ON CASCADED RECURSIVE LEAST SQUARES: AN INTERPRETABLE AND CHEAPER ALTERNATIVE TO STANDARD RLS

Basil Liekens *Arnout Roebben* *Toon van Waterschoot* *Marc Moonen*

KU Leuven, Department of Electrical Engineering (ESAT)
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
Kasteelpark Arenberg 10, 3001 Leuven, Belgium

ABSTRACT

The recursive least squares (RLS) algorithm is a well-known adaptive filtering algorithm. In this paper the specific case is considered where the RLS input vector is partitioned into two parts, e.g. in practice a first part containing loudspeaker signals and a second part containing microphone signals. A filter defined over the entire RLS input vector is called an integrated filter. It has been shown that an optimal integrated filter for a given batch of data is equivalent to a cascade of optimal filters defined over the two individual parts, producing the same output. However, when adaptively updating the individual filters in the cascade using standard RLS updates, the equivalence to the integrated filter is lost. In this paper, it is shown that it is possible to retain equivalence by introducing a modification in the RLS updates in the cascade. In addition, it is shown that updating the cascade is actually cheaper than running the standard RLS algorithm. Finally, it is shown that the proposed algorithm is a generalization of the recursive modified Gram-Schmidt algorithm.

***Index Terms*—** Adaptive filtering, Recursive least squares, Recursive modified Gram-Schmidt algorithm

1. INTRODUCTION

In adaptive filtering the task can be to predict the output of a system or to identify a system model [1]. In this paper, the standard recursive least squares (RLS) algorithm [1, Ch. 10] will be considered for this task, in particular the special case where the RLS input vector is partitioned into two parts, e.g. a first part containing loudspeaker signals and a second part containing microphone signals. A filter defined over the entire RLS input vector is referred to as an integrated filter. It has been shown in [2, 3] that an optimal integrated filter for a given batch of data is equivalent to a cascade of optimal filters defined over the two individual parts of the input vector in the sense that they produce the same output. This cascade then leads to physically interpretable models.

However, when updating the integrated filter and the individual filters in the corresponding cascade using standard RLS updates the equivalence between the integrated filter and cascade is lost. To the best of our knowledge an RLS algorithm that considers this cascade and its equivalence to the standard RLS algorithm has not yet been proposed. In [4] an algorithm has been proposed that also acts on a partitioning of the RLS input vector, but to retain equivalence with the integrated filter an orthogonality assumption is required. In [5, 6] the filter itself is split into multiple parts using a Kronecker product, but this requires the integrated filter to be low-rank. In this paper it is shown that by slightly modifying the RLS updates of the filters in the cascade the equivalence can be retained, regardless of the properties of the input signal or integrated filter. It will also turn out that updating the cascade is actually cheaper than updating the integrated filter using the standard RLS algorithm. Finally, it is shown that the proposed algorithm is a generalization of the recursive modified Gram-Schmidt (RMGS) [7] algorithm.

To illustrate the interpretability, two running examples will be used throughout this paper. In [8] a joint echo cancellation and noise reduction problem is considered. The two parts in the input vector partitioning would then contain the loudspeaker signals and noise references, respectively. In [9] a joint dereverberation and noise reduction problem is considered. The two parts would then contain the delayed microphone signals and noise references, respectively.

The rest of the paper is structured as follows. Section 2 introduces the notation used in this paper and recaps the standard RLS problem. Section 3 shows how to update the cascade while retaining equivalence with the integrated filter. Section 4 discusses the computational complexity of both approaches. Section 5 discusses the equivalence to the RMGS algorithm. Section 6 gives some simulation results with code made available in [10]. Section 7 concludes the paper.

2. STANDARD RECURSIVE LEAST SQUARES

The standard RLS algorithm solves a, possibly weighted, least squares problem recursively [1, Ch. 10]. Given an input signal vector $\mathbf{u}[n] \in \mathbb{R}^L$, with n the time index, and a desired signal vector $\mathbf{d}[n] \in \mathbb{R}^M$, the filter $\mathbf{W}[n] \in \mathbb{R}^{L\times M}$ is the solution to the following least squares problem

$$\min_{\mathbf{W}} \sum_{i=0}^{n} \frac{\lambda^{n-i}}{w[i]} \|\mathbf{d}[i] - \mathbf{W}^{\mathrm{T}}\mathbf{u}[i]\|_2^2 + \lambda^{n+1}\delta\|\mathbf{W}\|_F^2, \quad (1)$$

with $\lambda \in (0, 1]$ a forgetting factor, $\delta > 0$ a regularization and $w[n] \in \mathbb{R}$ a, possibly time-varying, weighting factor. Given the dimension of the input and output vectors, this is also referred to as an RLS problem of size $L \times M$. Different choices of $w[n]$ lead to, e.g., exponentially weighted RLS [1, Ch. 10.3] or variance-normalized RLS [11–13]. It should also be noted that no specific assumptions about the temporal structure of the input vector are made in this paper, in contrast to, e.g. [14–16]. Introducing the following correlation matrices

$$\mathbf{R}_{\mathbf{uu}}[n] = \sum_{i=0}^{n} \frac{\lambda^{n-i}}{w[i]} \mathbf{u}[i]\mathbf{u}^{\mathrm{T}}[i] + \lambda^{n+1}\delta\mathbf{I}, \tag{2}$$

$$\mathbf{R}_{\mathbf{ud}}[n] = \sum_{i=0}^{n} \frac{\lambda^{n-i}}{w[i]} \mathbf{u}[i]\mathbf{d}^{\mathrm{T}}[i], \tag{3}$$

then leads to [13]

$$\mathbf{W}[n] = \mathbf{R}_{\mathbf{uu}}^{-1}[n]\mathbf{R}_{\mathbf{ud}}[n]. \tag{4}$$

Suppose now that at time $n-1$ the filter $\mathbf{W}[n-1]$ and the inverse correlation matrix $\mathbf{R}_{\mathbf{uu}}^{-1}[n-1]$ are known. When new data comes in at time n it is possible to update the filter recursively instead of computing (4), leading to the well-known standard RLS algorithm in Table 1 [11, 13]. The algorithm is initialized with $\mathbf{R}_{\mathbf{uu}}[-1] = \delta I$, $\mathbf{R}_{\mathbf{ud}}[-1] = \mathbf{0}$ and, correspondingly, $\mathbf{W}[-1] = \mathbf{0}$ [13].

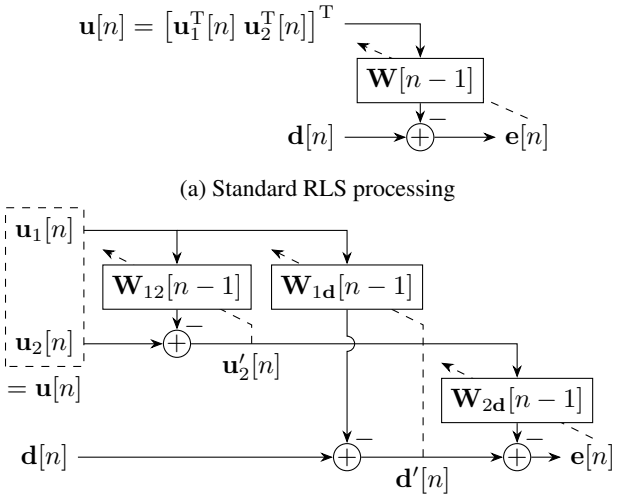
	computation	real MACs
1.	$\mathbf{e}[n] = \mathbf{d}[n] - \mathbf{W}^{\mathrm{T}}[n-1]\mathbf{u}[n]$	LM
2.	$\widetilde{\mathbf{R}}_{\mathbf{uu}}^{-1}[n-1] = \mathbf{R}_{\mathbf{uu}}^{-1}[n-1]/\lambda$	$L^2/2 + L/2$
3.	$\mathbf{v}[n] = \widetilde{\mathbf{R}}_{\mathbf{uu}}^{-1}[n-1]\mathbf{u}[n]$	L^2
4.	$\zeta[n] = w[n] + \mathbf{u}^{\mathrm{T}}[n]\mathbf{v}[n]$	L
5.	$\mathbf{k}[n] = \mathbf{v}[n]/\zeta[n]$	L
6.	$\mathbf{R}_{\mathbf{uu}}^{-1}[n] = \widetilde{\mathbf{R}}_{\mathbf{uu}}^{-1}[n-1] - \mathbf{k}[n]\mathbf{v}^{\mathrm{T}}[n]$	$L^2/2 + L/2$
7.	$\mathbf{W}[n] = \mathbf{W}[n-1] + \mathbf{k}[n]\mathbf{e}^{\mathrm{T}}[n]$	LM

Table 1: Standard RLS algorithm with computational complexity in terms of real multiply-and-accumulates (MACs).

3. CASCADED RECURSIVE LEAST SQUARES

In this section it is shown that an RLS problem of size $L \times M$ can be split into a cascade of two smaller RLS problems, leading to a cascaded RLS algorithm that is equivalent to the standard RLS algorithm. Define $\mathbf{u}_1[n] \in \mathbb{R}^{L_1}$ and $\mathbf{u}_2[n] \in \mathbb{R}^{L_2}$ with $\mathbf{u}[n] = [\mathbf{u}_1^{\mathrm{T}}[n]\ \mathbf{u}_2^{\mathrm{T}}[n]]^{\mathrm{T}}$, i.e. $L_1 + L_2 = L$. It should be noted that no assumptions about the temporal structure of $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ or the relation between them is made.

Applying this to the running examples, in [8] $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ would be the loudspeaker signals and noise references, respectively. Similarly, in [9] $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ would be the delayed microphone signals and noise references, respectively. Section 3.1 recaps how an integrated filter obtained by the RLS problem discussed in the previous section can be decomposed into a cascade at time $n-1$ as discussed in, e.g., [3]. Section 3.2 then shows how to update the cascade.



(a) Standard RLS processing

(b) Cascaded RLS processing

Fig. 1: Block scheme representation of standard and cascaded RLS processing.

3.1. Equivalence to cascaded filters

Using the partitioning of $\mathbf{u}[n]$, (2) and (3) can correspondingly be written in a partitioned form as

$$\mathbf{R}_{\mathbf{uu}}[n] = \begin{bmatrix} \mathbf{R}_{11}[n] & \mathbf{R}_{12}[n] \\ \mathbf{R}_{21}[n] & \mathbf{R}_{22}[n] \end{bmatrix}, \ \mathbf{R}_{\mathbf{ud}}[n] = \begin{bmatrix} \mathbf{R}_{1\mathbf{d}}[n] \\ \mathbf{R}_{2\mathbf{d}}[n] \end{bmatrix}. \tag{5}$$

Defining the following two auxiliary correlation matrices

$$\mathbf{\Sigma}_{22}[n] = \mathbf{R}_{22}[n] - \mathbf{R}_{21}[n]\mathbf{R}_{11}^{-1}[n]\mathbf{R}_{12}[n], \tag{6}$$

$$\mathbf{\Xi}_{2\mathbf{d}}[n] = \mathbf{R}_{2\mathbf{d}}[n] - \mathbf{R}_{21}[n]\mathbf{R}_{11}^{-1}[n]\mathbf{R}_{1\mathbf{d}}[n], \tag{7}$$

and subsequently introducing the following filter definitions

$$\mathbf{W}_{1\mathbf{d}}[n] = \mathbf{R}_{11}^{-1}[n]\mathbf{R}_{1\mathbf{d}}[n], \tag{8}$$

$$\mathbf{W}_{12}[n] = \mathbf{R}_{11}^{-1}[n]\mathbf{R}_{12}[n], \tag{9}$$

$$\mathbf{W}_{2\mathbf{d}}[n] = \mathbf{\Sigma}_{22}^{-1}[n]\mathbf{\Xi}_{2\mathbf{d}}[n], \tag{10}$$

then allows to use the 2×2 block-inverse formulas [17, (2.2)] to rewrite the error signal in step 1 of Table 1 as [3]

$$\begin{aligned} \mathbf{e}[n] = &\underbrace{\mathbf{d}[n] - \mathbf{W}_{1\mathbf{d}}^{\mathrm{T}}[n-1]\mathbf{u}_1[n]}_{\triangleq \mathbf{d}'[n]} \\ &- \mathbf{W}_{2\mathbf{d}}^{\mathrm{T}}[n-1] \underbrace{\left(\mathbf{u}_2[n] - \mathbf{W}_{12}^{\mathrm{T}}[n-1]\mathbf{u}_1[n]\right)}_{\triangleq \mathbf{u}_2'[n]}. \end{aligned} \tag{11}$$

It should be noted that in order to be able to make use of these 2×2 block inverses it is required that $\mathbf{R}_{\mathbf{uu}}[n-1]$ is of full rank. The cascaded RLS processing is also depicted graphically in Fig. 1b, contrasted with the standard RLS processing in Fig. 1a. This approach of cascading a filter is similar to the approaches presented in [2, 3], yet it is reiterated that the decomposition is considered only at a given time step.

3.2. Updating of cascaded filters

In this section RLS updates for the filters (8)-(10) will be derived. It will be shown that a specific modification to the weighting is required when updating $\mathbf{W}_{2\mathbf{d}}[n-1]$ and that, as a consequence, using standard RLS updates loses equivalence between the integrated filter and the individual filters in the cascade. In order to retain this equivalence, the correlation matrices in the filter definitions (8)-(10) should be updated in accordance with their definitions in the partitionings found in (5), i.e.

$$\mathbf{R}_{11}[n] = \lambda \mathbf{R}_{11}[n-1] + w^{-1}[n]\mathbf{u}_1[n]\mathbf{u}_1^{\mathrm{T}}[n], \tag{12}$$

and likewise for the other correlation matrices. For $\mathbf{W}_{1\mathbf{d}}[n]$ and $\mathbf{W}_{12}[n]$ this leads to standard RLS updates with $\mathbf{u}_1[n]$ as input and $\mathbf{d}[n]$ and $\mathbf{u}_2[n]$ as desired output

$$\mathbf{k}_1[n] = \frac{\mathbf{R}_{\mathbf{uu}}^{-1}[n-1]\mathbf{u}_1[n]}{\lambda w[n] + \mathbf{u}_1^{\mathrm{T}}[n]\mathbf{R}_{11}^{-1}[n-1]\mathbf{u}_1[n]}, \tag{13}$$

$$\mathbf{R}_{11}^{-1}[n] = \frac{\mathbf{R}_{11}^{-1}[n-1]}{\lambda} - \mathbf{k}_1[n]\frac{\mathbf{u}_1^{\mathrm{T}}[n]\mathbf{R}_{11}^{-1}[n-1]}{\lambda}, \tag{14}$$

$$\mathbf{W}_{1\mathbf{d}}[n] = \mathbf{W}_{1\mathbf{d}}[n-1] + \mathbf{k}_1[n]\mathbf{d}'^{\mathrm{T}}[n], \tag{15}$$

$$\mathbf{W}_{12}[n] = \mathbf{W}_{12}[n-1] + \mathbf{k}_1[n]\mathbf{u}_2'^{\mathrm{T}}[n], \tag{16}$$

with $\mathbf{d}'[n]$ and $\mathbf{u}_2'[n]$ defined in (11). This implies that a new, smaller, RLS update of size $L_1 \times (M + L_2)$ can be used to update $\mathbf{W}_{1\mathbf{d}}[n-1]$ and $\mathbf{W}_{12}[n-1]$ jointly.

To determine the update of $\mathbf{W}_{2\mathbf{d}}[n-1]$, the updates of $\mathbf{\Sigma}_{22}[n-1]$ and $\mathbf{\Xi}_{2\mathbf{d}}[n-1]$ have to be constructed first. However, these correlation matrices are not directly available in the partitionings found in (5). Therefore, their updates should be constructed by substituting the updates of their constituent parts in definitions (6) and (7) using (12) and (14). Introducing the following auxiliary definition

$$w'[n] = w[n] + \lambda^{-1}\mathbf{u}_1^{\mathrm{T}}[n]\mathbf{R}_{11}^{-1}[n-1]\mathbf{u}_1[n], \tag{17}$$

this then leads to

$$\mathbf{\Sigma}_{22}[n] = \lambda \mathbf{\Sigma}_{22}[n-1] + \frac{\mathbf{u}_2'[n]\mathbf{u}_2'^{\mathrm{T}}[n]}{w'[n]}, \tag{18}$$

$$\mathbf{\Xi}_{2\mathbf{d}}[n] = \lambda \mathbf{\Xi}_{2\mathbf{d}}[n-1] + \frac{\mathbf{u}_2'[n]\mathbf{d}'^{\mathrm{T}}[n]}{w'[n]}. \tag{19}$$

The fact that the denominator $w[n]$ is changed into $w'[n]$ can be interpreted as a correction for the fact that $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ have changed while (6) and (7) are correlation matrices that assume the same filter was applied to the entire batch of data. Thus, using (18) and (19), the RLS update for $\mathbf{W}_{2\mathbf{d}}[n-1]$ is

$$\mathbf{k}_2'[n] = \frac{\mathbf{\Sigma}_{22}^{-1}[n-1]\mathbf{u}_2'[n]}{\lambda w'[n] + \mathbf{u}_2'^{\mathrm{T}}[n]\mathbf{\Sigma}_{22}^{-1}[n-1]\mathbf{u}_2'[n]}, \tag{20}$$

$$\mathbf{\Sigma}_{22}^{-1}[n] = \frac{\mathbf{\Sigma}_{22}^{-1}[n-1]}{\lambda} - \mathbf{k}_2'[n]\frac{\mathbf{u}_2'^{\mathrm{T}}[n]\mathbf{\Sigma}_{22}^{-1}[n-1]}{\lambda}, \tag{21}$$

$$\mathbf{W}_{2\mathbf{d}}[n] = \mathbf{W}_{2\mathbf{d}}[n-1] + \mathbf{k}_2'[n]\mathbf{e}^{\mathrm{T}}[n]. \tag{22}$$

This means that the second update in the cascade is also an RLS update, of size $L_2 \times N$, albeit with the modification of changing $w[n]$ into $w'[n]$. As a consequence, updating $\mathbf{W}_{2\mathbf{d}}[n-1]$ using RLS without accounting for this modification would lead to a loss of equivalence between the integrated filter and the cascade. It should be noted that the factor $\lambda w'[n]$ in the denominator of (20) is equal to the denominator of (13), meaning no additional computational cost is incurred to compute $w'[n]$ when that denominator is propagated from the first RLS update (13)-(16) to the second one.

Applying this to the examples given before, the following two scenarios are obtained. In [8], with $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ the loudspeaker signals and noise references, respectively, $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ are echo cancellation filters for the noise and signal references, respectively. $\mathbf{W}_{2\mathbf{d}}[n]$ is then a noise reduction filter operating on these echo-cancelled signals. Similarly, in [9], with $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ the delayed microphone signals and noise references, respectively, $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ are dereverberation filters for the noise and signal references. $\mathbf{W}_{2\mathbf{d}}[n]$ is then a noise reduction filter operating on these dereverberated signals.

In summary, this section has shown that when a filter is split into a cascade of filters as done in [2, 3], it is possible to retain equivalence with the corresponding integrated filter if both are updated using RLS and a modified weighting is used to update $\mathbf{W}_{2\mathbf{d}}[n]$. Finally, it should be noted that even though the derivation in this section used a standard RLS update, any RLS update can be used for the updates provided that the modified weighting in the second update of the cascade is accounted for.

4. COMPUTATIONAL COMPLEXITY

In this section the computational complexity of the updates of the integrated filter and cascaded filters is compared when both are using standard RLS. Fast RLS implementations such as the ones proposed in [14–16] are not considered as they require the input vector to obey specific temporal structures, while the exposition in this paper is generally applicable.

The analysis is based on the one from [13] with three important differences. First, it is assumed that the weighting factor $w[n]$ is known as it is identical for both the integrated and cascaded RLS updates. Second, only the a-priori error $\mathbf{e}[n]$ is considered as in some applications only the filters themselves are focused upon. Third, since the involved correlation matrices are symmetric, only a triangular part has to be computed and stored which leads to computational savings.

For a standard RLS update the individual steps alongside their computational cost is given in Table 1. The total cost of a standard RLS update of size $L \times M$, i.e. an update of the integrated filter, in terms of real multiply-and-accumulates (MACs) is therefore $2L^2 + (2M + 3)L$. As discussed in Section 3.2, the cascaded implementation effectively turns an RLS update of size $L \times M$ into one of size $L_1 \times (M + L_2)$ with $\mathbf{u}_1[n]$ as input and $\mathbf{d}'[n]$ and $\mathbf{u}_2'[n]$ as outputs, followed

by a second RLS update of size $L_2 \times M$ with $\mathbf{u}_2'[n]$ as input and $\mathbf{e}[n]$ as output. The aforementioned reuse of the denominator means that in the second RLS update $w[n]$ in step 4 of the algorithm in Table 1 should be replaced with the $\zeta[n]$ computed in the first RLS update. The first RLS update will require $2L_1^2 + (2(M + L_2) + 3)L_1$ MACs while the second one will require $2L_2^2 + (2M + 3)L_2$ if the reuse is accounted for. This implies a total cost for the cascaded update of $2L^2 - 2L_1L_2 + (2M+3)L$, meaning that it will always be cheaper than the integrated update. The cost is minimal when $L_1 = L_2 = L/2$, leading to a cost of $3/2L^2 + (2M + 3)L$ compared to $2L^2 + (2M+3)L$ for an integrated RLS update.

5. EQUIVALENCE TO THE RMGS ALGORITHM

A natural extension of the proposed algorithm would be to partition the input vector into more than two parts, although it should be noted that this might lose the possible physical interpretation. This can be achieved by first splitting the filter into a part of the desired length and the remainder and repeating this procedure on the latter part until the entire input vector is partitioned. When the filter is split into x even parts, i.e. all parts have length L/x, this leads to an update cost of

$$\underbrace{2\left(1 - \frac{1}{x^2}\sum_{i=1}^{x}(x-i)\right)}_{=1+x^{-1}} L^2 + (2M+3)L. \qquad (23)$$

In the limit this can be repeated up until the level where each filter is acting on single input sample, i.e. $x = L$. For $M = 1$ this leads to a cost equal to the one reported in [7], suggesting that the algorithm proposed in this paper is actually a generalization of the RMGS algorithm in which subspaces are iteratively orthogonalized rather than vectors. Indeed, when replacing the a priori errors in the updating equations (18) and (19) with the a posteriori error divided by a so-called conversion factor [13] and writing the denominators in (13) and (20) in terms of the inverse correlation matrix at time n instead of $n-1$, it can be shown through induction that, when the cascade is split to operate on samples just like RMGS, the obtained update equations are identical.

6. SIMULATION RESULTS

In order to show that the proposed algorithm leads to interpretable results and is indeed equivalent to the corresponding integrated RLS, an experiment with the generalized echo and interference canceller (GEIC) proposed in [8] is used. As mentioned in Section 3.2, taking $\mathbf{u}_1[n]$ and $\mathbf{u}_2[n]$ to be the loudspeaker signals and noise references should lead to $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ being echo cancellation filters on the noise reference and signal reference paths, respectively.

A setup with two microphones is considered. Audio segments with a duration of 10 seconds at 16 kHz were obtained by concatenating sentences from the VCTK corpus [18]. Impulse responses were taken from the Hearpiece database [19], also resampled to 16 kHz and padded or truncated to 160 samples. The desired signal source was put at the 90 degree angle while the noise source and loudspeaker positions were varied. All three source signals were scaled to have the same power and measurement noise with an SNR of 20 dB w.r.t. the desired source was added. The quiescent beamformer and corresponding blocking matrix were taken to be $[1/2,\ 1/2]^{\mathrm{T}}$ and $\left[1/\sqrt{2}, -1/\sqrt{2}\right]^{\mathrm{T}}$, respectively. The implies that the expected echo canceller in the signal reference path is the average of the impulse responses from the loudspeaker to both microphones whereas for the noise reference path this is a scaled difference between the two. All RLS updates were performed using HRLS [13]. λ was set to 0.9999 and $w[n]$ was fixed at 1. An experiment runs both a standard integrated RLS update as well as a cascaded RLS update. It is expected that i) the output of the two is identical up to machine precision and ii) $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ converge to the expected filters.

In total 144 different combinations of source signals, noise source and loudspeaker locations and microphones were tested. During the simulations it was confirmed that the difference between the output of the integrated filter and the cascade was always around machine precision. The misalignment (MIS) [20] between the expected filters and $\mathbf{W}_{12}[n]$ and $\mathbf{W}_{1\mathbf{d}}[n]$ is plotted in Fig. 2. The curves indicate the median MIS while the shaded areas are bounded by the 25th and 75th percentile. This demonstrates that the filters in the cascade converge to the expected, physically interpretable filters. Both MIS were around -10 dB due to imperfect quiescent beamformers and blocking matrices.

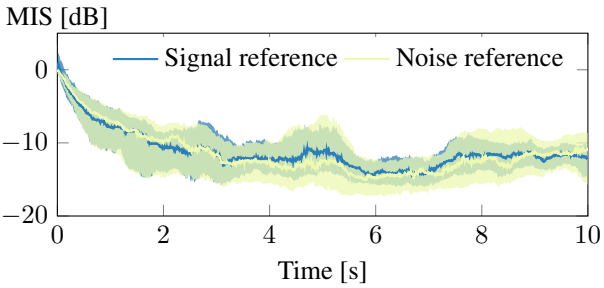


Fig. 2: Evolution of MIS between expected echo cancellation paths and filters obtained by cascaded RLS in GEIC.

7. CONCLUSION

In this paper it has been shown that when an optimal integrated filter is decomposed into a cascade of optimal filters, it is possible to update the filters in the cascade using RLS while retaining equivalence with the integrated filter, also updated using RLS, by changing the weighting in the second RLS update of the cascade. It was also shown that updating the cascade is actually computationally cheaper than performing a standard RLS update. Experimental results corroborate the fact that the two algorithms are equivalent. Finally, it was shown that the proposed algorithm is a generalization of the RMGS algorithm.

8. ACKNOWLEDGEMENTS

This research was carried out at the ESAT Laboratory of KU Leuven, in the frame of Research Council KU Leuven Project C14-21-0075 “A holistic approach to the design of integrated and distributed digital signal processing algorithms for audio and speech communication devices”, and Aspirant Grant 11PDH24N (for A. Roebben) from the Research Foundation - Flanders (FWO). The scientific responsibility is assumed by its authors.

9. REFERENCES

[1] S. S. Haykin, *Adaptive Filter Theory*, 5th ed. Upper Saddle River, NJ: Pearson, 2014.

[2] C. Boeddeker, T. Nakatani, K. Kinoshita, and R. Haeb-Umbach, “Jointly optimal dereverberation and beamforming,” in *Proc. 2020 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2020, pp. 216–220.

[3] A. Roebben, T. van Waterschoot, J. Wouters, and M. Moonen, “Integrated minimum mean squared error algorithms for combined acoustic echo cancellation and noise reduction,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 34, pp. 512–528, 2026.

[4] A.-Y. Wu and K. R. Liu, “Split recursive least-squares: algorithms, architectures, and applications,” *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 43, no. 9, pp. 645–658, 1996.

[5] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochina, “A multichannel recursive least-squares algorithm based on a kronecker product decomposition,” in *Proc. 2020 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, 2020, pp. 14–18.

[6] R.-A. Otopeleanu, C. Paleologu, J. Benesty, C.-L. Stanciu, J. R. Jensen, and M. Græsbøll Christensen, “An RLS algorithm using a third-order tensor decomposition and variable forgetting factors,” in *Proc. 2026 IEEE Int. Black Seas Conf. Commun. Netw. (BlackSeaCom)*, 2026, pp. 1–6.

[7] F. Ling, D. Manolakis, and J. Proakis, “A recursive modified gram-schmidt algorithm for least- squares estimation,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, no. 4, pp. 829–836, 1986.

[8] W. Herbordt, W. Kellermann, and S. Nakamura, “Joint optimization of lcmv beamforming and acoustic echo cancellation,” in *Proc. 2004 12th Eur. Signal Process. Conf. (EUSIPCO)*, 2004, pp. 2003–2006.

[9] T. Dietzen, S. Doclo, M. Moonen, and T. Van Waterschoot, “Joint multi-microphone speech dereverberation and noise reduction using integrated sidelobe cancellation and linear prediction,” in *Proc. 2018 16th Int. Workshop Acoust. Signal Enhancement (IWAENC)*, 2018, pp. 221–225.

[10] B. Liekens, “Interpretable cascaded rls,” 2026-09-08. [Online]. Available: https://github.com/BasilLiekens/interpretable-cascaded-RLS

[11] T. Yoshioka, H. Tachibana, T. Nakatani, and M. Miyoshi, “Adaptive dereverberation of speech signals with speaker-position change detection,” in *Proc. 2009 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2009, pp. 3733–3736.

[12] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, “NARA-WPE: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing,” in *Speech Commun.; 13th ITG-Symp.*, 2018, pp. 1–5.

[13] J. Wung, A. Jukić, S. Malik, M. Souden, R. Pichevar, J. Atkins, D. Naik, and A. Acero, “Robust multichannel linear prediction for online speech dereverberation using weighted householder least squares lattice adaptive filter,” *IEEE Trans. Signal Process.*, vol. 68, pp. 3559–3574, 2020.

[14] J. Cioffi and T. Kailath, “Fast, recursive-least-squares transversal filters for adaptive filtering,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 2, pp. 304–337, 1984.

[15] R. Merched, “Extended RLS lattice adaptive filters,” *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2294–2309, 2003.

[16] I. Skidmore and I. Proudler, “The KaGE RLS algorithm,” *IEEE Trans. Signal Process.*, vol. 51, no. 12, pp. 3094–3104, 2003.

[17] T.-T. Lu and S.-H. Shiou, “Inverses of 2×2 block matrices,” *Comput. & Math. Appl.*, vol. 43, no. 1, pp. 119–129, 2002.

[18] J. Yamagishi, C. Veaux, and K. MacDonald, “CSTR VCTK corpus: English multi-speaker corpus for cstr voice cloning toolkit.” [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443

[19] F. Denk and B. Kollmeier, “The hearpiece database of individual transfer functions of an in-the-ear earpiece for hearing device research,” *Acta Acust.*, vol. 5, p. 2, 2021.

[20] A. Spriet, “Adaptive filtering techniques for noise reduction and acoustic feedback cancellation in hearing aids,” PhD thesis, KU Leuven, Leuven, Belgium, 2004.