



Article Wiener Filter Using the Conjugate Gradient Method and a Third-Order Tensor Decomposition

Jacob Benesty ¹, Constantin Paleologu ^{2,*}, Cristian-Lucian Stanciu ², Ruxandra-Liana Costea ³, Laura-Maria Dogariu ² and Silviu Ciochină ²

- ¹ INRS-EMT, University of Quebec, Montreal, QC H5A 1K6, Canada; jacob.benesty@inrs.ca
- ² Department of Telecommunications, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania; cristian@comm.pub.ro (C.-L.S.); ldogariu@comm.pub.ro (L.-M.D.); silviu@comm.pub.ro (S.C.)
- ³ Department of Electrical Engineering, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania; ruxandra.costea@upb.ro
- * Correspondence: pale@comm.pub.ro; Tel.: +40-21-402-4687

Abstract: In linear system identification problems, the Wiener filter represents a popular tool and stands as an important benchmark. Nevertheless, it faces significant challenges when identifying long-length impulse responses. In order to address the related shortcomings, the solution presented in this paper is based on a third-order tensor decomposition technique, while the resulting sets of Wiener-Hopf equations are solved with the conjugate gradient (CG) method. Due to the decomposition-based approach, the number of coefficients (i.e., the parameter space of the filter) is greatly reduced, which results in operating with smaller data structures within the algorithm. As a result, improved robustness and accuracy can be achieved, especially in harsh scenarios (e.g., limited/incomplete sets of data and/or noisy conditions). Besides, the CG-based solution avoids matrix inversion operations, together with the related numerical and complexity issues. The simulation results are obtained in a network echo cancellation scenario and support the performance gain. In this context, the proposed iterative Wiener filter outperforms the conventional benchmark and also some previously developed counterparts that use matrix inversion or second-order tensor decompositions.

Keywords: system identification; Wiener filter; conjugate gradient method; tensor decomposition; Kronecker product; low-rank approximation; echo cancellation

1. Introduction

Linear system identification problems have to be worked out in the context of various applications [1,2], including echo cancellation, active noise control, interference reduction, and channel modeling among others. A benchmark technique used to address such problems is the well-known Wiener filter, which basically relies on solving a linear system (namely the Wiener–Hopf equations) using a set of statistics. The Wiener–Hopf equations involve a set of estimates for the covariance matrix of the input signal and the cross-correlation vector between the input and reference sequences. The problem is formulated following an optimization criterion in terms of minimizing the mean-squared error (MSE), while the error is defined as the difference between the reference sequence and the output signal. The resulting optimal filter also represents an important basis for the development of other related tools for system identification problems, such as adaptive filtering algorithms [3,4].

There are some inherent limitations associated with the conventional Wiener filter solution, which is obtained by directly solving (using matrix inversion) the Wiener–Hopf equations. First, the accuracy of the solution is highly influenced by the accuracy of the statistics' estimates. On the other hand, obtaining a reliable set of these estimates requires a large amount of data, i.e., much larger than the filter length. This could represent



Citation: Benesty, J.; Paleologu, C.; Stanciu, C.-L.; Costea, R.-L.; Dogariu, L.-M.; Ciochină, S. Wiener Filter Using the Conjugate Gradient Method and a Third-Order Tensor Decomposition. *Appl. Sci.* 2024, *14*, 2430. https:// doi.org/10.3390/app14062430

Academic Editor: Tomasz Figlus

Received: 27 November 2023 Revised: 12 March 2024 Accepted: 12 March 2024 Published: 13 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). a significant shortcoming when dealing with a limited (or incomplete) amount of data and/or a long-length filter. Second, the external noise (that is part of the reference signal) biases the Wiener filter solution, which becomes less accurate when the signal-to-noise ratio (SNR) decreases. This could be the case in noisy environments, where different types of perturbations are likely to emerge. Third, the conventional solution involves the covariance matrix inversion, which is a very challenging operation in terms of both computational complexity and numerical accuracy [5,6]. The difficulty could increase significantly when operating with long-length filters, which further entail large dimension matrices.

Most of the previously discussed limitations are connected to the length of the filter, which could be very large in many scenarios. For example, in applications like echo cancellation and noise reduction [7], the acoustic impulse responses to be identified have hundreds/thousands of coefficients when using the common sampling rate of 8 or 16 kHz. Therefore, dealing with such long-length filters could lead to significant limitations in terms of both the accuracy and complexity of the solution. In order to reformulate such highdimension system identification problems (with a large parameter space) more efficiently, a recently developed decomposition-based technique has been involved [8]. The main idea behind this technique is to exploit the low-rank feature of the system impulse response, in conjunction with its nearest Kronecker product (NKP) decomposition. As a result, a system identification problem featuring a large parameter space is designed as a combination of two shorter filters, with a significantly reduced number of coefficients. This further implies the operation with smaller matrices/vectors and, consequently, leads to improved robustness in terms of the accuracy of the final solution, even for the challenging cases mentioned above (e.g., a limited amount of data and/or low SNRs). Due to these important gains, the NKP-based approach has been involved in a wide range of applications, among which can be mentioned echo cancellation, adaptive beamforming, linear prediction, speech dereverberation, and microphone arrays, e.g., see [9-17] and the references therein.

Recently, the NKP technique has been applied in conjunction with a third-order tensor (TOT) decomposition of the impulse response [18], leading to a higher efficiency in terms of reducing the dimensionality of the system identification problem. This was not a straightforward extension of the low-rank approach presented in [8], since handling the rank of a tensor is a sensitive issue that usually involves different approximation techniques [19–28]. On the other hand, the solution proposed in [18] avoids such an approximation, by controlling and limiting the tensor rank to very small values. However, the resulting Wiener filter based on TOT decomposition solves the involved sets of Wiener–Hopf equations using the conventional approach, which relies on matrix inversion. Alternatively, different iterative techniques could be used to avoid such an operation [29–31], like the conjugate gradient (CG) method [32]. In [33], the CG algorithm has been applied in conjunction with the NKP-based technique from [8], showing improved performance. However, applying the CG method together with the TOT decomposition is a more challenging task, due to the particular connection between the three (shorter) component filters and the need for auxiliary variables within the algorithm.

Motivated by these aspects, in the current paper, we design an improved iterative version of the Wiener filter. The proposed algorithm involves the TOT-based decomposition, together with the CG method to solve three sets of Wiener–Hopf equations. As a result, it outperforms the counterpart version from [18], which uses the direct matrix inversion for solving the Wiener–Hopf equations, and also the CG-based solution from [33], which exploits the second-order NKP decomposition. Following this introduction, in Section 2 we provide some background on the conventional Wiener filter, the CG method (to avoid matrix inversion), and the TOT-based decomposition. Next, in Section 3, the proposed algorithm is developed. Simulation results provided in Section 4 support its performance and advantages compared to the existing solutions. The paper is summarized in Section 5, outlining the main conclusions and several perspectives for future works.

2. Conventional Wiener Filter, Conjugate Gradient Method, and Impulse Response Decomposition Based on a Third-Order Tensor

In this section, several backgrounds related to the upcoming developments are provided. First, we present the conventional Wiener filter for solving linear system identification problems. Next, the CG method is introduced as an efficient (iterative) alternative to avoid the matrix inversion required by the direct solution of Wiener–Hopf equations. Finally, the TOT-based decomposition of the impulse response is presented, outlining the main idea recently introduced in [18].

The main framework considered in this paper is related to a single-input single-output (SISO) linear system identification scenario, where all the involved signals are zero-mean and real-valued. In this context, the available signals are the input x(t) and the reference d(t), while t represents the discrete-time index. Under this scenario, there is a correlation between these two sequences, since the reference signal is obtained at the output of an unknown system driven by the input signal, while the output is corrupted by an additive noise, as shown in Figure 1. Thus ,

$$d(t) = \mathbf{h}^T \mathbf{x}(t) + v(t)$$
(1)
= $v(t) + v(t)$,

where the vector **h** contains the *L* coefficients of the unknown impulse response (with superscript ^{*T*} denoting transposition), $\mathbf{x}(t) = \begin{bmatrix} x(t) & x(t-1) & \cdots & x(t-L+1) \end{bmatrix}^T$ is a vector that contains the *L* most recent time samples of the input signal x(t), and v(t) is an additive noise, which is uncorrelated with x(t). In the second line of (1), $y(t) = \mathbf{h}^T \mathbf{x}(t)$ represents the output signal.



Figure 1. The reference signal obtained in a SISO scenario.

Based on the correlation between the reference sequence and the input signal, and following the MSE optimization criterion, an estimate of **h** can be obtained by solving the Wiener–Hopf equations [2], i.e.,

$$\mathbf{R}_{\mathbf{x}}\mathbf{h}_{\mathrm{W}}=\mathbf{r}_{\mathbf{x}d},\tag{2}$$

where

$$\mathbf{R}_{\mathbf{x}} = E\left[\mathbf{x}(t)\mathbf{x}^{T}(t)\right],\tag{3}$$

$$\mathbf{r}_{\mathbf{x}d} = E[\mathbf{x}(t)d(t)] \tag{4}$$

represent the covariance matrix of the input signal and the cross-correlation vector between the input and reference sequences, respectively, \mathbf{h}_W contains the coefficients of the Wiener filter (i.e., *L* parameters), while $E(\cdot)$ denotes mathematical expectation. Thus, the conventional Wiener filter results by using the matrix inversion operation, so that

$$\mathbf{h}_{\mathrm{W}} = \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}d}.$$
 (5)

In order to avoid matrix inversion, several alternative methods for solving (5) can be applied. The basic idea is to obtain the final solution in an iterative manner. In this context,

the CG method [32] represents a popular choice, being included in the category of exact line search methods [34,35]. Hence, considering the initialization $\mathbf{h}_{W(0)}$, an initial residual $\mathbf{z}_{(0)} = \mathbf{r}_{\mathbf{x}d} - \mathbf{R}_{\mathbf{x}}\mathbf{h}_{W(0)}$ can be computed. Also, in the initial step, it requires a conjugate vector $\mathbf{c}_{(0)} = \mathbf{z}_{(0)}$ and an auxiliary scalar $\gamma_{(0)} = \mathbf{z}_{(0)}^T \mathbf{z}_{(0)}$. Using this initialization, the CG algorithm runs for *k* steps, each one involving the relations:

$$\mathbf{q}_{(k)} = \mathbf{R}_{\mathbf{x}} \mathbf{c}_{(k-1)},\tag{6}$$

$$\boldsymbol{\alpha}_{(k)} = \frac{\boldsymbol{\gamma}_{(k-1)}}{\mathbf{c}_{(k-1)}^{T} \mathbf{q}_{(k)}},\tag{7}$$

$$\mathbf{h}_{\mathrm{W}(k)} = \mathbf{h}_{\mathrm{W}(k-1)} + \alpha_{(k)} \mathbf{c}_{(k-1)},\tag{8}$$

$$\mathbf{z}_{(k)} = \mathbf{z}_{(k-1)} - \alpha_{(k)} \mathbf{q}_{(k)}, \tag{9}$$

$$\gamma_{(k)} = \mathbf{z}_{(k)}^{\scriptscriptstyle L} \mathbf{z}_{(k)}, \tag{10}$$

$$\beta_{(k)} = \frac{\gamma_{(k)}}{\gamma_{(k-1)}},\tag{11}$$

$$\mathbf{c}_{(k)} = \mathbf{z}_{(k)} + \beta_{(k)} \mathbf{c}_{(k-1)}.$$
(12)

The stopping criterion can be related to a maximum number of updates (i.e., for k = 1, 2, ..., K) or a predefined threshold for the residual.

The convergence of the CG algorithm is influenced by the condition number of $\mathbf{R}_{\mathbf{x}}$ [5], i.e., the larger this number, the slower the convergence is. In order to improve the convergence rate, a preconditioning procedure could be applied to this matrix. There are different methods for choosing the so-called preconditioner (i.e., a matrix that multiplies $\mathbf{R}_{\mathbf{x}}$), like Jacobi, Gauss–Seidel, etc. Basically, the algorithm from (6)–(12) is reformulated, while incorporating the preconditioning directly into the iteration. On the other hand, this procedure involves an additional computational amount. Nevertheless, the purpose of this paper is not to analyze the influence of different preconditioners on the overall performance of the CG algorithm. Our primary goal is to develop the decomposition-based approach in conjunction with the CG method for solving the Wiener-Hopf equations. In this context, the main challenges are related to the connection between the component filters and the specific initialization (as will be shown in the next section), and not to the performance of the CG algorithm itself. Consequently, in the following, the basic CG algorithm from (6)-(12) is considered without preconditioning. However, in order to keep the positivedefinite character of the covariance matrix and to avoid any potential numerical/stability problems [6], it is recommended to add a very small positive constant to the elements of the main diagonal.

The maximum number of updates required by the Wiener filter using the CG method (namely WF-CG) to reach the solution of the conventional Wiener filter (WF) is generally much smaller compared to the filter length. This is supported in Figures 2 and 3, where the performances of the conventional WF and WF-CG are analyzed in two different scenarios for the identification of a network echo path of length L = 512 (using a sampling rate of 8 kHz). This impulse response results from the first cluster of ITU-T G168 Recommendation [36] concerning digital network echo cancellers. It contains 64 coefficients padded with zeros up to the full-length L. The required statistics ($\mathbf{R}_{\mathbf{x}}$ and $\mathbf{r}_{\mathbf{x}d}$) are estimated by averaging across N = ML data samples of x(t) and d(t), with M > 1. The reference signal is obtained according to (1), using a first-order autoregressive [AR(1)] process as input, which results from filtering white Gaussian noise through an AR(1) model; the pole of this model is set to 0.8. The additive noise is white and Gaussian, with SNR = σ_v^2 / σ_v^2 , where σ_y^2 and σ_v^2 stand for the variances of y(t) and v(t), respectively. The results are shown using a common performance measure involved in system identification scenarios, which is the normalized misalignment (in dB). It is defined as $20\log_{10}||\mathbf{h} - \mathbf{h}_W||_2 / ||\mathbf{h}||_2$ (where $\|\cdot\|_2$ denotes the Euclidean norm) and basically shows the "difference" between the true



impulse response and the estimated one. The lower this quantity, the better the accuracy of the estimate.

Figure 2. Normalized misalignment of the conventional WF and WF-CG using different amounts of data to estimate $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{r}_{\mathbf{x}d}$. These statistics are obtained by averaging across N = ML data samples (with L = 512 and different values of M), while SNR = 20 dB.



Figure 3. Normalized misalignment of the conventional WF and WF-CG for different SNRs. The estimates of the statistics $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{r}_{\mathbf{x}d}$ are obtained by averaging across N = 5L data samples, where L = 512.

In the first scenario considered in Figure 2, we evaluate the impact of using different amounts of data (N = ML) to estimate the statistics by varying the value of M. It can be noticed that a low amount of data significantly influences the accuracy of the Wiener solution. Nevertheless, the WF-CG converges toward the conventional WF after a small number of CG iterations (as compared to the filter length). Second, in Figure 3, the SNR influence is outlined. As expected, a lower SNR reduces the accuracy of the Wiener estimate. Similarly to the previous experiment, the WF-CG reaches the conventional WF for $K \ll L$. Both analyzed scenarios support the influence of the main factors that affect the behavior of

the Wiener filter, i.e., the amount of available data for estimating the statistics and the SNR level. Thus, it is of great importance to improve the overall performance and robustness related to these aspects.

In terms of computational complexity, the conventional Wiener solution based on matrix inversion requires $O(L^3)$ operations, while the iterative version that uses the CG method needs an $O(KL^2)$ amount, with $K \ll L$. Nevertheless, when identifying a long-length impulse response, a large value of *K* could be required for the CG iterations. This also represents a motivation for the dimensionality reduction of the problem, by reformulating a system identification scenario with a large parameter space (i.e., a large number of coefficients) as a combination of the estimates provided by shorter filters.

In this regard, the recent solution from [18] is based on a third-order tensor decomposition of the impulse response, namely TOT decomposition. As a result, the final estimate results as a combination (via the Kronecker product) of the coefficients associated with three filters, which are significantly shorter (as compared to the original impulse response). This idea is briefly explained in the following. First, let us consider that the length of the filter can be factorized as $L = L_1L_2$, with $L_1 \gg L_2$, so that the impulse response of the system results in [8]

$$\mathbf{h} = \sum_{i=1}^{L_2} \mathbf{h}_2^i \otimes \mathbf{h}_1^i.$$
(13)

Here, the shorter impulse responses \mathbf{h}_1^i and \mathbf{h}_2^i have the lengths L_1 and L_2 , respectively, while \otimes denotes the Kronecker product [37]. At this point, let us assume that \mathbf{h}_1^i is low rank [8]. Moreover, its length can be factorized as $L_1 = L_{11}L_{12}$ (with $L_{11} \ge L_{12}$). Consequently, the global impulse response results in

$$\mathbf{h} = \sum_{i=1}^{L_2} \sum_{j=1}^{P} \mathbf{h}_2^i \otimes \mathbf{h}_{12}^{ij} \otimes \mathbf{h}_{11}^{ij}, \qquad (14)$$

where the two impulse responses \mathbf{h}_{11}^{ij} and \mathbf{h}_{12}^{ij} have the lengths L_{11} and L_{12} , respectively, while $P < L_{12}$. It can be noticed that the coefficients of \mathbf{h} can be "rearranged" in the form of a third-order tensor, i.e.,

$$\mathcal{H} = \sum_{j=1}^{P} \left(\sum_{i=1}^{L_2} \mathbf{h}_{11}^{ij} \circ \mathbf{h}_{12}^{ij} \circ \mathbf{h}_2^i \right), \tag{15}$$

where \circ stands as the notation for the outer product. Furthermore, \mathcal{H} is in fact a sum of *P* third-order tensors, each one of rank L_2 [19]. As indicated in [18], the recommended values for L_2 are small (e.g., 2 or 3).

Summarizing, the identification of the global impulse response **h** of length *L* (i.e., with $L_{11}L_{12}L_2$ coefficients) is transformed into a combination of three (shorter) sets of impulse responses, i.e., \mathbf{h}_{11}^{ij} , \mathbf{h}_{12}^{ij} , and \mathbf{h}_2^i (for $i = 1, 2, ..., L_2$ and j = 1, 2, ..., P). As a result, the new parameter space of the filter involves only $PL_{11}L_2$, $PL_{12}L_2$, and L_2^2 coefficients, respectively. Since usually $P \ll L_{12}$ [18], the TOT-based decomposition approach leads to a significant dimensionality reduction, which is achieved especially when dealing with long-length filters (i.e., large values of *L*). While the conventional Wiener filter using matrix inversion involves a computational complexity proportional to $\mathcal{O}(L^3) = \mathcal{O}[(L_{11}^3 L_{12}^3 L_2^3)]$, the decomposition-based solution using the CG method combines the estimates of three shorter filters, which results in a computational complexity proportional to $\mathcal{O}[(PL_{11}L_2)^2 + (PL_{12}L_2)^2 + L_2^4]$, with $P \ll L_{12}$.

3. Iterative Wiener Filter Based on TOT and CG

The current section is dedicated to the development of the proposed solution, which results in the form of an iterative Wiener filter based on TOT decomposition and using the

CG method for solving the associated sets of Wiener–Hopf equations. For this purpose and a better readability of the upcoming developments, several preliminary elements from [18] and the specific notation are presented at the beginning of this section. These preliminaries are related to the TOT-based decomposition framework and the associated Wiener–Hopf equations. Further, the proposed solution is developed. The differences between the version from [18] and the current proposal based on the CG method are mainly related to (i) the specific initialization that involves auxiliary matrices and (ii) the connection between the component filters from one CG cycle to another within the main iterations of the proposed CG-based Wiener filter. Moreover, since the impulse responses from (14) have different lengths, the CG cycles corresponding to the component filters use different numbers of updates.

As shown in [18], the estimates of the component impulse responses from (14) can be obtained based on a multilinear optimization approach [38,39]. In other words, two of the component impulse responses are considered fixed, while optimizing the third (remaining) one. This approach leads to three sets of Wiener–Hopf equations, i.e.,

$$\left(\overline{\mathbf{G}}_{12,11}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{12,11} \underline{\mathbf{g}}_{2,W} = \left(\overline{\mathbf{G}}_{12,11}\right)^{T} \mathbf{r}_{\mathbf{x}d}, \tag{16}$$

$$\left(\overline{\mathbf{G}}_{2,11}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{2,11} \overline{\mathbf{g}}_{12,W} = \left(\overline{\mathbf{G}}_{2,11}\right)^{T} \mathbf{r}_{\mathbf{x}d},\tag{17}$$

$$\left(\overline{\underline{\mathbf{G}}}_{2,12}\right)^{T}\mathbf{R}_{\mathbf{x}}\overline{\underline{\mathbf{G}}}_{2,12}\underline{\overline{\mathbf{g}}}_{11,W} = \left(\overline{\underline{\mathbf{G}}}_{2,12}\right)^{T}\mathbf{r}_{\mathbf{x}d}.$$
(18)

The corresponding data structures and the associated notation are shown in Table 1, where $\mathbf{g}_{\star,W}$ generally denotes the estimate of \mathbf{h}_{\star} from (14), while $\mathbf{I}_{L_{\bullet}}$ is the identity matrix of size $L_{\bullet} \times L_{\bullet}$.

At this point, (16)–(18) are going to be solved with the CG method. The resulting solutions will then be sequentially iterated and combined (via the Kronecker product). Finally, the Wiener filter \mathbf{g}_{W} , which represents an estimate of \mathbf{h} , will be obtained as

$$\mathbf{g}_{W} = \sum_{i=1}^{L_{2}} \sum_{j=1}^{P} \mathbf{g}_{2,W}^{i} \otimes \mathbf{g}_{12,W}^{ij} \otimes \mathbf{g}_{11,W'}^{ij}$$
(19)

where $\mathbf{g}_{2,W}^{i}$, $\mathbf{g}_{12,W}^{ij}$, and $\mathbf{g}_{11,W}^{ij}$ are obtained from $\underline{\mathbf{g}}_{2,W}, \overline{\mathbf{g}}_{12,W}$ and $\overline{\mathbf{g}}_{11,W}$ respectively. All these steps of the designed algorithm are detailed in the following.

As mentioned before, the developed iterative Wiener filter is based on the TOT decomposition of the global impulse response, while the CG updates are used to efficiently solve (16), (17), and (18), respectively. To this purpose, the main iterations of the Wiener filter are denoted as superscripts ^(*n*), while the CG updates appear in subscripts _(*k*). The initialization of the algorithm concerns the three component filters, which are initially defined as

$$\underline{\mathbf{g}}_{2,W(K_2)}^{(0)} = \begin{bmatrix} \epsilon & \mathbf{0}_{L_2^2-1}^T \end{bmatrix}^T,$$
(20)

$$\overline{\underline{g}}_{12,W(K_1)}^{(0)} = \begin{bmatrix} \epsilon & \mathbf{0}_{PL_{12}L_2-1}^T \end{bmatrix}^T,$$
(21)

$$\overline{\mathbf{g}}_{11,W(K_1)}^{(0)} = \begin{bmatrix} \epsilon & \mathbf{0}_{PL_{11}L_2-1}^T \end{bmatrix}^T,$$
(22)

where K_1 and K_2 represent the maximum number of CG updates (for the component filters), ϵ is a very small positive number, while **0**• denotes an all-zeros vector with the length indicated in subscript. The reason for using $K_1 \neq K_2$ is that we are dealing with different lengths for the component filters. Among them, the length of $\underline{\mathbf{g}}_{2,W}$ (which has L_2^2 coefficients) could be significantly smaller, taking into account that $\overline{\underline{\mathbf{g}}}_{12,W}$ and $\overline{\underline{\mathbf{g}}}_{11,W}$ have $PL_{12}L_2$ and $PL_{11}L_2$ coefficients, respectively, while $L_2 \ll L_{11}L_{12}$.

Table 1. Specific Data Structures and Notation Related to Wiener–Hopf equations with TOT Decomposition.

$$\begin{split} & \frac{\text{Data structures and notation from (16) :}}{\overline{\mathbf{G}}_{12,11} = \left[\overline{\mathbf{G}}_{12,11}^{1} \overline{\mathbf{G}}_{12,11}^{2} \cdots \overline{\mathbf{G}}_{12,11}^{L_{2}} \right], \\ & \text{where } \overline{\mathbf{G}}_{12,11}^{i} = \sum_{j=1}^{p} \mathbf{G}_{12,11}^{ij}, i = 1, 2, \dots, L_{2}, \\ & \text{with } \mathbf{G}_{12,11}^{ij} = \mathbf{I}_{L_{2}} \otimes \mathbf{g}_{12,W}^{ij} \otimes \mathbf{g}_{11}^{ij,W}, i = 1, 2, \dots, L_{2}, j = 1, 2, \dots, P \\ & \underline{\mathbf{g}}_{2,W} = \left[\left(\mathbf{g}_{2,W}^{1} \right)^{T} \left(\mathbf{g}_{2,W}^{2} \right)^{T} \cdots \left(\mathbf{g}_{2,W}^{L_{2}} \right)^{T} \right]^{T} \\ & \underline{\text{Data structures and notation from (17) :}} \\ & \overline{\mathbf{G}}_{2,11} = \left[\overline{\mathbf{G}}_{2,11}^{1} \overline{\mathbf{G}}_{2,11}^{2} \cdots \overline{\mathbf{G}}_{2,11}^{L_{2}} \right], \\ & \text{where } \overline{\mathbf{G}}_{2,11}^{i} = \left[\mathbf{G}_{2,11}^{i} \mathbf{G}_{2,11}^{2} \cdots \mathbf{G}_{2,11}^{i} \right], i = 1, 2, \dots, L_{2} \\ & \text{with } \mathbf{G}_{2,11}^{ij} = \mathbf{g}_{2,W}^{i} \otimes \mathbf{I}_{L_{12}} \otimes \mathbf{g}_{11,W}^{ij}, i = 1, 2, \dots, L_{2}, j = 1, 2, \dots, P \\ & \overline{\mathbf{g}}_{12,W} = \left[\left(\overline{\mathbf{g}}_{12,W}^{1} \right)^{T} \left(\overline{\mathbf{g}}_{12,W}^{2} \right)^{T} \cdots \left(\overline{\mathbf{g}}_{12,W}^{L_{2}} \right)^{T} \right]^{T} \\ & \text{where } \overline{\mathbf{g}}_{12,W}^{i} = \left[\left(\mathbf{g}_{12,W}^{i1} \right)^{T} \left(\mathbf{g}_{12,W}^{i2} \right)^{T} \cdots \left(\mathbf{g}_{12,W}^{ip} \right)^{T} \right]^{T}, i = 1, 2, \dots, L_{2} \\ & \text{Data structures and notation from (18) :} \\ & \overline{\mathbf{G}}_{2,12} = \left[\overline{\mathbf{G}}_{2,12}^{1} \overline{\mathbf{G}}_{2,12}^{2} \cdots \overline{\mathbf{G}}_{2,12}^{L_{2}} \right], \\ & \text{where } \overline{\mathbf{g}}_{12,W}^{i} = \left[\mathbf{G}_{2,12}^{i1} \mathbf{G}_{2,12}^{i2} \cdots \mathbf{G}_{2,12}^{L_{2}} \right], \\ & \text{where } \overline{\mathbf{G}}_{2,12}^{i} = \mathbf{g}_{2,W}^{i} \otimes \mathbf{g}_{12,W}^{ij} \otimes \mathbf{I}_{L_{11}}, i = 1, 2, \dots, L_{2}, \\ & \text{where } \overline{\mathbf{G}}_{2,12}^{i} = \left[\mathbf{G}_{2,12}^{i1} \mathbf{G}_{2,12}^{i2} \cdots \mathbf{G}_{2,12}^{i2} \right], \\ & \text{where } \overline{\mathbf{G}}_{1,12}^{i} = \mathbf{g}_{2,W}^{i} \otimes \mathbf{g}_{12,W}^{ij} \otimes \mathbf{I}_{L_{11}}, i = 1, 2, \dots, L_{2}, \\ & \text{with } \mathbf{G}_{2,12}^{ij} = \mathbf{g}_{2,W}^{ij} \otimes \mathbf{g}_{12,W}^{ij} \otimes \mathbf{I}_{L_{11}}, i = 1, 2, \dots, L_{2}, \\ & \text{where } \overline{\mathbf{G}}_{1,1W}^{ij} = \left[\left(\mathbf{g}_{11,W}^{ij} \right)^{T} \cdots \left(\mathbf{g}_{11,W}^{ij} \right)^{T} \right]^{T}, \\ & \text{where } \overline{\mathbf{g}}_{11,W}^{ij} = \left[\left(\mathbf{g}_{11,W}^{ij} \right)^{T} (\mathbf{g}_{2,1}^{ij} \right)^{T} \cdots \left(\mathbf{g}_{11,W}^{ij} \right]^{T} \right]^{T} \\ & \text{where } \overline{\mathbf{g}}_{11,W}^{ij} = \left[\left(\mathbf{g}_{11,W}^{ij} \right)^{T$$

At this point, we need to introduce the auxiliary matrices (for $i = 1, 2, ..., L_2$ and j = 1, 2, ..., P):

$$\mathbf{M}_{12,11}^{ij(0)} = \mathbf{I}_{L_2} \otimes \begin{bmatrix} \epsilon & \mathbf{0}_{L_{12}-1}^T \end{bmatrix}^T \otimes \begin{bmatrix} \epsilon & \mathbf{0}_{L_{11}-1}^T \end{bmatrix}^T,$$
(23)

$$\mathbf{M}_{11}^{ij(0)} = \mathbf{I}_{L_{12}} \otimes \begin{bmatrix} \epsilon & \mathbf{0}_{L_{11}-1}^T \end{bmatrix}^T,$$
(24)

which will further facilitate the definition of matrices $\overline{\mathbf{G}}_{12,11}$ and $\overline{\mathbf{G}}_{2,11}$. Hence, in each main iteration $^{(n)}$ of the algorithm, we first construct using (23):

$$\mathbf{G}_{12,11}^{ij(n)} = \mathbf{M}_{12,11}^{ij(n-1)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P,$$
(25)

$$\overline{\mathbf{G}}_{12,11}^{i(n)} = \sum_{j=1}^{P} \mathbf{G}_{12,11}^{ij(n)}, \ i = 1, 2, \dots, L_2,$$
(26)

$$\overline{\underline{\mathbf{G}}}_{12,11}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{12,11}^{1(n)} & \overline{\mathbf{G}}_{12,11}^{2(n)} & \cdots & \overline{\mathbf{G}}_{12,11}^{L_2(n)} \end{bmatrix}.$$
(27)

These allow us to compute

$$\mathbf{R}_{2}^{(n)} = \left(\overline{\underline{\mathbf{G}}}_{12,11}^{(n)}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\underline{\mathbf{G}}}_{12,11}^{(n)}, \tag{28}$$

$$\mathbf{r}_{2}^{(n)} = \left(\overline{\underline{\mathbf{G}}}_{12,11}^{(n)}\right)^{T} \mathbf{r}_{\mathbf{x}d}.$$
(29)

The structures from (28) and (29) are used to process (16) with the CG method. Consequently, using (20), the initial settings are

$$\underline{g}_{2,W(0)}^{(n)} = \underline{g}_{2,W(K_2)}^{(n-1)},$$
(30)

$$\mathbf{z}_{2(0)}^{(n)} = \mathbf{r}_{2}^{(n)} - \mathbf{R}_{2}^{(n)} \underline{\mathbf{g}}_{2,W(0)}^{(n)},$$
(31)

$$\mathbf{c}_{2(0)}^{(n)} = \mathbf{z}_{2(0)}^{(n)},\tag{32}$$

$$\gamma_{2(0)}^{(n)} = \left(\mathbf{z}_{2(0)}^{(n)}\right)^T \mathbf{z}_{2(0)}^{(n)}.$$
(33)

Next, for $k_2 = 1, 2, ..., K_2$, we perform similar to (6)–(12):

$$\mathbf{q}_{2(k_2)}^{(n)} = \mathbf{R}_2^{(n)} \mathbf{c}_{2(k_2-1)}^{(n)}, \tag{34}$$

$$\alpha_{2(k_2)}^{(n)} = \frac{\gamma_{2(k_2-1)}^{(n)}}{\left(\mathbf{c}_{2(k_2-1)}^{(n)}\right)^T \mathbf{q}_{2(k_2)}^{(n)}},\tag{35}$$

$$\underline{\mathbf{g}}_{2,W(k_2)}^{(n)} = \underline{\mathbf{g}}_{2,W(k_2-1)}^{(n)} + \alpha_{2(k_2)}^{(n)} \mathbf{c}_{2(k_2-1)'}^{(n)}$$
(36)

$$\mathbf{z}_{2(k_{2})}^{(n)} = \mathbf{z}_{2(k_{2}-1)}^{(n)} - \alpha_{2(k_{2})}^{(n)} \mathbf{q}_{2(k_{2})}^{(n)},$$
(37)

$$\gamma_{2(k_2)}^{(n)} = \left(\mathbf{z}_{2(k_2)}^{(n)}\right)^{I} \mathbf{z}_{2(k_2)}^{(n)},\tag{38}$$

$$\beta_{2(k_2)}^{(n)} = \frac{\gamma_{2(k_2)}^{(n)}}{\gamma_{2(k_2-1)}^{(n)}},\tag{39}$$

$$\mathbf{c}_{2(k_2)}^{(n)} = \mathbf{z}_{2(k_2)}^{(n)} + \beta_{2(k_2)}^{(n)} \mathbf{c}_{2(k_2-1)}^{(n)}.$$
(40)

The final solution $\underline{\mathbf{g}}_{2,W(K_2)}^{(n)}$ will represent the initialization for the CG cycle associated with this filter [similar to (30)] in the next main iteration of the algorithm. Also, it is decomposed as

$$\underline{\mathbf{g}}_{2,W(K_2)}^{(n)} = \begin{bmatrix} \left(\mathbf{g}_{2,W(K_2)}^{1(n)} \right)^T & \left(\mathbf{g}_{2,W(K_2)}^{2(n)} \right)^T & \cdots & \left(\mathbf{g}_{2,W(K_2)}^{L_2(n)} \right)^T \end{bmatrix}^T, \quad (41)$$

which further allows the evaluation of

$$\mathbf{G}_{2,11}^{ij(n)} = \mathbf{g}_{2,W(K_2)}^{i(n)} \otimes \mathbf{M}_{11}^{ij(n-1)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P,$$
(42)

$$\overline{\mathbf{G}}_{2,11}^{i(n)} = \begin{bmatrix} \mathbf{G}_{2,11}^{i1(n)} & \mathbf{G}_{2,11}^{i2(n)} & \cdots & \mathbf{G}_{2,11}^{iP(n)} \end{bmatrix}, \ i = 1, 2, \dots, L_2,$$
(43)

$$\overline{\mathbf{G}}_{2,11}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{2,11}^{1(n)} & \overline{\mathbf{G}}_{2,11}^{2(n)} & \cdots & \overline{\mathbf{G}}_{2,11}^{L_2(n)} \end{bmatrix},$$
(44)

so that we can compute

$$\mathbf{R}_{12}^{(n)} = \left(\overline{\mathbf{G}}_{2,11}^{(n)}\right)^T \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{2,11}^{(n)},\tag{45}$$

$$\mathbf{r}_{12}^{(n)} = \left(\overline{\mathbf{G}}_{2,11}^{(n)}\right)^T \mathbf{r}_{\mathbf{x}d}.$$
(46)

The notation from (45) and (46) is used to process (17) with the CG updates. Hence, in this step, we follow the initial settings from (21), so that

$$\underline{\overline{\mathbf{g}}}_{12,\mathsf{W}(0)}^{(n)} = \underline{\overline{\mathbf{g}}}_{12,\mathsf{W}(K_1)}^{(n-1)},\tag{47}$$

$$\mathbf{z}_{12(0)}^{(n)} = \mathbf{r}_{12}^{(n)} - \mathbf{R}_{12}^{(n)} \overline{\mathbf{g}}_{12,W(0)'}^{(n)}$$
(48)

$$\mathbf{c}_{12(0)}^{(n)} = \mathbf{z}_{12(0)}^{(n)},\tag{49}$$

$$\gamma_{12(0)}^{(n)} = \left(\mathbf{z}_{12(0)}^{(n)}\right)^T \mathbf{z}_{12(0)}^{(n)}.$$
(50)

Consequently, the CG cycle for the second filter is defined by the relations:

$$\mathbf{q}_{12(k_1)}^{(n)} = \mathbf{R}_{12}^{(n)} \mathbf{c}_{12(k_1-1)}^{(n)}, \tag{51}$$

$$\alpha_{12(k_1)}^{(n)} = \frac{\gamma_{12(k_1-1)}^{(n)}}{\left(\mathbf{c}_{12(k_1-1)}^{(n)}\right)^T \mathbf{q}_{12(k_1)}^{(n)}},\tag{52}$$

$$\overline{\mathbf{g}}_{12,\mathsf{W}(k_1)}^{(n)} = \overline{\mathbf{g}}_{12,\mathsf{W}(k_1-1)}^{(n)} + \alpha_{12(k_1)}^{(n)} \mathbf{c}_{12(k_1-1)}^{(n)}, \tag{53}$$

$$\mathbf{z}_{12(k_1)}^{(n)} = \mathbf{z}_{12(k_1-1)}^{(n)} - \alpha_{12(k_1)}^{(n)} \mathbf{q}_{12(k_1)}^{(n)},$$
(54)

$$\gamma_{12(k_1)}^{(n)} = \left(\mathbf{z}_{12(k_1)}^{(n)}\right)^{I} \mathbf{z}_{12(k_1)}^{(n)},$$
(55)

$$\beta_{12(k_1)}^{(n)} = \frac{\gamma_{12(k_1)}^{(n)}}{\gamma_{12(k_1-1)}^{(n)}},\tag{56}$$

$$\mathbf{c}_{12(k_1)}^{(n)} = \mathbf{z}_{12(k_1)}^{(n)} + \beta_{12(k_1)}^{(n)} \mathbf{c}_{12(k_1-1)}^{(n)},$$
(57)

for $k_1 = 1, 2, ..., K_1$. The final solution $\overline{\mathbf{g}}_{12,W(K_1)}^{(n)}$ will represent the initial setting in the next main iteration of the algorithm [similar to (47)]. The decomposition of this impulse response is performed in two steps, i.e.,

$$\underline{\overline{\mathbf{g}}}_{12,\mathsf{W}(K_1)}^{(n)} = \left[\left(\overline{\mathbf{g}}_{12,\mathsf{W}(K_1)}^{1(n)} \right)^T \left(\overline{\mathbf{g}}_{12,\mathsf{W}(K_1)}^{2(n)} \right)^T \cdots \left(\overline{\mathbf{g}}_{12,\mathsf{W}(K_1)}^{L_2(n)} \right)^T \right]_T^T,$$
(58)

$$\overline{\mathbf{g}}_{12,W(K_1)}^{i(n)} = \left[\left(\mathbf{g}_{12,W(K_1)}^{i1(n)} \right)^T \left(\mathbf{g}_{12,W(K_1)}^{i2(n)} \right)^T \cdots \left(\mathbf{g}_{12,W(K_1)}^{iP(n)} \right)^T \right]^I, \ i = 1, 2, \dots, L_2.$$
(59)

At this point, having the components from (41) and (59), we continue with the development associated with the last component filter, starting with the evaluation of

$$\mathbf{G}_{2,12}^{ij(n)} = \mathbf{g}_{2,W(K_2)}^{i(n)} \otimes \mathbf{g}_{12,W(K_1)}^{ij(n)} \otimes \mathbf{I}_{L_{11}}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P,$$
(60)

$$\overline{\mathbf{G}}_{2,12}^{i(n)} = \begin{bmatrix} \mathbf{G}_{2,12}^{i1(n)} & \mathbf{G}_{2,12}^{i2(n)} & \cdots & \mathbf{G}_{2,12}^{iP(n)} \end{bmatrix}, \ i = 1, 2, \dots, L_2,$$
(61)

$$\overline{\mathbf{G}}_{2,12}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{2,12}^{1(n)} & \overline{\mathbf{G}}_{2,12}^{2(n)} & \cdots & \overline{\mathbf{G}}_{2,12}^{L_2(n)} \end{bmatrix}.$$
(62)

Therefore, introducing the notation:

$$\mathbf{R}_{11}^{(n)} = \left(\overline{\mathbf{G}}_{2,12}^{(n)}\right)^T \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{2,12}^{(n)}, \tag{63}$$

$$\mathbf{r}_{11}^{(n)} = \left(\overline{\mathbf{G}}_{2,12}^{(n)}\right)^{T} \mathbf{r}_{\mathbf{x}d},\tag{64}$$

we can further process (18) with the CG method. To this purpose, the initialization relies on (22), so that the settings are

$$\overline{\mathbf{g}}_{11,W(0)}^{(n)} = \overline{\mathbf{g}}_{11,W(K_1)}^{(n-1)},\tag{65}$$

$$\mathbf{z}_{11(0)}^{(n)} = \mathbf{r}_{11}^{(n)} - \mathbf{R}_{11}^{(n)} \overline{\mathbf{g}}_{11,W(0)'}^{(n)}$$
(66)

$$\mathbf{c}_{11(0)}^{(n)} = \mathbf{z}_{11(0)}^{(n)},\tag{67}$$

$$\gamma_{11(0)}^{(n)} = \left(\mathbf{z}_{11(0)}^{(n)}\right)^T \mathbf{z}_{11(0)}^{(n)}.$$
(68)

Thus, for $k_1 = 1, 2, ..., K_1$, the CG cycle for the third filter consists of the relations:

$$\mathbf{q}_{11(k_1)}^{(n)} = \mathbf{R}_{11}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)}, \tag{69}$$

$$\alpha_{11(k_1)}^{(n)} = \frac{\gamma_{11(k_1-1)}^{(n)}}{\left(\mathbf{c}_{11(k_1-1)}^{(n)}\right)^T \mathbf{q}_{11(k_1)}^{(n)}},\tag{70}$$

$$\overline{\mathbf{g}}_{11,W(k_1)}^{(n)} = \overline{\mathbf{g}}_{11,W(k_1-1)}^{(n)} + \alpha_{11(k_1)}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)},$$
(71)

$$\mathbf{z}_{11(k_1)}^{(n)} = \mathbf{z}_{11(k_1-1)}^{(n)} - \alpha_{11(k_1)}^{(n)} \mathbf{q}_{11(k_1)}^{(n)},$$
(72)

$$\gamma_{11(k_1)}^{(n)} = \left(\mathbf{z}_{11(k_1)}^{(n)}\right)^T \mathbf{z}_{11(k_1)}^{(n)},\tag{73}$$

$$\beta_{11(k_1)}^{(n)} = \frac{\gamma_{11(k_1)}^{(n)}}{\gamma_{11(k_1-1)}^{(n)}},\tag{74}$$

$$\mathbf{c}_{11(k_1)}^{(n)} = \mathbf{z}_{11(k_1)}^{(n)} + \beta_{11(k_1)}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)}.$$
(75)

The decomposition of the final solution $\overline{\mathbf{g}}_{11,W(K_1)}^{(n)}$ results in

$$\overline{\mathbf{g}}_{11,W(K_1)}^{(n)} = \begin{bmatrix} \left(\overline{\mathbf{g}}_{11,W(K_1)}^{1(n)}\right)^T & \left(\overline{\mathbf{g}}_{11,W(K_1)}^{2(n)}\right)^T & \cdots & \left(\overline{\mathbf{g}}_{11,W(K_1)}^{L_2(n)}\right)^T \end{bmatrix}^T,$$
(76)

$$\overline{\mathbf{g}}_{11,W(K_1)}^{i(n)} = \left[\left(\mathbf{g}_{11,W(K_1)}^{i1(n)} \right)^T \left(\mathbf{g}_{11,W(K_1)}^{i2(n)} \right)^T \cdots \left(\mathbf{g}_{11,W(K_1)}^{iP(n)} \right)^T \right]^i, i = 1, 2, \dots, L_2,$$
(77)

and provides the final elements for evaluating the estimated impulse response based on (19). Also, $\overline{\mathbf{g}}_{11,W(K_1)}^{(n)}$ represents the initialization for the next main iteration of the algorithm, according to (65).

Summarizing, using (41), (59), and (77), we obtain

$$\mathbf{g}_{W}^{(n)} = \sum_{i=1}^{L_2} \sum_{j=1}^{P} \mathbf{g}_{2,W(K_2)}^{i(n)} \otimes \mathbf{g}_{12,W(K_1)}^{ij(n)} \otimes \mathbf{g}_{11,W(K_1)}^{ij(n)}.$$
(78)

Finally, using the same components, we evaluate the auxiliary matrices (for $i = 1, 2, ..., L_2$ and j = 1, 2, ..., P):

$$\mathbf{M}_{12,11}^{ij(n)} = \mathbf{I}_{L_2} \otimes \mathbf{g}_{12,W(K_1)}^{ij(n)} \otimes \mathbf{g}_{11,W(K_1)}^{ij(n)},$$
(79)

$$\mathbf{M}_{11}^{ij(n)} = \mathbf{I}_{L_{12}} \otimes \mathbf{g}_{11,W(K_1)}^{ij(n)}.$$
(80)

These will be used in the next main iteration of the algorithm, in order to compute the structures from (25) and (42), respectively.

The resulting iterative Wiener filter (IWF) is based on TOT decomposition and uses the CG method, which will be referred to as IWF-TOT-CG. Its main steps are summarized in Table 2, while the CG cycles for solving (16)–(18) are detailed in Table 3.

Table 2. Iterative Wiener Filter Based on a Third-Order Tensor Decomposition and Using the CG Method (IWF-TOT-CG).

```
<u>Data</u>: \mathbf{R}_{\mathbf{x}}, \mathbf{r}_{\mathbf{x}d} (estimated statistics based on N data samples)
  L = L_{11}L_{12}L_2, \ L_{12} \le L_{11}, \ L_2 \ll L_{11}L_{12}, \ P < L_{12}
 Initialization:
 \underline{\mathbf{g}}_{2,\mathsf{W}(K_2)}^{(0)} = \begin{bmatrix} \epsilon & \mathbf{0}_{L_2^2-1}^T \end{bmatrix}^T, \ 0 < \epsilon \le 1
\frac{\underline{\mathbf{g}}_{2,W(K_2)}}{\underline{\mathbf{g}}_{12,W(K_1)}^{(0)}} = \begin{bmatrix} \epsilon & \mathbf{0}_{PL_{12}L_2-1}^T \end{bmatrix}_T^T
 \mathbf{\overline{g}}_{11,W(\mathbf{K}_1)}^{(0)} = \begin{bmatrix} \boldsymbol{\epsilon} & \mathbf{0}_{PL_{11}L_{2}-1}^T \end{bmatrix}^T \\ \text{for } i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P: \\ \mathbf{M}_{12,11}^{ij(0)} = \mathbf{I}_{L_2} \otimes \begin{bmatrix} \boldsymbol{\epsilon} & \mathbf{0}_{L_{12}-1}^T \end{bmatrix}^T \otimes \begin{bmatrix} \boldsymbol{\epsilon} & \mathbf{0}_{L_{11}-1}^T \end{bmatrix}^T \\ \mathbf{M}_{11}^{ij(0)} = \mathbf{I}_{L_{12}} \otimes \begin{bmatrix} \boldsymbol{\epsilon} & \mathbf{0}_{L_{11}-1}^T \end{bmatrix}^T 
  For n = 1, 2, ...
 \mathbf{G}_{12,11}^{ij(n)} = \mathbf{M}_{12,11}^{ij(n-1)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P
 \overline{\mathbf{G}}_{12,11}^{i(n)} = \sum_{j=1}^{p} \mathbf{G}_{12,11}^{ij(n)}, i = 1, 2, \dots, L_2
 \overline{\mathbf{G}}_{12,11}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{12,11}^{1(n)} & \overline{\mathbf{G}}_{12,11}^{2(n)} & \cdots & \overline{\mathbf{G}}_{12,11}^{L_2(n)} \end{bmatrix}
 \underline{\mathbf{g}}_{2,W(0)}^{(n)} = \underline{\underline{g}}_{2,W(K_2)}^{(n-1)}
 Solve (16) \xrightarrow{\text{CG } (K_2)}{\text{(see Table 3)}} \underbrace{\mathbf{g}_{2,W(K_2)}^{(n)}} \xrightarrow{(41)} \mathbf{g}_{2,W(K_2)}^{i(n)}, i = 1, 2, \dots, L_2
\mathbf{G}_{2,11}^{i(n)} = \mathbf{g}_{2,W(K_2)}^{i(n)} \otimes \mathbf{M}_{11}^{ij(n-1)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P
\overline{\mathbf{G}}_{2,11}^{i(n)} = \begin{bmatrix} \mathbf{G}_{2,11}^{i(n)} & \mathbf{G}_{2,11}^{i2(n)} & \cdots & \mathbf{G}_{2,11}^{iP(n)} \end{bmatrix}, \ i = 1, 2, \dots, L_2
\underline{\mathbf{G}}_{2,11}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{2,11}^{1(n)} & \overline{\mathbf{G}}_{2,11}^{i2(n)} & \cdots & \overline{\mathbf{G}}_{2,11}^{iP(n)} \end{bmatrix}, \ i = 1, 2, \dots, L_2
\underline{\mathbf{G}}_{2,11}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{2,11}^{1(n)} & \overline{\mathbf{G}}_{2,11}^{i2(n)} & \cdots & \overline{\mathbf{G}}_{2,11}^{iP(n)} \end{bmatrix}, \ i = 1, 2, \dots, L_2
 Solve (17) \xrightarrow{\text{CG}(K_1)}_{\text{(see Table 3)}} \underline{\overline{\mathbf{g}}}_{12,W(K_1)}^{(n)} \xrightarrow{(58),(59)} \mathbf{g}_{12,W(K_1)}^{ij(n)}, i = 1, 2, \dots, L_2, j = 1, 2, \dots, P
 \mathbf{G}_{2,12}^{ij(n)} = \mathbf{g}_{2,W(K_2)}^{i(n)} \otimes \mathbf{g}_{12,W(K_1)}^{ij(n)} \otimes \mathbf{I}_{L_{11}}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P

\overline{\mathbf{G}}_{2,12}^{i(n)} = \begin{bmatrix} \mathbf{G}_{2,12}^{i(1n)} & \mathbf{G}_{2,12}^{i(2n)} & \cdots & \mathbf{G}_{2,12}^{iP(n)} \\ \mathbf{G}_{2,12}^{i(1n)} & \mathbf{G}_{2,12}^{i(2n)} & \cdots & \mathbf{G}_{2,12}^{iP(n)} \end{bmatrix}, i = 1, 2, \dots, L_2

\overline{\mathbf{G}}_{2,12}^{(n)} = \begin{bmatrix} \overline{\mathbf{G}}_{2,12}^{1(n)} & \overline{\mathbf{G}}_{2,12}^{2(1)} & \cdots & \overline{\mathbf{G}}_{2,12}^{L_2(n)} \end{bmatrix}

 \overline{\mathbf{g}}_{11,W(0)}^{(n)} \stackrel{\mathsf{L}}{=} \overline{\mathbf{g}}_{11,W(K_1)}^{(n-1)}
 Solve (18) \xrightarrow{\text{CG}(K_1)}_{\text{(see Table 3)}} \overline{\underline{\mathbf{g}}}_{11,W(K_1)}^{(n)} \xrightarrow{(76),(77)} \mathbf{g}_{11,W(K_1)}^{ij(n)}, i = 1, 2, \dots, L_2, j = 1, 2, \dots, P
 \mathbf{M}_{12,11}^{ij(n)} = \mathbf{I}_{L_2} \otimes \mathbf{g}_{12,W(K_1)}^{ij(n)} \otimes \mathbf{g}_{11,W(K_1)}^{ij(n)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P
 \mathbf{M}_{11}^{ij(n)} = \mathbf{I}_{L_{12}} \otimes \mathbf{g}_{11,W(K_1)}^{ij(n)}, \ i = 1, 2, \dots, L_2, \ j = 1, 2, \dots, P
 \mathbf{g}_{\mathsf{W}}^{(n)} = \sum_{i=1}^{L_2} \sum_{j=1}^{p} \mathbf{g}_{2,\mathsf{W}(K_2)}^{(n)} \otimes \mathbf{g}_{12,\mathsf{W}(K_1)}^{ij(n)} \otimes \mathbf{g}_{11,\mathsf{W}(K_1)}^{ij(n)}
```

Table 3. CG Solutions of the Wiener-Hopf Equations within IWF-TOT-CG.

Solution of (16): $\overline{\mathbf{R}_{2}^{(n)} = \left(\overline{\mathbf{G}}_{12,11}^{(n)}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{12,11}^{(n)}, \ \mathbf{r}_{2}^{(n)} = \left(\overline{\mathbf{G}}_{12,11}^{(n)}\right)^{T} \mathbf{r}_{\mathbf{x}d}}$ $\underline{\mathbf{g}}_{2,W(0)}^{(n)} = \underline{\mathbf{g}}_{2,W(K_2)}^{(n-1)}, \ \mathbf{z}_{2(0)}^{(n)} = \mathbf{r}_2^{(n)} - \mathbf{R}_2^{(n)} \underline{\mathbf{g}}_{2,W(0)}^{(n)}, \ \mathbf{c}_{2(0)}^{(n)} = \mathbf{z}_{2(0)}^{(n)}, \ \gamma_{2(0)}^{(n)} = \left(\mathbf{z}_{2(0)}^{(n)}\right)^T \mathbf{z}_{2(0)}^{(n)}$ For $k_2 = 1, 2, \dots, K_2$: $\begin{aligned} \mathbf{p}_{2(k_{2})}^{(n)} &= \mathbf{R}_{2}^{(n)} \mathbf{c}_{2(k_{2}-1)}^{(n)}, \, \alpha_{2(k_{2})}^{(n)} = \frac{\gamma_{2(k_{2}-1)}^{(n)}}{\left(\mathbf{c}_{2(k_{2}-1)}^{(n)}\right)^{T} \mathbf{q}_{2(k_{2})}^{(n)}} \\ \underline{\mathbf{g}}_{2,\mathbf{W}(k_{2})}^{(n)} &= \underline{\mathbf{g}}_{2,\mathbf{W}(k_{2}-1)}^{(n)} + \alpha_{2(k_{2})}^{(n)} \mathbf{c}_{2(k_{2}-1)}^{(n)} \\ \mathbf{z}_{2(k_{2})}^{(n)} &= \mathbf{z}_{2(k_{2}-1)}^{(n)} - \alpha_{2(k_{2})}^{(n)} \mathbf{q}_{2(k_{2})}^{(n)} \\ \gamma_{2(k_{2})}^{(n)} &= \left(\mathbf{z}_{2(k_{2})}^{(n)}\right)^{T} \mathbf{z}_{2(k_{2})}^{(n)}, \, \beta_{2(k_{2})}^{(n)} = \frac{\gamma_{2(k_{2}-1)}^{(n)}}{\gamma_{2(k_{2}-1)}^{(n)}} \end{aligned}$ $\begin{aligned} \mathbf{c}_{2(k_{2})}^{(n)} &= \mathbf{z}_{2(k_{2})}^{(n)} + \beta_{2(k_{2})}^{(n)} \mathbf{c}_{2(k_{2}-1)}^{(n)} \\ \frac{\text{Solution of } (17)}{\mathbf{R}_{12}} &: \\ \mathbf{R}_{12}^{(n)} &= \left(\overline{\mathbf{G}}_{2,11}^{(n)}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{2,11}^{(n)}, \ \mathbf{r}_{12}^{(n)} &= \left(\overline{\mathbf{G}}_{2,11}^{(n)}\right)^{T} \mathbf{r}_{\mathbf{x}d} \end{aligned}$ $\underline{\overline{g}}_{12,W(0)}^{(n)} = \underline{\overline{g}}_{12,W(K_1)}^{(n-1)}, \mathbf{z}_{12(0)}^{(n)} = \mathbf{r}_{12}^{(n)} - \mathbf{R}_{12}^{(n)} \underline{\overline{g}}_{12,W(0)}^{(n)}, \mathbf{c}_{12(0)}^{(n)} = \mathbf{z}_{12(0)}^{(n)}, \gamma_{12(0)}^{(n)} = \left(\mathbf{z}_{12(0)}^{(n)}\right)^T \mathbf{z}_{12(0)}^{(n)}$ For $k_1 = 1, 2, \dots, K_1$: $\mathbf{q}_{12(k_1)}^{(n)} = \mathbf{R}_{12}^{(n)} \mathbf{c}_{12(k_1-1)}^{(n)}, \ \boldsymbol{\alpha}_{12(k_1)}^{(n)} = \frac{\gamma_{12(k_1-1)}^{(n)}}{\left(\mathbf{c}_{12(k_1-1)}^{(n)}\right)^T \mathbf{q}_{12(k_1)}^{(n)}}$ $\overline{\mathbf{g}}_{12,W(k_1)}^{(n)} = \overline{\mathbf{g}}_{12,W(k_1-1)}^{(n)} + \alpha_{12(k_1)}^{(n)} \mathbf{c}_{12(k_1-1)}^{(n)} \\ \mathbf{z}_{12(k_1)}^{(n)} = \mathbf{z}_{12(k_1-1)}^{(n)} - \alpha_{12(k_1)}^{(n)} \mathbf{q}_{12(k_1)}^{(n)}$ $\gamma_{12(k_1)}^{(n)} = \left(\mathbf{z}_{12(k_1)}^{(n)}\right)^T \mathbf{z}_{12(k_1)}^{(n)}, \ \beta_{12(k_1)}^{(n)} = \frac{\gamma_{12(k_1)}^{(n)}}{\gamma_{12(k_1-1)}^{(n)}}$ $\mathbf{c}_{12(k_{1})}^{(n)} = \mathbf{z}_{12(k_{1})}^{(n)} + \beta_{12(k_{1})}^{(n)} \mathbf{c}_{12(k_{1}-1)}^{(n)}$ $\mathbf{solution of (18):}$ $\mathbf{R}_{11}^{(n)} = \left(\overline{\mathbf{G}}_{2,12}^{(n)}\right)^{T} \mathbf{R}_{\mathbf{x}} \overline{\mathbf{G}}_{2,12}^{(n)}, \mathbf{r}_{11}^{(n)} = \left(\overline{\mathbf{G}}_{2,12}^{(n)}\right)^{T} \mathbf{r}_{\mathbf{x}d}$ $\overline{\mathbf{g}}_{11,W(0)}^{(n)} = \overline{\mathbf{g}}_{11,W(K_{1})}^{(n-1)}, \mathbf{z}_{11(0)}^{(n)} = \mathbf{r}_{11}^{(n)} - \mathbf{R}_{11}^{(n)} \overline{\mathbf{g}}_{11,W(0)}^{(n)}, \mathbf{c}_{11(0)}^{(n)} = \mathbf{z}_{11(0)}^{(n)}, \gamma_{11(0)}^{(n)} = \left(\mathbf{z}_{11(0)}^{(n)}\right)^{T} \mathbf{z}_{11(0)}^{(n)}$ For $k_{1} = 1, 2, \dots, K_{1}$ (n) $\mathbf{q}_{11(k_1)}^{(n)} = \mathbf{R}_{11}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)}, \ \alpha_{11(k_1)}^{(n)} = \frac{\gamma_{11(k_1-1)}^{(n)}}{\left(\mathbf{c}_{11(k_1-1)}^{(n)}\right)^T \mathbf{q}_{11(k_1)}^{(n)}}$ $\overline{\mathbf{g}}_{11,W(k_1)}^{(n)} = \overline{\mathbf{g}}_{11,W(k_1-1)}^{(n)} + \alpha_{11(k_1)}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)}$ $\mathbf{z}_{11(k_1)}^{(n)} = \mathbf{z}_{11(k_1-1)}^{(n)} - \alpha_{11(k_1)}^{(n)} \mathbf{q}_{11(k_1)}^{(n)}$ $\gamma_{11(k_1)}^{(n)} = \left(\mathbf{z}_{11(k_1)}^{(n)}\right)^T \mathbf{z}_{11(k_1)}^{(n)}, \ \beta_{11(k_1)}^{(n)} = \frac{\gamma_{11(k_1)}^{(n)}}{\gamma_{11(k_1)}^{(n)}}, \\ \mathbf{c}_{11(k_1)}^{(n)} = \mathbf{z}_{11(k_1)}^{(n)} + \beta_{11(k_1)}^{(n)} \mathbf{c}_{11(k_1-1)}^{(n)}$

4. Simulation Results

The experimental setup is based on a network echo cancellation scenario [36], as previously described in Section 2, related to the results reported in Figures 2 and 3. The experiments were performed using MATLAB R2018b (for programming and graphic representations), running on a GIGABYTE AORUS 15G XC device (Windows 10 OS) sourced by GYGABYTE, Taipei, Taiwan, which has an Intel Core i7-10870H CPU with 8 Cores, 16 Logical Processors (@2.21 GHz base speed), and 32 GB of RAM. As a performance measure, the normalized misalignment (in dB) is involved in all the following experiments. Summarizing, the main goal is to identify an impulse response **h** of length L = 512 (corresponding to a network echo path), while the input signal x(t) is an AR(1) process. In this context, the reference signal d(t) is obtained based on (1), using a white Gaussian additive noise v(t) with different SNRs. Specifically, three SNR levels are used, i.e., 20 dB, 10 dB, and 0 dB. The first one corresponds to good SNR conditions, where the noise level is mild, thus expecting a good accuracy of the Wiener filter. Second, using SNR = 10 dB corresponds to moderate noisy conditions, while noticing the influence in reducing the accuracy of the solution (i.e., an increase in the misalignment). Finally, heavy noise conditions are considered when SNR = 0 dB, critically influencing the reliability of the Wiener filter.

4.1. IWF-TOT-CG versus WF-CG

In the first set of simulations, the performance of the proposed IWF-TOT-CG is analyzed according to its main parameters, i.e., K_1 , K_2 , and P. The benchmark algorithm involved in comparisons is the WF-CG, using K = 50 CG updates. As shown in Figures 2 and 3, this value of K is sufficient for the WF-CG to reach the conventional WF solution. Since L = 512, the decomposition of the IWF-TOT-CG is performed using $L_{11} = L_{12} = 16$ and $L_2 = 2$.

In Figure 4, the influence of K_1 is assessed, i.e., the maximum number of CG updates used for the two longer filters within the IWF-TOT-CG. Since the third filter (of length $L_2^2 = 4$) is very short and the corresponding CG parameter (K_2) should be much smaller than its length, it is natural to use the smallest value $K_2 = 1$. In this simulation, the decomposition parameter of the IWF-TOT-CG is set to P = 2; the influence of this parameter will be analyzed in an upcoming experiment. Also, the amount of data available for estimating the statistics is N = 5L, while SNR = 20 dB. These represent good conditions for the WF-CG (i.e., the comparing algorithm) to obtain a reliable estimate in terms of accuracy. Nevertheless, as we can notice from Figure 4, the proposed IWF-TOT-CG reaches a significantly lower misalignment level (i.e., a better accuracy) for all the values of K_1 . A larger value of this parameter leads to a faster convergence rate, but up to a certain limit. On the other hand, a larger number of iterations also increases the computational complexity in terms of the number of operations. Consequently, a compromise should be made. As we can notice from Figure 4, increasing the value of K_1 to more than 12 (e.g., 14) does not lead to performance improvements. Besides, the difference between the cases $K_1 = 10$ and $K_1 = 12$ is not so apparent (as compared to the difference between $K_1 = 8$ and $K_1 = 10$), slightly improving the convergence rate, while reaching the same misalignment level (i.e., accuracy). In this context, using $K_1 = 10$ or 12 are reasonable choices.



Figure 4. Normalized misalignment of the WF-CG (after K = 50 CG updates) and IWF-TOT-CG with P = 2, $K_2 = 1$, and using different values of K_1 . The required statistics are obtained by averaging across N = 5L data samples, L = 512, and SNR = 20 dB.

A similar experiment is considered in Figure 5, but using a fixed value for the first CG parameter (i.e., $K_1 = 10$) and two different values for K_2 . These represent the minimum and maximum values for this parameter, i.e., $K_2 = 1$ and $K_2 = 4$, respectively. The maximum value is associated with the length of the corresponding filter (L_2^2) . The other conditions remain the same as in the previous simulation. Under these circumstances, it can be noticed in Figure 5 that a higher value of K_2 does not significantly influence the overall performance of the IWF-TOT-CG so it is natural to use $K_2 = 1$ in the following experiments. The experiment from Figure 5 was performed to assess the influence of K_2 , i.e., the number of CG iterations for the shorter filter. We should note that similar conclusions can be obtained when using other values of K_1 (for the two longer filters).



Figure 5. Normalized misalignment of the WF-CG (after K = 50 CG updates) and IWF-TOT-CG with P = 2, $K_1 = 10$, and using different values of K_2 . The required statistics are obtained by averaging across N = 5L data samples, L = 512, and SNR = 20 dB.

Next, the influence of the decomposition parameter *P* is analyzed in Figure 6. Based on the previous experiments, the setup used for the IWF-TOT-CG is $K_1 = 12$ and $K_2 = 1$, while N = 5L and SNR = 20 dB. As shown in Section 2, the decomposition parameter is chosen such that $P \ll L_{12}$, relying on the low-rank feature of the impulse response. For all the values of *P* considered in Figure 6, the IWF-TOT-CG outperforms the WF-CG. Even the minimum value P = 1 produces a reasonable attenuation of the normalized misalignment, showing improved accuracy as compared to the benchmark algorithm. Moreover, we can notice that increasing the value of *P* beyond a certain value does not improve the overall performance, which also certifies the low-rank approach.

Finally, the last experiment from this first set concerns the influence of different conditions on the performance of the proposed IWF-TOT-CG, as compared to the WF-CG. To this purpose, in Figure 7, several scenarios are considered using different amounts of data for estimating the statistics and lower SNRs. The IWF-TOT-CG uses the same CG parameters as in the previous simulation, while P = 1. This represents a very advantageous setup as compared to the WF-CG counterpart. While the benchmark algorithm involves a single filter of length L = 512, the proposed version combines the estimates provided by three filters of lengths $PL_{11}L_2$, $PL_{12}L_2$, and L_2^2 , which have 32, 32, and 4 coefficients, respectively. Consequently, there is an important reduction in terms of the parameter space, using only 68 coefficients instead of 512. As a result, due to the significantly smaller data structures used within the IWF-TOT-CG, the proposed algorithm is much more robust in harsh conditions, as compared to the WF-CG. This is supported in Figure 7, where the

IWF-TOT-CG outperforms the WF-CG algorithm, especially when using a low amount of data for estimating the statistics (e.g., N = L in Figure 7b) and/or in noisy environments (e.g., SNR = 10 or 0 dB in Figure 7c and d, respectively).



Figure 6. Normalized misalignment of the WF-CG (after K = 50 CG updates) and IWF-TOT-CG with $K_1 = 12$, $K_2 = 1$, and using different values of *P*. The required statistics are obtained by averaging across N = 5L data samples, L = 512, and SNR = 20 dB.



Figure 7. Normalized misalignment of the WF-CG (after K = 50 CG updates) and IWF-TOT-CG with $K_1 = 12$, $K_2 = 1$, P = 1, and using different amounts of data (N = ML, with L = 512) to estimate the required statistics, under different SNR conditions. (**a**) M = 2 and SNR = 20 dB; (**b**) M = 1 and SNR = 20 dB; (**c**) M = 2 and SNR = 10 dB and (**d**) M = 5 and SNR = 0 dB.

4.2. IWF-TOT-CG versus IWF-TOT

The second set of experiments focuses on the comparison between the proposed IWF-TOT-CG and its counterpart recently developed in [18], namely IWF-TOT. This iterative Wiener filter is also based on TOT decomposition but uses direct matrix inversion to solve the associated sets of Wiener–Hopf equations. Therefore, in terms of their decomposition, both algorithms will use the same setup, i.e., $L = L_{11}L_{12}L_2$, with $L_{11} = L_{12} = 16$ and $L_2 = 2$. Besides, the IWF-TOT-CG involves its specific CG cycles, using the settings $K_1 = 12$ and $K_2 = 1$ (as in the previous set of simulations).

In Figure 8, the two TOT-based algorithms are compared for different values of the decomposition parameter P under favorable conditions, i.e., N = 5L data samples (to estimate the statistics) and SNR = 20 dB. While for P = 1 the performances are very similar, the CG-based version achieves a better accuracy for a larger value of P. This supports the advantage of the line search methods (like CG), as compared to the traditional matrix inversion approach, which is also indicated in other previous works [33–35]. As shown in the experiment related to Figure 6, increasing the value of P beyond a certain value does not lead to performance improvement, while increasing the computational complexity on the other hand. In fact, this trade-off relies on the low-rank approach. As indicated in [18], for network impulse responses, the rank of the corresponding matrices is much lower than L_{12} , e.g., usually less than $L_{12}/5$. In our scenario, this leads to a range for P between 1 and 3. This was also previously supported in Figure 6, where we can notice that the misalignment curves for P = 2 and P = 3 are very similar. Consequently, there is no reason for choosing P beyond these values, since there is no performance gain while paying in terms of computational complexity.



Figure 8. Normalized misalignment of the IWF-TOT [18] and IWF-TOT-CG (with $K_1 = 12$ and $K_2 = 1$) using different values of *P*. The required statistics are obtained by averaging across N = 5L data samples, L = 512, and SNR = 20 dB.

The robustness of the line search methods in noisy environments is also an important feature to be considered. The experiment provided in Figure 9 outlines this gain. As we can notice, even for P = 1 (which led to similar results in good SNR conditions), the IWF-TOT-CG performs better as compared to the previous IWF-TOT [18] in low SNR environments.



Figure 9. Normalized misalignment of the IWF-TOT [18] and IWF-TOT-CG (with $K_1 = 12$ and $K_2 = 1$) using P = 1. The required statistics are obtained by averaging across N = 5L data samples (with L = 512), while (**a**) SNR = 10 dB and (**b**) SNR = 0 dB.

4.3. IWF-TOT-CG versus IWF-NKP-CG

The last set of experiments aims to evaluate the Wiener filters based on the CG method in conjunction with the decomposition-based approaches. The proposed IWF-TOT-CG is compared with a recently designed version of the iterative Wiener filter [33]. This algorithm (namely IWF-NKP-CG) also uses the CG method to solve the associated Wiener–Hopf equations, but it relies on the second-order NKP decomposition, thus following the initial approach from [8]. In this case, the length of the filter is factorized as $L = L_1^*L_2^*$, while the low-rank approach relies on the decomposition parameter $P^* < L_2^*$. In our scenario that considers an impulse response of length L = 512, this decomposition is performed using $L_1^* = 32$ and $L_2^* = 16$. The IWF-NKP-CG involves two sets of Wiener–Hopf equations, which correspond to two component Wiener filters of lengths $P^*L_1^*$ and $P^*L_2^*$. Their solutions are obtained based on the CG method, using a maximum number of updates (denoted by K^*). Since in the involved setup, we have $L_2^* = L_{12}$, it is reasonable to use $K^* = K_1$.

In Figure 10, the required statistics are obtained by averaging across N = 5L available data samples, while SNR = 20 dB. These represent reasonably good conditions, which favor reliable estimates. Nevertheless, the IWF-NKP-CG [33] is outperformed by the proposed IWF-TOT-CG for different values of their decomposition parameters. This performance gain results from the TOT decomposition compared to the second-order NKP-based approach, which further supports the initial findings from [18]. For this experiment, the values of *P* and *P*^{*} were selected according to the low-rank approach. Concerning the values of *P* (for IWF-TOT-CG), these considerations were previously indicated related to Figures 6 and 8. Similar aspects can be outlined when choosing the values of *P*^{*} (for IWF-NKP-CG). As recently indicated in [33] and previously supported in [8], for network impulse responses, the rank of the corresponding matrix of size $L_1^* \times L_2^*$ (where $L_1^* \times L_2^* = L$) associated with the reshaped impulse response is much lower than L_2^* , e.g., usually less than $L_2^*/6$. In this context, it is reasonable to consider *P*^{*} < 3 in Figure 10, in order to properly address the trade-off between performance and complexity. Otherwise, using larger values of *P*^{*} will not justify the performance gain, while increasing the computational amount.



Figure 10. Normalized misalignment of the IWF-NKP-CG [33] (with $K^* = 12$) and IWF-TOT-CG (with $K_1 = 12$ and $K_2 = 1$) using different values of P^* and P, respectively. The required statistics are obtained by averaging across N = 5L data samples, L = 512, and SNR = 20 dB.

Finally, the IWF-NKP-CG [33] and the proposed IWF-TOT-CG (using $P^* = P = 1$) are compared in more challenging conditions, when using smaller amounts of data for estimating the required statistics, i.e., N = 2L and N = L. As we can notice in Figure 11, the performance gain of the IWF-TOT-CG is more apparent in these cases, which shows the robustness of the TOT-based decomposition in conjunction with the CG method.



Figure 11. Normalized misalignment of the IWF-NKP-CG [33] (with $K^* = 12$) and IWF-TOT-CG (with $K_1 = 12$ and $K_2 = 1$) using $P^* = P = 1$, while SNR = 20 dB. The required statistics are obtained by averaging across N = ML data samples (with L = 512), where (**a**) M = 2 and (**b**) M = 1.

5. Conclusions and Future Works

In this paper, we have developed an iterative version of the Wiener filter using the CG method and exploiting a tensorial decomposition of the impulse response. The resulting

IWF-TOT-CG combines the solutions of three sets of Wiener–Hopf equations which are solved with CG updates, thus avoiding matrix inversion operations. An important gain is related to the dimensionality reduction in a long-length system identification problem, which can be reformulated using a reduced set of coefficients, corresponding to three (much) shorter filters. This approach fits very well with the identification of low-rank impulse responses, like in echo cancellation.

In terms of its performance, the proposed IWF-TOT-CG outperforms the conventional Wiener filter, especially in some challenging scenarios. Here, we can mention the cases when dealing with a small amount of available data (to estimate the statistics) or working in low SNR environments. While the accuracy of the conventional Wiener filter is highly affected in such conditions, the proposed version (which operates with smaller data structures) is still robust and provides reliable solutions. Moreover, the IWF-TOT-CG performs better as compared to the previously developed IWF-TOT [18], which involves matrix inversion operations to solve the Wiener–Hopf equations. Also, the proposed algorithm provides improved performance as compared to its counterpart based on the second-order NKP decomposition, i.e., the IWF-NKP-CG [33].

Future works will focus on three main directions. First, we can exploit other line search methods to solve the Wiener–Hopf equations, like those based on the coordinate descent technique [34,35]. A comparison between these methods is beyond the scope of this paper; however, this could open the path toward using inexact line search methods in conjunction with decomposition-based algorithms. Among them, we can mention the dichotomous coordinate descent technique [40–44], which is very appealing in terms of computational efficiency. Second, another direction for future works targets the extension to higher-order tensors, which could lead to improved decomposition-based solutions and higher dimensionality reduction. Third, it would be highly useful to extend the decomposition-based approach and the tensorial framework to other potential solutions used for system identification problems, like the Kalman filter and different adaptive filtering algorithms. These developments could be further used in real-world applications, like echo cancellation, active noise control, and interference reduction.

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

Funding: This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS–UEFISCDI, project number PN-III-P4-PCE-2021-0438, within PNCDI III, and by a grant from the National Program for Research of the National Association of Technical Universities -GNAC ARUT 2023 (no. 36/09.10.2023, code 133).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

References

- 1. Ljung, L. System Identification: Theory for the User, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 1999.
- 2. Haykin, S. *Adaptive Filter Theory*, 4th ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2002.
- 3. Sayed, A.H. Adaptive Filters; Wiley: New York, NY, USA, 2008.
- 4. Diniz, P.S.R. *Adaptive Filtering: Algorithms and Practical Implementation*, 4th ed.; Springer: New York, NY, USA, 2013.
- 5. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 3rd ed.; The John Hopkins University Press: Baltimore, MD, USA, 1996.
- Hansen, P.C. Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion; SIAM: Philadelphia, PA, USA, 1998.
- 7. Hänsler, E.; Schmidt, G. Acoustic Echo and Noise Control-A Practical Approach; Wiley: Hoboken, NJ, USA, 2004.
- Paleologu, C.; Benesty, J.; Ciochină, S. Linear system identification based on a Kronecker product decomposition. *IEEE/ACM Trans. Audio Speech Lang. Process.* 2018, 26, 1793–1808. [CrossRef]

- Bhattacharjee, S.S.; George, N.V. Nearest Kronecker product decomposition based normalized least mean square algorithm. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 476–480.
- Bhattacharjee, S.S.; Kumar, K.; George, N.V. Nearest Kronecker product decomposition based generalized maximum correntropy and generalized hyperbolic secant robust adaptive filters. *IEEE Signal Process. Lett.* 2020, 27, 1525–1529. [CrossRef]
- 11. Yang, W.; Huang, G.; Chen, J.; Benesty, J.; Cohen, I.; Kellermann, W. Robust dereverberation with Kronecker product based multichannel linear prediction. *IEEE Signal Process. Lett.* **2021**, *28*, 101–105. [CrossRef]
- Bhattacharjee, S.S.; George, N.V. Fast and efficient acoustic feedback cancellation based on low rank approximation. *Signal Process*. 2021, 182, 107984. [CrossRef]
- 13. Bhattacharjee, S.S.; George, N.V. Nearest Kronecker product decomposition based linear-in-the-parameters nonlinear filters. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 2111–2122. [CrossRef]
- 14. Wang, X.; Benesty, J.; Chen, J.; Huang, G.; Cohen, I. Beamforming with cube microphone arrays via Kronecker product decompositions. *IEEE/ACM Trans. Audio Speech Lang. Process.* 2021, 29, 1774–1784. [CrossRef]
- 15. Huang, G.; Benesty, J.; Cohen, I.; Chen, J. Kronecker product multichannel linear filtering for adaptive weighted prediction error-based speech dereverberation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *30*, 1277–1289. [CrossRef]
- 16. Vadhvana, S.; Yadav, S.K.; Bhattacharjee, S.S.; George, N.V. An improved constrained LMS algorithm for fast adaptive beamforming based on a low rank approximation. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 3605–3609. [CrossRef]
- Bhattacharjee, S.S.; Patel, V.; George, N.V. Nonlinear spline adaptive filters based on a low rank approximation. *Signal Process*. 2022, 201, 108726. [CrossRef]
- Benesty, J.; Paleologu, C.; Ciochină, S. Linear system identification based on a third-order tensor decomposition. *IEEE Signal Process. Lett.* 2023, 30, 503–507. [CrossRef]
- 19. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. SIAM Rev. 2009, 51, 455–500. [CrossRef]
- 20. Comon, P. Tensors: A brief introduction. IEEE Signal Process. Mag. 2014, 31, 44-53. [CrossRef]
- 21. Vervliet, N.; Debals, O.; Sorber, L.; Lathauwer, L.D. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Process. Mag.* **2014**, *31*, 71–79. [CrossRef]
- 22. Friedland, S.; Tammali, V. Low-rank approximation of tensors. In *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*; Benner, P., Bollhöfer, M., Kressner, D., Mehl, C., Stykel, T., Eds.; Springer: Cham, Switzerland, 2015; pp. 377–411.
- 23. Cichocki, A.; Mandic, D.P.; Phan, A.; Caiafa, C.F.; Zhou, G.; Zhao, Q.; Lathauwer, L.D. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163. [CrossRef]
- Becker, H.; Albera, L.; Comon, P.; Gribonval, R.; Wendling, F.; Merlet, I. Brain-source imaging: From sparse to tensor models. *IEEE Signal Process. Mag.* 2015, *32*, 100–112. [CrossRef]
- Bousse, M.; Debals, O.; Lathauwer, L.D. A tensor-based method for large-scale blind source separation using segmentation. *IEEE Trans. Signal Process.* 2017, 65, 346–358. [CrossRef]
- Sidiropoulos, N.; Lathauwer, L.D.; Fu, X.; Huang, K.; Papalexakis, E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* 2017, 65, 3551–3582. [CrossRef]
- 27. Chang, S.Y.; Wei, Y. Generalized T-product tensor Bernstein bounds. Ann. Appl. Math. 2022, 38, 25–61.
- 28. Bozorgmanesh, H.; Hajarian, M.; Chronopoulos, A.T. The relation between a tensor and its associated semi-symmetric form. *Numer. Math. Theory Methods Appl.* **2022**, *15*, 530–564. [CrossRef]
- 29. Zheng, L.; Yang, L.; Liang, Y. A conjugate gradient projection method for solving equations with convex constraints. *J. Comput. Appl. Math.* **2020**, *375*, 112781. [CrossRef]
- Damale, P.U.; Chong, E.K.P.; Scharf, L.L. Wiener filtering without covariance matrix inversion. In Proceedings of the ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; 5p.
- Damale, P.U.; Chong, E.K.P.; Scharf, L.L. Wiener filter approximations without covariance matrix inversion. *IEEE Open J. Signal Process.* 2023, 4, 366–374. [CrossRef]
- 32. Hestenes, M.R.; Stiefel, E. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436. [CrossRef]
- 33. Stanciu, C.L.; Benesty, J.; Paleologu, C.; Costea, R.L.; Dogariu, L.M.; Ciochină, S. Decomposition-based Wiener filter using the Kronecker product and conjugate gradient method. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 124–138. [CrossRef]
- 34. Zakharov, Y.V.; Albu, F. Coordinate descent iterations in fast affine projection algorithm. *IEEE Signal Process. Lett.* **2005**, *12*, 353–356. [CrossRef]
- Zakharov, Y.V.; White, G.P.; Liu, J. Low-complexity RLS algorithms using dichotomous coordinate descent iterations. *IEEE Trans.* Signal Process. 2008, 56, 3150–3161. [CrossRef]
- Digital Network Echo Cancellers. ITU-T Recommendation G.168. 2012. Available online: www.itu.int/rec/T-REC-G.168 (accessed on 11 March 2024).
- 37. Loan, C.F.V. The ubiquitous Kronecker product. J. Comput. Appl. Math. 2000, 123, 85–100. [CrossRef]
- 38. Bertsekas, D.P. Nonlinear Programming, 2nd ed.; Athena Scientific: Belmont, MA, USA, 1999.
- Rupp, M.; Schwarz, S. A tensor LMS algorithm. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 3347–3351.

- Zakharov, Y.V.; Nascimento, V.H. DCD-RLS adaptive filters with penalties for sparse identification. *IEEE Trans. Signal Process.* 2013, 61, 3198–3213. [CrossRef]
- 41. Kim, G.; Lee, H.; Chung, J.; Lee, J. A delay relaxed RLS-DCD algorithm for real-time implementation. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 61–65. [CrossRef]
- 42. Zhang, Y.; Wu, T.; Zakharov, Y.V.; Li, J. MMP-DCD-CV based sparse channel estimation algorithm for underwater acoustic transform domain communication system. *Appl. Acoust.* **2019**, *154*, 43–52. [CrossRef]
- 43. Yu, Y.; Lu, L.; Zheng, Z.; Wang, W.; Zakharov, Y.V.; de Lamare, R.C. DCD-based recursive adaptive algorithms robust against impulsive noise. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 1359–1363. [CrossRef]
- 44. Liao, M.; Zakharov, Y.V. DCD-based joint sparse channel estimation for OFDM in virtual angular domain. *IEEE Access* 2021, 9, 102081–102090. [CrossRef]

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