)
- Stanciu, C.; Paleologu, C.; Benesty, J.; Gaensler, T.; Ciochina, S. An efficient RLS algorithm for stereophonic acoustic echo cancellation with the widely linear model. In Proceedings of the 41st International Congress and Exposition on Noise Control Engineering, INTERNOISE, New York, NY, USA, 19–22 August 2012; Volume 9, pp. 7756–7767.
- Breining, C.; Dreiscitel, P.; Hänsler, E.; Mader, A.; Nitsch, B.; Puder, H.; Schertler, T.; Schmidt, G.; Tilp, J. Acoustic echo control. an application of very-high-order adaptive filters. *IEEE Signal Process. Mag.* **1999**, *16*, 42–69. [\[CrossRef\]](#)
- Zakharov, Y.V.; White, G.P.; Liu, J. Low-Complexity RLS Algorithms Using Dichotomous Coordinate Descent Iterations. *IEEE Trans. Signal Process.* **2008**, *56*, 3150–3161. [\[CrossRef\]](#)
- Zakharov, Y.V.; Nascimento, V.H. DCD-RLS Adaptive Filters with Penalties for Sparse Identification. *IEEE Trans. Signal Process.* **2013**, *61*, 3198–3213. [\[CrossRef\]](#)
- Paleologu, C.; Benesty, J.; Ciochină, S. Data-Reuse Recursive Least-Squares Algorithms. *IEEE Signal Process. Lett.* **2022**, *29*, 752–756. [\[CrossRef\]](#)
- Ficiu, I.D.; Stanciu, C.L.; Elisei-Iliescu, C.; Anghel, C.; Udrea, R.M. Low-Complexity Implementation of a Data-Reuse RLS Algorithm. In Proceedings of the 2022 45th International Conference on Telecommunications and Signal Processing (TSP), Virtual, 13–15 July 2022; pp. 289–293. [\[CrossRef\]](#)
- The Open Speech Repository. Available online: <https://www.itu.int/rec/T-REC-G.168/en> (accessed on 5 November 2022).

26. Paleologu, C.; Benesty, J.; Ciochina, S. A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification. *IEEE Signal Process. Lett.* **2008**, *15*, 597–600. [[CrossRef](#)]
27. Stanciu, C.; Paleologu, C.; Benesty, J.; Ciochina, S.; Albu, F. Variable-forgetting factor RLS for stereophonic acoustic echo cancellation with widely linear model. In Proceedings of the 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 1960–1964.
28. Paleologu, C.; Benesty, J.; Ciochină, S. Linear System Identification Based on a Kronecker Product Decomposition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1793–1808. [[CrossRef](#)]
29. Elisei-Iliescu, C.; Paleologu, C.; Benesty, J.; Stanciu, C.; Anghel, C.; Ciochină, S. Recursive Least-Squares Algorithms for the Identification of Low-Rank Systems. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 903–918. [[CrossRef](#)]
30. Dogariu, L.M.; Benesty, J.; Paleologu, C.; Ciochină, S. Identification of Room Acoustic Impulse Responses via Kronecker Product Decompositions. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 2828–2841. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.