# Online Block-Term Decomposition with Separable Gauss–Newton Updates for Long Impulse Responses

...

***Abstract*—We propose an adaptive algorithm that leverages higher-order tensor decomposition to identify large room acoustic systems, i.e., long room impulse responses (RIRs), more efficiently. In room-acoustic applications, long RIRs may have tens of thousands of coefficients, which makes their estimation computationally demanding. Based on the observations that RIRs in room acoustics exhibit low-rank structure when reshaped into tensor form, tensor decomposition methods have been proposed as a means to reduce the number of parameters in the estimation problem, where the state-of-the-art method recursive least squares for third-order tensor decomposition (RLS-TOT) is based on third-order Kronecker product tensor decomposition. Here, we make use of the block term decomposition (BTD), a tensor decomposition that is a generalization of the canonical polyadic decomposition (CPD)/Kronecker product decomposition and extend the use of tensor decomposition-based estimation to arbitrary orders. We formulate the convolutive signal model of input-output system identification using tensor algebra forming a BTD filter model, and propose an adaptive algorithm based on alternating higher-order orthogonal iteration (HOOI) and three-operator splitting to solve the resulting mean square error (MSE) optimization problem. We introduce dampened second-order updates using Kronecker decompositions of the Hessian of the full parameter space to achieve efficient updates in the decomposed parameter space. We analyse the computational complexity of the proposed algorithm and compare it to the state-of-the-art method RLS-TOT. Numerical experiments validate that our approach strikes a favourable balance between reducing the number of parameters, computational complexity and maintaining estimation accuracy, making it a promising and flexible tool for complex acoustic system identification tasks.**

***Index Terms*—adaptive filtering, block term decomposition, Kronecker product, low-rank approximation, room acoustics, system identification**

## I. INTRODUCTION

The task of estimating acoustic room impulse responses (RIRs) from observed input-output data is a commonly required problem in the field of room acoustics as RIRs describe the acoustic path from a source to a microphone and comprise the direct path, if existent, reflections from the room boundaries and objects within the room, and are therefore a complete characterization of the room acoustic system. This estimation problem can refer to a static system, where the RIRs are assumed time-invariant, or to a time-varying system, where the RIRs change over time due to changes in source or microphone position or changes of the room itself. Downstream applications of estimated RIRs are numerous in audio signal processing and room acoustics, including enhancement tasks such as dereverberation [1], source separation [2], acoustic feedback and echo cancellation [3] or for room acoustics analysis: the computation of room acoustic parameters, such as the reverberation time $T_{60}$ or speech intelligibility predictors $C_{50}$ [4], room geometry estimation [5], [6] and source localisation [7].

Room acoustic systems are commonly modelled as a linear filters driven by the source. In practice, finite impulse response (FIR) models are standard for room paths, especially in acoustic echo cancellation (AEC)/active noise control (ANC), because they are straightforward to estimate and guaranteed stable, albeit often very long; hence frequency-domain [8] and partitioned implementations [9], [10], [11] are popular to cope with complexity. In contrast, infinite impulse response (IIR) models can give compact representations but risk numerical/robustness issues in identification for high-order systems; FIR-to-IIR conversion methods such as Prony/Stieglitz–McBride [12] illustrate the potential and the stability caveats in practice. These trade-offs underpin the interest in low-rank/tensor models to reduce parameter count without changing the linear convolutional structure. The FIR approach is more common in audio signal processing applications, as it is straightforward to implement and estimate, however, in typical room acoustic scenarios, the RIRs can comprise several thousands of taps, leading to issues with dimensionality in estimation and downstream tasks. There are various existing approaches to reduce dimensionality and computational complexity in FIR-based room acoustic system identification, such as frequency-domain processing [8] or low-rank approximations [13].

Here, we focus on the latter, formulating an estimation problem using tensor decompositions [14], [15], specifically the block term decomposition (BTD) [16], [17], [18], a generalization of the canonical polyadic decomposition (CPD) used in [19], [13], [20] and the (generalized) singular value decomposition (SVD) used in [21] and [15].

In adaptive filtering for acoustic paths, first-order methods (least mean squares (LMS) and variants) and second-order methods like recursive least squares (RLS) are well established; second-order updates improve convergence but scale poorly with filter length. To reduce complexity, decomposition-based approaches factor the long filter into Kronecker-structured components, e.g., third-order tensor models estimated by recursive least squares for third-order tensor decomposition (RLS-TOT) [22], achieving faster convergence and large savings vs. a monolithic RLS. However, this approach is still limited in its scalability to long

This research work was carried out at the ESAT Laboratory of KU Leuven, and has received funding from KU Leuven internal funds (C14/21/075), Research Foundation Flanders (FWO projects S005525N and G0A0424N), the Flemish Government (AI Research Program), and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956369.

RIRs. Our contribution generalizes this idea to arbitrary-order BTD models and replaces exact RLS curvature with Kronecker-preconditioned Levenberg-Marquardt (LM)/Gauss-Newton (GN) updates in the factor and core spaces; this preserves much of the second-order behavior at a fraction of the cost, akin in spirit to Kronecker-factored curvature approximations used in large-scale optimization [23], [24].

We propose an adaptive algorithm based on alternating higher-order orthogonal iteration (HOOI) [14], [16] and three-operator splitting [25] (forward-backward-backward) proximal algorithm [26] that generalizes the two- and three-dimensional tensor decomposition approaches in [22] to arbitrary-order tensor decompositions. We make use of Kronecker product approximations of Hessians in the decomposition's factor and core spaces [24] and use these to precondition the LM-style forward updates in the alternating optimization scheme. This results in efficient approximated second-order updates, which are lighter to compute than the RLS approach in [22]. Expectedly, there is a trade-off between the approximation quality of the second-order updates, with accompanying computational complexity savings, and the convergence speed. However, we show that the proposed method strikes a good balance between these two aspects. We name the proposed method Separable-Curvature Online BTD (SeCO-BTD).

This paper contains the following contributions: First, we introduce the low-rank signal model of a convolutional system using the BTD and formulate the estimation problem. We propose the cost function to comprise a signal term, a recursive Tikhonov regularization to avoid swamp behaviour (flat regions due to decomposition indeterminacy) [27], and a Stiefel manifold constraint to enforce the orthogonality of the factor matrices. Factor orthogonality is not a strict requirement for a feasible decomposition, but it is a common constraint to avoid scaling indeterminacies and to improve the conditioning of the optimization problem. Secondly, we introduce the proposed algorithm to solve the resulting optimization problem, which is based on alternating HOOI and three-operator splitting. We show how the proposed algorithm's computational complexity scales with the length of filters, the number of modes and the rank of the decomposition, and compare it to the RLS-TOT method's complexity. Finally, we validate the proposed algorithm in numerical experiments using simulated room-acoustical environments.

The remainder of this paper is structured as follows: Section II introduces the signal model and formulates the estimation problem using tensor-based notation. Section III introduces the proposed algorithm to solve the estimation problem. Section IV analyses the computational complexity of the proposed algorithm and compares it to the RLS-TOT method. Section V validates the proposed algorithm in numerical experiments and we conclude in Section VI.

## II. Problem Formulation

Let us consider the problem of estimating the impulse response of a room acoustic system from input-output data. First, we introduce the signal model, then formulate the estimation problem using tensor-based notation.

**Notation:** We use lowercase letters for scalar values ($x$), lowercase bold letters for vectors ($\mathbf{x} \in \mathbb{R}^I$), uppercase bold letters for matrices ($\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$), and calligraphic bold letters for $D$-modal tensors ($\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$). Each of which can be discrete-time-dependent, e.g. $x(n)$ or $\boldsymbol{\mathcal{X}}(n)$. More general linear operators are denoted as uppercase letters (e.g., $A : \mathbb{R} \to \mathbb{R}$). Addressing specific elements of a vector or matrix is done by using brackets, e.g. $\mathbf{x}[i]$, $\mathbf{X}[i_1, i_2]$, and $\boldsymbol{\mathcal{X}}[i_1, \ldots, i_D]$ for the $i$-th, the $(i_1, i_2)$-th, and the $(i_1, \ldots, i_D)$-th element of the vector, matrix, and tensor, respectively. The Kronecker product of vectors/matrices/tensors is denoted with $\otimes$. We denote convolution in the time-domain with $*$. The expectation operator is denoted as $\mathbb{E}[\cdot]$.
Of special importance are tensor-specific operations: Let the map between a vector $\mathbf{x}$ and tensor $\boldsymbol{\mathcal{X}}$, with $L = \prod_{d=1}^{D} I_d$, be the bijective reshaping operator $\mathcal{R}_D : \mathbb{R}^L \to \mathbb{R}^{I_1 \times \cdots \times I_D}$ such that $\boldsymbol{\mathcal{X}} = \mathcal{R}_D(\mathbf{x})$ and its inverse operation, which is vectorization, $\mathbf{x} = \mathrm{vec}(\boldsymbol{\mathcal{X}})$. Unfolding the tensor into a matrix $\mathbf{X}_{(d)} \in \mathbb{R}^{I_d \times \prod_{\delta \neq d} I_\delta}$ along mode $d$ is defined as the matrix whose columns are the mode-$d$ fibres of the tensor $\boldsymbol{\mathcal{X}}$. Further, the $d$-mode product of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_d}$ is a matrix-vector product along mode $d$ such that

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_d \mathbf{A} \in \mathbb{R}^{I_1 \times \cdots \times I_{d-1} \times J \times I_{d+1} \times \cdots \times I_D}$$
$$\iff \mathbf{Y}_{(d)} = \mathbf{A}\mathbf{X}_{(d)} \in \mathbb{R}^{J \times \prod_{\delta \neq d} I_\delta}.$$

We denote the multiple-mode product on a tensor with $D \geq E$ modes as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{A}_1 \cdots \times_E \mathbf{A}_E \equiv \boldsymbol{\mathcal{X}} \times_{d \in [E]} \mathbf{A}_d.$$

We denote the multiple-mode product with all modes except $d$ as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_{\delta \neq d} \mathbf{A}_\delta \equiv \boldsymbol{\mathcal{X}} \times_{\delta \in [D] \setminus \{d\}} \mathbf{A}_\delta.$$

For tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_K}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times \cdots \times J_L}$, where $K, L$ are integers $\geq 1$ and index sets $\mathcal{K} \subseteq \{1, \ldots, K\}$ and $\mathcal{L} \subseteq \{1, \ldots, L\}$ with a 1-1 correspondence and matching sizes, write

$$\boldsymbol{\mathcal{Z}} = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{\mathcal{K}|\mathcal{L}}$$

for the tensor obtained by contracting $\boldsymbol{\mathcal{X}}$ along modes in $\mathcal{K}$ and $\boldsymbol{\mathcal{Y}}$ along modes in $\mathcal{L}$. The result keeps the remaining modes of $\boldsymbol{\mathcal{X}}$ in order, then the remaining modes of $\boldsymbol{\mathcal{Y}}$ in order. The inner product of two tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ is the special case of the contracted tensor product with contraction over all modes $1, \ldots, D$ (denoted $1 : D$)

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle \coloneqq \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle_{1:D|1:D} \in \mathbb{R} \tag{1}$$

We use the shorthand $\langle \cdot, \cdot \rangle_{\delta \neq d}$ for contraction of two tensors over all modes except $d$. More detail on these definitions can be found in [14] and [28].
Since we use superscripts to index blocks and factor matrices in tensor decompositions, we denote transpose and inverse operations on the same level, e.g. $\mathbf{A}^{(d)\top}$, $\mathbf{A}^{(d)-1}$, for convenience. The operator $\mathrm{sym}(\cdot)$ symmetrizes a matrix such that $\mathrm{sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^\top)/2$.

### A. Signal Model

Suppose we have an acoustic system that can be modelled by

$$y(n) = x(n) * h(n) + v(n) \tag{2}$$

where $y(n)$ is the system's output, $x(n)$ is the input signal, $h(n)$ is the system's impulse response, $v(n)$ is the noise (assumed independent white Gaussian noise (WGN)). We denote the discrete-time index as $n$. We assume the length of the impulse responses is $L$.

Let the last $L$ samples of the input signal $x(n)$ be collected in a signal vector $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^\top$ and the impulse response $h(n)$ be collected in a vector $\mathbf{h} = [h(0), h(1), \ldots, h(L-1)]^\top$. The tensor representation of the input signal vector is the tensorization $\boldsymbol{\mathcal{X}}(n) = \mathcal{R}_D\big(\mathbf{x}(n)\big) \in$

$\mathbb{R}^{I_1 \times \cdots \times I_D}$ where the tensor dimensions $I_1, \ldots, I_D$ are such that $L = \prod_{d=1}^{D} I_d$. Analogously, we define $\boldsymbol{\mathcal{H}} = \mathcal{R}_D(\mathbf{h})$ such that we can rewrite the signal model in (2) using tensor notation as

$$\begin{aligned} y(n) &= \mathbf{x}^\top(n)\mathbf{h} + v(n) \\ &= \left\langle \boldsymbol{\mathcal{X}}(n), \boldsymbol{\mathcal{H}} \right\rangle + v(n). \end{aligned} \tag{3}$$

Let us consider a BTD of the tensor $\boldsymbol{\mathcal{H}}$ with $R$ blocks, where each block $r$ comprises a core tensor $\boldsymbol{\mathcal{G}}^{(r)} \in \mathbb{R}^{K_1 \times \ldots \times K_D}$ and $D$ factor matrices $\mathbf{A}^{(r,d)} \in \mathbb{R}^{I_d \times K_d}$ for each mode $d$. The core dimensions $K_1, \ldots, K_D$ define the multilinear rank of the block. The low-rank tensor approximation is then

$$\boldsymbol{\mathcal{H}} \approx \widehat{\boldsymbol{\mathcal{H}}}(\boldsymbol{\Theta}) = \sum_{r=1}^{R} \boldsymbol{\mathcal{G}}^{(r)} \times_{d \in [D]} \mathbf{A}^{(r,d)}. \tag{4}$$

with $\boldsymbol{\Theta} = (\{\boldsymbol{\mathcal{G}}^{(r)}\}, \{\mathbf{A}^{(r,d)}\})$ as the set of all decomposition parameters. Note that the core tensor dimensions need not be equal across blocks, however to avoid notational overload we will assume so, w.l.o.g. The factors matrices are constrained to have orthonormal columns, such that $\mathbf{A}^{(r,d)\top}\mathbf{A}^{(r,d)} = \mathbf{I}_{K_d}$, which implies $K_d \leq I_d$. This orthogonality constraint is not a strict requirement for a feasible decomposition, but it is a common constraint to avoid scaling indeterminacies and to improve the conditioning of the optimization problem.

For any input signal $x$ and its tensorization $\boldsymbol{\mathcal{X}}(n)$ and time $n$, the projection of the input signal onto the core space is

$$\boldsymbol{\mathcal{W}}^{(r)}(x, n) \coloneqq \boldsymbol{\mathcal{X}}(n) \times_{d \in [D]} \mathbf{A}^{(r,d)\top}. \tag{5}$$

The channel output $y(n)$ can then be approximated using the BTD filter

$$\hat{y}(n) = F[x](n) + v(n), \tag{6}$$

which we define as the multilinear (linear in each factor and core) operator

$$F[x](n) \coloneqq \sum_{r=1}^{R} \left\langle \boldsymbol{\mathcal{G}}^{(r)}, \boldsymbol{\mathcal{W}}^{(r)}(x, n) \right\rangle. \tag{7}$$

With these model definitions, we can now formulate the optimization problem to estimate the parameters of the BTD filter.

### B. Optimization Problem

To estimate the set of parameters, we formulate an optimization problem which yields the BTD filter parameters that best fit the observed data in the mean square error (MSE) sense. Computation of the BTD of a given tensor in literature is based on the HOOI or alternating least squares (ALS) algorithm [14], however for a given target tensor. Here we have a signal model with a BTD filter, and we want to estimate the parameters of the filter based on observed input-output data.

Let us define the optimization problem as the minimization

$$\min_{\boldsymbol{\Theta}} \quad f(\boldsymbol{\Theta}) + g_{\mathrm{q}}(\boldsymbol{\Theta}) + g_{\mathrm{o}}(\boldsymbol{\Theta}) \tag{8}$$

with the signal term $f(\boldsymbol{\Theta})$ defined as the MSE, i.e.,

$$f(\boldsymbol{\Theta}, n) = \mathbb{E}\left[e^2(\boldsymbol{\Theta}, n)\right], \tag{9}$$

$$e(\boldsymbol{\Theta}, n) = y(n) - F[x](n) \quad \forall\, m = 1, \ldots, M. \tag{10}$$

Note that within the proposed methods, the expectation operator $\mathbb{E}[\cdot]$ will be approximated by exponential averaging operations. The recursive Tikhonov penalty $g_{\mathrm{q}}(\boldsymbol{\Theta})$ [27] is

$$g_{\mathrm{q}}(\boldsymbol{\Theta}) = \frac{\gamma}{2} \sum_{(r,d)} \left[ \|\mathbf{A}^{(r,d)} - \mathbf{A}_{\text{old}}^{(r,d)}\|_F^2 \right], \tag{11}$$

with penalty parameter $\gamma \geq 0$ to improve convergence by avoiding flat regions („swamps") in the cost function caused by the scaling and conjugate indeterminacies of the BTD [27]; $\mathbf{A}_{\text{old}}^{(r,d)}$ denotes the previous iteration. The orthonormality penalty term $g_{\mathrm{o}}(\boldsymbol{\Theta})$ is the sum of indicator functions on the Stiefel manifolds, $\mathrm{St}(I_d, K_d)$ for each factor matrix, the the set of $I_d \times K_d$ matrices with orthonormal columns, defined as

$$g_{\mathrm{o}}(\boldsymbol{\Theta}) = \sum_{r,d} \iota_{\mathrm{St}}(\mathbf{A}^{(r,d)}) \tag{12}$$

where

$$\iota_{\mathrm{St}}(\mathbf{A}) = \begin{cases} 0 & \text{if } \mathbf{A}^\top\mathbf{A} = \mathbf{I} \\ \infty & \text{otherwise} \end{cases}. \tag{13}$$

Orthonormal columns of the factors make it straightforward to interpret, as one block of the BTD (a Tucker decomposition (TD)) is then a higher-order generalization of the generalized SVD (generalized low-rank approximation of matrices (GLRAM)) also known as the higher-order singular value decomposition (HOSVD) [18], [14], [29].

## III. Method

To solve the optimization problem (8), we propose an adaptive algorithm that iterates over all ranks and modes of the tensor decomposition forward-backward-backward steps based on the three-operator splitting strategy [25], [26]. We apply a HOOI-style ALS optimization procedure [14], [18] with preconditioned forward steps with subsequent proximal backward steps for regularization and orthonormality constraint. Let us state the distinct steps:

**Forward** on $f(\boldsymbol{\Theta})$: Perform updates on the cores and factors $\{\boldsymbol{\mathcal{G}}^{(r)}\}$, $\{\mathbf{A}^{(r,d)}\}$ based on signal data.

**Backward** on $g_{\mathrm{q}}(\boldsymbol{\Theta})$: Perform a proximal step on factors with Tikhonov/RALS regularization.

**Backward** on $g_{\mathrm{o}}(\boldsymbol{\Theta})$: Each factor is projected onto the Stiefel manifold to enforce orthonormality and the cores are realigned accordingly.

Note that in practice the backward steps are not required to be performed at every iteration, but rather can be performed every $T_b$ iterations for some integer $T_b \geq 1$ to reduce computational load.

### A. Forward Update

Although each factor and core sub-problem is linear in its respective variables, the joint problem is non-convex. Due to its overall nonlinearity, we propose to use LM-type [30] updates, and using approximate second-order information for better convergence speed and stability. The orthonormality

constraints yields a geometry-constrained optimization problem where the updates of the factors follow the Stiefel manifold. We therefore project factor updates onto the tangent space of the Stiefel manifold [31] before applying them and subsequently project the updated factors back onto the Stiefel manifold.

The second-order information in $\mathbb{R}^L$ space is approximated by Kronecker blocks in core and factor spaces. This approximation for large parameter spaces is established in large-scale optimization and machine learning literature [23], [24]. Let us introduce these approximations for factors and cores, and derive preconditioned update steps.

*1) Factors:* To derive the factor update steps, let us define the mode-$d$ factor regressor as the partial derivative of the BTD filter operator w.r.t. the mode-$d$ factor matrix,

$$\begin{aligned}\boldsymbol{\Phi}^{(r,d)}(x,n) &\coloneqq \frac{\partial F[x](n)}{\partial \mathbf{A}^{(r,d)}} \\ &= \left\langle \boldsymbol{\mathcal{Z}}^{(r,d)}(x,n), \boldsymbol{\mathcal{G}}^{(r)} \right\rangle_{\delta \neq d} \in \mathbb{R}^{I_d \times K_d}\end{aligned} \tag{14}$$

where

$$\boldsymbol{\mathcal{Z}}^{(r,d)}(x,n) \coloneqq \boldsymbol{\mathcal{X}}(n) \times_{\delta \neq d} \mathbf{A}^{(r,\delta)\top} \tag{15}$$

is the mode-$d$ partially projected tensor, i.e., the input projected through all mode factors except $d$. For the subsequent explanations, we will omit the time and input dependencies for notational convenience, i.e., we write $\boldsymbol{\Phi}^{(r,d)}$ instead of $\boldsymbol{\Phi}^{(r,d)}(x,n)$. For the mode-$d$ factor update, let $\boldsymbol{\Phi}^{(r,d)}$, be the back-projected regressor and define the matrices

$$\boldsymbol{\Sigma}_{\text{col}}^{(r,d)} = \mathbb{E}\left[\boldsymbol{\Phi}^{(r,d)}(\boldsymbol{\Phi}^{(r,d)})^\top\right] \in \mathbb{R}^{I_d \times I_d}, \tag{16}$$

$$\boldsymbol{\Sigma}_{\text{row}}^{(r,d)} = \mathbb{E}\left[(\boldsymbol{\Phi}^{(r,d)})^\top \boldsymbol{\Phi}^{(r,d)}\right] \in \mathbb{R}^{K_d \times K_d}. \tag{17}$$

Following standard ALS/HOOI normal equations for BTD, we approximate the covariance matrix (GN Hessian) in the parameter space for $\operatorname{vec}(\boldsymbol{\Phi}) \in \mathbb{R}^{I_d K_d}$ by a Kronecker product,

$$\boldsymbol{\Sigma}_{\mathbf{A}}^{(r,d)} = \mathbb{E}\left[\operatorname{vec}(\boldsymbol{\Phi}^{(r,d)})\operatorname{vec}(\boldsymbol{\Phi}^{(r,d)})^\top\right] \approx \boldsymbol{\Sigma}_{\text{row}}^{(r,d)} \otimes \boldsymbol{\Sigma}_{\text{col}}^{(r,d)}. \tag{18}$$

Using the properties of Kronecker product [32], [23], we can state that for a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{I_d \times K_d}$ and square invertible matrices $\boldsymbol{\Sigma}_{\text{col}}^{(r,d)}$ and $\boldsymbol{\Sigma}_{\text{row}}^{(r,d)}$, that

$$\operatorname{vec}\left(\boldsymbol{\Sigma}_{\text{col}}^{(r,d)} \boldsymbol{\Phi} \boldsymbol{\Sigma}_{\text{row}}^{(r,d)\top}\right) = \left(\boldsymbol{\Sigma}_{\text{row}}^{(r,d)} \otimes \boldsymbol{\Sigma}_{\text{col}}^{(r,d))}\right) \operatorname{vec}(\boldsymbol{\Phi}) \tag{19}$$

and with $(\mathbf{U} \otimes \mathbf{V})^{-1} = \mathbf{U}^{-1} \otimes \mathbf{V}^{-1}$, we can approximate the GN preconditioning of $\boldsymbol{\Phi}^{(r,d)}$ with (18) as

$$\begin{aligned}&\operatorname{vec}^{-1}\left(\boldsymbol{\Sigma}_{\mathbf{A}}^{(r,d)-1} \operatorname{vec}(\boldsymbol{\Phi}^{(r,d)})\right) \\ &\approx (\boldsymbol{\Sigma}_{\text{col}}^{(r,d)} + \alpha \mathbf{I})^{-1} \boldsymbol{\Phi}^{(r,d)} (\boldsymbol{\Sigma}_{\text{row}}^{(r,d)} + \alpha \mathbf{I})^{-\top} =: \tilde{\boldsymbol{\Phi}}^{(r,d)},\end{aligned} \tag{20}$$

where $\alpha \geq 0$ is a dampening parameter and $\tilde{\boldsymbol{\Phi}}^{(r,d)}$ is the conditioned regressor. The matrices $\boldsymbol{\Sigma}_{\text{col}}^{(r,d)}$ and $\boldsymbol{\Sigma}_{\text{row}}^{(r,d)}$ are recursively estimated over the sequence of input data by

$$\boldsymbol{\Sigma}_{\text{col}}^{(r,d)} \leftarrow \lambda \boldsymbol{\Sigma}_{\text{col}}^{(r,d)} + (1-\lambda) \boldsymbol{\Phi}^{(r,d)} (\boldsymbol{\Phi}^{(r,d)})^\top, \tag{21}$$

$$\boldsymbol{\Sigma}_{\text{row}}^{(r,d)} \leftarrow \lambda \boldsymbol{\Sigma}_{\text{row}}^{(r,d)} + (1-\lambda) (\boldsymbol{\Phi}^{(r,d)})^\top \boldsymbol{\Phi}^{(r,d)}, \tag{22}$$

where $\lambda$ is the forgetting factor. We define the factor update step as a dampened LM step, with the update direction

$$\delta \mathbf{A}^{(r,d)} = -\frac{e}{\beta_{\text{f}} + \left\langle \boldsymbol{\Phi}^{(r,d)}, \tilde{\boldsymbol{\Phi}}^{(r,d)} \right\rangle} \tilde{\boldsymbol{\Phi}}^{(r,d)} \tag{23}$$

where $\beta_{\text{f}} \geq 0$ is a dampening parameter. The update direction is then projected onto the tangent space of the Stiefel manifold at the current factor estimate by

$$\delta \mathbf{A}^{(r,d)} \leftarrow \delta \mathbf{A}^{(r,d)} - \mathbf{A}^{(r,d)} \operatorname{sym}\{\mathbf{A}^{(r,d)\top} \delta \mathbf{A}^{(r,d)}\} \tag{24}$$

which is the canonical Riemannian gradient mapping on the Stiefel manifold [31].

The update step is then

$$\mathbf{A}^{(r,d)} \leftarrow \mathbf{A}^{(r,d)} - \mu_{\text{f}}\, \delta \mathbf{A}^{(r,d)}, \tag{25}$$

with the step size $\mu_{\text{f}} > 0$.

*2) Cores:* The core regressor, i.e., the partial derivative of the operator (7) w.r.t. a core is

$$\frac{\partial F[x](n)}{\partial \boldsymbol{\mathcal{G}}^{(r)}} = \boldsymbol{\mathcal{W}}^{(r)}(x,n) \tag{26}$$

due to the linearity of the tensor inner product. Let there be the covariance matrix (GN Hessian) in vectorized-core space,

$$\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{G}}}^{(r)} = \mathbb{E}\left[\operatorname{vec}(\boldsymbol{\mathcal{W}}^{(r)})\operatorname{vec}(\boldsymbol{\mathcal{W}}^{(r)})^\top\right] \approx \bigotimes_{d=1}^{D} \boldsymbol{\Sigma}_{\text{core}}^{(r,d)}, \tag{27}$$

again approximated by a Kronecker product of $D$ smaller blocks, defined as

$$\boldsymbol{\Sigma}_{\text{core}}^{(r,d)} = \mathbb{E}\left[\mathbf{W}_{(d)}^{(r)} (\mathbf{W}_{(d)}^{(r)})^\top\right], \tag{28}$$

where $\mathbf{W}_{(d)}$ is the mode-$d$ unfolding of the core regressor tensor (26). We can again define the dampened LM/GN preconditioning in the full core parameter space using the Kronecker blocks, applying the same inversion rule as before:

$$\begin{aligned}&\operatorname{vec}^{-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{G}}}^{(r)-1} \operatorname{vec}(\boldsymbol{\mathcal{W}})^{(r)}\right) \\ &\approx \boldsymbol{\mathcal{W}} \times_{d \in [D]} (\boldsymbol{\Sigma}_{\text{core}}^{(r,d)} + \alpha \mathbf{I})^{-1} =: \tilde{\boldsymbol{\mathcal{W}}}^{(r)},\end{aligned} \tag{29}$$

which is the generalization of the matrix (2D tensor) case (19) to $D$ modes. The matrices $\boldsymbol{\Sigma}_{\text{core}}^{(r,d)}$ are recursively estimated by

$$\boldsymbol{\Sigma}_{\text{core}}^{(r,d)} \leftarrow \lambda \boldsymbol{\Sigma}_{\text{core}}^{(r,d)} + (1-\lambda) \mathbf{W}_{(d)}^{(r)} (\mathbf{W}_{(d)}^{(r)})^\top, \tag{30}$$

where $\lambda$ is the forgetting factor again. The core update direction is then

$$\delta \boldsymbol{\mathcal{G}}^{(r)} = -\frac{e(n)}{\beta_{\text{c}} + \left\langle \boldsymbol{\mathcal{W}}^{(r)}, \tilde{\boldsymbol{\mathcal{W}}}^{(r)} \right\rangle} \tilde{\boldsymbol{\mathcal{W}}}^{(r)}, \tag{31}$$

with the dampening parameter as $\beta_{\text{c}} \geq 0$, and the core update is given by

$$\boldsymbol{\mathcal{G}}^{(r)} \leftarrow \boldsymbol{\mathcal{G}}^{(r)} - \mu_{\text{c}}\, \delta \boldsymbol{\mathcal{G}}^{(r)} \tag{32}$$

with the step size $\mu_{\text{c}} > 0$.

In practice, we accumulate the update directions (24) and (31) over multiple iterations $T_f \geq 1$ before applying the (25) and (32) to stabilize updates. Approximating the Hessian by separation into a Kronecker product of smaller blocks reduces the complexity of estimating and inverting the Hessian from

a factor and core parameter space of dimensions $I_d K_d$ and $\prod_d K_d$ respectively, to smaller parameter spaces of dimensions $I_d$ and $K_d$. For scalar cores, i.e., $K_d = 1$ for all modes $d$, the core Hessian and the row-side conditioning of the factors reduces to scalar operations. Conceptually, these approximations map to the full Hessian in the original parameter space $\mathbb{R}^L$ of the represented RIR, and the updates correspond to (approximate) GN steps in that space, restricted to the decomposition manifold. We show this mapping in more detail in Appendix-A.

### B. Backwards: Recursive Tikhonov Regularization

The backwards step every $T_b$-th iteration applies the proximal operator of the quadratic penalty term $g_{\mathrm{q}}(\boldsymbol{\Theta})$ to the factors, which can be computed in closed form as

$$\mathbf{A}^{(r,d)} \leftarrow \mathrm{prox}_{g_{\mathrm{q}}}\big(\mathbf{A}^{(r,d)}\big) = \frac{\mathbf{A}^{(r,d)} + \gamma^{(r,d)}\mathbf{A}^{(r,d)}_{\mathrm{old}}}{1+\gamma^{(r,d)}}. \quad (33)$$

The proximal step size is chosen as 1 [26].

### C. Backwards: Orthonormalization

Every $T_b$-th iteration, the factors are re-orthogonalized by the proximal operator of the Stiefel manifold penalty,

$$\mathbf{A}^{(r,d)} \leftarrow \mathrm{prox}_{g_{\mathrm{o}}}\big(\mathbf{A}^{(r,d)}\big), \quad (34)$$

which can be computed by an orthogonalization procedure, i.e., using a Polar decomposition, or an approximation such as the QR decomposition (QR) [32], see Appendix-B. Note that the projection of the updates onto the tangent space of the Stiefel manifold (24) keeps the factors close to orthonormality, however, not exactly on the manifold. Further, the proximal step (33) can drift the factors away from orthonormality. To compensate for the non-smooth orthonormalization, let the right normalizer be such that

$$\mathbf{A}^{(r,d)} = \mathrm{prox}_{g_{\mathrm{o}}}\big(\mathbf{A}^{(r,d)}\big)\mathbf{R}^{(r,d)}, \quad (35)$$

then the cores are realigned

$$\boldsymbol{\mathcal{G}}^{(r)} \leftarrow \boldsymbol{\mathcal{G}}^{(r)} \times_d \mathbf{R}^{(r,d)\top}. \quad (36)$$

and the composed map of orthonormalization and realignment is constant at that step, i.e., it leaves the represented tensor unchanged only shifting values internally. The overall algorithm remains stable.

The full procedure of the proposed SeCO-BTD method is summarized in Algorithm 1.

**Algorithm 1** Separable-Curvature Online BTD (SeCO-BTD)

**Input:** stream $x(n)$, decomposition parameters $R$, $I_d, K_d \,\forall\, d$, step sizes $\mu_f, \mu_c$, dampings $\beta_f, \beta_c$, forgetting factor $\lambda$,
**Init:** $\{\boldsymbol{\mathcal{G}}^{(r)}\}$, $\{\mathbf{A}^{(r,d)}\}$, stats $\boldsymbol{\Sigma}^{(r,d)}_{\mathrm{col}}, \boldsymbol{\Sigma}^{(r,d)}_{\mathrm{row}}$ $\boldsymbol{\Sigma}^{(r,d)}_{\mathrm{core}}$,
**for** each time $n$ **do**
  // Forward local update
  **for** all $r, d$ **do**
    Compute $\boldsymbol{\Phi}^{(r,d)}(n)$ (14), $\boldsymbol{\mathcal{W}}^{(r)}(n)$ (5)
    Compute $e(n)$ (10)
    Update $\boldsymbol{\Sigma}^{(r,d)}_{\mathrm{col}}$ (21), $\boldsymbol{\Sigma}^{(r,d)}_{\mathrm{row}}$ (22), $\boldsymbol{\Sigma}^{(r,d)}_{\mathrm{core}}$ (30)
    Compute $\tilde{\boldsymbol{\Phi}}_{(r,d)}$, $\tilde{\boldsymbol{\mathcal{W}}}^{(r)}$
    Aggregate $\delta\mathbf{A}^{(r,d)}$ updates (23) (24)
    Aggregate $\delta\boldsymbol{\mathcal{G}}^{(r)}$ updates (31)
    **if** mod$(n, T_f)$ == 0 **then**
      // Batched forward update
      Update $\mathbf{A}^{(r,d)}$ (25)
      Update $\boldsymbol{\mathcal{G}}^{(r)}$ (32)
    **end if**
  **end for**
  **if** mod$(n, T_b)$ == 0 **then**
    // Backward penalty update
    Regularization on $\mathbf{A}^{(r,d)}$ (33)
    // Orthonormalization & core realignment
    **for** all $r, d$ **do**
      // Orthonormalize factors and realign cores
      Orthonormalize $\mathbf{A}^{(r,d)}$ (34)
      Realign $\boldsymbol{\mathcal{G}}^{(r)}$ (36)
    **end for**
  **end if**
**end for**

## IV. Computational Complexity

While we introduce quite a few conceptual steps in the proposed method, we will show that the overall computational complexity is relatively low, and we make the comparison to the state-of-the-art RLS-TOT method introduced in [22]. We will state the dominant terms for each relevant step of the algorithm in big-$\mathcal{O}(\cdot)$ notation, then give the overall dominant terms for $L, D, I_d, K_d$.

The forward step comprises the following relevant computations: The contractions (5) and (15), which have a cost that is linear in the length of the filters $L$ and the number of modes $D$, as $\mathcal{O}(LD)$. The cost of Kronecker preconditioning, which is dominated by the cost of solving the linear systems for the row- and column-side Hessian factors (20), which is $\mathcal{O}(R\sum_d(I_d K_d^2 + K_d^2 + K_d I_d^2 + I_d^2))$ for all ranks and modes, and preconditioning of the core regressor (29) which is $\mathcal{O}(R\prod_d K_d \sum_d K_d^2 + R\sum_d K_d^3)$. This cost can be amortized by performing the factor updates in batches of $T_f$ iterations, which we do in practice. If diagonal-only preconditioning is used, the cost reduces to $\mathcal{O}(R\sum_d I_d K_d)$, and $\mathcal{O}(R\prod_d K_d)$, respectively.

The cost for the factor penalty proximal step (33) is $\mathcal{O}(R\sum_d I_d K_d)$ for all ranks and modes.

The cost incurred by the Stiefel geometry constraint, which is the tangent space projection (24) and the re-orthogonalization step (34), are $\mathcal{O}(R\sum_d(I_d K_d^2 + K_d^3))$ and $\mathcal{O}(R\sum_d K_d^3)$ respectively, if we consider the cost of the Polar decomposition for the re-orthogonalization step. If we consider the QR decomposition for the re-orthogonalization step, the cost reduces to $\mathcal{O}(R\sum_d I_d K_d^2)$. To realign the cores (36), the cost is $\mathcal{O}(R\sum_d K_d^3)$ by mode product of the core with the right-normalizers.

The overall cost of the proposed method is dominated by

the terms

$$\mathcal{O}(LD) + \mathcal{O}(R\sum_d (I_d K_d^2 + K_d^3 + K_d I_d^2 + I_d^3)), \tag{37}$$

which is the worst-case complexity for full Kronecker preconditioning and no batching/amortization.

As a direct comparison, we use the RLS-TOT approach introduced in [22], which has cost of $\mathcal{O}(L_2^4 + (PL_{11}L_2)^2 + (PL_{12}L_2)^2)$ for tensor shape $\mathbb{R}^{L_{11}\times L_{12}\times L_2}$. RLS-TOT implements RLS updates for the three decomposition modes, while the proposed method uses an approximation of second-order information for the factor and core updates, which is more efficient to compute. Figure 1 shows the comparison of computational complexity of the algorithms. We use the recommended selection of decomposition parameters for RLS-TOT (keep $L_2$ as small as possible) as described in the original paper. For the proposed method, we show the same shapes as for RLS-TOT for $K_d = 1$ (scalar cores), and show The proposed method scales better with the filter length $L$ if the decomposition shape is balanced, otherwise, the cost of the proposed method can be higher than RLS-TOT for some decomposition shapes, which we show as counter examples in the plot.

Memory requirements of the proposed method during estimation can be stated as the following: i) the number of parameters to be estimated, which is

$$p = R\sum_d I_d K_d + R\prod_d K_d, \tag{38}$$

and ii) conditioning covariances, which is $R\sum_d K_d^2 + I_d^2$ for the factors and $R\sum_d K_d^2$ for the cores if we consider the full Kronecker preconditioning case, and $R\sum_d K_d + I_d$ for the factors and $R\sum_d K_d$ for the cores if we consider the diagonal-only preconditioning. The number of parameters to be estimated defines the compression ratio as

$$C = 1 - \frac{p}{L}. \tag{39}$$

## V. NUMERICAL EXPERIMENTS

To validate the proposed method SeCO-BTD, we perform numerical experiments. We evaluate the method's performance with and estimation task on simulated RIRs of varying lengths, tensor shapes and decomposition ranks. As the performance metric for the estimation and reconstruction error, we use the normalized misalignment (NM) defined as

$$\text{NM} = 10\log_{10}\frac{\|\hat{\mathbf{h}} - \mathbf{h}\|_2^2}{\|\mathbf{h}\|_2^2} \quad [\text{dB}] \tag{40}$$

where $\hat{\mathbf{h}}$ is the estimated RIR (reconstructed from the tensor decomposition) and $\mathbf{h}$ is the true RIR.

We compare the proposed SeCO-BTD filter method to the state-of-the-art RLS-TOT [22]. It is important to note that RLS-TOT as described in the original paper tends to instability when using the recommended parameters, therefore, we slightly modify the method by introducing a step size to the update steps of the $L_{11}, L_{12}$ partial filters, which stabilizes the method and allows us to do a fair comparison. We tuned

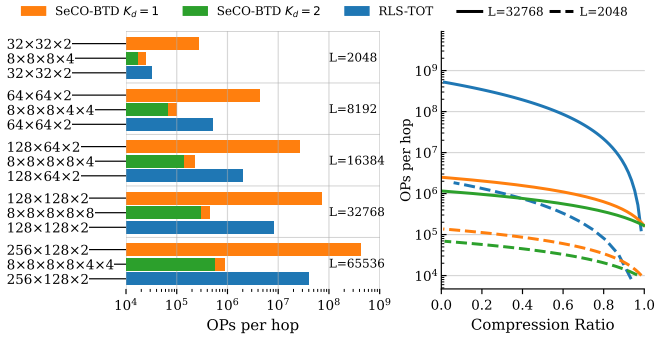


Fig. 1. Complexity comparison of the proposed method and RLS-TOT [22] for different filter lengths and decomposition shapes. For comparison, we tuned the proposed method's $R$ and RLS-TOT's $P$ for to achieve the compression ratio $C = 0.87$ (reference is the $P = 2$ case from [33]) for each filter length $L \in \{2048, 8192, 16384, 32768, 65536\}$ and tensor shape (labelled on the left). We show the complexity proxy for scalar cores $K_d = 1$, and indicate the increase for $K_d = 2$, again tuned to achieve the same compression ratio, on some of the points. We can see that the proposed method scales better with the filter length $L$ and the decomposition rank $R$ than RLS-TOT if the decomposition's shape is balanced, i.e., when the tensor dimensions are approximately equal (see counter examples). The right plot shows the complexity over the compression ratio $C$ for $L \in \{2048, 32768\}$ for RLS-TOT shapes $\mathbb{R}^{32\times32\times2}$ and $\mathbb{R}^{128\times128\times2}$, and the proposed method's $\mathbb{R}^{8\times8\times8\times4}$ and $\mathbb{R}^{8\times8\times8\times8\times8}$, showing the significant reduction in complexity towards higher compression ratios.

the step size for RLS-TOT to the best performance, leading to the value of 0.2. We generate RIRs using the randomized image source method (RISM) [34] for a shoebox room of size 8.00 m × 5.00 m × 3.00 m and two scenarios:

- a short RIR scenario, with a sampling rate of 8 kHz and a filter length of $L = 2048$ samples, which corresponds to a reverberation time of $T_{60} \approx 0.2$ s. This is to tie into the RLS-TOT paper [22], which uses this scenario for evaluation.
- a long RIR scenario, with a sampling rate of 32 kHz and a filter length of $L = 16384$ samples, which corresponds to a reverberation time of $T_{60} \approx 0.5$ s. This is to evaluate the methods in a more challenging scenario.

Figure 2 shows an example RIR and its energy decay curve (EDC) for the second, long scenario.

We generate the input signals using both WGN and colored noise generated by an autoregressive (AR)(1) process as input signal. The AR(1) process is defined as $x(n) = 0.8x(n-1) + w(n)$, where $w(n)$ is WGN. Then the observed signal is generated by convolving the input signal with the RIR and adding WGN at a specified signal-to-noise ratio (SNR) of 10 dB. The results are averaged over 10 runs, where at each run, the source and receiver positions are randomly placed in the room. The computational effort of the RLS-TOT method is high for the long RIR scenario, therefore, we need to limit the number of runs for that scenario to 5.

Figure 3 shows the results for the short RIR scenario for both input signal types. We chose the decomposition parameters for the proposed method to a) match the compression ratio of RLS-TOT at $P \in \{2, 4\}$ and b) to be balanced, i.e., with approximately equal tensor dimensions, which is the optimal shape for the proposed method. Further, we compare

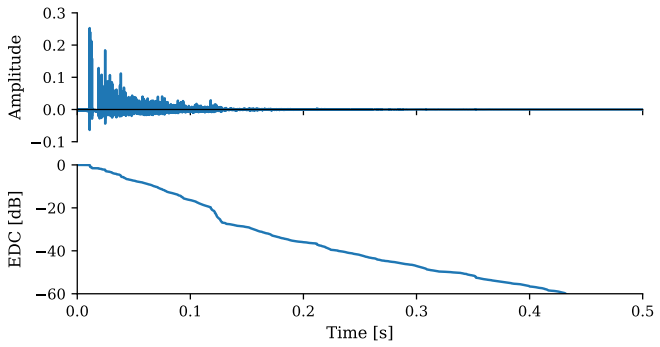


Fig. 2. Example RIR and the normalized EDC computed by backwards integration for the long scenario with $L = 16384$ samples, at a sampling rate of 32 kHz.

to a decomposition with lower compression ratio, but approximately matching the complexity of RLS-TOT for the same filter length. These results show that the proposed method achieves similar albeit slightly worse estimation performance than RLS-TOT for the same compression ratio, but reduces the computational complexity. It is to note that the length of the RIR in this scenario is still relatively short, and the application of either tensor-decomposition-based method is questionable, as methods such as partitioned-block frequency-domain adaptive filter (PBFDAF)/multidelay block frequency-domain adaptive filter (MDF) might be a better choice. Nonetheless, the results show that the proposed method is competitive with the state-of-the-art method in this scenario, while reducing the computational complexity.

The results for the long RIR scenario are shown in Figure 4 for the WGN and colored noise input signals respectively. RLS-TOT becomes computationally very expensive for this scenario, as the cost scales worse than the proposed method. The computational advantage of the proposed method is more significant in this scenario, one to two orders of magnitude depending on the decomposition parameters, while approaching the estimation performance of RLS-TOT for the same compression ratio. Expectedly, the estimation performance of both methods degrades for the colored noise input and the convergence speed of the proposed method is slower due to approximated second-order information. RLS-TOT shows instability in early iterations for the colored noise input.

## VI. CONCLUSION

We proposed a novel adaptive algorithm for efficient input-output estimation of long RIRs based on the BTD. We first show how to formulate the convolutional model using BTD and tensor algebra and then derive an adaptive algorithm based on three-operator splitting to solve the resulting optimization problem. It uses approximate second-order information by using Kronecker-block approximations in the decompositions core and factor spaces. We use LM/GN preconditioned update steps, which are more efficient to compute than RLS and the state-of-the-art decomposition-based RLS-TOT method. The proposed method is flexible in terms of the decomposition parameters, making it applicable to arbitrary decomposition shapes and ranks and can therefore be applied to tensor-valued signals with various structures, also outside RIR estimation. The caveat is that the computational complexity is minimal for balanced decomposition shapes. Experiments on simulated RIRs show that the proposed method achieves similar estimation performance to the state-of-the-art RLS-TOT method for the same compression ratio, while significantly reducing the computational complexity, trading off convergence speed.

Within the scope of the proposed method, future extensions include scattered updates, i.e., different updating schedules for the different factors and cores, and further minimizing computational complexity by exploiting repeating tensor fibres during the signal sequence. An obvious next step beyond the current work is to formulate an RLS version of the proposed method for arbitrary decomposition shapes and ranks, which would further improve the convergence speed at the cost of increased computational complexity, and compare with state-of-the-art RLS-TOT and the proposed method. Also, as shown in literature, multiple RIRs within the same room show significant similarities in their tensor decompositions, which can be exploited by extending the proposed method to consider shared factors between the decompositions of different RIRs, reducing the parameter count and increasing model efficiency.

## APPENDIX

### A. Mapping of Decomposition Parameters to the Hessian Approximation

Using the model $\widehat{\boldsymbol{\mathcal{H}}}(\boldsymbol{\Theta})$ (4) with parameters $\boldsymbol{\Theta}$, let the reconstructed RIR be $\hat{\mathbf{h}}(\boldsymbol{\Theta}) = \operatorname{vec}\big(\widehat{\boldsymbol{\mathcal{H}}}(\boldsymbol{\Theta})\big) \in \mathbb{R}^{L}$. Use the multilinear reconstruction map and its Jacobian

$$\Psi : \boldsymbol{\Theta} \mapsto \hat{\mathbf{h}}(\boldsymbol{\Theta}), \quad \mathbf{J}(\boldsymbol{\Theta}) = \frac{\partial \Psi(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \in \mathbb{R}^{L \times p}, \tag{41}$$

where $p$ is the number of parameters in the decomposition (38) and the $\mathbf{J}(\boldsymbol{\Theta})$ stacks (5) and (14) for all $r, d$ such that the Jacobian is then

$$\mathbf{J}(\boldsymbol{\Theta}) = \left[\mathbf{J}_{\boldsymbol{\mathcal{G}}}^{(1)}, \ldots, \mathbf{J}_{\boldsymbol{\mathcal{G}}}^{(R)}, \mathbf{J}_{\mathbf{A}}^{(1,1)}, \ldots, \mathbf{J}_{\mathbf{A}}^{(R,D)}\right], \tag{42}$$

with $\mathbf{J}_{\boldsymbol{\mathcal{G}}}^{(r)} \in \mathbb{R}^{L \times \prod_d K_d}$ and $\mathbf{J}_{\mathbf{A}}^{(r,d)} \in \mathbb{R}^{L \times I_d K_d}$. Accumulating over samples, the corresponding rows are $\operatorname{vec}\big(\boldsymbol{\mathcal{W}}^{(r)}(x,n)\big)$ and $\operatorname{vec}\big(\boldsymbol{\Phi}^{(r,d)}(x,n)\big)$, respectively. Then, with $y(n) = \big\langle \boldsymbol{\mathcal{X}}(n), \boldsymbol{\mathcal{H}} \big\rangle + v(n) = \mathbf{x}^{\top}(n)\mathbf{h} + v(n)$ (3), the MSE cost is $\mathbb{E}\left[e^2(n)\right]$, with $e(n) = y(n) - \mathbf{x}^{\top}(n)\hat{\mathbf{h}}(\boldsymbol{\Theta})$. Using standard MSE results and the chain rule, the reduced covariance is

$$\mathbf{C}_p(\boldsymbol{\Theta}) = \mathbf{J}(\boldsymbol{\Theta})^{\top}\mathbf{C}_L\mathbf{J}(\boldsymbol{\Theta}) \in \mathbb{R}^{p \times p}, \tag{43}$$

where $\mathbf{C}_L = \mathbb{E}\left[\mathbf{x}\mathbf{x}^{\top}\right]$ is the covariance matrix of the input signal. Further, using (42), and assuming cross-mode and cross-rank independence, each factor's Hessian approximation (18) can be derived as the corresponding block of the reduced Hessian $\mathbf{C}_p$ as

$$\boldsymbol{\Sigma}_{\mathbf{A}}^{(r,d)} = \mathbf{J}_{\mathbf{A}}^{(r,d)\top}\mathbf{C}_L\mathbf{J}_{\mathbf{A}}^{(r,d)} \approx \boldsymbol{\Sigma}_{\text{row}}^{(r,d)} \otimes \boldsymbol{\Sigma}_{\text{col}}^{(r,d)}, \tag{44}$$

and each core's Hessian approximation (28) can be derived as

$$\boldsymbol{\Sigma}_{\boldsymbol{\mathcal{G}}}^{(r)} = \mathbf{J}_{\boldsymbol{\mathcal{G}}}^{(r)\top}\mathbf{C}_L\mathbf{J}_{\boldsymbol{\mathcal{G}}}^{(r)} \approx \bigotimes_{d=1}^{D} \boldsymbol{\Sigma}_{\text{core}}^{(r,d)}. \tag{45}$$

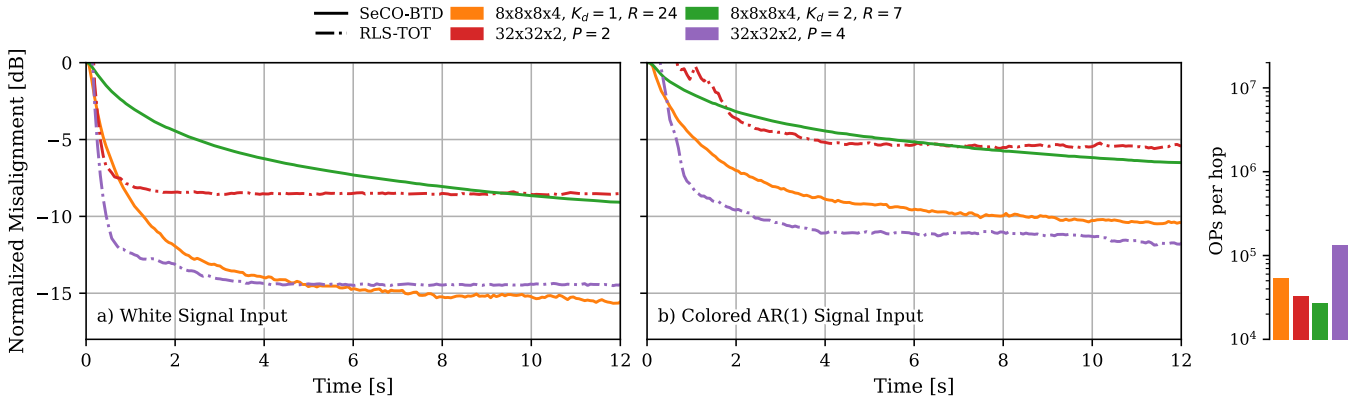


Fig. 3. RIR estimation results for simulated rooms with $L = 2048$ showing how the proposed SeCO-BTD method with Kronecker-block Hessian approximation compares to the state-of-the-art RLS-TOT method. The left figure shows results for a WGN input signal for a selected set of decomposition parameters, while the right figure shows the corresponding computational complexity. The SNR is 10 dB.

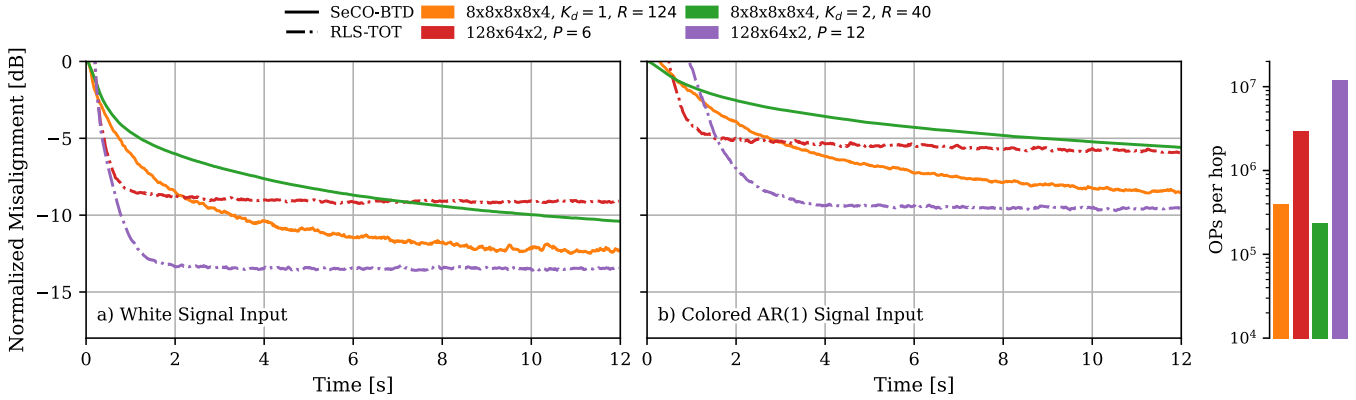


Fig. 4. RIR estimation results for simulated rooms with $L = 16384$ showing how the proposed SeCO-BTD method with Kronecker-block Hessian approximation compares to the state-of-the-art RLS-TOT method. The left figure shows results for a WGN input signal for a selected set of decomposition parameters, while the right figure shows the corresponding computational complexity. The SNR is 10 dB.

Then the dampened LM/GN solves the linear system

$$(\mathbf{C}_p(\boldsymbol{\Theta}) + \beta\mathbf{I})\,\Delta\boldsymbol{\Theta} = \mathbf{J}(\boldsymbol{\Theta})^\top \mathbf{g}, \quad \mathbf{g} = \mathbb{E}\left[\mathbf{x}e\right], \tag{46}$$

which then implies that the update direction in the full RIR parameter space is given by

$$\Delta\mathbf{h} = \mathbf{J}(\boldsymbol{\Theta})\Delta\boldsymbol{\Theta} \tag{47}$$

which corresponds to updates on the decomposition manifold $\{\Psi(\boldsymbol{\Theta})\} \subset \mathbb{R}^L$ [31], [35].

### B. Proximal Operator of the Stiefel Manifold Indicator

The proximal operator of the indicator function $\iota_{\mathrm{St}}$ on the Stiefel set $\mathrm{St}(m,n) = \{\mathbf{X} \in \mathbb{R}^{m\times n} \mid \mathbf{X}^\top\mathbf{X} = \mathbf{I}\}$ is defined as [26]

$$\mathrm{prox}_{\iota_{\mathrm{St}}}(\mathbf{Y}) = \arg\min_{\mathbf{X}} \left\{ \iota_{\mathrm{St}}(\mathbf{X}) + \frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 \right\}, \tag{48}$$

where $\iota_{\mathrm{St}}(\mathbf{X}) = 0$ if $\mathbf{X}^\top\mathbf{X} = \mathbf{I}$ and $\infty$ else. With $\mathbf{Y} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, the optimal solution is

$$\mathrm{prox}_{\iota_{\mathrm{St}}}(\mathbf{Y}) = \mathbf{U}\mathbf{V}^\top = \mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-1/2} \tag{49}$$

which is the left factors of the polar decomposition of $\mathbf{Y}$ [36], [32] or also done using the SVD. Using the QR decomposition instead of the Polar decomposition or SVD yields a retraction onto the Stiefel manifold as well, however, it is not the closest-point projection and therefore not the exact proximal operator.

## REFERENCES

[1] P. A. Naylor and N. D. Gaubitch, Eds., *Speech Dereverberation*, ser. Signals and Communication Technology. Springer Science & Business Media, 2010. [Online]. Available: http://link.springer.com/10.1007/978-1-84996-056-4

[2] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A Consolidated Perspective on Multimicrophone Speech Enhancement and Source Separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[3] T. van Waterschoot, "Design and evaluation of digital signal processing algorithms for acoustic feedback and echo cancellation," *J. Acoust. Soc. Am.*, vol. 126, no. 6, pp. 3373–3373, Dec. 2009. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.3257646

[4] J. S. Bradley, "Predictors of speech intelligibility in rooms," *J. Acoust. Soc. Am.*, vol. 80, no. 3, pp. 837–845, Sep. 1986. [Online]. Available: https://doi.org/10.1121/1.393907

[5] Y. E. Baba, A. Walther, and E. A. P. Habets, "3D Room Geometry Inference Based on Room Impulse Response Stacks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 5, pp. 857–872, May 2018.

[6] K. MacWilliam, F. Elvander, and T. van Waterschoot, "Simultaneous Acoustic Echo Sorting and 3-D Room Geometry Inference," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '23)*. Rhodes Island, Greece: IEEE, 2023, pp. 1–5.

[7] C. Evers, H. W. Löllmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, and W. Kellermann, "The LOCATA Challenge: Acoustic Source Localization and Tracking," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1620–1643, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9079214/

[8] Y. Avargel and I. Cohen, "System Identification in the Short-Time Fourier Transform Domain With Crossband Filtering," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 4, pp. 1305–1319, May 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4156182/

[9] J.-S. Soo and K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 373–376, Feb. 1990. [Online]. Available: https://ieeexplore.ieee.org/document/103078/?arnumber=103078&tag=1

[10] E. Moulines, O. Ait Amrane, and Y. Grenier, "The generalized multidelay adaptive filter: Structure and convergence analysis," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 14–28, Jan. 1995. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/365282

[11] G. Bernardi, T. van Waterschoot, J. Wouters, and M. Moonen, "Adaptive Feedback Cancellation Using a Partitioned-Block Frequency-Domain Kalman Filter Approach With PEM-Based Signal Prewhitening," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1784–1798, Sep. 2017.

[12] A. P. Adamov, S. G. Sementsov, and A. A. Adamova, "IIR-filter room response modeling," in *AIP Conference Proceedings*, vol. 2833. AIP Publishing LLC, 2023, p. 020004.

[13] M. Jalmby, F. Elvander, and T. van Waterschoot, "Low-Rank Room Impulse Response Estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, pp. 1–13, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10028766/

[14] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009. [Online]. Available: http://epubs.siam.org/doi/10.1137/07070111X

[15] M. Jälmby, F. Elvander, and T. van Waterschoot, "Multi-Channel Low-Rank Convolution of Jointly Compressed Room Impulse Responses," *IEEE Open J. Signal Process.*, vol. 5, pp. 850–857, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10549780/?arnumber=10549780&tag=1

[16] L. De Lathauwer and D. Nion, "Decompositions of a Higher-Order Tensor in Block Terms—Part III: Alternating Least Squares Algorithms," *SIAM J. Matrix Anal. & Appl.*, vol. 30, no. 3, pp. 1067–1083, Jan. 2008. [Online]. Available: http://epubs.siam.org/doi/10.1137/070690730

[17] L. De Lathauwer, "Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness," *SIAM J. Matrix Anal. & Appl.*, vol. 30, no. 3, pp. 1033–1066, Jan. 2008. [Online]. Available: http://epubs.siam.org/doi/10.1137/070690729

[18] ——, "Decompositions of a Higher-Order Tensor in Block Terms—Part I: Lemmas for Partitioned Matrices," *SIAM J. Matrix Anal. & Appl.*, vol. 30, no. 3, pp. 1022–1032, Jan. 2008. [Online]. Available: http://epubs.siam.org/doi/10.1137/060661685

[19] M. Jälmby, F. Elvander, and T. van Waterschoot, "Low-Rank Tensor Modeling of Room Impulse Responses," in *Proc. 29th European Signal Process. Conf. (EUSIPCO '21)*, Aug. 2021, pp. 111–115.

[20] ——, "Compression of room impulse responses for compact storage and fast low-latency convolution," *EURASIP J. Audio, Speech, Music Process.*, vol. 2024, no. 1, p. 45, Sep. 2024. [Online]. Available: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-024-00363-5

[21] G. Huang, J. Benesty, and J. Chen, "Dimensionality Reduction of Room Acoustic Impulse Responses and Applications to System Identification," *IEEE Signal Process. Lett.*, vol. 30, pp. 1107–1111, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10224274

[22] 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, p. 109216, Dec. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0165168423002906

[23] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, Jul. 2015, pp. 2408–2417. [Online]. Available: https://proceedings.mlr.press/v37/martens15.html

[24] M. Fahrbach, G. Fu, and M. Ghadiri, "Subquadratic kronecker regression with applications to tensor decomposition," in *Adv. Neural Inf. Process. (NeurIPS)*, vol. 35. Curran Associates, Inc., 2022, pp. 28 776–28 789. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/b9121bbb3112975d33c527f046ae68f2-Paper-Conference.pdf

[25] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-valued var. anal.*, vol. 25, no. 4, pp. 829–858, 2017.

[26] N. Parikh and S. Boyd, "Proximal Algorithms," *Found. Trends Optim*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[27] C. Navasca, L. De Lathauwer, and S. Kindermann, "Swamp reducing technique for tensor decomposition," in *Proc. 16th European Signal Process. Conf. (EUSIPCO '08)*, Aug. 2008, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/7080724/?arnumber=7080724

[28] D. Pandey, A. Venugopal, and H. Leib, "Linear to multi-linear algebra and systems using tensors," Dec. 2023. [Online]. Available: http://arxiv.org/abs/2304.10658

[29] J. Ye, "Generalized Low Rank Approximations of Matrices," in *Proc. Twenty-First Int. Conf. Mach. Learn.*, ser. Icml '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 112. [Online]. Available: https://doi.org/10.1145/1015330.1015347

[30] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun. 1963. [Online]. Available: http://epubs.siam.org/doi/10.1137/0111030

[31] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton: Princeton University Press, 2008.

[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, fourth edition ed., ser. Johns Hopkins Studies in the Mathematical Sciences. Baltimore: The Johns Hopkins University Press, 2013.

[33] C. Paleologu, J. Benesty, and S. Ciochină, "Linear System Identification Based on a Kronecker Product Decomposition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1793–1808, Oct. 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8369106

[34] E. De Sena, N. Antonello, M. Moonen, and T. van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE Trans. Audio Speech Lang. Process.*, vol. 23, no. 4, pp. 774–786, 2015.

[35] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

[36] N. J. Higham, *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Jan. 2008. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9780898717778

**SECOND B. AUTHOR,** photograph and biography not available at the time of publication.

**THIRD C. AUTHOR JR.** (Member, IEEE), photograph and biography not available at the time of publication.