SCIENTIFIC REPORT
PROJECT PN-III-P4-PCE-2021-0438

Novel Sparsity-Aware Adaptive Algorithms for Acoustic Applications (NARALAN)

Stage 2023 – Development of RLS Algorithms with Individual Control Factors Based on Tensorial Decomposition

1 RLS Adaptive Algorithms for System Identification

Adaptive filtering algorithms are frequently involved in many real-word system identification problems [1]–[15]. Among them, the recursive least-squares (RLS) algorithm represents a very appealing choice due to its fast convergence rate, which can be achieved even for highly correlated input signals [16]. However, the price to pay is a higher computational complexity, as compared to the counterparts that belong to the least-mean-square (LMS) family.

In this framework, the overall difficulty increases when dealing with the identification of long length impulse responses, which raise significant challenges in terms of the complexity, convergence/tracking, and accuracy of the solution. Even the "fast" (i.e., less complex) versions of the RLS algorithm face performance limitations in such scenarios [17]. A well-known example is related to acoustic echo cancellation, where the acoustic echo paths are usually modeled as finite impulse response filters that can reach thousands of coefficients [18].

System identification problems that involve a large parameter space were previously addressed in the framework of linearly separable systems [19]–[24], which rely on the modeling of rank-1 tensors. For example, this technique can be applied in case of a long length impulse response $\mathbf{g}$ with $L = \prod_{s=1}^{S} L_s$ coefficients, which can be (perfectly) decomposed as a combination of S shorter impulse responses $\mathbf{g}_s$ of lengths L_s, $s = 1, 2 \ldots, S$, via the Kronecker product, i.e., $\mathbf{g} = \mathbf{g}_1 \otimes \mathbf{g}_2 \otimes \cdots \otimes \mathbf{g}_S$. Clearly, combining these sets of coefficients using the outer product results in the rank-1 tensor $\boldsymbol{\mathcal{G}} = \mathbf{g}_S \circ \mathbf{g}_{S-1} \circ \cdots \circ \mathbf{g}_1$ of size $L_S \times L_{S-1} \times \cdots \times L_1$. Furthermore, the components $\mathbf{g}_s$ ($s = 1, 2 \ldots, S$) can be individually processed and estimated, resulting in an improved convergence rate of the global solution [19]. This approach allows the design of different tensorial adaptive algorithms [20] and iterative Wiener filters [21]. As a result, it finds applications in several frameworks, like multiple-input multiple-output (MIMO) wireless communication systems [22], [23] and separable linearly constrained beamformers for large-scale antenna systems [24]. The main advantage of this decomposition-based technique that involves the modeling of rank-1 tensors is related to the dimensionality reduction. Basically, a large problem of size $\prod_{s=1}^{S} L_s$ is reformulated and solved as a combination of smaller solutions, thus resulting in a parameter space with $\sum_{s=1}^{S} L_s$ coefficients. On the other hand, this method can be successfully applied only for linearly separable systems, which is not the case in many real-world applications, including network and acoustic echo cancellation.

The realistic network and acoustic echo paths cannot be perfectly decomposed as linearly separable systems and further modeled as rank-1 tensors. However, exploiting the characteristics of the systems to be identified represents a natural path to follow, in order to overcome the main challenges related to the estimation of long length impulse responses. In this context, several recent works have focused on decomposition-based techniques that involve the nearest Kronecker product (NKP) and low-rank approxi-

mations, e.g., see [25] and the references therein. These methods exploit the singular value decomposition (SVD) of the corresponding matrix of the reshaped impulse response (obtained by inverse vectorization) and its low rank feature. As a result, while the conventional system identification approach relies on a single long length adaptive filter, the NKP-based method obtains the solution using a combination of two (much) shorter filters. This fits very well in conjunction with fast converging algorithms like RLS. In addition, since a wide range of real-world systems own low-rank features, this decomposition-based technique can be applied in the framework of other important applications, e.g., [26]–[30]. Nevertheless, the NKP-based technique used in these works relies on the second-order decomposition, exploiting the SVD of the (low-rank) corresponding matrix associated to the impulse response. Extending this technique to a higher-order decomposition faces the difficulty of dealing with a higher-rank tensor, where its rank results from a sum of rank-1 tensors. In this context, while the rank of a matrix (related to the second-order decomposition) is upper limited by the number of its columns, the rank of a higher-order tensor could exceed its largest dimension. Therefore, the direct extension of the NKP-based technique to a higher-order decomposition may not be a feasible approach.

Recently, the extension of the NKP-based approach to a third-order decomposition level has been addressed in [2], targeting a higher dimensionality reduction (i.e., a combination of even shorter filters). As explained before, this was not a straightforward development, since the direct solution would involve a third-order tensor, while approximating its rank would be a sensitive issue [31]–[33]. Contrary, in [2], the rank of this tensor is controlled in terms of a matrix rank, which is limited to small values. The resulting iterative Wiener filter developed based on this approach has proved to outperform its counterpart that relies on the previous second-order NKP-based decomposition.

Nevertheless, there are some inherent limitations of the Wiener filter, which are mainly related to the statistics' estimates and the time-invariant framework. These further influence the accuracy of the solution, respectively the applicability in real-time applications working in time-variant environments. Therefore, an adaptive filtering RLS-based algorithm that exploits the third-order tensor decomposition method would represent an appealing practical choice in many real-world applications. We should note that there is not a straightforward extension from the Wiener filter to the RLS algorithm under the third-order decomposition framework, due to the particular connection between the components filters and the specific parameters. In addition, while the iterative Wiener filter developed in [2] sequentially alternates between the solutions resulted from three sets of Wiener-Hopf equations, it would be desirable for the adaptive algorithm to update its component filters in parallel, which is highly useful in terms of the modularity of implementation. Due to these features (i.e., parallel updates and specific combination of three shorter filters), such an adaptive filtering solution would be an appealing choice among the family of RLS-based algorithms, especially in the framework of system identification problems that deal with long length impulse responses. In addition, the solution could be applied in the framework of other applications that require good tracking capabilities, like those related to the control of unmanned aerial vehicles [34], [35].

Motivated by these aspects, we further build on the third-order tensor decomposition method and design an RLS algorithm that exploits this efficient technique. The resulting algorithm updates in parallel and further combines the coefficients of three adaptive filters of much shorter lengths (as compared to the length of the original impulse response), which leads to important advantages in terms of the main performance criteria. In the following, the third-order tensor decomposition technique is summarized in Section 2. Next, the proposed RLS-based algorithm is developed in Section 3. Simulations results are provided in Section 4, in the context of echo cancellation. Finally, an overall discussion and several important remarks that follow these results are summarized in Section 5.

2 Third-Order Tensor Decomposition of the Impulse Response

In this section, we summarize the background from [2], which recently introduced the third-order tensor decomposition-based approach, in the framework of linear system identification. To this purpose, let us consider a single-input single-output (SISO) scenario with real-valued signals. The main goal is to identify an unknown impulse response with L real-valued coefficients, which are grouped into the vector $\mathbf{h} = \begin{bmatrix} h_1 & h_2 & \cdots & h_L \end{bmatrix}^T$, where the superscript ^{T} denotes the transpose operator. Thus, at the discrete-

time index n, the reference (or desired) signal results in

$$d(n) = \mathbf{h}^T \mathbf{x}(n) + w(n) = y(n) + w(n), \tag{1}$$

where $\mathbf{x}(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-L+1) \end{bmatrix}^T$ contains the L most recent time samples of the zero-mean input signal $x(n)$, $y(n) = \mathbf{h}^T \mathbf{x}(n)$ is the output signal, and $w(n)$ is a zero-mean additive noise, which is uncorrelated with $x(n)$.

Next, let us consider that the length of $\mathbf{h}$ can be expressed as $L = L_{11} L_{12} L_2$, with $L_{11} \geq L_{12}$ and $L_{11} L_{12} \gg L_2$. Therefore, the impulse response can be equivalently decomposed as [2]

$$\mathbf{h} = \sum_{i=1}^{L_2} \mathbf{h}_2^i \otimes \left(\sum_{j=1}^{L_{12}} \mathbf{h}_{12}^{ij} \otimes \mathbf{h}_{11}^{ij} \right) = \sum_{i=1}^{L_2} \sum_{j=1}^{L_{12}} \mathbf{h}_2^i \otimes \mathbf{h}_{12}^{ij} \otimes \mathbf{h}_{11}^{ij}, \tag{2}$$

where the (shorter) component impulse responses $\mathbf{h}_{11}^{ij}$, $\mathbf{h}_{12}^{ij}$, and $\mathbf{h}_2^i$ have the lengths L_{11}, L_{12}, and L_2, respectively, while $\otimes$ denotes the Kronecker product. At this point, since $\mathbf{h}_2^i$ is considered very short, there is no need to decompose it but to consider that the Kronecker product of the other two components is low rank [36]. Consequently, (2) becomes

$$\mathbf{h} = \sum_{l=1}^{L_2} \mathbf{h}_2^l \otimes \left(\sum_{p=1}^{P} \mathbf{h}_{12}^{lp} \otimes \mathbf{h}_{11}^{lp} \right) = \sum_{l=1}^{L_2} \sum_{p=1}^{P} \mathbf{h}_2^l \otimes \mathbf{h}_{12}^{lp} \otimes \mathbf{h}_{11}^{lp}, \tag{3}$$

where $P < L_{12}$. This is a reasonable approach, since in practice most of the system impulse responses are low rank [25].

As we can notice, an equivalent representation of (3) in a third-order tensor form is

$$\boldsymbol{\mathcal{H}} = \sum_{l=1}^{L_2} \sum_{p=1}^{P} \mathbf{h}_{11}^{lp} \circ \mathbf{h}_{12}^{lp} \circ \mathbf{h}_2^l = \sum_{p=1}^{P} \left(\sum_{l=1}^{L_2} \boldsymbol{\mathcal{H}}^{lp} \right) = \sum_{p=1}^{P} \underline{\boldsymbol{\mathcal{H}}}^p, \tag{4}$$

where $\circ$ is the outer product, while $\boldsymbol{\mathcal{H}}^{lp} = \mathbf{h}_{11}^{lp} \circ \mathbf{h}_{12}^{lp} \circ \mathbf{h}_2^l$ and $\underline{\boldsymbol{\mathcal{H}}}^p = \sum_{l=1}^{L_2} \boldsymbol{\mathcal{H}}^{lp}$ are third-order tensors (of different ranks), which belong to $\mathbb{R}^{L_{11} \times L_{12} \times L_2}$. As we can notice, $\underline{\boldsymbol{\mathcal{H}}}^p$ corresponds to the canonical polyadic (CP) decomposition [37], i.e., the factorization of a third-order tensor into a sum of rank-1 third-order tensors, $\boldsymbol{\mathcal{H}}^{lp}$, $l = 1, 2, \ldots, L_2$. In this context, L_2 represents the rank of $\underline{\boldsymbol{\mathcal{H}}}^p$, i.e., the minimum number of rank-1 tensors that generates $\underline{\boldsymbol{\mathcal{H}}}^p$. Hence, $\boldsymbol{\mathcal{H}}$ is the sum of P third-order tensors of rank L_2. On the other hand, based on (2), the representation of $\mathbf{h}$ in a matrix form is

$$\mathbf{H} = \sum_{l=1}^{L_2} \left(\sum_{j=1}^{L_{12}} \mathbf{h}_{12}^{lj} \otimes \mathbf{h}_{11}^{lj} \right) \left(\mathbf{h}_2^l \right)^T = \sum_{l=1}^{L_2} \left(\sum_{j=1}^{L_{12}} \mathbf{h}_{12}^{lj} \otimes \mathbf{h}_{11}^{lj} \right) \circ \mathbf{h}_2^l. \tag{5}$$

The size of this matrix is $L_{11} L_{12} \times L_2$, while its maximum rank is L_2 (with $L_2 \ll L_{11} L_{12}$). Concluding, based on (4) and (5), the rank of a third-order tensor is controlled from the rank of a matrix, which is limited to small values. In other words, the sensitive problem of approximating the rank of a tensor is avoided thanks to the control of a matrix rank.

In this decomposition framework, the main focus is the identification of the component impulse responses from (3). Thus, the identification of the impulse response $\mathbf{h}$ (with $L = L_{11} L_{12} L_2$ coefficients) is reformulated as a combination of three sets of coefficients, i.e., $\mathbf{h}_2^l$ of length L_2 (with $l = 1, 2, \ldots, L_2$), $\mathbf{h}_{12}^{lp}$ of length L_{12}, and $\mathbf{h}_{11}^{lp}$ of length L_{11} (with $l = 1, 2, \ldots, L_2$ and $p = 1, 2, \ldots, P$). Correspondingly, related to these three sets, we need to estimate L_2^2, $P L_{12} L_2$, and $P L_{11} L_2$ coefficients, respectively. For the common decomposition setup that involves $L_{11} L_{12} \gg L_2$ and $P \ll L_{12}$ [2], this represents an important dimensionality reduction, especially for large values of L.

3 System Identification with the Tensorial RLS Algorithm

The objective of the conventional SISO system identification problem from (1) is to estimate $\mathbf{h}$ given the input signal vector, $\mathbf{x}(n)$, and the reference signal, $d(n)$. On the other hand, the decomposition-based approach presented in Section 2 targets the estimation of three shorter sets of coefficients, which are combined together. To this purpose, an RLS adaptive filter that exploits this decomposition technique will be further developed in this section.

In this context, let us consider that $\widehat{\mathbf{h}}(n)$ is an estimate of the impulse response $\mathbf{h}$ at the discrete-time index n, so that the a priori error signal results in $e(n) = d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$. On the other hand, using the impulse response decomposition from (3), we can also express its estimate as $\widehat{\mathbf{h}}(n) = \sum_{l=1}^{L_2}\sum_{p=1}^{P} \widehat{\mathbf{h}}_2^l(n) \otimes \widehat{\mathbf{h}}_{12}^{lp}(n) \otimes \widehat{\mathbf{h}}_{11}^{lp}(n)$, where $\widehat{\mathbf{h}}_{11}^{lp}(n)$, $\widehat{\mathbf{h}}_{12}^{lp}(n)$, and $\widehat{\mathbf{h}}_2^l(n)$ are three shorter impulse responses of length L_{11}, L_{12}, and L_2, respectively. Next, in order to construct the cost functions of the RLS-based algorithm, we can rewrite $e(n)$ in three equivalent ways, with the purpose of "extracting" each individual component. In the sequel, $\mathbf{I}_{L_\bullet}$ denotes the identity matrix of size $L_\bullet \times L_\bullet$.

First, we can develop the a priori error as

$$\begin{aligned} e(n) &= d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\sum_{p=1}^{P}\left[\mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{12}^{lp}(n-1) \otimes \widehat{\mathbf{h}}_{11}^{lp}(n-1)\right]\widehat{\mathbf{h}}_2^l(n-1) \\ &= d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\sum_{p=1}^{P}\widehat{\mathbf{H}}_{12,11}^{lp}(n-1)\widehat{\mathbf{h}}_2^l(n-1) = d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\widehat{\overline{\mathbf{H}}}_{12,11}^{l}(n-1)\mathbf{h}_2^l(n-1) \\ &= d(n) - \mathbf{x}^T(n)\widehat{\overline{\mathbf{H}}}_{12,11}(n-1)\widehat{\underline{\mathbf{h}}}_2(n-1) = d(n) - \widehat{\underline{\mathbf{h}}}_2^T(n-1)\mathbf{x}_{12,11}(n), \end{aligned} \tag{6}$$

where $\widehat{\mathbf{H}}_{12,11}^{lp}(n) = \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{12}^{lp}(n) \otimes \widehat{\mathbf{h}}_{11}^{lp}(n)$ is a matrix of size $L \times L_2$, $\widehat{\overline{\mathbf{H}}}_{12,11}^{l}(n) = \sum_{p=1}^{P}\widehat{\mathbf{H}}_{12,11}^{lp}(n)$, the matrix $\widehat{\overline{\mathbf{H}}}_{12,11}(n) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{12,11}^{1}(n) & \widehat{\overline{\mathbf{H}}}_{12,11}^{2}(n) & \cdots & \widehat{\overline{\mathbf{H}}}_{12,11}^{L_2}(n) \end{array}\right]$ has the size $L \times L_2^2$, the vector $\widehat{\underline{\mathbf{h}}}_2(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_2^1\right)^T(n) & \left(\widehat{\mathbf{h}}_2^2\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_2^{L_2}\right)^T(n) \end{array}\right]^T$ has the length L_2^2, and $\mathbf{x}_{12,11}(n) = \widehat{\overline{\mathbf{H}}}_{12,11}^T(n-1)\mathbf{x}(n)$. As we can notice, the form from (6) targets the extraction and separation of $\widehat{\mathbf{h}}_2^l(n),\ l = 1, 2, \ldots, L_2$.

Second, this error signal can be alternatively rewritten as

$$\begin{aligned} e(n) &= d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\sum_{p=1}^{P}\left[\widehat{\mathbf{h}}_2^l(n-1) \otimes \mathbf{I}_{L_{12}} \otimes \widehat{\mathbf{h}}_{11}^{lp}(n-1)\right]\widehat{\mathbf{h}}_{12}^{lp}(n-1) \\ &= d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\sum_{p=1}^{P}\widehat{\mathbf{H}}_{2,11}^{lp}(n-1)\widehat{\mathbf{h}}_{12}^{lp}(n-1) = d(n) - \mathbf{x}^T(n)\sum_{l=1}^{L_2}\widehat{\overline{\mathbf{H}}}_{2,11}^{l}(n-1)\widehat{\overline{\mathbf{h}}}_{12}^{l}(n-1) \\ &= d(n) - \mathbf{x}^T(n)\widehat{\overline{\mathbf{H}}}_{2,11}(n-1)\widehat{\underline{\mathbf{h}}}_{12}(n-1) = d(n) - \widehat{\underline{\mathbf{h}}}_{12}^T(n-1)\mathbf{x}_{2,11}(n), \end{aligned} \tag{7}$$

where $\widehat{\mathbf{H}}_{2,11}^{lp}(n) = \widehat{\mathbf{h}}_2^l(n) \otimes \mathbf{I}_{L_{12}} \otimes \widehat{\mathbf{h}}_{11}^{lp}(n)$ is a matrix of size $L \times L_{12}$, $\widehat{\overline{\mathbf{H}}}_{2,11}^{l}(n) = \left[\begin{array}{cccc} \widehat{\mathbf{H}}_{2,11}^{l1}(n) & \widehat{\mathbf{H}}_{2,11}^{l2}(n) & \cdots & \widehat{\mathbf{H}}_{2,11}^{lP}(n) \end{array}\right]$ is a matrix of size $L \times PL_{12}$, the vector $\widehat{\overline{\mathbf{h}}}_{12}^{l}(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{12}^{l1}\right)^T(n) & \left(\widehat{\mathbf{h}}_{12}^{l2}\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_{12}^{lP}\right)^T(n) \end{array}\right]^T$ has the length PL_{12}, the matrix $\widehat{\overline{\mathbf{H}}}_{2,11}(n) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{2,11}^{1}(n) & \widehat{\overline{\mathbf{H}}}_{2,11}^{2}(n) & \cdots & \widehat{\overline{\mathbf{H}}}_{2,11}^{L_2}(n) \end{array}\right]$ has the size $L \times PL_{12}L_2$, the vector $\widehat{\underline{\mathbf{h}}}_{12}(n) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{12}^{1}\right)^T(n) & \left(\widehat{\overline{\mathbf{h}}}_{12}^{2}\right)^T(n) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{12}^{L_2}\right)^T(n) \end{array}\right]^T$ has the length $PL_{12}L_2$, and $\mathbf{x}_{2,11}(n) = \widehat{\overline{\mathbf{H}}}_{2,11}^T(n-1)\mathbf{x}(n)$. Clearly, (7) is designed for the extraction of $\widehat{\mathbf{h}}_{12}^{lp}(n),\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P$.

The third and final way to process the a priori error results in

$$\begin{aligned} e(n) &= d(n) - \mathbf{x}^T(n) \sum_{l=1}^{L_2} \sum_{p=1}^{P} \left[\widehat{\mathbf{h}}_2^l(n-1) \otimes \widehat{\mathbf{h}}_{12}^{lp}(n-1) \otimes \mathbf{I}_{L_{11}} \right] \widehat{\mathbf{h}}_{11}^{lp}(n-1) \\ &= d(n) - \mathbf{x}^T(n) \sum_{l=1}^{L_2} \sum_{p=1}^{P} \widehat{\mathbf{H}}_{2,12}^{lp}(n-1) \widehat{\mathbf{h}}_{11}^{lp}(n-1) = d(n) - \mathbf{x}^T(n) \sum_{l=1}^{L_2} \widehat{\overline{\mathbf{H}}}_{2,12}^{l}(n-1) \widehat{\overline{\mathbf{h}}}_{11}^{l}(n-1) \\ &= d(n) - \mathbf{x}^T(n) \widehat{\overline{\mathbf{H}}}_{2,12}(n-1) \widehat{\underline{\overline{\mathbf{h}}}}_{11}(n-1) = d(n) - \widehat{\underline{\overline{\mathbf{h}}}}_{11}^T(n-1) \mathbf{x}_{2,12}(n), \end{aligned} \tag{8}$$

where $\widehat{\mathbf{H}}_{2,12}^{lp}(n) = \widehat{\mathbf{h}}_2^l(n) \otimes \widehat{\mathbf{h}}_{12}^{lp}(n) \otimes \mathbf{I}_{L_{11}}$ is a matrix of size $L \times L_{11}$, $\widehat{\overline{\mathbf{H}}}_{2,12}^{l}(n) = \left[\begin{array}{cccc} \widehat{\mathbf{H}}_{2,12}^{l1}(n) & \widehat{\mathbf{H}}_{2,12}^{l2}(n) & \cdots & \widehat{\mathbf{H}}_{2,12}^{lP}(n) \end{array} \right]$ is a matrix of size $L \times PL_{11}$, $\widehat{\overline{\mathbf{h}}}_{11}^{l}(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{11}^{l1}\right)^T(n) & \left(\widehat{\mathbf{h}}_{11}^{l2}\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_{11}^{lP}\right)^T(n) \end{array} \right]^T$ is a vector of length PL_{11}, $\widehat{\underline{\overline{\mathbf{H}}}}_{2,12}(n) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{2,12}^{1}(n) & \widehat{\overline{\mathbf{H}}}_{2,12}^{2}(n) & \cdots & \widehat{\overline{\mathbf{H}}}_{2,12}^{L_2}(n) \end{array} \right]$ is a matrix of size $L \times PL_{11}L_2$, the vector $\widehat{\underline{\overline{\mathbf{h}}}}_{11}(n) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{11}^{1}\right)^T(n) & \left(\widehat{\overline{\mathbf{h}}}_{11}^{2}\right)^T(n) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{11}^{L_2}\right)^T(n) \end{array} \right]^T$ has the length $PL_{11}L_2$, and $\mathbf{x}_{2,12}(n) = \widehat{\underline{\overline{\mathbf{H}}}}_{2,12}^T(n-1)\mathbf{x}(n)$. The expression from (8) separates $\widehat{\mathbf{h}}_{11}^{lp}(n),\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P$.

At this point, based on (6)–(8) and using the least-squares (LS) criterion [16], we can define three cost functions associated to the component filters, as follows:

$$\widetilde{\mathcal{J}}\left[\widehat{\underline{\mathbf{h}}}_2(n) | \widehat{\underline{\overline{\mathbf{h}}}}_{12}, \widehat{\underline{\overline{\mathbf{h}}}}_{11}\right] = \sum_{k=1}^{n} \lambda_2^{n-k} \left[d(k) - \widehat{\underline{\mathbf{h}}}_2^T(n) \mathbf{x}_{12,11}(k) \right]^2, \tag{9}$$

$$\widetilde{\mathcal{J}}\left[\widehat{\underline{\overline{\mathbf{h}}}}_{12}(n) | \widehat{\underline{\mathbf{h}}}_2, \widehat{\underline{\overline{\mathbf{h}}}}_{11}\right] = \sum_{k=1}^{n} \lambda_{12}^{n-k} \left[d(k) - \widehat{\underline{\overline{\mathbf{h}}}}_{12}^T(n) \mathbf{x}_{2,11}(k) \right]^2, \tag{10}$$

$$\widetilde{\mathcal{J}}\left[\widehat{\underline{\overline{\mathbf{h}}}}_{11}(n) | \widehat{\underline{\mathbf{h}}}_2, \widehat{\underline{\overline{\mathbf{h}}}}_{12}\right] = \sum_{k=1}^{n} \lambda_{11}^{n-k} \left[d(k) - \widehat{\underline{\overline{\mathbf{h}}}}_{11}^T(n) \mathbf{x}_{2,12}(k) \right]^2, \tag{11}$$

where λ_2, λ_{12}, and λ_{11} are the forgetting factors, i.e., positive constants smaller than or equal to one. As the notation from the left-hand sides of (9)–(11) suggests, two out of the three component impulse responses are considered fixed within the LS criterion at a time, similar to a multilinear optimization strategy [19]–[24].

Based on these considerations, three sets of normal equations are obtained from the minimization of (9), (10), and (11) with respect to $\widehat{\underline{\mathbf{h}}}_2(n)$, $\widehat{\underline{\overline{\mathbf{h}}}}_{12}(n)$, and $\widehat{\underline{\overline{\mathbf{h}}}}_{11}(n)$, respectively, i.e.,

$$\mathbf{R}_{12,11}(n) \widehat{\underline{\mathbf{h}}}_2(n) = \mathbf{r}_{12,11}(n), \tag{12}$$

$$\mathbf{R}_{2,11}(n) \widehat{\underline{\overline{\mathbf{h}}}}_{12}(n) = \mathbf{r}_{2,11}(n), \tag{13}$$

$$\mathbf{R}_{2,12}(n) \widehat{\underline{\overline{\mathbf{h}}}}_{11}(n) = \mathbf{r}_{2,12}(n), \tag{14}$$

where

$$\mathbf{R}_{12,11}(n) = \sum_{k=1}^{n} \lambda_2^{n-k} \mathbf{x}_{12,11}(k)\mathbf{x}_{12,11}^T(k) = \lambda_2 \mathbf{R}_{12,11}(n-1) + \mathbf{x}_{12,11}(n)\mathbf{x}_{12,11}^T(n), \tag{15}$$

$$\mathbf{r}_{12,11}(n) = \sum_{k=1}^{n} \lambda_2^{n-k} \mathbf{x}_{12,11}(k)d(k) = \lambda_2 \mathbf{r}_{12,11}(n-1) + \mathbf{x}_{12,11}(n)d(n), \tag{16}$$

$$\mathbf{R}_{2,11}(n) = \sum_{k=1}^{n} \lambda_{12}^{n-k} \mathbf{x}_{2,11}(k)\mathbf{x}_{2,11}^T(k) = \lambda_{12} \mathbf{R}_{2,11}(n-1) + \mathbf{x}_{2,11}(n)\mathbf{x}_{2,11}^T(n), \tag{17}$$

$$\mathbf{r}_{2,11}(n) = \sum_{k=1}^{n} \lambda_{12}^{n-k} \mathbf{x}_{2,11}(k)d(k) = \lambda_{12} \mathbf{r}_{2,11}(n-1) + \mathbf{x}_{2,11}(n)d(n), \tag{18}$$

$$\mathbf{R}_{2,12}(n) = \sum_{k=1}^{n} \lambda_{11}^{n-k} \mathbf{x}_{2,12}(k)\mathbf{x}_{2,12}^T(k) = \lambda_{11} \mathbf{R}_{2,12}(n-1) + \mathbf{x}_{2,12}(n)\mathbf{x}_{2,12}^T(n), \tag{19}$$

$$\mathbf{r}_{2,12}(n) = \sum_{k=1}^{n} \lambda_{11}^{n-k} \mathbf{x}_{2,12}(k)d(k) = \lambda_{11} \mathbf{r}_{2,12}(n-1) + \mathbf{x}_{2,12}(n)d(n). \tag{20}$$

In order to avoid matrix inversion operations for solving (12)–(14), the Woodbury formula (i.e., the matrix inversion lemma) [16] can be used in (15), (17), and (19), respectively, thus obtaining the recursive updates:

$$\mathbf{P}_{12,11}(n) = \lambda_2^{-1} \left[\mathbf{I}_{L_2^2} - \mathbf{k}_{12,11}(n)\mathbf{x}_{12,11}^T(n) \right] \mathbf{P}_{12,11}(n-1), \tag{21}$$

$$\mathbf{P}_{2,11}(n) = \lambda_{12}^{-1} \left[\mathbf{I}_{PL_{12}L_2} - \mathbf{k}_{2,11}(n)\mathbf{x}_{2,11}^T(n) \right] \mathbf{P}_{2,11}(n-1), \tag{22}$$

$$\mathbf{P}_{2,12}(n) = \lambda_{11}^{-1} \left[\mathbf{I}_{PL_{11}L_2} - \mathbf{k}_{2,12}(n)\mathbf{x}_{2,12}^T(n) \right] \mathbf{P}_{2,12}(n-1), \tag{23}$$

where the notation $\mathbf{P}_{12,11}(n)$, $\mathbf{P}_{2,11}(n)$, and $\mathbf{P}_{2,12}(n)$ stand for $\mathbf{R}_{12,11}^{-1}(n)$, $\mathbf{R}_{2,11}^{-1}(n)$, and $\mathbf{R}_{2,12}^{-1}(n)$, respectively, while $\mathbf{k}_{12,11}(n) = \mathbf{P}_{12,11}(n)\mathbf{x}_{12,11}(n)$, $\mathbf{k}_{2,11}(n) = \mathbf{P}_{2,11}(n)\mathbf{x}_{2,11}(n)$, and $\mathbf{k}_{2,12}(n) = \mathbf{P}_{2,12}(n)\mathbf{x}_{2,12}(n)$ are the Kalman gain vectors [16], which are evaluated as

$$\mathbf{k}_{12,11}(n) = \frac{\mathbf{P}_{12,11}(n-1)\mathbf{x}_{12,11}(n)}{\lambda_2 + \mathbf{x}_{12,11}^T(n)\mathbf{P}_{12,11}(n-1)\mathbf{x}_{12,11}(n)}, \tag{24}$$

$$\mathbf{k}_{2,11}(n) = \frac{\mathbf{P}_{2,11}(n-1)\mathbf{x}_{2,11}(n)}{\lambda_{12} + \mathbf{x}_{2,11}^T(n)\mathbf{P}_{2,11}(n-1)\mathbf{x}_{2,11}(n)}, \tag{25}$$

$$\mathbf{k}_{2,12}(n) = \frac{\mathbf{P}_{2,12}(n-1)\mathbf{x}_{2,12}(n)}{\lambda_{11} + \mathbf{x}_{2,12}^T(n)\mathbf{P}_{2,12}(n-1)\mathbf{x}_{2,12}(n)}. \tag{26}$$

Finally, the filters' updates result by developing the normal equations (12)–(14). For example, sequentially using (16), (21), and (6) in (12), while considering the available solution from the previous time index, i.e., $\widehat{\underline{\mathbf{h}}}_2(n-1) = \mathbf{P}_{12,11}(n-1)\mathbf{r}_{12,11}(n-1)$, we can develop

$$\begin{aligned}\widehat{\underline{\mathbf{h}}}_2(n) &= \mathbf{P}_{12,11}(n)\mathbf{r}_{12,11}(n) = \lambda_2^{-1} \left[\mathbf{I}_{L_2^2} - \mathbf{k}_{12,11}(n)\mathbf{x}_{12,11}^T(n) \right] \mathbf{P}_{12,11}(n-1) \left[\lambda_2 \mathbf{r}_{12,11}(n-1) + \mathbf{x}_{12,11}(n)d(n) \right] \\ &= \widehat{\underline{\mathbf{h}}}_2(n-1) + \mathbf{k}_{12,11}(n) \left[d(n) - \mathbf{x}_{12,11}^T(n)\widehat{\underline{\mathbf{h}}}_2(n-1) \right] = \widehat{\underline{\mathbf{h}}}_2(n-1) + \mathbf{k}_{12,11}(n)e(n). \end{aligned} \tag{27}$$

In a similar manner, the updates for the other two component filters result by using (18), (22), and (7) in (13), and then (20), (23), and (8) in (14), which finally lead to

$$\widehat{\underline{\mathbf{h}}}_{12}(n) = \widehat{\underline{\mathbf{h}}}_{12}(n-1) + \mathbf{k}_{2,11}(n)e(n), \tag{28}$$

$$\widehat{\underline{\mathbf{h}}}_{11}(n) = \widehat{\underline{\mathbf{h}}}_{11}(n-1) + \mathbf{k}_{2,12}(n)e(n). \tag{29}$$

The resulting RLS algorithm based on third-order tensor (TOT) decomposition, namely RLS-TOT, is summarized in Table 1. The specific data structures used within the algorithm are constructed as

Table 1 Recursive Least-Squares Algorithm Based on a Third-Order Tensor Decomposition (RLS-TOT).

Data:

$\mathbf{x}(n),\ d(n)$

Decomposition setup:

$L = L_{11}L_{12}L_2,$ with $L_{12} \leq L_{11},\ L_2 \ll L_{11}L_{12},$ and $P < L_{12}$

Parameters:

forgetting factors: $0 \ll \lambda_2 \leq 1,\ 0 \ll \lambda_{12} < 1,\ 0 \ll \lambda_{11} < 1$

Initialization:

Initialize $\widehat{\underline{\mathbf{h}}}_2(0),\ \widehat{\underline{\underline{\mathbf{h}}}}_{12}(0),$ and $\widehat{\underline{\underline{\mathbf{h}}}}_{11}(0)$ as shown in Table 2

$\mathbf{P}_{12,11}(0) = \delta^{-1}\mathbf{I}_{L_2^2},\ \mathbf{P}_{2,11}(0) = \delta^{-1}\mathbf{I}_{PL_{12}L_2},\ \mathbf{P}_{2,12}(0) = \delta^{-1}\mathbf{I}_{PL_{11}L_2},\ \delta > 0$

For discrete-time index $n = 1, 2, \ldots$

Construct $\widehat{\underline{\underline{\mathbf{H}}}}_{12,11}(n-1),\ \widehat{\underline{\underline{\mathbf{H}}}}_{2,11}(n-1),$ and $\widehat{\underline{\underline{\mathbf{H}}}}_{2,12}(n-1)$ as shown in Table 2

$$\mathbf{x}_{12,11}(n) = \widehat{\underline{\underline{\mathbf{H}}}}_{12,11}^{T}(n-1)\mathbf{x}(n)$$

$$\mathbf{x}_{2,11}(n) = \widehat{\underline{\underline{\mathbf{H}}}}_{2,11}^{T}(n-1)\mathbf{x}(n)$$

$$\mathbf{x}_{2,12}(n) = \widehat{\underline{\underline{\mathbf{H}}}}_{2,12}^{T}(n-1)\mathbf{x}(n)$$

$$\begin{aligned} e(n) &= d(n) - \widehat{\underline{\mathbf{h}}}_2^{T}(n-1)\mathbf{x}_{12,11}(n) \\ &= d(n) - \widehat{\underline{\underline{\mathbf{h}}}}_{12}^{T}(n-1)\mathbf{x}_{2,11}(n) \\ &= d(n) - \widehat{\underline{\underline{\mathbf{h}}}}_{11}^{T}(n-1)\mathbf{x}_{2,12}(n) \end{aligned}$$

$$\mathbf{k}_{12,11}(n) = \frac{\mathbf{P}_{12,11}(n-1)\mathbf{x}_{12,11}(n)}{\lambda_2 + \mathbf{x}_{12,11}^{T}(n)\mathbf{P}_{12,11}(n-1)\mathbf{x}_{12,11}(n)}$$

$$\mathbf{k}_{2,11}(n) = \frac{\mathbf{P}_{2,11}(n-1)\mathbf{x}_{2,11}(n)}{\lambda_{12} + \mathbf{x}_{2,11}^{T}(n)\mathbf{P}_{2,11}(n-1)\mathbf{x}_{2,11}(n)}$$

$$\mathbf{k}_{2,12}(n) = \frac{\mathbf{P}_{2,12}(n-1)\mathbf{x}_{2,12}(n)}{\lambda_{11} + \mathbf{x}_{2,12}^{T}(n)\mathbf{P}_{2,12}(n-1)\mathbf{x}_{2,12}(n)}$$

$$\mathbf{P}_{12,11}(n) = \lambda_2^{-1}\left[\mathbf{I}_{L_2^2} - \mathbf{k}_{12,11}(n)\mathbf{x}_{12,11}^{T}(n)\right]\mathbf{P}_{12,11}(n-1)$$

$$\mathbf{P}_{2,11}(n) = \lambda_{12}^{-1}\left[\mathbf{I}_{PL_{12}L_2} - \mathbf{k}_{2,11}(n)\mathbf{x}_{2,11}^{T}(n)\right]\mathbf{P}_{2,11}(n-1)$$

$$\mathbf{P}_{2,12}(n) = \lambda_{11}^{-1}\left[\mathbf{I}_{PL_{11}L_2} - \mathbf{k}_{2,12}(n)\mathbf{x}_{2,12}^{T}(n)\right]\mathbf{P}_{2,12}(n-1)$$

$$\widehat{\underline{\mathbf{h}}}_2(n) = \widehat{\underline{\mathbf{h}}}_2(n-1) + \mathbf{k}_{12,11}(n)e(n)$$

$$\widehat{\underline{\underline{\mathbf{h}}}}_{12}(n) = \widehat{\underline{\underline{\mathbf{h}}}}_{12}(n-1) + \mathbf{k}_{2,11}(n)e(n)$$

$$\widehat{\underline{\underline{\mathbf{h}}}}_{11}(n) = \widehat{\underline{\underline{\mathbf{h}}}}_{11}(n-1) + \mathbf{k}_{2,12}(n)e(n)$$

Decompose $\widehat{\underline{\mathbf{h}}}_2(n),\ \widehat{\underline{\underline{\mathbf{h}}}}_{12}(n),$ and $\widehat{\underline{\underline{\mathbf{h}}}}_{11}(n)$ as shown in Table 2

shown in Table 2. It is known that the computational complexity order of the RLS-based algorithms is proportional to the square of the filter length [16]. Therefore, as compared to the conventional RLS algorithm [i.e., $\mathcal{O}(L^2) = \mathcal{O}(L_{11}L_{12}L_2)^2$], the proposed RLS-TOT has a lower computational complexity order, i.e., $\mathcal{O}\left[L_2^4 + (PL_{11}L_2)^2 + (PL_{12}L_2)^2\right]$, for the common setup of the decomposition parameters. This is supported in Fig. 1, where the complexity orders of the conventional RLS algorithm and RLS-TOT (with different decomposition setups) are depicted, considering a long length impulse response with $L = 2048$. Since the value of P is usually significantly smaller than L_{12} (as also supported in simulations), while $L_{12} \ll L$, the overall complexity of the RLS-TOT can be much lower as compared to its conventional counterpart.

Table 2 Data Structures for the RLS-TOT Algorithm.

Initialization: ($\mathbf{0}_{L_\bullet}$ generally denotes an all-zeros column vector of length $L_\bullet$)

$\widehat{\mathbf{h}}_2^l(0) = \left[\begin{array}{cc} \epsilon & \mathbf{0}_{L_2-1}^T \end{array} \right]^T,\ p = 1, 2, \ldots, L_2,\ 0 < \epsilon \leq 1$

$\underline{\widehat{\mathbf{h}}}_2(0) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_2^1\right)^T(0) & \left(\widehat{\mathbf{h}}_2^2\right)^T(0) & \cdots & \left(\widehat{\mathbf{h}}_2^{L_2}\right)^T(0) \end{array} \right]^T$

$\widehat{\mathbf{h}}_{12}^{lp}(0) = \left[\begin{array}{cc} \epsilon & \mathbf{0}_{L_{12}-1}^T \end{array} \right]^T,\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P,\ 0 < \epsilon \leq 1$

$\widehat{\overline{\mathbf{h}}}_{12}^{l}(0) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{12}^{l1}\right)^T(0) & \left(\widehat{\mathbf{h}}_{12}^{l2}\right)^T(0) & \cdots & \left(\widehat{\mathbf{h}}_{12}^{lP}\right)^T(0) \end{array} \right]^T,\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{h}}}}_{12}(0) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{12}^{1}\right)^T(0) & \left(\widehat{\overline{\mathbf{h}}}_{12}^{2}\right)^T(0) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{12}^{L_2}\right)^T(0) \end{array} \right]^T$

$\widehat{\mathbf{h}}_{11}^{lp}(0) = \left[\begin{array}{cc} \epsilon & \mathbf{0}_{L_{11}-1}^T \end{array} \right]^T,\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P,\ 0 < \epsilon \leq 1$

$\widehat{\overline{\mathbf{h}}}_{11}^{l}(0) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{11}^{l1}\right)^T(0) & \left(\widehat{\mathbf{h}}_{11}^{l2}\right)^T(0) & \cdots & \left(\widehat{\mathbf{h}}_{11}^{lP}\right)^T(0) \end{array} \right]^T,\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{h}}}}_{11}(0) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{11}^{1}\right)^T(0) & \left(\widehat{\overline{\mathbf{h}}}_{11}^{2}\right)^T(0) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{11}^{L_2}\right)^T(0) \end{array} \right]^T$

For discrete-time index $n = 1, 2, \ldots$

$\widehat{\mathbf{H}}_{12,11}^{lp}(n-1) = \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{12}^{lp}(n-1) \otimes \widehat{\mathbf{h}}_{11}^{lp}(n-1),\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P$

$\widehat{\overline{\mathbf{H}}}_{12,11}^{l}(n-1) = \sum_{p=1}^{P} \widehat{\mathbf{H}}_{12,11}^{lp}(n-1),\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{H}}}}_{12,11}(n-1) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{12,11}^{1}(n-1) & \widehat{\overline{\mathbf{H}}}_{12,11}^{2}(n-1) & \cdots & \widehat{\overline{\mathbf{H}}}_{12,11}^{L_2}(n-1) \end{array} \right]$

$\widehat{\mathbf{H}}_{2,11}^{lp}(n-1) = \widehat{\mathbf{h}}_{2}^{l}(n-1) \otimes \mathbf{I}_{L_{12}} \otimes \widehat{\mathbf{h}}_{11}^{lp}(n-1),\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P$

$\widehat{\overline{\mathbf{H}}}_{2,11}^{l}(n-1) = \left[\begin{array}{cccc} \widehat{\mathbf{H}}_{2,11}^{l1}(n-1) & \widehat{\mathbf{H}}_{2,11}^{l2}(n-1) & \cdots & \widehat{\mathbf{H}}_{2,11}^{lP}(n-1) \end{array} \right],\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{H}}}}_{2,11}(n-1) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{2,11}^{1}(n-1) & \widehat{\overline{\mathbf{H}}}_{2,11}^{2}(n-1) & \cdots & \widehat{\overline{\mathbf{H}}}_{2,11}^{L_2}(n-1) \end{array} \right]$

$\widehat{\mathbf{H}}_{2,12}^{lp}(n-1) = \widehat{\mathbf{h}}_{2}^{l}(n-1) \otimes \widehat{\mathbf{h}}_{12}^{lp}(n-1) \otimes \mathbf{I}_{L_{11}},\ l = 1, 2, \ldots, L_2,\ p = 1, 2, \ldots, P$

$\widehat{\overline{\mathbf{H}}}_{2,12}^{l}(n-1) = \left[\begin{array}{cccc} \widehat{\mathbf{H}}_{2,12}^{l1}(n-1) & \widehat{\mathbf{H}}_{2,12}^{l2}(n-1) & \cdots & \widehat{\mathbf{H}}_{2,12}^{lP}(n-1) \end{array} \right],\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{H}}}}_{2,12}(n-1) = \left[\begin{array}{cccc} \widehat{\overline{\mathbf{H}}}_{2,12}^{1}(n-1) & \widehat{\overline{\mathbf{H}}}_{2,12}^{2}(n-1) & \cdots & \widehat{\overline{\mathbf{H}}}_{2,12}^{L_2}(n-1) \end{array} \right]$

Obtain $\underline{\widehat{\mathbf{h}}}_2(n)$, $\underline{\widehat{\overline{\mathbf{h}}}}_{12}(n)$, and $\underline{\widehat{\overline{\mathbf{h}}}}_{11}(n)$ as shown in Table 1

$\underline{\widehat{\mathbf{h}}}_2(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_2^1\right)^T(n) & \left(\widehat{\mathbf{h}}_2^2\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_2^{L_2}\right)^T(n) \end{array} \right]^T$

$\underline{\widehat{\overline{\mathbf{h}}}}_{12}(n) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{12}^{1}\right)^T(n) & \left(\widehat{\overline{\mathbf{h}}}_{12}^{2}\right)^T(n) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{12}^{L_2}\right)^T(n) \end{array} \right]^T$

$\widehat{\overline{\mathbf{h}}}_{12}^{l}(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{12}^{l1}\right)^T(n) & \left(\widehat{\mathbf{h}}_{12}^{l2}\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_{12}^{lP}\right)^T(n) \end{array} \right]^T,\ l = 1, 2, \ldots, L_2$

$\underline{\widehat{\overline{\mathbf{h}}}}_{11}(n) = \left[\begin{array}{cccc} \left(\widehat{\overline{\mathbf{h}}}_{11}^{1}\right)^T(n) & \left(\widehat{\overline{\mathbf{h}}}_{11}^{2}\right)^T(n) & \cdots & \left(\widehat{\overline{\mathbf{h}}}_{11}^{L_2}\right)^T(n) \end{array} \right]^T$

$\widehat{\overline{\mathbf{h}}}_{11}^{l}(n) = \left[\begin{array}{cccc} \left(\widehat{\mathbf{h}}_{11}^{l1}\right)^T(n) & \left(\widehat{\mathbf{h}}_{11}^{l2}\right)^T(n) & \cdots & \left(\widehat{\mathbf{h}}_{11}^{lP}\right)^T(n) \end{array} \right]^T,\ l = 1, 2, \ldots, L_2$

$\widehat{\mathbf{h}}(n) = \sum_{l=1}^{L_2} \sum_{p=1}^{P} \widehat{\mathbf{h}}_2^l(n) \otimes \widehat{\mathbf{h}}_{12}^{lp}(n) \otimes \widehat{\mathbf{h}}_{11}^{lp}(n)$

4 Simulation Results

The experimental framework considered in this section is echo cancellation, aiming to identify two echo paths ($\mathbf{h}$) with different characteristics, as shown in Fig. 2. The first impulse response used in simulations represents a network echo path and it is obtained from the first cluster of G168 Recommendation [38], which

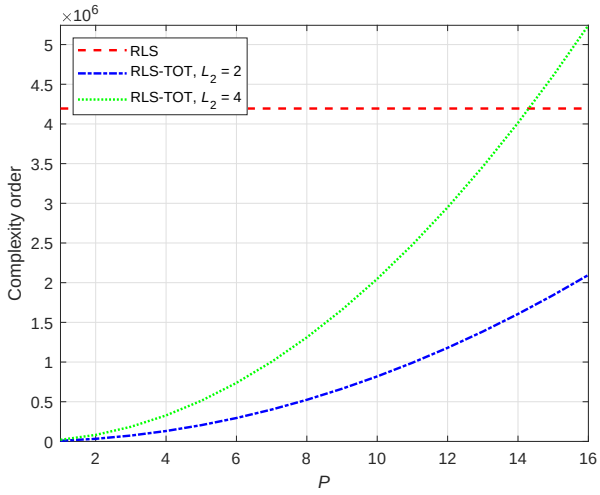

Figure 1: Computational complexity order of the conventional RLS algorithm (with $L = 2048$) and the proposed RLS-TOT using different values of L_2. For $L_2 = 2$, the decomposition setup of the RLS-TOT involves $L_{11} = L_{12} = 32$, while for $L_2 = 4$ we use $L_{11} = 32$ and $L_{12} = 16$.

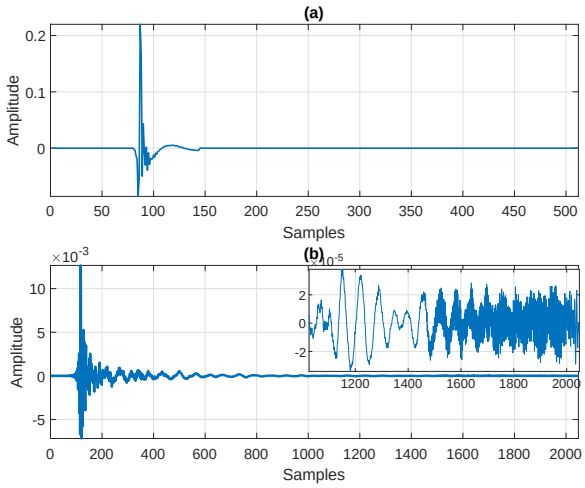

Figure 2: Impulse responses used in simulations, corresponding to (a) a network echo path of length $L = 512$ and (b) a measured acoustic echo path of length $L = 2048$ (a magnified portion of the last $L/2$ coefficients is shown in the top-right of the subfigure).

is padded with zeros up to the length $L = 512$. The second impulse response characterizes a measured acoustic echo path of length $L = 2048$. The input signal $x(n)$ is either a first order autoregressive process, AR(1) [which is obtained by filtering a white Gaussian noise through an AR(1) transfer function with the pole at 0.8] or a speech signal; the sampling rate is 8 kHz. The output of the echo path, $y(n)$, is corrupted by a white Gaussian noise, $w(n)$, with different signal-to-noise ratios (SNRs); a nonstationary background noise is also used in the last experiment. In order to test the tracking capabilities of the algorithms, an abrupt change of the impulse response is considered in most of the experiments, by changing the sign of the coefficients after 2.5 seconds. The performance measure is the normalized misalignment (in dB), which is evaluated as $20\log_{10}\left[\left\|\mathbf{h} - \widehat{\mathbf{h}}(n)\right\| / \|\mathbf{h}\|\right]$, where $\|\cdot\|$ denotes the Euclidean norm.

As indicated in Section 3, we consider that the length of the adaptive filter (L) is equal to the length of the impulse response $\mathbf{h}$. This setup is frequently used in most of the works related to the design of adaptive filtering algorithms, in order to simplify the development and facilitate the notation. In practice, the most challenging scenario corresponds to the under-modeling case, when the length of the adaptive filter is smaller than the length of $\mathbf{h}$. In such a scenario, the tail of the impulse response that cannot

be modeled by the adaptive filter acts like an additional "noise" (i.e., residual echo), thus increasing the misalignment. Due to this reason, the length of the adaptive filter (i.e., the echo canceler) should be large enough to attenuate this effect. Regarding the decomposition of L, it is recommended to factorize it using powers of two (to facilitate the implementation), taking into account that $L_2 \ll L_{11}L_{12}$ and L_{11} should be closer or equal to L_{12}.

The first set of experiments involves the network impulse response of length $L = 512$ [from Fig. 2(a)]. The comparisons are performed using the conventional RLS algorithm as a benchmark, together with the previously developed RLS algorithm using the NKP decomposition (namely RLS-NKP) [36], and the proposed RLS-TOT. The conventional RLS algorithm uses a single adaptive filter of length $L = 512$. Since the RLS-TOT is developed following the line of the conventional RLS algorithm (as shown in Section 3), we do not include in comparisons other fast RLS algorithms. In general, these fast RLS versions [with a complexity order of $\mathcal{O}(L)$] usually target the same performance as the conventional RLS algorithm, but using a lower computational complexity. Moreover, they could require some additional parameters, besides the forgetting factors [17]. In addition, as shown in [36], the RLS-NKP outperforms both the conventional RLS algorithm and its fast version, i.e., the RLS using dichotomous coordinate descent (DCD) iterations, namely RLS-DCD [39]. Several results from [36] also indicate that the conventional RLS algorithm and the RLS-DCD version performs very similarly. Due to these reasons, in the first set of experiments, we limit our comparisons to the conventional RLS benchmark and the recently developed RLS-NKP version [36].

The RLS-NKP algorithm is based on a second-order decomposition of the global impulse response and combines the estimates provided by two adaptive filters of lengths $P^*L_1^*$ and $P^*L_2^*$, with $L = L_1^*L_2^*$ and $P^* < L_2^*$ [36]; in this set of experiments, its decomposition setup uses $L_1^* = 32$ and $L_2^* = 16$. As explained in Section 3, the proposed RLS-TOT algorithm involves three adaptive filters of lengths $PL_{11}L_2$, $PL_{12}L_2$, and L_2^2, where $L = L_{11}L_{12}L_2$, with $L_2 \ll L_{11}L_{12}$ and $P < L_{12}$. As shown in [2], the value of L_2 should be very small, e.g., 2, 3, or 4, while the performance differences are minor. Consequently, in the following experiments that involve the network impulse response of length $L = 512$, we set $L_{11} = L_{12} = 16$ and $L_2 = 2$.

The main parameters of the RLS-based algorithms used in comparisons are the forgetting factors. While the conventional RLS algorithm involves a single forgetting factor (denoted by λ), the RLS-NKP algorithm [36] uses two forgetting factors, λ_1^* and λ_2^*, which correspond to the two adaptive filters. Accordingly, the proposed RLS-TOT algorithm requires three forgetting factors, i.e., λ_{11}, λ_{12}, and λ_2. Choosing the value of a forgetting factor involves a compromise between the main performance criteria, i.e., fast convergence/tracking and low misalignment. In general, the value of a forgetting factor $\lambda_\bullet$ can be related to the associated filter length (generally denoted by $L_\bullet$), according to the relation [36]:

$$\lambda_\bullet = 1 - (KL_\bullet)^{-1}, \tag{30}$$

with $K > 1$. In other words, the longer the filter, the larger the value of the forgetting factor should be used. A higher value of K leads to $\lambda_\bullet$ closer to one, which improves the accuracy of the solution (i.e., lower misalignment), but sacrifices the tracking behavior. We should also outline that the initial convergence rate is not always relevant for the RLS-based algorithms, while the tracking capability represents the true assessment [16], [17]. Relation (30) is not limited to the decomposition-based algorithms that involve low-rank approximations, but it can also be applied to the general case, e.g., for setting the values of the forgetting factor for the conventional RLS algorithm [i.e., $\lambda = 1 - (KL)^{-1}$], which updates a single (long length) adaptive filter with L coefficients. In some of the following experiments, by varying the value of K, we basically adjust the forgetting factors for detailed comparisons of the algorithms. For example, by matching the same convergence rate/tracking of the algorithms, we can evaluate and distinguish their performance in terms of misalignment (i.e., accuracy). Also, when reaching a similar misalignment level (i.e., a similar accuracy), the differences between the algorithms can be assessed in terms of the tracking capability.

In the first experiment reported in Fig. 3, all the algorithms use $K = 30$ in (30), for their associated forgetting factors. The input signal is an AR(1) process and SNR = 20 dB. The decomposition-based algorithms, i.e., RLS-NKP and RLS-TOT, use different values of their specific parameters, P^* and P, respectively. It can be noticed that both these algorithms track much faster as compared to the conventional

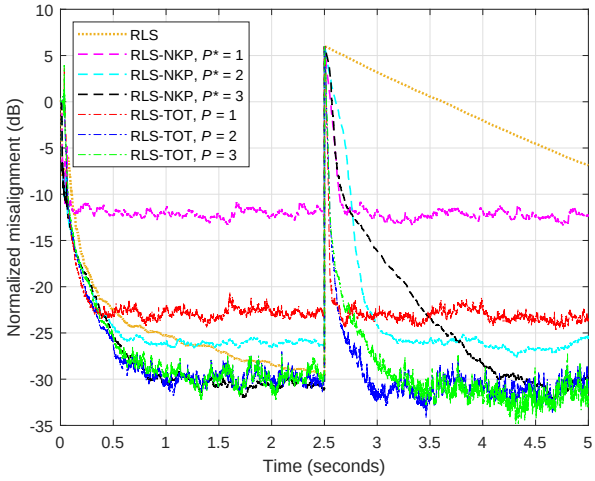

Figure 3: Normalized misalignment of the conventional RLS algorithm, RLS-NKP [36], and proposed RLS-TOT, for the identification of a network impulse response of length $L = 512$ [from Fig. 2(a)]. The forgetting factors are set based on (30), using $K = 30$. The RLS-NKP uses $L_1^* = 32$, $L_2^* = 16$, and different values of P^*. The proposed RLS-TOT uses $L_{11} = L_{12} = 16$, $L_2 = 2$, and different values of P. The input signal is an AR(1) process, SNR = 20 dB, and the echo path changes after 2.5 seconds.

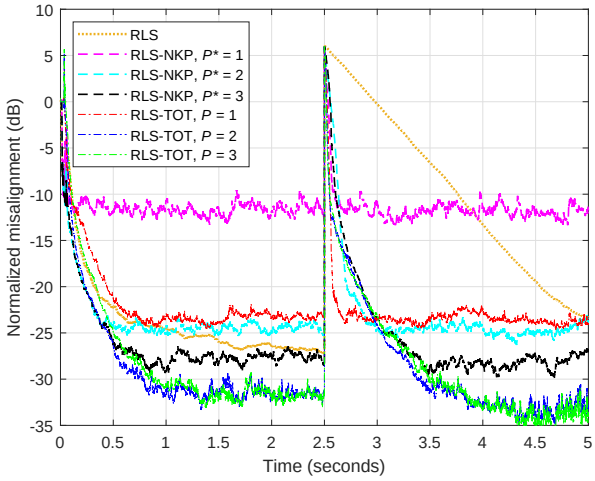

Figure 4: Normalized misalignment of the conventional RLS algorithm, RLS-NKP [36], and proposed RLS-TOT, for the identification of a network impulse response of length $L = 512$ [from Fig. 2(a)]. The forgetting factors are set based on (30), using $K = 10$ for the conventional RLS algorithm, $K = 15$ for the RLS-NKP, and $K = 45$ for the RLS-TOT. The other conditions are the same as in Fig. 3.

RLS benchmark, for a similar misalignment level (which is reached before the change of the system). Among them, the proposed RLS-TOT outperforms the previously developed RLS-NKP in terms of both convergence/tracking capability and misalignment level, in most cases when $P^* = P$. The exception corresponds to $P^* = P = 3$, when the decomposition-based algorithms reach the same misalignment level, but the RLS-TOT tracks faster as compared to the RLS-NKP. In fact, this represents a performance limit (in terms of accuracy/misalignment), since the rank of the corresponding matrix of $\mathbf{h}$ [from Fig. 2(a)] is 3 [36]. Increasing the values of the decomposition parameters beyond this limit will not reduce the misalignment level. Overall, this experiment indicates that the RLS-TOT can use a lower value of the decomposition parameter (P), as compared to its RLS-NKP counterpart (P^*). Even for $P = 1$, the RLS-TOT provides a good attenuation of the misalignment, with a very fast tracking behavior.

The good tracking capability of the RLS-TOT algorithm suggests that we could further increase the

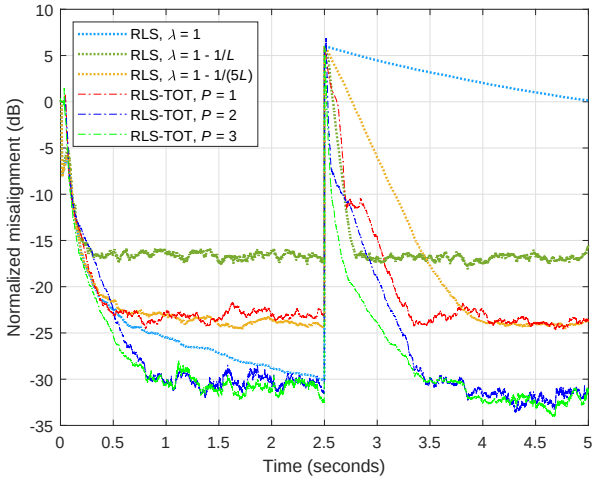

Figure 5: Normalized misalignment of the conventional RLS algorithm (using different values of the forgetting factor λ) and proposed RLS-TOT, for the identification of a network impulse response of length $L = 512$ [from Fig. 2(a)]. The RLS-TOT uses λ_{11} and λ_{12} set based on (30), with $K = 30$, while $\lambda_2 = 1$. The other conditions are the same as in Fig. 3.

values of its forgetting factors, aiming to improve the attenuation of the misalignment, while slightly sacrificing in terms of tracking. On the other hand, the forgetting factors of the conventional RLS algorithm and the RLS-NKP should be increased in order to improve their tracking behavior, while paying in terms of accuracy (i.e., higher misalignment). Such an approach is considered in Fig. 4, where the RLS-TOT uses $K = 45$ in (30) for computing its forgetting factors, while the conventional RLS algorithm and the RLS-NKP use $K = 10$ and $K = 15$, respectively. The other conditions are the same as in the previous experiment. As we can notice, the tracking capability of the RLS-NKP is now close to the RLS-TOT, for $P^* = P$. However, the proposed RLS-TOT outperforms its RLS-NKP counterpart in terms accuracy, reaching lower misalignment levels. Even if the tracking capability of the conventional RLS algorithm has been improved when using a smaller forgetting factor (i.e., a smaller value of K), its tracking reaction is still significantly slower as compared to the decomposition-based algorithms (for similar misalignment levels).

An interesting strategy that could be considered in case of the proposed RLS-TOT is to set the maximum value of the forgetting factor for the shortest filter (of length L_2^2), i.e., $\lambda_2 = 1$. Since this filter has only a few coefficients, this would not impact too much the tracking behavior of the global filter, while still improving its misalignment. This is supported in Fig. 5, where the RLS-TOT is compared to the conventional RLS algorithm using different values of the forgetting factor. Based on the results of this experiment, several conclusions can be outlined, as follows. First, the conventional RLS algorithm using its maximum forgetting factor (i.e., $\lambda = 1$) could reach a misalignment level similar to the RLS-TOT using $P = 2$ or 3, but it is significantly slower in tracking the abrupt change of the system. The value of λ should be significantly increased (e.g., $\lambda = 1 - 1/L$) for the conventional RLS algorithm, in order to match the tracking reaction of the proposed RLS-TOT. However, the misalignment of the conventional benchmark is significantly higher in this case. Finally, the conventional RLS algorithm using a larger value of the forgetting factor [e.g., $\lambda = 1 - 1/(5L)$] can reach the misalignment level of the RLS-TOT using $P = 1$, but it is still outperformed by the proposed algorithm in terms of tracking. Moreover, for $P = 1$, the RLS-TOT uses a combination of three filters of lengths 32, 32, and 4, while the conventional RLS algorithm involves a single adaptive filter with 512 coefficients. Thus, the computational amount of the RLS-TOT is far more advantageous.

In the last simulation from the first set of experiments, we consider a noisy scenario by increasing the level of the additive noise, such that SNR = 10 dB. The forgetting factors of the algorithms are set based on (30), using $K = 5$ for the conventional RLS algorithm and $K = 45$ for the decomposition-based versions, i.e., RLS-NKP and RLS-TOT. The other conditions are the same as in Fig. 3. As we can notice

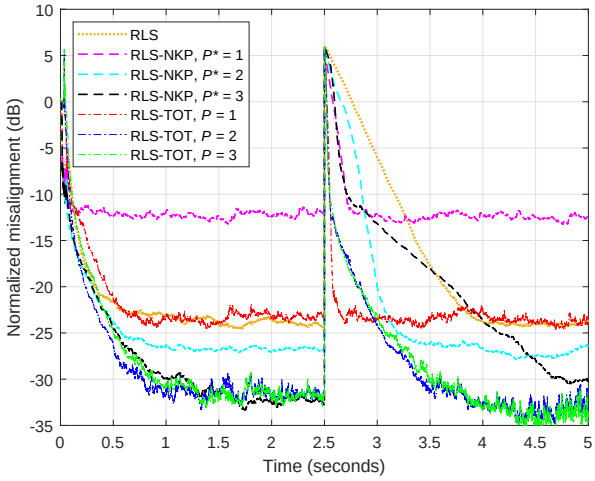

Figure 6: Normalized misalignment of the conventional RLS algorithm, RLS-NKP [36], and proposed RLS-TOT, for the identification of a network impulse response of length $L = 512$ [from Fig. 2(a)], when SNR = 10 dB. The forgetting factors are set based on (30), using $K = 5$ for the conventional RLS algorithm, and $K = 45$ for the RLS-NKP and RLS-TOT. The other conditions are the same as in Fig. 3.

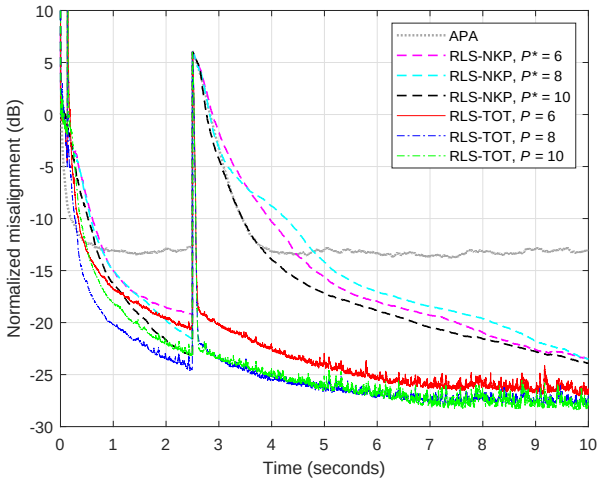

Figure 7: Normalized misalignment of the APA [40], RLS-NKP [36], and proposed RLS-TOT, for the identification of an acoustic impulse response of length $L = 2048$ [from Fig. 2(b)]. The projection order of APA is $M = 8$ and its step-size parameter is $\mu = 1$. The forgetting factors of the RLS-NKP and RLS-TOT are set based on (30), using $K = 50$. The RLS-NKP uses $L_1^* = 64$, $L_2^* = 32$, and different values of P^*. The proposed RLS-TOT uses $L_{11} = L_{12} = 32$, $L_2 = 2$, and different values of P. The input signal is an AR(1) process, SNR = 20 dB, and the echo path changes after 2.5 seconds.

from Fig. 6, the performance gain (in terms of both misalignment and tracking) is more apparent for the proposed RLS-TOT, as compared to the RLS-NKP counterpart. This supports the conclusions from [2] (obtained in the framework of the Wiener filter), which indicate that the TOT-based decomposition is more efficient in challenging scenarios, e.g., in low SNR environments.

The second set of experiments is dedicated to the identification of an acoustic impulse response of length $L = 2048$ [from Fig. 2(b)]. In this case, the decomposition setup of the RLS-NKP [36] considers $L_1^* = 64$ and $L_2^* = 32$, while the proposed RLS-TOT uses $L_{11} = L_{12} = 32$ and $L_2 = 2$. Due to its high computational complexity, the conventional RLS algorithm is prohibitive in such scenarios that involve very long length impulse responses. Therefore, for the purpose of comparisons, other more practical algorithms will be indicated and involved in the following simulations.

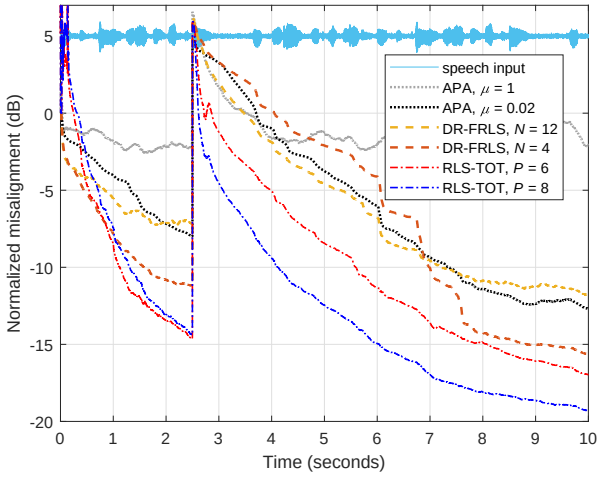

Figure 8: Normalized misalignment of the APA [40], DR-FRLS algorithm [41], and proposed RLS-TOT (using different values of P), for the identification of an acoustic impulse response of length $L = 2048$ [from Fig. 2(b)], when SNR = 10 dB. The projection order of APA is $M = 8$ and different values of the step-size parameter are used. The forgetting factor of the DR-FRLS is $\lambda = 1$ and different values of the data-reuse parameter N are considered. The RLS-TOT uses λ_{11} and λ_{12} set based on (30), with $K = 100$, while $\lambda_2 = 1$. The decomposition setup of the RLS-TOT is the same from Fig. 7. The input signal is speech and the echo path changes after 2.5 seconds.

In the first experiment from this set, we use the affine projection algorithm (APA) [40] as a benchmark, which is a very popular choice in the framework of acoustic echo cancellation. The main reason is that it outperforms the well-known normalized least-mean-square (NLMS) algorithm [16], especially for highly correlated input signals. Also, the APA is less computationally expensive as compared to the conventional RLS algorithm. The main parameters of APA are the step-size (denoted by μ; this is a positive constant smaller than or equal to one) and the projection order (referred as M; for $M = 1$, the APA reduces to the NLMS algorithm). Higher values of μ and M improve the convergence rate and tracking of the algorithm, but also increase the misalignment. In our experiment, we set $\mu = 1$ (i.e., the fastest convergence mode) and $M = 8$. The input signal is an AR(1) process and SNR = 20 dB. The comparing algorithms are RLS-NKP [36] and the proposed RLS-TOT, with different values of the decomposition parameters (P^* and P, respectively). Since the acoustic echo paths are closer to full-rank [25], higher values of P^* and P should be used, as compared to the previous scenario that involved the network impulse response; however, these values are still significantly smaller than L_2^* and L_{12}, respectively. The results are presented in Fig. 7, where we can firstly notice that the tracking capability of the RLS-TOT is clearly better as compared to both RLS-NKP and APA. Besides, both RLS-based algorithms reach significantly lower misalignment levels than the APA. Finally, it can be noticed that increasing the value of P beyond a certain limit (e.g., $P = 8$ in Fig. 7) does not improve the overall performance; this threshold is in the vicinity of $L_{12}/4$.

Another interesting solution that can be used as a low-complexity alternative to the conventional RLS is the recently developed data-reuse fast RLS (DR-FRLS) algorithm [41], with a complexity order of $\mathcal{O}(L)$. Due to its data-reuse features, this algorithm can operate with the maximum value of the forgetting factor (i.e., $\lambda = 1$), while tuning its tracking capability based on the data-reuse parameter, denoted by N (i.e., the number of data-reuse iterations). In addition, using $\lambda = 1$ basically overcomes the main stability issues related to the fast versions of the RLS algorithm, which usually lead to divergence, stalling, etc., especially for highly correlated and nonstationary input signals [16]. In the experiment reported in Fig. 8, we consider such a challenging scenario, by using a noisy speech signal as input, while also operating in a noisy environment with SNR = 10 dB. The other algorithms involved in the comparisons are the APA with $M = 8$ and different step-size parameters, and the proposed RLS-TOT using different values of P. Two forgetting factors of the RLS-TOT are set based on (30), i.e., λ_{11} and λ_{12}, while $\lambda_2 = 1$. Using larger values of the forgetting factors is recommended for the RLS-based algorithms, when dealing with

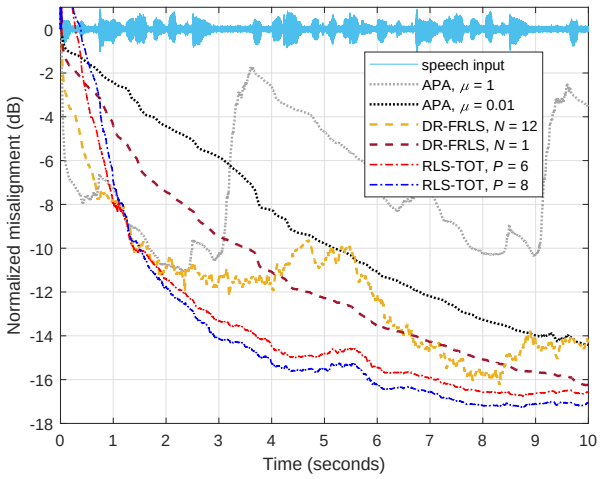

Figure 9: Normalized misalignment of the APA [40], DR-FRLS algorithm [41], and proposed RLS-TOT (using different values of P), for the identification of an acoustic impulse response of length $L = 2048$ [from Fig. 2(b)] that evolves based on a random walk model. The background noise is a recorded highway noise and the input signal is a noisy speech sequence. The projection order of APA is $M = 8$ and different values of the step-size parameter are used. The forgetting factor of the DR-FRLS is $\lambda = 1$ and different values of the data-reuse parameter N are considered. The RLS-TOT uses λ_{11} and λ_{12} set based on (30), with $K = 100$, while $\lambda_2 = 1$. The decomposition setup of the RLS-TOT is the same from Fig. 7.

nonstationary input signals like speech. As we can notice, the RLS-TOT outperforms both APA and the DR-FRLS algorithm, in terms of convergence rate/tracking and misalignment level, even for a small value of the decomposition parameter (e.g., $P = 6$, which is less than $L_{12}/5$). Reducing the step-size parameter of APA decreases the misalignment level, but also the tracking capability. The DR-FRLS algorithm, using a larger value of the date-reuse parameter N, improves the tracking, but sacrifices the accuracy of its estimate (i.e., increasing the misalignment). In conclusion, the proposed RLS-TOT achieves a more proper compromise between the main performance criteria.

In all the previous experiments, the tracking capabilities of the algorithms were tested by considering an abrupt change of the impulse response, which represents a challenging scenario for the adaptive filter. However, in the framework of acoustic echo cancellation, the impulse response could also slowly vary in time due to small disturbances or natural changes in the environment. These include potential variations of temperature, pressure, and humidity of the air [18]. In this context, a random walk model could be used to emulate these changes. Consequently, in the final experiment, we consider that the acoustic impulse response evolves according to such a model, i.e., $\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{q}(n)$, where $\mathbf{h}(0)$ is the echo path from Fig. 2(b) and $\mathbf{q}(n)$ contains the samples of a zero-mean white Gaussian, with variance $\sigma^2_{\mathbf{q}}$. In our scenario, we set $\sigma_{\mathbf{q}} = \varepsilon \, \|\mathbf{h}(0)\|$, using $\varepsilon = 10^{-5}$. In addition, the background noise could be nonstationary and it is also amplified by the microphone of a hands-free device. Therefore, we use a recorded highway noise, similar to a scenario in which there is an open window in the near-end room, which is located close to a highway. The results of this experiment are shown in Fig. 9, using the comparing algorithms from the previous simulation. The DR-FRLS algorithm using $N = 1$ is equivalent to the conventional fast RLS (FRLS) algorithm [15], [16]. We can notice that the proposed RLS-TOT is more robust under this scenario, as compared to the other algorithms. Forcing a fast convergence rate is not a feasible approach under these nonstationary conditions, where the robustness of the algorithm should be the main target. For example, this reflects in the large fluctuations of APA using $\mu = 1$ (i.e., the fastest convergence mode). A similar behavior can be noticed when increasing the data-reuse parameter of the DR-FRLS algorithm (e.g., $N = 12$), targeting a higher convergence rate and tracking, while sacrificing the robustness. Overall, the RLS-TOT achieves a better performance compromise among the comparing algorithms, in terms of the convergence/tracking, misalignment, and robustness features.

5 Discussion

The proposed RLS-TOT algorithm is based on a third-order decomposition of the impulse response and combines (via the Kronecker product) the estimates provided by three shorter adaptive filters. As compared to the previously developed iterative Wiener filter from [2], the RLS-TOT does not require any statistics' estimates and is able to operate in time-variant environments. Besides, the three component adaptive filters of the RLS-TOT are updated in parallel, contrary to the iterative Wiener filter that sequentially alternates its solutions related to the three sets of Wiener-Hopf equations. This parallelism represents an advantage in terms of the modularity of implementation. Since the overall performance of any adaptive filter is influenced by its length, the proposed RLS-based version owns several specific advantages, as follows.

First, as compared to the conventional RLS algorithm, the RLS-TOT has a much lower computational complexity for the common setup of its decomposition parameters. While the complexity order of the conventional RLS is proportional to $\mathcal{O}(L^2)$, the RLS-TOT requires $\mathcal{O}\left[L_2^4 + (PL_{11}L_2)^2 + (PL_{12}L_2)^2\right]$ operations, where $L = L_{11}L_{12}L_2$, with $L_2 \ll L_{11}L_{12}$ and $P < L_{12}$. For example, in echo cancellation scenarios, L can reach the order of thousands. Consequently, such a decomposition setup could lead to significant computational reductions, since L_{11} and L_{12} are on the order of tens, L_2 is limited to small values (e.g., 2 or 3), while $P \ll L_{12}$ in most of the scenarios.

Second, the shorter the adaptive filter, the faster its convergence rate and tracking. Therefore, since the global estimate of the RLS-TOT results based on a combination of the coefficients provided by three shorter adaptive filters, it clearly outperforms the tracking capability of the conventional algorithms (e.g., RLS, APA, and DR-FRLS [41]), which update a single long length adaptive filter. In addition, the proposed RLS-TOT also outperforms the previously developed RLS-NKP [36], which is based on a second-order decomposition (i.e., it combines the estimates of only two adaptive filters).

Third, there is always a compromise between the convergence rate/tracking and accuracy (i.e., misalignment) of the solution, in terms of setting the parameters of the adaptive filter (e.g., the forgetting factors of RLS-based algorithms). From this point of view, the proposed RLS-TOT is more robust to this compromise as compared to the comparing algorithms, since its forgetting factors can reach larger values (i.e., closer to one), slightly sacrificing the tracking capability, but improving the accuracy of its estimate. For example, one possible solution is to set $\lambda_2 = 1$ (i.e., the maximum value), which corresponds to the shortest component filter of length L_2^2, while tuning only the other two forgetting factors (λ_{11} and λ_{12}). Besides, using larger values of the forgetting factors is also a benefit in terms of the numerical robustness of the RLS-based algorithms [15], [16], especially when dealing with nonstationary input signals like speech.

References

[1] C. Paleologu, J. Benesty, C. L. Stanciu, J. R. Jensen, M. G. Christensen, and S. Ciochină, "Recursive least-squares algorithm based on a third-order tensor decomposition for low-rank system identification," *Signal Processing*, vol. 213, id. 109216, Dec. 2023 (10 pages).

[2] J. Benesty, C. Paleologu, and S. Ciochină, "Linear system identification based on a third-order tensor decomposition," *IEEE Signal Processing Lett.*, vol. 30, pp. 503–507, May 2023.

[3] I. D. Fîciu, C. L. Stanciu, C. Paleologu, and J. Benesty, "Low-complexity data-reuse RLS algorithm for stereophonic acoustic echo cancellation," *Applied Sciences*, vol. 13, id. 2227, Feb. 2023 (16 pages).

[4] J. Benesty, C. Paleologu, and L. M. Dogariu, "An iterative Wiener filter based on a fourth-order tensor decomposition," *Symmetry*, vol. 15, id. 1560, Aug. 2023 (18 pages).

[5] C. Paleologu, J. Benesty, C. L. Stanciu, R. L. Costea, L. M. Dogariu, and S. Ciochină, "Iterative Wiener filter based on third-order tensor decomposition and coordinate descent method," in *Proc. IEEE International Symposium on Electrical and Electronics Engineering* (ISEEE), 2023 pp. 156–161.

[6] C. L. Stanciu, C. Anghel, and C. Elisei-Iliescu, "Regularized RLS adaptive algorithm with conjugate gradient method," in *Proc. IEEE International Conference on Speech Technology and Human-Computer Dialogue* (SpeD), 2023, pp. 18–23.

[7] B. Moroşanu, M. Negru, A. Neacşu, C. Negrescu, C. Paleologu, and C. Burileanu, "Personalized multi-track leveling algorithm," in *Proc. IEEE International Conference on Speech Technology and Human-Computer Dialogue* (SpeD), 2023, pp. 24–29.

[8] L. M. Dogariu, C. Paleologu, J. Benesty, C. L. Stanciu, and S. Ciochină, "A decomposition-based Kalman filter for the identification of acoustic impulse responses," in *Proc. European Signal Processing Conference* (EUSIPCO), 2023, pp. 351–355.

[9] C. L. Stanciu, C. Paleologu, J. Benesty, R. L. Costea, L. M. Dogariu, and S. Ciochină, "Iterative Wiener filter using a Kronecker product decomposition and the coordinate descent method," in *Proc. IEEE International Symposium on Signals, Circuits and Systems* (ISSCS), 2023 (4 pages)a.

[10] I. D. Fîciu, C. L. Stanciu, C. Elisei-Iliescu, C. Anghel, L. Stanciu, and R. M. Udrea, "Efficient RLS algorithms with line search methods for stereophonic acoustic echo cancellation," in *Proc. IEEE International Symposium on Signals, Circuits and Systems* (ISSCS), 2023 (4 pages).

[11] E. V. Dumitraşcu, R. A. Dobre, and I. D. Fîciu, "Comparative analysis of the RLS algorithm with Kronecker product decomposition for acoustic impulse response identification," in *Proc. International Conference on Telecommunications and Signal Processing* (TSP), 2023, pp. 139–143.

[12] R. A. Dobre, C. Paleologu, J. Benesty, and F. Albu, "On the decomposition parameter of the RLS algorithm based on the nearest Kronecker product," in *Proc. IEEE International Conference on Electronics, Computers and Artificial Intelligence* (ECAI), 2023 (4 pages).

[13] I. D. Fîciu, C. L. Stanciu, C. Paleologu, J. Benesty, C. Elisei-Iliescu, C. Anghel, and S. Ciochină, "Stereophonic acoustic echo cancellation with the RLS algorithm using the conjugate gradient method," in *Proc. International Conference on Advances in Signal Processing and Artificial Intelligence* (ASPAI), 2023, pp. 55–57.

[14] I. D. Fîciu, C. L. Stanciu, C. Paleologu, and F. Albu, "On the performance of a low-complexity data-reuse RLS algorithm for stereophonic acoustic echo cancellation," in *Proc. International Conference on Digital Telecommunications* (ICDT), 2023 (4 pages).

[15] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Fourth Edition, New York, NY: Springer-Verlag, 2013.

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

[17] Y. V. Zakharov and V. H. Nascimento, "DCD-RLS adaptive filters with penalties for sparse identification," *IEEE Trans. Signal Processing*, vol. 61, pp. 3198–3213, June 2013.

[18] E Hänsler and G Schmidt, *Acoustic Echo and Noise Control–A Practical Approach*. Hoboken, NJ: Wiley 2004.

[19] M. Rupp and S. Schwarz, "Gradient-based approaches to learn tensor products," in *Proc. EUSIPCO*, 2015, pp. 2486–2490.

[20] M. Rupp and S. Schwarz, "A tensor LMS algorithm," in *Proc. IEEE ICASSP*, 2015, pp. 3347–3351.

[21] L. N. Ribeiro, A. L. F. de Almeida, and J. C. M. Mota, "Identification of separable systems using trilinear filtering," in *Proc. IEEE CAMSAP*, 2015, pp. 189–192.

[22] L. N. Ribeiro, S. Schwarz, M. Rupp, A. L. F. de Almeida, and J. C. M. Mota, "A low-complexity equalizer for massive MIMO systems based on array separability," in *Proc. EUSIPCO*, 2017, pp. 2522–2526.

[23] M. N. da Costa, G. Favier, and J. M. T. Romano, "Tensor modelling of MIMO communication systems with performance analysis and Kronecker receivers," *Signal Processing*, vol. 145, pp. 304–316, Apr. 2018.

[24] L. N. Ribeiro, A. L. F. de Almeida, and J. C. M. Mota, "Separable linearly constrained minimum variance beamformers," *Signal Processing*, vol. 158, pp. 15–25, May 2019.

[25] L.-M. Dogariu, J. Benesty, C. Paleologu, and S. Ciochină, "Identification of room acoustic impulse responses via Kronecker product decompositions," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 30, pp. 2828–2841, Sept. 2022.

[26] S. S. Bhattacharjee and N. V. George, "Fast and efficient acoustic feedback cancellation based on low rank approximation," *Signal Processing*, vol. 182, id. 107984, May 2021.

[27] S. S. Bhattacharjee and N. V. George, "Nearest Kronecker product decomposition based linear-in-the-parameters nonlinear filters," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 29, pp. 2111–2122, May 2021.

[28] G. Huang, J. Benesty, I. Cohen, and J. Chen, "Kronecker product multichannel linear filtering for adaptive weighted prediction error-based speech dereverberation," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 30, pp. 1277–1289, Mar. 2022.

[29] S. Vadhvana, S. K. Yadav, S. S. Bhattacharjee, and N. V. George, "An improved constrained LMS algorithm for fast adaptive beamforming based on a low rank approximation," *IEEE Trans. Circuits Systems II: Express Briefs*, vol. 69, pp. 3605–3609, Aug. 2022.

[30] S. S. Bhattacharjee, V. Patel, and N. V. George, "Nonlinear spline adaptive filters based on a low rank approximation," *Signal Processing*, vol. 201, id. 108726, Dec. 2022.

[31] A. Cichocki, D. P. Mandic, A. Phan, C. F. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications," *IEEE Signal Processing Mag.*, pp. 145–163, Mar. 2015.

[32] A. P. da Silva, P. Comon, and A. L. F. de Almeida, "A finite algorithm to compute rank-1 tensor approximations," *IEEE Signal Processing Lett.*, vol. 23, pp. 959–963, July 2016.

[33] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Processing*, vol. 65, pp. 3551–3582, July 2017.

[34] A. Altan and R. Hacioğlu, "Hammerstein model performance of three axes gimbal system on Unmanned Aerial Vehicle (UAV) for route tracking," in *Proc. IEEE SIU*, 2018, 4 pages.

[35] A. Altan and R. Hacioğlu, "Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances," *Mechanical Systems and Signal Processing*, vol. 138, id. 106548, 23 pages, Apr. 2020.

[36] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină, "Recursive least-squares algorithms for the identification of low-rank systems," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 27, pp. 903–918, May 2019.

[37] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
[38] *Digital Network Echo Cancellers*, ITU-T Recommendation G.168, 2012.
[39] Y. V. Zakharov, G. P. White, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Processing*, vol. 56, pp. 3150–3161, July 2008.
[40] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Jpn.*, vol. 67-A, pp. 19–27, May 1984.
[41] C. Paleologu, J. Benesty, and S. Ciochină, "Data-reuse recursive least-squares algorithms," *IEEE Signal Processing Lett.*, vol. 29, pp. 752–756, Mar. 2022.