Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 58(72), Fascicola 1, 2013

On a Robust Dual-Path DCD-RLS Algorithm for Stereophonic Acoustic Echo Cancellation

Cristian Stanciu[1], Constantin Paleologu[1], Jacob Benesty[2], and Silviu Ciochină[1]

Abstract – The standard model for stereophonic acoustic echo cancellation (SAEC) consists of a two-input/two-output system with real random variables, which can be modeled using four adaptive filters. Recently, we used the widely linear (WL) model in the context of SAEC. In this case, we deal with a single-input/single-output system with complex random variables, which involves only one adaptive filter. In this paper, the functionality of the WL-SAEC scheme is shown using the recursive least-squares (RLS)-dichotomous coordinate descent (DCD) algorithm. Moreover, a dual path filtering methodology is used to make the algorithm more robust.

Keywords: stereophonic acoustic echo cancellation (SAEC), widely linear (WL) model, recursive least-squares (RLS) algorithm, dichotomous coordinate descent (DCD), transfers logic (TL).

I. INTRODUCTION

Stereo transmissions make use of independent audio channels to enhance the sound directionality [1], using a typical configuration of two loudspeakers and two microphones. Nonetheless, a compromise is made as a result of the increased complexity of the stereophonic acoustic echo cancellation (SAEC) scheme. The system has two-inputs and two-outputs (i.e., the loudspeakers and microphones signals [2]) with four acoustic echo paths (i.e., the impulse responses from the loudspeakers to the microphones) to be estimated using four adaptive filters [2], [3]. Moreover, the two input signals (i.e., the loudspeaker signals) are linearly related [2], [3] and the system of equations solved by the adaptive algorithm can be singular, producing more than one possible solution [2]. In order to avoid the potential instability of the system, a preprocessing method must be employed to reduce the coherence between the two signals, conserving as much as possible the signal quality and the stereo effect.

Recently, we recast the SAEC problem using the widely linear (WL) model [2], [3]. The two-input/two-output system with real random variables was restructured as a single-input/single-output scheme with complex random variables (CRVs) [2]. Also, the four real-valued impulse responses are linked into one complex-valued impulse response. In addition, we used the recursive least-squares (RLS) algorithm [2], [4], [5] in the WL context [3]. This algorithm is a good choice for SAEC scenarios thanks to its fast convergence rate.

In order to decrease the prohibitive arithmetic complexity of the RLS algorithm, the dichotomous coordinate descent (DCD) method [6], [7] can be used to perform the necessary matrix inversion using only additions. Furthermore, the algorithm can be made more robust in double-talk scenarios [2] by using a two-path filtering methodology governed by transfer logic (TL) [8]. Some preliminary work can be found in [9]. In this paper, we investigate two versions of the DCD method and their key parameters. Simulations were performed to validate the proposed solutions.

II. PROBLEM FORMULATION

Let us denote by $x_{\mathrm{L}}(n)$ ("left") and $x_{\mathrm{R}}(n)$ ("right") the input (or loudspeaker) signals and by $\mathbf{h}_{\mathrm{t,LL}}$, $\mathbf{h}_{\mathrm{t,LR}}$, $\mathbf{h}_{\mathrm{t,RL}}$, $\mathbf{h}_{\mathrm{t,RR}}$ the L-dimensional vectors of the loudspeaker-to-microphone "true" acoustic impulse responses [2], [3]. Thus, we define the two echo signals:

$$y_{\mathrm{L}}(n) = \mathbf{h}_{\mathrm{t,LL}}^T \mathbf{x}_{\mathrm{L}}(n) + \mathbf{h}_{\mathrm{t,RL}}^T \mathbf{x}_{\mathrm{R}}(n), \quad (1)$$

$$y_{\mathrm{R}}(n) = \mathbf{h}_{\mathrm{t,RR}}^T \mathbf{x}_{\mathrm{L}}(n) + \mathbf{h}_{\mathrm{t,LR}}^T \mathbf{x}_{\mathrm{R}}(n), \quad (2)$$

where ^{T} signifies transpose and the most recent L input samples form the vectors:

$$\mathbf{x}_{\mathrm{L}}(n) = [x_{\mathrm{L}}(n) \;\; x_{\mathrm{L}}(n-1) \; \ldots \; x_{\mathrm{L}}(n-L+1)]^T, \quad (3)$$

$$\mathbf{x}_{\mathrm{R}}(n) = [x_{\mathrm{R}}(n) \;\; x_{\mathrm{R}}(n-1) \; \ldots \; x_{\mathrm{R}}(n-L+1)]^T. \quad (4)$$

[1] Department of Telecommunications, University Politehnica of Bucharest, 1-3 Iuliu Maniu Blvd., Bucharest, 061071, Romania
e-mail {cristian, pale, silviu}@comm.pub.ro
[2] INRS-EMT, University of Quebec, Montreal, Quebec, H5A 1K6, Canada
e-mail benesty@emt.inrs.ca

The WL model combines the real random variables into CRVs. First, the input complex signal is

$$x(n) = x_{\mathrm{L}}(n) + jx_{\mathrm{R}}(n), \tag{5}$$

where $j = \sqrt{-1}$. Next, the complex input vector is defined using (3) and (4):

$$\mathbf{x}(n) = \mathbf{x}_{\mathrm{L}}(n) + j\mathbf{x}_{\mathrm{R}}(n). \tag{6}$$

Additionally, the complex echo signal [2], [3] is written as

$$\begin{aligned} y(n) &= y_{\mathrm{L}}(n) + jy_{\mathrm{R}}(n) \\ &= \mathbf{h}_{\mathrm{t}}^{H}\mathbf{x}(n) + \mathbf{h}_{\mathrm{t}}^{'H}\mathbf{x}^{*}(n), \end{aligned} \tag{7}$$

where the superscripts ^{H} and * denote transpose-conjugate and conjugate; the complex vectors

$$\mathbf{h}_{\mathrm{t}} = \mathbf{h}_{\mathrm{t},1} + j\mathbf{h}_{\mathrm{t},2}, \tag{8}$$

$$\mathbf{h}_{\mathrm{t}}^{'} = \mathbf{h}_{\mathrm{t},1}^{'} + j\mathbf{h}_{\mathrm{t},2}^{'}, \tag{9}$$

are formed with linear combinations of the true impulse responses:

$$\mathbf{h}_{\mathrm{t},1} = \frac{\mathbf{h}_{\mathrm{t,LL}} + \mathbf{h}_{\mathrm{t,RR}}}{2},\ \mathbf{h}_{\mathrm{t},2} = \frac{\mathbf{h}_{\mathrm{t,RL}} + \mathbf{h}_{\mathrm{t,LR}}}{2}, \tag{10}$$

$$\mathbf{h}_{\mathrm{t},1}^{'} = \frac{\mathbf{h}_{\mathrm{t,LL}} - \mathbf{h}_{\mathrm{t,RR}}}{2},\ \mathbf{h}_{\mathrm{t},2}^{'} = -\frac{\mathbf{h}_{\mathrm{t,RL}} + \mathbf{h}_{\mathrm{t,LR}}}{2}. \tag{11}$$

Furthermore, the complex echo can be expressed in a simpler manner as

$$y(n) = \tilde{\mathbf{h}}_{\mathrm{t}}^{H}\tilde{\mathbf{x}}(n), \tag{12}$$

where

$$\tilde{\mathbf{h}}_{\mathrm{t}} = \left[\mathbf{h}_{\mathrm{t}}^{T}, \mathbf{h}_{\mathrm{t}}^{'T}\right]^{T},\ \tilde{\mathbf{x}}(n) = [\mathbf{x}^{T}(n), \mathbf{x}^{H}(n)]^{T}. \tag{13}$$

As a result, the complex microphone signal [2], [3] (or the complex output) is defined as

$$d(n) = y(n) + v(n) = \tilde{\mathbf{h}}_{\mathrm{t}}^{H}\tilde{\mathbf{x}}(n) + v(n), \tag{14}$$

where $v(n) = v_{\mathrm{L}}(n) + jv_{\mathrm{R}}(n)$ is the complex near-end signal [2], which can be noise, speech, or a combination of both.

The classic SAEC scheme has two-input and two-output signals (i.e., a total number of four real random variables), which are now compacted into a one-input/one-output system with CRVs [1], [2]. Moreover, the four L dimensional vectors representing the true real-valued echo paths (that have to be estimated) are combined into a single vector (i.e., $\tilde{\mathbf{h}}_{\mathrm{t}}$) with $2L$ complex values.

Regardless of the adaptive algorithm used to estimate the complex echo path, the linear system of equations to be solved might not have a unique solution; the reason is the linear relation between the two input signals $x_{\mathrm{L}}(n)$ and $x_{\mathrm{R}}(n)$ [2], [3]. A remedy is to use a pre-distortion method, as the one proposed in [2] and [3], which modifies the phase of the complex input signal $x(n)$ and introduces a compromise between the nonlinearity amount and the stereo effect of the audio signal. More details are available in [2] and [3].

III. THE DCD-RLS ALGORITHM

In this paper, we use of the RLS algorithm to estimate the complex impulse response $\tilde{\mathbf{h}}_{\mathrm{t}}$. It has a fast convergence rate but also a high arithmetic complexity which is proportional to $4L^2$. For the RLS algorithm in the context of the WL linear, the $2L$-dimensional vectors $\tilde{\mathbf{h}}_{\mathrm{t}}$ and $\tilde{\mathbf{x}}(n)$ are the interleaved versions of the ones defined in (13) [2], [3].

We have the set of coefficients $\tilde{\mathbf{h}}(n)$ of the adaptive filter used to estimate the true complex echo path $\tilde{\mathbf{h}}_{\mathrm{t}}$. At any time instance n, the complex echo estimate is defined using

$$\hat{y}(n) = \tilde{\mathbf{h}}^{H}(n-1)\tilde{\mathbf{x}}(n). \tag{15}$$

Thus, the *a-priori* error signal [2] is defined:

$$e(n) = d(n) - \hat{y}(n). \tag{16}$$

Considering the defined complex-valued signals, the least-squares error criterion (also called the cost function) [2], [4] is expressed as

$$J\left[\tilde{\mathbf{h}}(n)\right] = \sum_{i=1}^{n}\lambda^{n-i}\left|d(i) - \tilde{\mathbf{h}}^{H}(n)\tilde{\mathbf{x}}(i)\right|^{2}, \tag{17}$$

where λ is the *forgetting factor* $(0 \ll \lambda < 1)$, which directly impacts the memory in the statistical estimates [2], [4]; the special case of $\lambda = 1$ corresponds to infinite memory.

It can be shown that the minimization of the cost function $J\left[\tilde{\mathbf{h}}(n)\right]$, with respect to $\tilde{\mathbf{h}}(n)$, leads to the well-known normal equations [2]:

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n)\tilde{\mathbf{h}}(n) = \mathbf{p}_{\tilde{\mathbf{x}}d}(n), \tag{18}$$

where

$$\mathbf{R}_{\tilde{\mathbf{x}}}(n) = \sum_{i=1}^{n} \lambda^{n-i} \tilde{\mathbf{x}}(i)\tilde{\mathbf{x}}^{H}(i), \tag{19}$$

$$\mathbf{p}_{\tilde{\mathbf{x}}d}(n) = \sum_{i=1}^{n} \lambda^{n-i} \tilde{\mathbf{x}}(i) d^{*}(i), \tag{20}$$

have the dimensions $2L \times 2L$ and $2L \times 1$, respectively.

The RLS algorithm performs the matrix inversion necessary to determine the complex-valued filter coefficients $\tilde{\mathbf{h}}(n)$ with an arithmetic complexity proportional to $4L^2$. This order is too high for hardware implementations. The fast RLS (FRLS) algorithm [2] was developed to reduce the number of computations to an order of $2L$. Nevertheless, it is not a viable option because its behavior is extremely unstable when operating with non-stationary signals.

Recently, a new adaptive algorithm was involved [5], which combines the RLS with the dichotomous coordinate descent (DCD) method. The resulting algorithm is called DCD-RLS [5], [7], [9]; its working principle is presented in Table 1. The solution for (18) is computed iteratively using the auxiliary system of equations [7] in step 4, which generates the solution vector $\Delta\tilde{\mathbf{h}}(n)$, also called the increment of the filter weights [7]. Besides, the DCD uses the residual vector $\mathbf{r}(n)$ [6], [7], which is updated in step 3 using the forgetting factor. The matrix $\mathbf{R}_{\tilde{\mathbf{x}}}(n)$ is initialized using a positive constant δ multiplying $\mathbf{I}_{2L}$, the $2L \times 2L$ identity matrix.

In step 4, the complex-valued DCD [5], [6] performs the matrix inversion, using four important parameters. The first one is denoted by M_{b} and it represents the number of bits used for the fixed point representation of the values in the solution vector $\Delta\tilde{\mathbf{h}}(n)$. The second important parameter is H, where $[-H;H]$ is the expected amplitude range for the real and imaginary values in $\Delta\tilde{\mathbf{h}}(n)$. By choosing the value of H to be a power of two, the step size α (the third critical parameter of the DCD) is initialized with H and it is halved at some iterations. Thus, the step size has the connotation of bits used in the fixed point representation of $\Delta\tilde{\mathbf{h}}(n)$ and any operations made with it (multiplications or divisions) can be replaced by bit-shifts. As a result, the DCD computes the solution vector using only additions, with a maximum number of so-called "successful iterations," denoted by N_{u}; these iterations, which update the solution vector $\Delta\tilde{\mathbf{h}}(n)$ in four potential coordinate directions (positive real, positive imaginary, negative real, or imaginary), include the main load of arithmetic computations. The necessary value for N_{u} is expected to be small as the increment of the filter weights is computed, instead of the direct solution to the normal equations in (18) [6].

Table 1

Initialization	$\tilde{\mathbf{h}}(0) = \mathbf{0}, \mathbf{r}(0) = \mathbf{0}, \mathbf{R}_{\tilde{\mathbf{x}}}(0) = \delta\mathbf{I}_{2L}$
	For $n = 1, 2, \ldots$
Step 1	$\mathbf{R}_{\tilde{\mathbf{x}}}(n) = \lambda\mathbf{R}_{\tilde{\mathbf{x}}}(n-1) + \tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^{H}(n)$
Step 2	$e(n) = d(n) - \tilde{\mathbf{h}}^{H}(n-1)\tilde{\mathbf{x}}(n)$
Step 3	$\mathbf{r}(n) = \lambda\mathbf{r}(n-1) + \tilde{\mathbf{x}}(n)e^{*}(n)$
Step 4	$\mathbf{R}_{\tilde{\mathbf{x}}}(n)\Delta\tilde{\mathbf{h}}(n) = \mathbf{r}(n) \Rightarrow \Delta\tilde{\mathbf{h}}(n), \mathbf{r}(n)$ (solved with DCD iterations)
Step 5	$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) + \Delta\tilde{\mathbf{h}}(n)$

The DCD method has two versions. The *cyclic* complex valued DCD [6] updates $\Delta\tilde{\mathbf{h}}(n)$ from the most significant bits to least important ones by checking each position for a possible update. The arithmetic complexity is upper-limited by the value:

$$N_1 = 2L(2N_{\mathrm{u}} + M_{\mathrm{b}} - 1) + N_{\mathrm{u}}, \tag{21}$$

where N_1 represents the number of real valued additions, $N_{\mathrm{u}} \ll L$, and $M_{\mathrm{b}} \ll L$. The cyclic DCD does not perform the updates in an efficient manner, hence a second version, called DCD with a *leading element*, was proposed [6]. It searches for the likely positions in $\Delta\tilde{\mathbf{h}}(n)$ to be changed by choosing the locations of the maximum absolute values (real and imaginary parts) of the residual vector $\mathbf{r}(n)$. The accuracy of the leading DCD can be slightly lower as some updates can be "overlooked" [6], especially in scenarios with speech signal input. The computational effort has a new highest limit:

$$N_2 = (4L + 1)N_{\mathrm{u}} + M_{\mathrm{b}}. \tag{22}$$

The contribution of the term LM_{b} is eliminated in N_2 and the arithmetic complexity decreases (i.e., $N_2 < N_1$).

Both versions of the DCD method stop performing modifications to $\Delta\tilde{\mathbf{h}}(n)$ when the maximum number of allowed updates (N_{u}) is achieved or all the bits of the solution vector have been processed. More details about the DCD-RLS algorithm can be found in [5], [6], [7].

Furthermore, the arithmetic complexity can be reduced in step 1 by an order of $2L$, if we consider that $\tilde{\mathbf{x}}(n)$ has the time shift property and the matrix $\mathbf{R}_{\tilde{\mathbf{x}}}(n)$ is Hermitian [5], [7], [9].

IV. THE DUAL PATH DCD-RLS ALGORITHM

The DCD-RLS algorithm [9] is modified to include a two-path filtering methodology; the new algorithm is entitled the dual-path DCD-RLS. We use a pair of filters to mitigate the effects of high disturbance situations, such as the double-talk scenarios. The DCD-RLS filter coefficients represent now the so-called background (BK) filter [8], [9]. For every time instant n they are the product of the continuously updating adaptive filter. A secondary set of coefficients, named the foreground (FG) filter [8], [9], are used for the actual echo cancellation. The separation is intended to segregate any BK filter perturbation from the FG. Thus, a transfer logic (TL) [8], [9] is used, which acts as a set of four conditions that must be met for a number of Q consecutive iterations in order to make an update to the FG filter, i.e., to copy the BK coefficients to the FG. The transfer may not take place for each iteration.

In order to better understand the conditions in the TL, we denote the cross-correlation between any two complex signals $a(n)$ and $b(n)$ as

$$r_{ab}(n) = \mathrm{E}\left[a(n)b^{*}(n)\right], \tag{23}$$

where $\mathrm{E}[\bullet]$ denotes the mathematical expectation. Correspondingly, the variance of $a(n)$ is expressed as:

$$\sigma_{a}^{2}(n) = \mathrm{E}\left[|a(n)|^{2}\right]. \tag{24}$$

Moreover, two new complex signals are defined for the FG filter, the complex echo estimate [8], [9]:

$$\hat{y}_{\mathrm{f}}(n) = \tilde{\mathbf{h}}_{\mathrm{f}}^{H}(n-1)\tilde{\mathbf{x}}(n), \tag{25}$$

and the FG error signal [8], [9]:

$$e(n) = d(n) - \hat{y}_{\mathrm{f}}(n). \tag{26}$$

Therefore, the first condition of the TL states that enough input energy must exist, i.e.,

$$\sigma_{x}^{2}(n) > T_{1}, \tag{27}$$

where T_1 is a chosen threshold.

Conditions two and three compare the performance of the BK and the FG filters. Thus, the deviation measures for the two filters are computed as [8], [9]

$$m_{\mathrm{f}}(n) = \left|\frac{r_{\hat{y}_{\mathrm{f}}e_{\mathrm{f}}}(n)}{r_{\hat{y}_{\mathrm{f}}d}(n)}\right|, \tag{28}$$

$$m_{\mathrm{b},D}(n) = \left|\frac{r_{\hat{y}_{D}e_{D}}(n)}{r_{\hat{y}_{D}d}(n)}\right|, \tag{29}$$

where D is a delay constant, $e_D(n) = d(n) - \hat{y}_D(n)$, $\hat{y}_D(n) = \tilde{\mathbf{h}}^{H}(n+D)\tilde{\mathbf{x}}(n)$; $\tilde{\mathbf{h}}(n+D)$ are the BK filter coefficients at time instant $n+D$. Additional information about the purpose of the delay constant can be found in [8] and references therein. Condition two states that the deviation measure for the FG filter has a greater value, in order to permit a transfer from the BK [8], [9]:

$$m_{\mathrm{f}}(n) > m_{\mathrm{b},D}(n). \tag{30}$$

Likewise, in condition three, the variances of the two error signals are compared [8], [9]:

$$\sigma_{e_{\mathrm{f}}}^{2}(n) > \sigma_{e_{D}}^{2}(n). \tag{31}$$

Finally, the forth condition of the TL plays the role of a double-talk detector (i.e., it determines when the near-end signal is speech) [8], [9]:

$$1 - \frac{\left|r_{de_{D}}\right|}{\sigma_{d}^{2}(n)} > T_{2}, \tag{32}$$

where T_2 is a second chosen threshold.

The overall purpose of the TL is to prevent any transfer to the FG when the BK filter is affected by high disturbance. In such situations, the SAEC must be performed with the last "good" estimate of the complex echo path. The use of a pair of filters governed by the TL adds to the arithmetic complexity, but it makes the DCD-RLS more robust in double-talk scenarios.

V. SIMULATIONS

Simulations were performed to demonstrate the validity of the theoretical models. The input signal is speech with a sampling frequency of 8 kHz and preprocessed with the method mentioned in the end of Section II. The four real-valued echo paths used in the experiments have the length L = 512. Besides, for the near-end signal we generated a Gaussian noise with the stereo-echo-to-noise ratio (SENR) of 30 dB. The forgetting factor for the RLS algorithm was set to $\lambda = 1 - 1/(14L)$. For the fixed point representation of the values in the solution vector we used $H = 2$ and $M_b = 16$.

The parameters of the TL were set to $Q = 3$, $T_1 = 10^{-8}$, and $T_2 = 0.99$. The estimates for (23) and (24) were computed using an exponential window:

$$\hat{r}_{ab}(n) = q\hat{r}_{ab}(n-1) + (1-q)a(n)b^*(n), \tag{33}$$

with $q = 0.95$. Finally, the performance of the adaptive algorithms was measured using the normalized misalignment expressed in dB:

$$\mathrm{Mis}(n) = 20\log_{10} \left\| \tilde{\mathbf{h}}_t - \tilde{\mathbf{h}}(n) \right\|_2 \Big/ \left\| \tilde{\mathbf{h}}_t \right\|_2, \tag{34}$$

where $\|\bullet\|_2$ is the l_2 norm.

In Fig. 1, we compared the behavior of the classical RLS algorithm with the two versions of DCD-RLS (the BK filters), i.e., the cyclic DCD-RLS and the leading DCD-RLS; we used a maximum number of allowed updates $N_u = 8$. A tracking situation occurs after 30 seconds; the four real-valued echo paths are suddenly changed. The misalignment shows that both proposed methods outperform the classical RLS. The comparison between the two DCD-RLS versions reveals that the leading DCD is a good compromise between arithmetic complexity and performance. Fig. 2 illustrates the behavior of the FG filters for the cyclic and the leading versions of the DCD-RLS in the tracking scenario. It is obvious that the TL allows an efficient transfer from the BK to the FG, even when the echo paths change. Frequent FG filter updates are necessary when crucial (required) variations take place in the BK filter.

In Fig. 3, we showed (for the same tracking scenario) the performance of the FG leading DCD-RLS algorithm with different values of the parameter

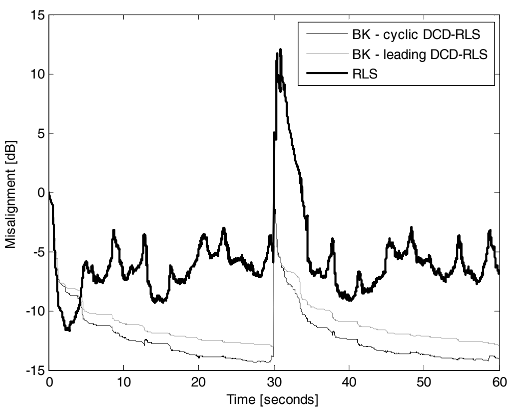

Fig.1. Tracking scenario, comparison between the DCD-RLS versions (BK filters) and the classical RLS, N_u=8.

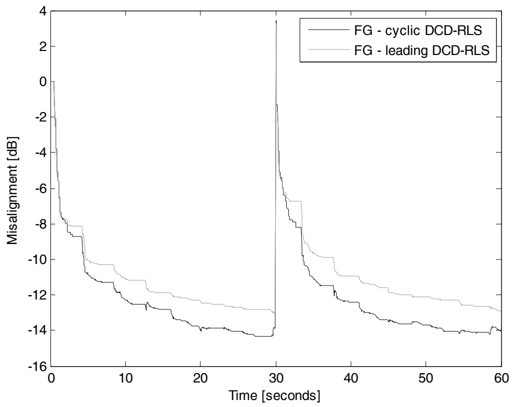

Fig.2. Tracking scenario, comparison between the DCD-RLS versions (FG filters), N_u=8.

N_u. Because the arithmetic complexity is proportional to N_u, a compromise is necessary between the computational effort and performance. It can be noticed that $N_u = 8$ is a good choice for the matrix inversion task.

Furthermore, we performed simulations for a double-talk setting. The secondary speech sequence is present in the microphone signal $d(n)$ during the time interval [25; 28.75] seconds. Fig. 4 shows the misalignment of the BK and FG filters for the leading DCD-RLS algorithm. It can be observed that the TL does not allow any transfer from the BK to the FG when the BK filter is affected by the strong perturbation. The FG filter performs the SAEC using a "good" set of coefficients copied from the BK just before the double-talk situation occurs.

The dual filtering method provides good protection against the double-talk circumstances. However, an important drawback is the fact that the BK filter recovers slowly from the perturbation, making the FG vulnerable to any echo path changes. In Fig. 5, we repeated the simulation for the double-talk setting. We added for the BK filter a reset condition which acts when the left-hand side of (32) is negative; the reset can occur only once for every second (8000 iterations). The BK filter recovers faster

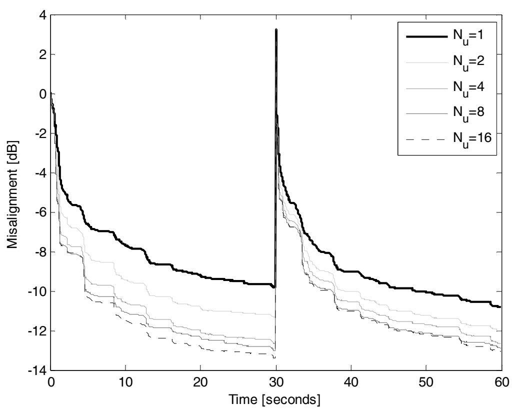

Fig.3. Tracking scenario, leading DCD-RLS, different values for N_u, FG filters.

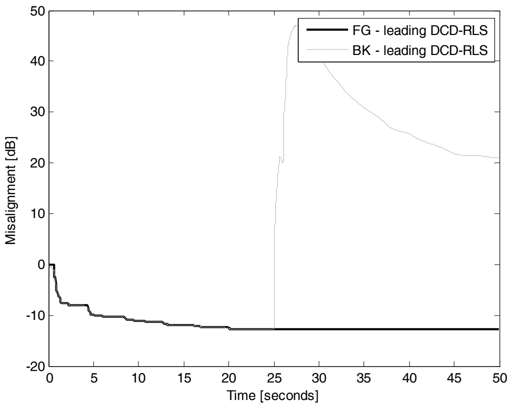

Fig.4. Double-talk scenario, leading DCD-RLS, N_u=8, FG and BK filters.

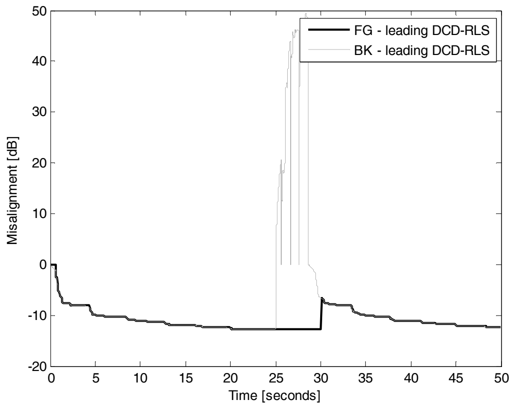

Fig.5. Double-talk scenario, leading DCD-RLS, N_u=8, FG and BK filters (BK with reset condition).

after the perturbation ends, making the adaptive filter more prepared for any necessary re-convergence process.

VI. CONCLUSIONS

In this paper, an alternative for the classical RLS algorithm was presented, i.e., the DCD-RLS, in the context of the WL model for SAEC. Two versions of the DCD method were analyzed, i.e., the cyclic complex valued DCD and the DCD with a leading element, respectively. Both solutions have lower arithmetic complexity, performing the necessary matrix inversion using only additions and bit-shifts. Moreover, a dual-path filtering approach was introduced to enhance the robustness of the DCD-RLS algorithm in high disturbance scenarios like the double-talk.

Simulations showed that the dual-path DCD-RLS algorithm is a viable solution to the SAEC problem, with good performance and attractive computational effort for hardware implementations.

REFERENCES

[1] 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.

[2] J. Benesty, C. Paleologu, T. Gänsler, and S. Ciochină, *A Perspective on Stereophonic Acoustic Echo Cancellation*. Berlin, Germany: Springer-Verlag, 2011.

[3] C. Stanciu, J. Benesty, C. Paleologu, T. Gänsler, and S. Ciochină, “A novel perspective on stereophonic acoustic echo cancellation,” in *Proc. IEEE ICASSP*, 2012, pp.25–28.

[4] S. Haykin, *Adaptive Filter Theory*. Fourth Edition, Upper Saddle River, NJ: Prentice-Hall, 2002.

[5] C. Stanciu, J. Benesty, C. Paleologu, T. Gänsler, and S. Ciochină, “A widely linear model for stereophonic acoustic echo cancellation,” *Signal Processing*, vol. 93, pp. 511–516, Feb. 2013

[6] J. Liu, Y. V. Zakharov, and B. Weaver, “Architecture and FPGA design of dichotomous coordinate descent algorithms,” *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 56, pp. 2425–2438, Nov. 2009.

[7] Y. V. Zakharov, G. P. White, and J. Liu, “Low-complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Trans. Signal Process.*, vol. 56, pp. 3150–3161, July 2008.

[8] C. Schüldt, F. Lindström, and I. Claesson, “Robust low-complexity transfer logic for two-path echo cancellation,” in *Proc. IEEE ICASSP*, 2012, pp. 173–176.

[9] C. Stanciu, C. Paleologu, J. Benesty, T. Gänsler, and S. Ciochină, “An efficient RLS algorithm for stereophonic acoustic echo cancellation with the widely linear model,” in *Proc. Internoise*, 2012.