

Article

# System Identification Based on Tensor Decompositions: A Trilinear Approach

Laura-Maria Dogariu <sup>1,\*</sup>, Silviu Ciochină <sup>1</sup>, Jacob Benesty <sup>2</sup> and Constantin Paleologu <sup>1</sup>

<sup>1</sup> Department of Telecommunications, University Politehnica of Bucharest, 1–3, Iuliu Maniu Blvd., 061071 Bucharest, Romania; silviu@comm.pub.ro (S.C.); pale@comm.pub.ro (C.P.)

<sup>2</sup> INRS-EMT, University of Quebec, Montreal, QC H5A 1K6, Canada; benesty@emt.inrs.ca

\* Correspondence: ldogariu@comm.pub.ro

Received: 28 March 2019; Accepted: 14 April 2019; Published: 17 April 2019



**Abstract:** The theory of nonlinear systems can currently be encountered in many important fields, while the nonlinear behavior of electronic systems and devices has been studied for a long time. However, a global approach for dealing with nonlinear systems does not exist and the methods to address this problem differ depending on the application and on the types of nonlinearities. An interesting category of nonlinear systems is one that can be regarded as an ensemble of (approximately) linear systems. Some popular examples in this context are nonlinear electronic devices (such as acoustic echo cancellers, which are used in applications for two-party or multi-party voice communications, e.g., videoconferencing), which can be modeled as a cascade of linear and nonlinear systems, similar to the Hammerstein model. Multiple-input/single-output (MISO) systems can also be regarded as separable multilinear systems and be treated using the appropriate methods. The high dimension of the parameter space in such problems can be addressed with methods based on tensor decompositions and modelling. In recent work, we focused on a particular type of multilinear structure—namely the bilinear form (i.e., two-dimensional decompositions)—in the framework of identifying spatiotemporal models. In this paper, we extend the work to the decomposition of more complex systems and we propose an iterative Wiener filter tailored for the identification of trilinear forms (where third-order tensors are involved), which can then be further extended to higher order multilinear structures. In addition, we derive the least-mean-square (LMS) and normalized LMS (NLMS) algorithms tailored for such trilinear forms. Simulations performed in the context of system identification (based on the MISO system approach) indicate the good performance of the proposed solution, as compared to conventional approaches.

**Keywords:** nonlinear systems; multiple-input/single-output (MISO) systems; Wiener filter, least-mean-square (LMS) algorithm; normalized LMS (NLMS) algorithm; trilinear forms; tensors

## 1. Introduction

The approximation of nonlinear systems can be performed using a finite sum of the Volterra series expansion that relates the system's inputs and outputs. This method has been studied since the 1960s [1–3] and used in different applications, for example, Reference [4–6], among others. In this context, bilinear forms have been used to approximate a large class of nonlinear systems, where the bilinear component is interpreted in terms of an input-output relation (meaning that it is defined with respect to the data). Consequently, the bilinear system may be regarded as one of the simplest recursive nonlinear systems.

More recently, a new approach was introduced in Reference [7], where the bilinear term is considered within the framework of a multiple-input/single-output (MISO) system, and it is defined with respect to the spatiotemporal model's impulse responses. A particular case of this type of system

is the Hammerstein model [8]; in this scenario, a single-input signal passes through a nonlinear block and a linear system, which are cascaded. Recently, adaptive solutions were proposed for the study of such bilinear forms in the context of system identification [9–11]. The bilinear approach is suitable for a particular form of the decomposition, which involves only two terms (i.e., two systems). In some cases, it would be useful to exploit a higher-order decomposition, which could improve the overall performance in terms of both complexity and efficiency.

Motivated by the good performance of these approaches in the study of bilinear forms, we aim to further extend this approach to higher-order multilinear in parameters' systems. Applications, such as multichannel equalization [12], nonlinear acoustic devices for echo cancellation [13], multiple-input/multiple-output (MIMO) communication systems [14,15] and others, can be addressed within the framework of multilinear systems. Because many of these applications can be formulated in terms of system identification problems [16], it is of interest to estimate a model based on the available and observed data, which are usually the input and the output of the system. The challenges that usually occur in such system identification tasks may be represented by a high length of the finite impulse response filter [17,18], which may employ hundreds or even thousands of coefficients. Moreover, another possible issue is the large parameter space [19,20]. Nowadays, such scenarios are related to very important topics, for example, big data [21], machine learning [22], and source separation [23]. Usually, they are addressed based on tensor decompositions and modelling [24–27]. These techniques mainly exploit the Kronecker product decomposition [28], which further allows the reformulation of a high-dimension problem as low-dimension models, which can be tensorized together.

In this paper, we propose an iterative Wiener filter tailored for third-order systems (i.e., trilinear forms), together with adaptive solutions for such problems—namely the least-mean-square (LMS) and normalized LMS (NLMS) algorithms. Following this development, the solution can then be extended to higher-order multilinear systems. The proposed approach presents an advantage from the perspective of exploiting the decomposition of the global impulse response. One potential limitation of such a method is related to the particular form of the global impulse response to be identified, which is a result of separable systems. In perspective, it would be useful to extend this approach to identify more general forms of impulse responses. Recently, we developed such ideas in References [29,30], based on the Wiener filter and the recursive least-squares (RLS) algorithm, by exploiting the nearest Kronecker product decomposition and the related low-rank approximation.

The rest of this paper is organized as follows: In Section 2, a brief theory of tensors is provided, which is necessary for the following developments; next, in Section 3, the iterative Wiener filter tailored for the identification of trilinear forms is proposed; Section 4 contains the derivation of the trilinear forms for the LMS and NLMS adaptive algorithms; simulation results are provided in Section 5, while Section 6 concludes the paper.

## 2. Background on Tensors

A tensor is a multidimensional array of data the entries of which are referred by using multiple indices [31,32]. A tensor, a matrix, a vector, and a scalar can be denoted by  $\mathcal{A}$ ,  $\mathbf{A}$ ,  $\mathbf{a}$ , and  $a$ , respectively. In this paper, we are only interested in the third-order tensor  $\mathcal{A} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ , meaning that its elements are real-valued and its dimension is  $L_1 \times L_2 \times L_3$ . For a third-order tensor, the first and second indices  $l_1$  and  $l_2$  correspond to the row and column, respectively—as in a matrix—while the third index  $l_3$  corresponds to the tube and describes its depth. These three indices describe the three different modes. The entries of the different order tensors are denoted by  $(\mathcal{A})_{l_1 l_2 l_3} = a_{l_1 l_2 l_3}$ ,  $(\mathbf{A})_{l_1 l_2} = a_{l_1 l_2}$ , and  $(\mathbf{a})_{l_1} = a_{l_1}$ , for  $l_1 = 1, 2, \dots, L_1$ ,  $l_2 = 1, 2, \dots, L_2$ , and  $l_3 = 1, 2, \dots, L_3$ .

The notion of vectorization, consisting of transforming a matrix into a vector, is very well known. Matricization does somewhat the same thing but from a third-order tensor into a large

matrix. Depending on which index's elements are considered first, we have matricization along three different modes [24,25]:

$$\begin{aligned}\mathbf{A}_{[1]} &= \mathbf{A}_{:,1:L_2,1:L_3} \\ &= \begin{bmatrix} \mathbf{A}_{::1} & \cdots & \mathbf{A}_{::L_3} \end{bmatrix}, \mathbf{A}_{[1]} \in \mathbb{R}^{L_1 \times L_2 L_3}, \\ \mathbf{A}_{[2]} &= \mathbf{A}_{1:L_1, :, 1:L_3}, \mathbf{A}_{[2]} \in \mathbb{R}^{L_2 \times L_1 L_3}, \\ \mathbf{A}_{[3]} &= \mathbf{A}_{1:L_1, 1:L_2, :}, \mathbf{A}_{[3]} \in \mathbb{R}^{L_3 \times L_1 L_2},\end{aligned}$$

where  $\mathbf{A}_{::l_3} \in \mathbb{R}^{L_1 \times L_2}$ ,  $l_3 = 1, 2, \dots, L_3$  are the frontal slices. Hence, the vectorization of a tensor is

$$\text{vec}(\mathcal{A}) = \text{vec}(\mathbf{A}_{[1]}) = \begin{bmatrix} \text{vec}(\mathbf{A}_{::1}) \\ \vdots \\ \text{vec}(\mathbf{A}_{::L_3}) \end{bmatrix} \in \mathbb{R}^{L_1 L_2 L_3}.$$

Let  $\mathbf{b}_1$ ,  $\mathbf{b}_2$ , and  $\mathbf{b}_3$  be vectors of length  $L_1$ ,  $L_2$ , and  $L_3$ , respectively, whose elements are  $b_{1l_1}$ ,  $b_{2l_2}$ , and  $b_{3l_3}$ . A rank-1 tensor (of dimension  $L_1 \times L_2 \times L_3$ ) is defined as

$$\mathcal{B} = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3, \quad (1)$$

where  $\circ$  is the vector outer product and the elements of  $\mathcal{B}$  are given by  $(\mathcal{B})_{l_1 l_2 l_3} = b_{1l_1} b_{2l_2} b_{3l_3}$ . The frontal slices of  $\mathcal{B}$  in Equation (1) are  $\mathbf{B}_{::l_3} = \mathbf{b}_1 \mathbf{b}_2^T b_{3l_3} \in \mathbb{R}^{L_1 \times L_2}$ ,  $l_3 = 1, 2, \dots, L_3$ . In particular, we have  $\mathbf{B} = \mathbf{b}_1 \circ \mathbf{b}_2 = \mathbf{b}_1 \mathbf{b}_2^T$ , where  $T$  is the transpose operator. Therefore, the rank of a tensor  $\mathcal{A}$ , denoted  $\text{rank}(\mathcal{A})$ , is defined as the minimum number of rank-1 tensors that generate  $\mathcal{A}$  as their sum.

The inner product between two tensors  $\mathcal{A}$  and  $\mathcal{B}$  of the same dimension is

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \sum_{l_3=1}^{L_3} a_{l_1 l_2 l_3} b_{l_1 l_2 l_3} = \text{vec}^T(\mathcal{B}) \text{vec}(\mathcal{A}).$$

It is important to be able to multiply a tensor with a matrix [26,27]. Let the tensor be  $\mathcal{A} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$  and the matrix be  $\mathbf{M}_1 \in \mathbb{R}^{M_1 \times L_1}$ . The mode-1 product between the tensor  $\mathcal{A}$  and the matrix  $\mathbf{M}_1$  gives the tensor:

$$\mathcal{U} = \mathcal{A} \times_1 \mathbf{M}_1, \mathcal{U} \in \mathbb{R}^{M_1 \times L_2 \times L_3}, \quad (2)$$

whose entries are  $u_{m_1 l_2 l_3} = \sum_{l_1=1}^{L_1} a_{l_1 l_2 l_3} m_{m_1 l_1}$ , for  $m_1 = 1, 2, \dots, M_1$ , and  $\mathbf{U}_{[1]} = \mathbf{M}_1 \mathbf{A}_{[1]}$ . In the same way, with the matrix  $\mathbf{M}_2 \in \mathbb{R}^{M_2 \times L_2}$ , the mode-2 product between the tensor  $\mathcal{A}$  and the matrix  $\mathbf{M}_2$  gives the tensor:

$$\mathcal{U} = \mathcal{A} \times_2 \mathbf{M}_2, \mathcal{U} \in \mathbb{R}^{L_1 \times M_2 \times L_3}, \quad (3)$$

whose entries are  $u_{l_1 m_2 l_3} = \sum_{l_2=1}^{L_2} a_{l_1 l_2 l_3} m_{m_2 l_2}$ , for  $m_2 = 1, 2, \dots, M_2$ , and  $\mathbf{U}_{[2]} = \mathbf{M}_2 \mathbf{A}_{[2]}$ . Finally, with the matrix  $\mathbf{M}_3 \in \mathbb{R}^{M_3 \times L_3}$ , the mode-3 product between the tensor  $\mathcal{A}$  and the matrix  $\mathbf{M}_3$  gives the tensor:

$$\mathcal{U} = \mathcal{A} \times_3 \mathbf{M}_3, \mathcal{U} \in \mathbb{R}^{L_1 \times L_2 \times M_3}, \quad (4)$$

whose entries are  $u_{l_1 l_2 m_3} = \sum_{l_3=1}^{L_3} a_{l_1 l_2 l_3} m_{m_3 l_3}$ , for  $m_3 = 1, 2, \dots, M_3$ , and  $\mathbf{U}_{[3]} = \mathbf{M}_3 \mathbf{A}_{[3]}$ .

The multiplication of  $\mathcal{A}$  with the row vectors  $\mathbf{b}_1^T$ ,  $\mathbf{b}_2^T$ , and  $\mathbf{b}_3^T$  (see the components of  $\mathcal{B}$  in Equation (1)) gives the scalar:

$$c = \mathcal{A} \times_1 \mathbf{b}_1^T \times_2 \mathbf{b}_2^T \times_3 \mathbf{b}_3^T = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \sum_{l_3=1}^{L_3} a_{l_1 l_2 l_3} b_{1l_1} b_{2l_2} b_{3l_3}. \quad (5)$$

In particular, we have

$$c = \mathbf{A} \times_1 \mathbf{b}_1^T \times_2 \mathbf{b}_2^T = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} a_{l_1 l_2} b_{1l_1} b_{2l_2} = \mathbf{b}_1^T \mathbf{A} \mathbf{b}_2 \quad (6)$$

and

$$c = \mathbf{a} \times_1 \mathbf{b}_1^T = \sum_{l_1=1}^{L_1} a_{l_1} b_{1l_1} = \mathbf{b}_1^T \mathbf{a}. \quad (7)$$

It is easy to check that Equations (5)–(7) are trilinear (with respect to  $\mathbf{b}_1$ ,  $\mathbf{b}_2$ , and  $\mathbf{b}_3$ ), bilinear (with respect to  $\mathbf{b}_1$  and  $\mathbf{b}_2$ ), and linear (with respect to  $\mathbf{b}_1$ ) forms, respectively.

We can express Equation (6) as

$$\begin{aligned} c &= \text{tr}(\mathbf{b}_2 \mathbf{b}_1^T \mathbf{A}) = \text{tr} \left[ \left( \mathbf{b}_1 \mathbf{b}_2^T \right)^T \mathbf{A} \right] = \text{vec}^T(\mathbf{b}_1 \mathbf{b}_2^T) \text{vec}(\mathbf{A}) \\ &= (\mathbf{b}_2 \otimes \mathbf{b}_1)^T \text{vec}(\mathbf{A}), \end{aligned} \quad (8)$$

where  $\text{tr}(\cdot)$  represents the trace of a square matrix and  $\otimes$  denotes the Kronecker product. Expression (5) can also be written in a more convenient way. Indeed, we have

$$\begin{aligned} c &= \langle \mathcal{A}, \mathcal{B} \rangle = \text{vec}^T(\mathcal{B}) \text{vec}(\mathcal{A}) \\ &= \text{vec}^T(\mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3) \text{vec}(\mathcal{A}) \\ &= (\mathbf{b}_3 \otimes \mathbf{b}_2 \otimes \mathbf{b}_1)^T \text{vec}(\mathcal{A}), \end{aligned} \quad (9)$$

where  $\mathcal{B}$  is defined in Equation (1).

### 3. Trilinear Wiener Filter

Let us consider the following signal model, which is used in system identification tasks:

$$\begin{aligned} d(t) &= y(t) + v(t) \\ &= \mathcal{X}(t) \times_1 \mathbf{h}_1^T \times_2 \mathbf{h}_2^T \times_3 \mathbf{h}_3^T + v(t) \\ &= \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \sum_{l_3=1}^{L_3} x_{l_1 l_2 l_3}(t) h_{1l_1} h_{2l_2} h_{3l_3} + v(t), \end{aligned} \quad (10)$$

where  $d(t)$  is the desired (also known as reference) signal at time index  $t$ ,  $y(t)$  is the output signal of a MISO system and  $v(t)$  is a zero-mean additive noise, uncorrelated with the input signals. The zero-mean input signals can be described in a tensorial form  $\mathcal{X}(t) \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ :

$$(\mathcal{X})_{l_1 l_2 l_3}(t) = x_{l_1 l_2 l_3}(t), \quad l_k = 1, 2, \dots, L_k, \quad k = 1, 2, 3,$$

and the three impulse responses  $\mathbf{h}_k$ ,  $k = 1, 2, 3$ , of lengths  $L_1$ ,  $L_2$ , and  $L_3$ , respectively, can be written as

$$\mathbf{h}_k = \begin{bmatrix} h_{k1} & h_{k2} & \cdots & h_{kL_k} \end{bmatrix}^T, \quad k = 1, 2, 3.$$

It can be seen that  $y(t)$  represents a trilinear form because it is a linear function of each of the vectors  $\mathbf{h}_k$ ,  $k = 1, 2, 3$ , if the other two are kept fixed. The trilinear form can be regarded as an extension of the bilinear form [7].

Starting from the three impulse responses of the MISO system, a rank-1 tensor of dimension  $L_1 \times L_2 \times L_3$  can be constructed in the following way:

$$\mathcal{H} = \mathbf{h}_1 \circ \mathbf{h}_2 \circ \mathbf{h}_3, \quad (11)$$

where

$$(\mathcal{H})_{l_1 l_2 l_3} = h_{1l_1} h_{2l_2} h_{3l_3}, \quad l_k = 1, 2, \dots, L_k, \quad k = 1, 2, 3.$$

Consequently, the output signal becomes

$$y(t) = \text{vec}^T(\mathcal{H}) \text{vec}[\mathcal{X}(t)], \quad (12)$$

where

$$\text{vec}(\mathcal{H}) = \begin{bmatrix} \text{vec}(\mathbf{H}_{::1}) \\ \vdots \\ \text{vec}(\mathbf{H}_{::L_3}) \end{bmatrix} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1 \triangleq \mathbf{h}, \quad (13)$$

$$\text{vec}[\mathcal{X}(t)] = \begin{bmatrix} \text{vec}[\mathbf{X}_{::1}(t)] \\ \vdots \\ \text{vec}[\mathbf{X}_{::L_3}(t)] \end{bmatrix} \triangleq \mathbf{x}(t), \quad (14)$$

$\mathbf{H}_{::l_3}$  and  $\mathbf{X}_{::l_3}(t)$  (with  $l_3 = 1, 2, \dots, L_3$ ) are the frontal slices of  $\mathcal{H}$  and  $\mathcal{X}(t)$ , respectively, while  $\mathbf{h}$  and  $\mathbf{x}(t)$  denote two long vectors, each of them having  $L_1 L_2 L_3$  elements. Hence, the output signal can be expressed as

$$y(t) = \mathbf{h}^T \mathbf{x}(t). \quad (15)$$

In this framework, the aim is to estimate the global impulse response  $\mathbf{h}$ . For that, first we define the error signal:

$$e(t) = d(t) - \hat{y}(t) = d(t) - \hat{\mathbf{h}}^T \mathbf{x}(t), \quad (16)$$

where  $\hat{y}(t) = \hat{\mathbf{h}}^T \mathbf{x}(t)$  represents the estimated signal, obtained using the impulse response  $\hat{\mathbf{h}}$  of length  $L_1 L_2 L_3$ .

Based on Equation (16), let us consider the mean-squared error (MSE) optimization criterion, that is, the minimization of the cost function:

$$J(\hat{\mathbf{h}}) = E[e^2(t)], \quad (17)$$

where  $E[\cdot]$  denotes mathematical expectation. Using Equation (16) in Equation (17), together with the notation  $\sigma_d^2 = E[d^2(t)]$  (the variance of the reference signal),  $\mathbf{p} = E[\mathbf{x}(t)d(t)]$  (the cross-correlation vector between the input and the reference signals) and  $\mathbf{R} = E[\mathbf{x}(t)\mathbf{x}^T(t)]$  (the covariance matrix of the input signal), the cost function can be developed as

$$J(\hat{\mathbf{h}}) = \sigma_d^2 - 2\hat{\mathbf{h}}^T \mathbf{p} + \hat{\mathbf{h}}^T \mathbf{R} \hat{\mathbf{h}}. \quad (18)$$

By minimizing Equation (18), we obtain the well-known Wiener filter:

$$\hat{\mathbf{h}}_W = \mathbf{R}^{-1}\mathbf{p}. \quad (19)$$

As we can see, the dimension of the covariance matrix is  $L_1L_2L_3 \times L_1L_2L_3$ , thus requiring a large amount of data (much more than  $L_1L_2L_3$  samples) to obtain a good estimate of it. Furthermore,  $\mathbf{R}$  could be very ill-conditioned because of its huge size. As a result, the solution  $\hat{\mathbf{h}}_W$  will be very inaccurate, to say the least, in practice.

On the other hand, as we notice from Equation (13), the global impulse response  $\mathbf{h}$  (with  $L_1L_2L_3$  coefficients) results based on a combination of the shorter impulse responses  $\mathbf{h}_k$ ,  $k = 1, 2, 3$ , with  $L_1$ ,  $L_2$ , and  $L_3$  coefficients, respectively. In fact, we only need  $L_1 + L_2 + L_3$  different elements to form  $\mathbf{h}$ , even though this global impulse response is of length  $L_1L_2L_3$ . This represents the motivation behind an alternative approach to the conventional Wiener solution. Similar to Equation (13), the estimate of the global system can be decomposed as

$$\hat{\mathbf{h}} = \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1, \quad (20)$$

where  $\hat{\mathbf{h}}_k$ ,  $k = 1, 2, 3$  are three impulse responses of lengths  $L_1$ ,  $L_2$ , and  $L_3$ , respectively, which represent the estimates of the individual impulse responses  $\mathbf{h}_k$ ,  $k = 1, 2, 3$ . Nevertheless, we should note that there is no unique solution related to the decomposition in Equation (20), since for any constants  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ , with  $\alpha_1\alpha_2\alpha_3 = 1$ , we have  $\hat{\mathbf{h}} = \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1 = \alpha_3\hat{\mathbf{h}}_3 \otimes \alpha_2\hat{\mathbf{h}}_2 \otimes \alpha_1\hat{\mathbf{h}}_1$ . Consequently,  $\alpha_k\hat{\mathbf{h}}_k$ ,  $k = 1, 2, 3$  also represents a set of solutions for our problem. Nevertheless, the global system impulse response— $\mathbf{h}$ —can be identified with no scaling ambiguity.

Next, we propose an iterative alternative to the conventional Wiener filter, following the decomposition from Equation (20). First, we can easily verify that

$$\hat{\mathbf{h}} = \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right) \hat{\mathbf{h}}_1 \quad (21)$$

$$= \left( \hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1 \right) \hat{\mathbf{h}}_2 \quad (22)$$

$$= \left( \mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1 \right) \hat{\mathbf{h}}_3, \quad (23)$$

where  $\mathbf{I}_{L_k}$  is the identity matrix of size  $L_k \times L_k$ , for  $k = 1, 2, 3$ . Based on the previous relations, the cost function from Equation (18) can be expressed in three different ways. For example, using Equation (21), we obtain

$$J \left( \hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3 \right) = \sigma_d^2 - 2\hat{\mathbf{h}}_1^T \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right)^T \mathbf{p} \\ + \hat{\mathbf{h}}_1^T \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right)^T \mathbf{R} \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right) \hat{\mathbf{h}}_1. \quad (24)$$

When  $\hat{\mathbf{h}}_2$  and  $\hat{\mathbf{h}}_3$  are fixed, we can rewrite Equation (24) as

$$J_{\hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3} \left( \hat{\mathbf{h}}_1 \right) = \sigma_d^2 - 2\hat{\mathbf{h}}_1^T \mathbf{p}_1 + \hat{\mathbf{h}}_1^T \mathbf{R}_1 \hat{\mathbf{h}}_1, \quad (25)$$

where

$$\mathbf{p}_1 = \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right)^T \mathbf{p}, \quad (26)$$

$$\mathbf{R}_1 = \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right)^T \mathbf{R} \left( \hat{\mathbf{h}}_3 \otimes \hat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1} \right). \quad (27)$$

In this case, the partial cost function from Equation (25) is a convex one and can be minimized with respect to  $\hat{\mathbf{h}}_1$ .

Similarly, using Equations (22) and (23), the cost function from Equation (18) becomes

$$J(\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3) = \sigma_d^2 - 2\hat{\mathbf{h}}_2^T (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1)^T \mathbf{p} + \hat{\mathbf{h}}_2^T (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1)^T \mathbf{R} (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1) \hat{\mathbf{h}}_2 \quad (28)$$

$$= \sigma_d^2 - 2\hat{\mathbf{h}}_3^T (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1)^T \mathbf{p} + \hat{\mathbf{h}}_3^T (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1)^T \mathbf{R} (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1) \hat{\mathbf{h}}_3. \quad (29)$$

Also, when  $\hat{\mathbf{h}}_1$  and  $\hat{\mathbf{h}}_3$  are fixed, Equation (28) becomes

$$J_{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_3}(\hat{\mathbf{h}}_2) = \sigma_d^2 - 2\hat{\mathbf{h}}_2^T \mathbf{p}_2 + \hat{\mathbf{h}}_2^T \mathbf{R}_2 \hat{\mathbf{h}}_2, \quad (30)$$

where

$$\mathbf{p}_2 = (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1)^T \mathbf{p}, \quad (31)$$

$$\mathbf{R}_2 = (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1)^T \mathbf{R} (\hat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1), \quad (32)$$

while when  $\hat{\mathbf{h}}_1$  and  $\hat{\mathbf{h}}_2$  are fixed, the cost function from Equation (29) results in

$$J_{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2}(\hat{\mathbf{h}}_3) = \sigma_d^2 - 2\hat{\mathbf{h}}_3^T \mathbf{p}_3 + \hat{\mathbf{h}}_3^T \mathbf{R}_3 \hat{\mathbf{h}}_3, \quad (33)$$

where

$$\mathbf{p}_3 = (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1)^T \mathbf{p}, \quad (34)$$

$$\mathbf{R}_3 = (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1)^T \mathbf{R} (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2 \otimes \hat{\mathbf{h}}_1). \quad (35)$$

In both cases, the partial cost functions from Equations (30) and (33) can be minimized with respect to  $\hat{\mathbf{h}}_2$  and  $\hat{\mathbf{h}}_3$ , respectively.

The previous procedure suggests an iterative approach. To start the algorithm, a set of initial values should be provided for two of the estimated impulse responses. For example, we can choose  $\hat{\mathbf{h}}_2^{(0)} = (1/L_2) [1 \ 1 \ \dots \ 1]^T$  and  $\hat{\mathbf{h}}_3^{(0)} = (1/L_3) [1 \ 1 \ \dots \ 1]^T$ . Hence, based on Equations (26) and (27), one may compute

$$\mathbf{p}_1^{(0)} = (\hat{\mathbf{h}}_3^{(0)} \otimes \hat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1})^T \mathbf{p}, \quad (36)$$

$$\mathbf{R}_1^{(0)} = (\hat{\mathbf{h}}_3^{(0)} \otimes \hat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1})^T \mathbf{R} (\hat{\mathbf{h}}_3^{(0)} \otimes \hat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1}). \quad (37)$$

In the first iteration, the first cost function to be minimized results from Equation (25) (using Equations (36) and (37)), that is,

$$J_{\hat{\mathbf{h}}_2, \hat{\mathbf{h}}_3}(\hat{\mathbf{h}}_1^{(1)}) = \sigma_d^2 - 2(\hat{\mathbf{h}}_1^{(1)})^T \mathbf{p}_1^{(0)} + (\hat{\mathbf{h}}_1^{(1)})^T \mathbf{R}_1^{(0)} \hat{\mathbf{h}}_1^{(1)},$$

which leads to the solution  $\hat{\mathbf{h}}_1^{(1)} = (\mathbf{R}_1^{(0)})^{-1} \mathbf{p}_1^{(0)}$ . Also, since  $\hat{\mathbf{h}}_1^{(1)}$  and  $\hat{\mathbf{h}}_3^{(0)}$  are now available, we can evaluate (based on Equations (31) and (32))

$$\mathbf{p}_2^{(1)} = (\hat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(1)})^T \mathbf{p}, \quad (38)$$

$$\mathbf{R}_2^{(1)} = (\hat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(1)})^T \mathbf{R} (\hat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(1)}), \quad (39)$$

so that the cost function from Equation (30) becomes

$$J_{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_3}(\hat{\mathbf{h}}_2^{(1)}) = \sigma_d^2 - 2(\hat{\mathbf{h}}_2^{(1)})^T \mathbf{p}_2^{(1)} + (\hat{\mathbf{h}}_2^{(1)})^T \mathbf{R}_2^{(1)} \hat{\mathbf{h}}_2^{(1)},$$

while its minimization leads to  $\hat{\mathbf{h}}_2^{(1)} = (\mathbf{R}_2^{(1)})^{-1} \mathbf{p}_2^{(1)}$ . Finally, using the solutions  $\hat{\mathbf{h}}_1^{(1)}$  and  $\hat{\mathbf{h}}_2^{(1)}$ , we can find  $\hat{\mathbf{h}}_3^{(1)}$  in a similar manner. First, we evaluate (based on Equations (34) and (35))

$$\mathbf{p}_3^{(1)} = (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(1)} \otimes \hat{\mathbf{h}}_1^{(1)})^T \mathbf{p}, \quad (40)$$

$$\mathbf{R}_3^{(1)} = (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(1)} \otimes \hat{\mathbf{h}}_1^{(1)})^T \mathbf{R} (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(1)} \otimes \hat{\mathbf{h}}_1^{(1)}). \quad (41)$$

Then, we minimize the cost function (which results from Equation (33)):

$$J_{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2}(\hat{\mathbf{h}}_3^{(1)}) = \sigma_d^2 - 2(\hat{\mathbf{h}}_3^{(1)})^T \mathbf{p}_3^{(1)} + (\hat{\mathbf{h}}_3^{(1)})^T \mathbf{R}_3^{(1)} \hat{\mathbf{h}}_3^{(1)},$$

which leads to the solution  $\hat{\mathbf{h}}_3^{(1)} = (\mathbf{R}_3^{(1)})^{-1} \mathbf{p}_3^{(1)}$ . Continuing the iterative procedure, at iteration number  $n$ , we obtain the estimates of the impulse responses based on the following steps:

$$\begin{aligned} \hat{\mathbf{h}}_1^{(n)} &= (\mathbf{R}_1^{(n-1)})^{-1} \mathbf{p}_1^{(n-1)}, \\ \mathbf{p}_2^{(n)} &= (\hat{\mathbf{h}}_3^{(n-1)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(n)})^T \mathbf{p}, \\ \mathbf{R}_2^{(n)} &= (\hat{\mathbf{h}}_3^{(n-1)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(n)})^T \mathbf{R} (\hat{\mathbf{h}}_3^{(n-1)} \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1^{(n)}), \\ \hat{\mathbf{h}}_2^{(n)} &= (\mathbf{R}_2^{(n)})^{-1} \mathbf{p}_2^{(n)}, \\ \mathbf{p}_3^{(n)} &= (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(n)} \otimes \hat{\mathbf{h}}_1^{(n)})^T \mathbf{p}, \\ \mathbf{R}_3^{(n)} &= (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(n)} \otimes \hat{\mathbf{h}}_1^{(n)})^T \mathbf{R} (\mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2^{(n)} \otimes \hat{\mathbf{h}}_1^{(n)}), \\ \hat{\mathbf{h}}_3^{(n)} &= (\mathbf{R}_3^{(n)})^{-1} \mathbf{p}_3^{(n)}. \end{aligned}$$

Thus, the global impulse response at iteration  $n$  results in

$$\hat{\mathbf{h}}^{(n)} = \hat{\mathbf{h}}_3^{(n)} \otimes \hat{\mathbf{h}}_2^{(n)} \otimes \hat{\mathbf{h}}_1^{(n)}. \quad (42)$$

The proposed iterative Wiener filter for trilinear forms represents an extension of the solution presented in Reference [7] (in the context of bilinear forms). However, when the MISO system identification problem results are based on Equation (10), it is more advantageous to use the algorithm tailored for trilinear forms instead of reformulating the problem in terms of multiple bilinear forms. The trilinear approach has some similarities (to some extent) with that introduced in Reference [33]. However, the batch Trilinear Wiener-Hopf (TriWH) algorithm from Reference [33] is more related to an adaptive approach, since the statistics are estimated within the algorithm. On the other hand, in the case of our iterative Wiener filter, the estimates of the statistics are considered to be *a priori* available (see also the related discussion in the next section), which is basically in the spirit of the Wiener filter.

#### 4. LMS and NLMS Algorithms for Trilinear Forms

It is well-known that the Wiener filter presents several limitations that may make it unsuitable to be used in practice (e.g., the matrix inversion operation, the correlation matrix estimation, etc.). For this reason, a more convenient manner of treating the system identification problem is through adaptive



filtering. One of the simplest types of adaptive algorithms is the LMS, which will be presented in the following, tailored for the new trilinear form approach.

First, let us consider the three estimated impulse responses  $\hat{\mathbf{h}}_k$ ,  $k = 1, 2, 3$ , together with the corresponding *a priori* error signals:

$$e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = d(t) - \hat{\mathbf{h}}_1^T(t-1)\mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t), \quad (43)$$

$$e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = d(t) - \hat{\mathbf{h}}_2^T(t-1)\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t), \quad (44)$$

$$e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) = d(t) - \hat{\mathbf{h}}_3^T(t-1)\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t), \quad (45)$$

where

$$\mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = \left[ \hat{\mathbf{h}}_3(t-1) \otimes \hat{\mathbf{h}}_2(t-1) \otimes \mathbf{I}_{L_1} \right] \mathbf{x}(t), \quad (46)$$

$$\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = \left[ \hat{\mathbf{h}}_3(t-1) \otimes \mathbf{I}_{L_2} \otimes \hat{\mathbf{h}}_1(t-1) \right] \mathbf{x}(t), \quad (47)$$

$$\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) = \left[ \mathbf{I}_{L_3} \otimes \hat{\mathbf{h}}_2(t-1) \otimes \hat{\mathbf{h}}_1(t-1) \right] \mathbf{x}(t). \quad (48)$$

It can be checked that  $e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t)$ . In this context, the LMS updates for the three filters are the following:

$$\begin{aligned} \hat{\mathbf{h}}_1(t) &= \hat{\mathbf{h}}_1(t-1) - \frac{\mu_{\hat{\mathbf{h}}_1}}{2} \times \frac{\partial e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}^2(t)}{\partial \hat{\mathbf{h}}_1(t-1)} \\ &= \hat{\mathbf{h}}_1(t-1) + \mu_{\hat{\mathbf{h}}_1} \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t), \end{aligned} \quad (49)$$

$$\begin{aligned} \hat{\mathbf{h}}_2(t) &= \hat{\mathbf{h}}_2(t-1) - \frac{\mu_{\hat{\mathbf{h}}_2}}{2} \times \frac{\partial e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}^2(t)}{\partial \hat{\mathbf{h}}_2(t-1)} \\ &= \hat{\mathbf{h}}_2(t-1) + \mu_{\hat{\mathbf{h}}_2} \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t), \end{aligned} \quad (50)$$

$$\begin{aligned} \hat{\mathbf{h}}_3(t) &= \hat{\mathbf{h}}_3(t-1) - \frac{\mu_{\hat{\mathbf{h}}_3}}{2} \times \frac{\partial e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}^2(t)}{\partial \hat{\mathbf{h}}_3(t-1)} \\ &= \hat{\mathbf{h}}_3(t-1) + \mu_{\hat{\mathbf{h}}_3} \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t), \end{aligned} \quad (51)$$

where  $\mu_{\hat{\mathbf{h}}_1} > 0$ ,  $\mu_{\hat{\mathbf{h}}_2} > 0$ ,  $\mu_{\hat{\mathbf{h}}_3} > 0$  represent the step-size parameters. Relations (49)–(51) define the LMS algorithm for trilinear forms, namely LMS-TF.

For the initialization of the estimated impulse responses, we use

$$\hat{\mathbf{h}}_1(0) = [1 \ 0 \ \dots \ 0]^T, \quad (52)$$

$$\hat{\mathbf{h}}_2(0) = \frac{1}{L_2} [1 \ 1 \ \dots \ 1]^T, \quad (53)$$

$$\hat{\mathbf{h}}_3(0) = \frac{1}{L_3} [1 \ 1 \ \dots \ 1]^T. \quad (54)$$

In the end, we can obtain the global filter using Relation (20). Alternatively, this global impulse response may be identified directly with the regular LMS algorithm by using the following update:

$$\hat{\mathbf{h}}(t) = \hat{\mathbf{h}}(t-1) + \mu_{\hat{\mathbf{h}}} \mathbf{x}(t) e(t), \quad (55)$$

where

$$e(t) = d(t) - \hat{\mathbf{h}}(t-1)\mathbf{x}(t) \quad (56)$$

and  $\mu_{\hat{\mathbf{h}}}$  is the global step-size parameter.

However, an observation needs to be made regarding the update in Relation (55): this involves the presence of an adaptive filter of length  $L_1L_2L_3$ , whereas the LMS-TF algorithm, which is defined by the update Relations (49)–(51), uses three shorter filters of lengths  $L_1$ ,  $L_2$ , and  $L_3$ , respectively. Basically, a system identification problem of size  $L_1L_2L_3$  (as in the regular approach) was reformulated in terms of three shorter filters of lengths  $L_1$ ,  $L_2$ , and  $L_3$ . Taking into account that we usually have  $L_1 + L_2 + L_3 \ll L_1L_2L_3$ , the advantage of the trilinear approach (in terms of reducing the complexity) could be important. Therefore, the complexity of this new approach is lower and the convergence rate is expected to be faster.

The step-size parameters in Relations (49)–(51) take constant values, chosen such that they ensure the convergence of the algorithm and a good compromise between convergence speed and steady-state misadjustment. Nevertheless, when dealing with non-stationary signals, it may be more appropriate to use time-dependent step-sizes, which lead to the following update relations:

$$\hat{\mathbf{h}}_1(t) = \hat{\mathbf{h}}_1(t-1) + \mu_{\hat{\mathbf{h}}_1}(t) \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t), \quad (57)$$

$$\hat{\mathbf{h}}_2(t) = \hat{\mathbf{h}}_2(t-1) + \mu_{\hat{\mathbf{h}}_2}(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t), \quad (58)$$

$$\hat{\mathbf{h}}_3(t) = \hat{\mathbf{h}}_3(t-1) + \mu_{\hat{\mathbf{h}}_3}(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t). \quad (59)$$

For deriving the expressions of the step-size parameters, we take into consideration the stability conditions and we target to cancel the following expressions, which represent the *a posteriori* error signals [34]:

$$\varepsilon_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = d(t) - \hat{\mathbf{h}}_1^T(t) \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t), \quad (60)$$

$$\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = d(t) - \hat{\mathbf{h}}_2^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t), \quad (61)$$

$$\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) = d(t) - \hat{\mathbf{h}}_3^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t). \quad (62)$$

By replacing Relation (49) in (60), Relation (50) in (61), and Relation (51) in (62), respectively, and by imposing the conditions  $\varepsilon_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = 0$ ,  $\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = 0$ , and  $\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) = 0$ , we obtain that

$$\varepsilon_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) = e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) \left[ 1 - \mu_{\hat{\mathbf{h}}_1}(t) \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) \right] = 0, \quad (63)$$

$$\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) = e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) \left[ 1 - \mu_{\hat{\mathbf{h}}_2}(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) \right] = 0, \quad (64)$$

$$\varepsilon_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) = e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) \left[ 1 - \mu_{\hat{\mathbf{h}}_3}(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) \right] = 0. \quad (65)$$

Consequently, assuming that  $e_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t) \neq 0$ ,  $e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t) \neq 0$ , and  $e_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t) \neq 0$ , the following expressions for the step-size parameters result:

$$\mu_{\hat{\mathbf{h}}_1}(t) = \frac{1}{\mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_2\hat{\mathbf{h}}_3}(t)}, \quad (66)$$

$$\mu_{\hat{\mathbf{h}}_2}(t) = \frac{1}{\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_3}(t)}, \quad (67)$$

$$\mu_{\hat{\mathbf{h}}_3}(t) = \frac{1}{\mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1\hat{\mathbf{h}}_2}(t)}. \quad (68)$$

In order to obtain a good balance between convergence rate and misadjustment, three positive constants,  $0 < \alpha_{\hat{\mathbf{h}}_1} < 1$ ,  $0 < \alpha_{\hat{\mathbf{h}}_2} < 1$ , and  $0 < \alpha_{\hat{\mathbf{h}}_3} < 1$ , are employed [35]. In addition, three regularization constants  $\delta_{\hat{\mathbf{h}}_1} > 0$ ,  $\delta_{\hat{\mathbf{h}}_2} > 0$ , and  $\delta_{\hat{\mathbf{h}}_3} > 0$ , usually chosen to be proportional

to the variance of the input signal [36], are added to the denominators of the step-size parameters. Finally, the updates of the NLMS algorithm for trilinear forms (NLMS-TF) become

$$\hat{\mathbf{h}}_1(t) = \hat{\mathbf{h}}_1(t-1) + \frac{\alpha_{\hat{\mathbf{h}}_1}(t) \mathbf{x}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3}(t)}{\mathbf{x}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_2 \hat{\mathbf{h}}_3}(t) + \delta_{\hat{\mathbf{h}}_1}}, \quad (69)$$

$$\hat{\mathbf{h}}_2(t) = \hat{\mathbf{h}}_2(t-1) + \frac{\alpha_{\hat{\mathbf{h}}_2}(t) \mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3}(t) e_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3}(t)}{\mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_3}(t) + \delta_{\hat{\mathbf{h}}_2}}, \quad (70)$$

$$\hat{\mathbf{h}}_3(t) = \hat{\mathbf{h}}_3(t-1) + \frac{\alpha_{\hat{\mathbf{h}}_3}(t) \mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2}(t) e_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2}(t)}{\mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2}^T(t) \mathbf{x}_{\hat{\mathbf{h}}_1 \hat{\mathbf{h}}_2}(t) + \delta_{\hat{\mathbf{h}}_3}}. \quad (71)$$

The initializations of the estimated filters may be the same as Equations (52)–(54). In a similar way as for the LMS algorithm, the global impulse response can be identified using the regular NLMS:

$$\hat{\mathbf{h}}(t) = \hat{\mathbf{h}}(t-1) + \frac{\alpha_{\hat{\mathbf{h}}} \mathbf{x}(t) e(t)}{\mathbf{x}^T(t) \mathbf{x}(t) + \delta_{\hat{\mathbf{h}}}}, \quad (72)$$

where  $e(t)$  is given in Equation (56). The parameters  $\alpha_{\hat{\mathbf{h}}}$  and  $\delta_{\hat{\mathbf{h}}}$  represent the normalized step-size parameter and the regularization constant for the global filter, respectively. As previously shown in Reference [9] for bilinear forms, the global misalignment can be controlled by using a constraint on the sum of the normalized step-sizes, and this sum should be smaller than 1. In this way, for different values of  $\alpha_{\hat{\mathbf{h}}_1}$ ,  $\alpha_{\hat{\mathbf{h}}_2}$ ,  $\alpha_{\hat{\mathbf{h}}_3}$  fulfilling this condition, the misalignment of the global filter is the same. On the other hand, in the case when  $\alpha_{\hat{\mathbf{h}}_1} = \alpha_{\hat{\mathbf{h}}_2} = \alpha_{\hat{\mathbf{h}}_3}$ , the three filters achieve the same level of the misalignment.

Again, we notice that the global impulse response identification involves the use of a filter of length  $L_1 L_2 L_3$ . Because the trilinear approach uses three much shorter impulse responses of lengths  $L_1$ ,  $L_2$ , and  $L_3$ , respectively, it is expected that this new solution will yield a faster convergence. This will be shown through simulations.

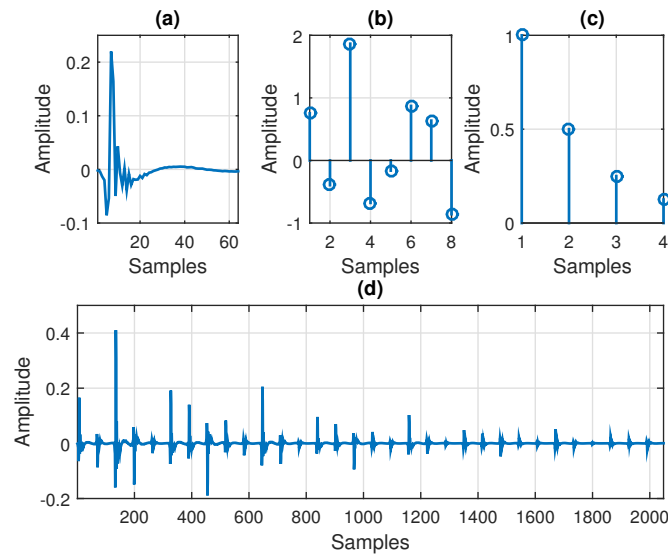
The NLMS-TF algorithm proposed here is similar to that presented in Reference [19]. However, our choice of the system impulse responses used in simulations is different from that in Reference [19]. On the contrary, we aim to show the performance of the algorithm in a scenario that includes a real echo path. In addition, we also study the tracking capability of the algorithm.

## 5. Simulation Results

In order to show the performance of our approach, we perform simulations in which we compare the trilinear forms of the proposed algorithms with their regular counterparts.

### 5.1. Iterative Wiener Filter

In this section, the performance of the proposed iterative Wiener filter for trilinear forms is evaluated in the context of system identification. The input signals that form  $\mathcal{X}(t)$  are AR(1) processes, which are obtained by generating white Gaussian noises and then filtering them through a first-order system  $1/(1 - 0.9z^{-1})$ . The additive noise  $v(t)$ , corrupting the output signal  $y(t)$ , is white and Gaussian, with the variance set to  $\sigma_v^2 = 0.01$ . The impulse responses used in simulations are depicted in Figure 1. The impulse response  $\mathbf{h}_1$  is the first impulse response from the G168 Recommendation [37], of length  $L_1 = 64$  (see Figure 1a). Next,  $\mathbf{h}_2$  is a random impulse response (with Gaussian distribution) of length  $L_2 = 8$  (as shown in Figure 1b). Finally, the coefficients of the impulse response  $\mathbf{h}_3$  (depicted in Figure 1c) are evaluated as  $h_{3l_3} = 0.5^{l_3-1}$ ,  $l_3 = 1, 2, \dots, L_3$ , using  $L_3 = 4$ . Therefore, the global impulse response from Figure 1d results in  $\mathbf{h} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1$  and its length is  $L = L_1 L_2 L_3 = 2048$ . As we can see, this global impulse response is similar (to some extent) to a channel with echoes, similar to an acoustic echo path.



**Figure 1.** Impulse responses used in simulations: (a)  $\mathbf{h}_1$  of length  $L_1 = 64$  (the first impulse response from G168 Recommendation [37]), (b)  $\mathbf{h}_2$  of length  $L_2 = 8$  (random impulse response with Gaussian distribution), (c)  $\mathbf{h}_3$  of length  $L_3 = 4$  (its elements are evaluated as  $h_{3l_3} = 0.5^{l_3-1}$ ,  $l_3 = 1, \dots, L_3$ ), and (d) the global impulse response  $\mathbf{h} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1$  of length  $L = L_1 L_2 L_3 = 2048$ .

In order to evaluate the identification of the individual filters  $\mathbf{h}_k$ ,  $k = 1, 2, 3$ , we should use the normalized projection misalignment (NPM) [38]:

$$\text{NPM}(\mathbf{h}_k, \hat{\mathbf{h}}_k) = 1 - \left( \frac{\mathbf{h}_k^T \hat{\mathbf{h}}_k}{\|\mathbf{h}_k\| \|\hat{\mathbf{h}}_k\|} \right)^2, \quad (73)$$

where  $\|\cdot\|$  denotes the Euclidean norm. For the identification of the global impulse response,  $\mathbf{h}(t)$ , we use the normalized misalignment (NM):

$$\text{NM}(\mathbf{h}, \hat{\mathbf{h}}) = \frac{\|\mathbf{h} - \hat{\mathbf{h}}\|^2}{\|\mathbf{h}\|^2}. \quad (74)$$

We consider that the covariance matrix  $\mathbf{R}$  and the cross-correlation vector  $\mathbf{p}$  are estimated based on  $N$  data samples:

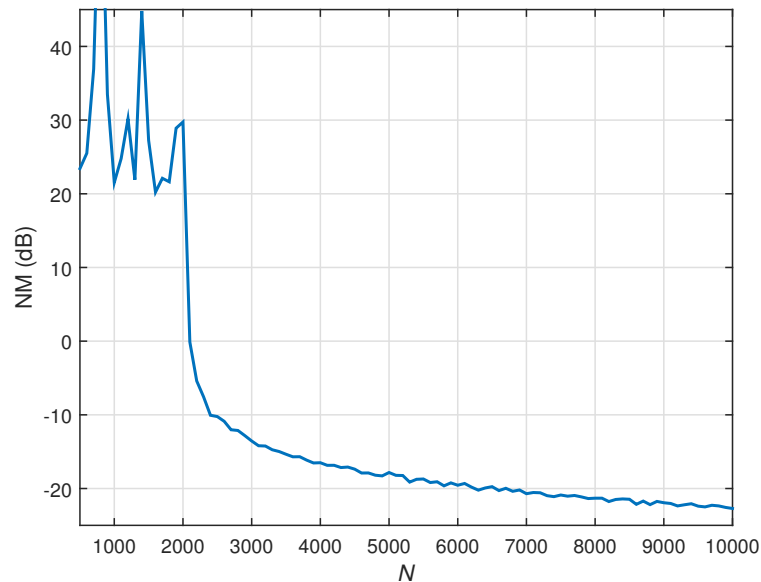
$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{t=1}^N \mathbf{x}(t) \mathbf{x}^T(t), \quad (75)$$

$$\hat{\mathbf{p}} = \frac{1}{N} \sum_{t=1}^N \mathbf{x}(t) d(t). \quad (76)$$

These two terms are *a priori* computed and they are used afterwards (instead of  $\mathbf{R}$  and  $\mathbf{p}$ ) for both the conventional and iterative Wiener filters.

The matrix involved in the linear system, to be solved in the case of the conventional Wiener filter from Equation (19), is of size  $L \times L$ ; hence, a number of data samples larger than  $L$  are needed to estimate the statistics in Equations (75) and (76), in order to obtain a good solution. This is shown in Figure 2, where different values of  $N$  (that is, the available amount of data in Equations (75) and (76)) are used and the solution provided by the conventional Wiener filter is evaluated for each of these values. Similar to Equation (74), the performance measure in Figure 2 is the NM (in dB), which is defined as  $10 \log_{10} \left( \frac{\|\mathbf{h} - \hat{\mathbf{h}}_{\text{W}}\|^2}{\|\mathbf{h}\|^2} \right)$ ; in this case, the conventional Wiener solution  $\hat{\mathbf{h}}_{\text{W}}$  results

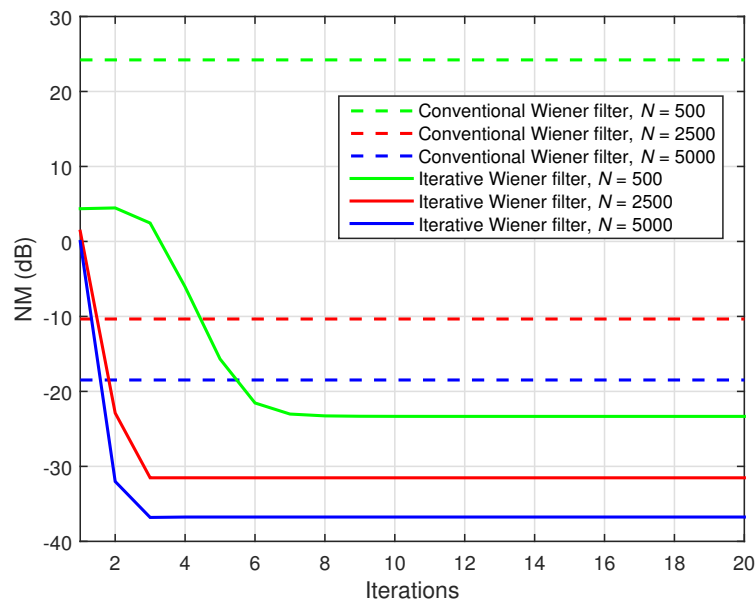
from Equation (19) (using Equations (75) and (76)). As we can see, the conventional Wiener filter achieves a reasonable decrease in misalignment only when a large amount of data (i.e.,  $N > L$ ) are used to estimate the statistics in Equations (75) and (76).



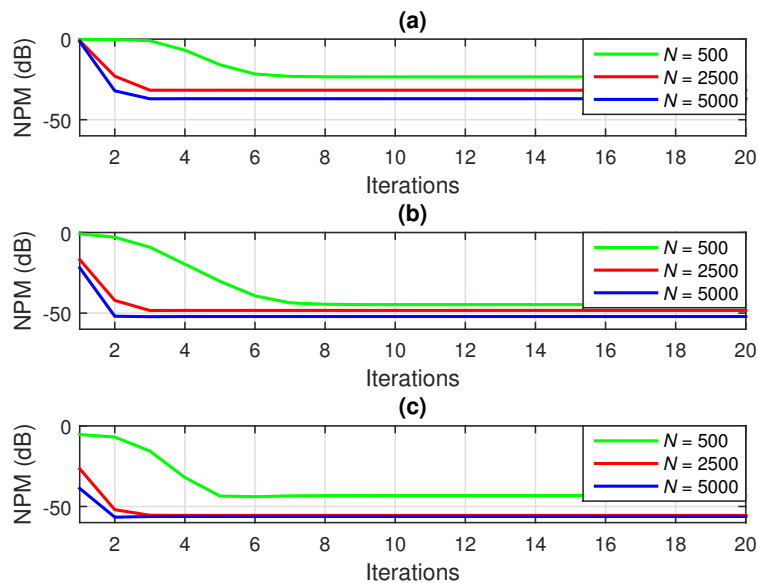
**Figure 2.** NM of the conventional Wiener filter as a function of the number of available data samples used to estimate the statistics ( $N$ ), for the identification of the global impulse response from Figure 1d. The input signals are AR(1) processes,  $L = 2048$ , and  $\sigma_v^2 = 0.01$ .

The main advantage provided by the iterative Wiener filter is that it operates with much shorter filters (due to the decomposition in Equation (20)) and, consequently, the dimensions of the linear systems of equations to be solved are significantly reduced. Therefore, even with a small amount of data (i.e.,  $N < L$ ), the iterative Wiener filter is able to obtain a reliable estimation. This advantage is outlined in Figure 3, where the solution provided by the conventional Wiener filter (based on Equation (19) and using Equations (75) and (76)) is compared to the iterative Wiener filter from Equation (42). The performance measure is the NM (in dB), which is evaluated based on Equation (74), for the identification of the global system  $\mathbf{h}$ . Three amounts of data are considered in this experiment, that is,  $N = 500, 2500$ , and  $5000$ . Clearly, in the first case ( $N = 500$ ), the conventional Wiener filter leads to an inaccurate solution due to the small amount of data (as compared to  $L = 2048$ ). When the amount of data slightly exceeds the value of  $L$  (e.g.,  $N = 2500$ ), the conventional Wiener filter provides a more reliable solution, that is, the misalignment attenuation is approximately  $-10$  dB. Finally, for a large amount of data ( $N = 5000$ ), this conventional solution is improved in terms of accuracy (e.g., the misalignment is close to  $-20$  dB). On the other hand, in all the previous cases, the proposed iterative Wiener filter achieves a much more accurate solution (with only a few iterations), which outperforms by far the conventional one (even in the case when a small amount of data are available, e.g.,  $N = 500$ ). For example, the iterative Wiener filter which uses  $N = 500$  yields a lower misalignment level with respect to the conventional Wiener filter with  $N = 5000$ .

In Figure 4, the performance of the iterative Wiener filter is also illustrated using the NPMs (in dB), based on Equation (73), for the identification of the individual impulse responses from Figure 1a–c. Basically, the same conclusion applies, that is, only a few iterations are required by the iterative Wiener filter to achieve a reliable solution (even for a small amount of data).



**Figure 3.** NM of the conventional and iterative Wiener filters, for different values of the number of available data samples used to estimate the statistics ( $N$ ), for the identification of the global impulse response from Figure 1d. The input signals are AR(1) processes,  $L = 2048$ , and  $\sigma_v^2 = 0.01$ .

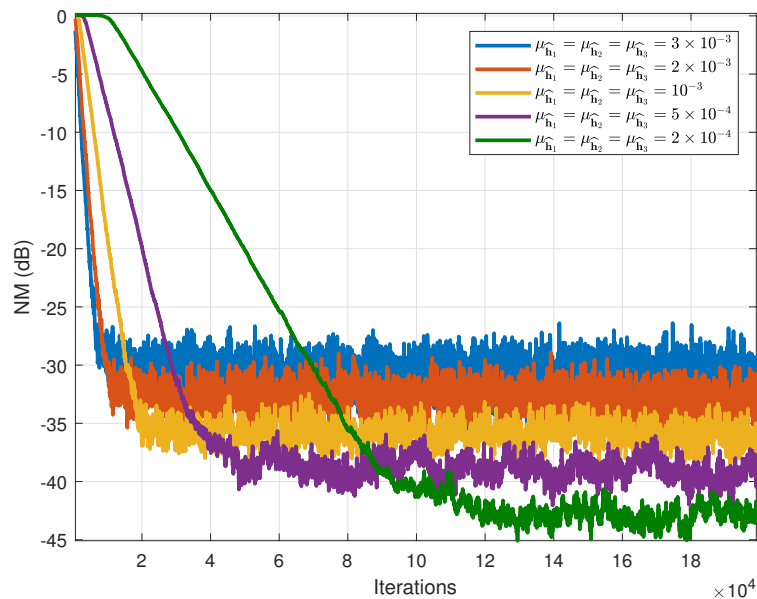


**Figure 4.** NPM of the iterative Wiener filter, for different values of the number of available data samples used to estimate the statistics ( $N$ ), for the identification of the individual impulse responses from Figure 1a–c: (a) NPM  $(\mathbf{h}_1, \hat{\mathbf{h}}_1^{(n)})$ , (b) NPM  $(\mathbf{h}_2, \hat{\mathbf{h}}_2^{(n)})$ , and (c) NPM  $(\mathbf{h}_3, \hat{\mathbf{h}}_3^{(n)})$ . The input signals are AR(1) processes,  $L_1 = 64$ ,  $L_2 = 8$ ,  $L_3 = 4$ , and  $\sigma_v^2 = 0.01$ .

## 5.2. LMS-TF and NLMS-TF

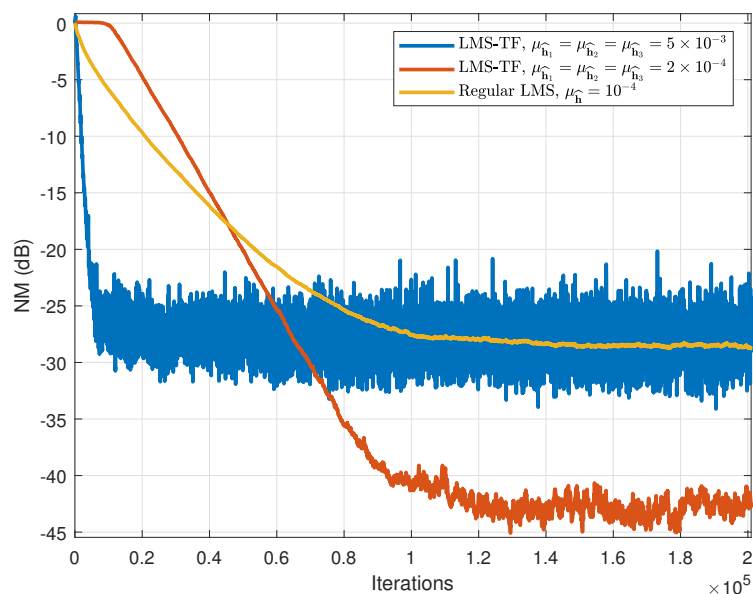
For the second set of simulations, the setup is the same as in the previous experiments. First, we aim to show the influence of the constant step-size values on the performance of the LMS-TF algorithm. The performance in terms of the NM (in dB) is shown in Figure 5.

It can be seen that if the step-sizes take large values, the LMS-TF algorithm reaches convergence after less than  $10^4$  iterations. Then, as these values decrease, the convergence becomes slower but the steady-state value of the NM also decreases, highlighting the compromise between convergence rate and NM value.



**Figure 5.** NM of the LMS-TF algorithm using different values of the step-size parameters.

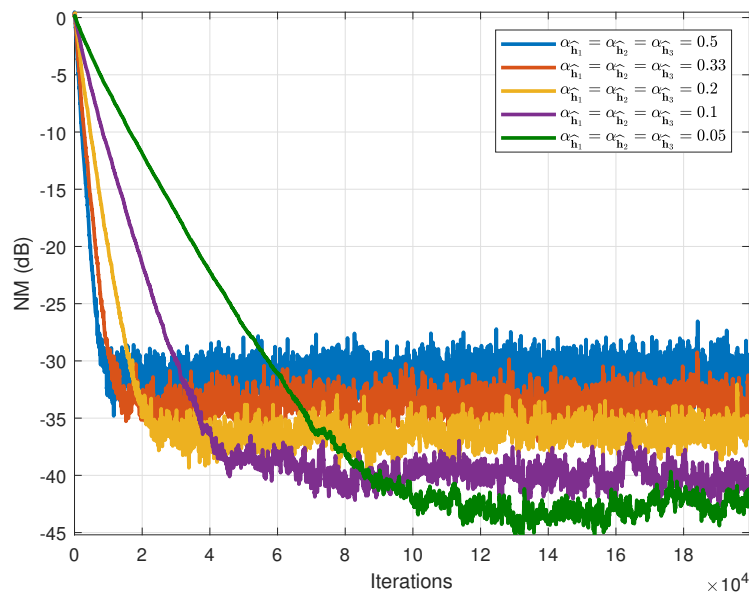
Next, we illustrate the improvement brought by the proposed solution, by comparing the LMS-TF algorithm to its regular counterpart, applied for the identification of the global filter. Figure 6 shows the values of the NM for the regular LMS filter and the LMS-TF. The first observation is that, in order to reach the same steady-state value of the NM, the regular LMS algorithm needs many more iterations. On the other hand, for a similar convergence speed, the final NM provided by the LMS-TF is much lower than that offered by its regular counterpart. This proves that the proposed solution offers a significant improvement with respect to the classical approach.



**Figure 6.** NM of the LMS-TF and regular LMS algorithms.

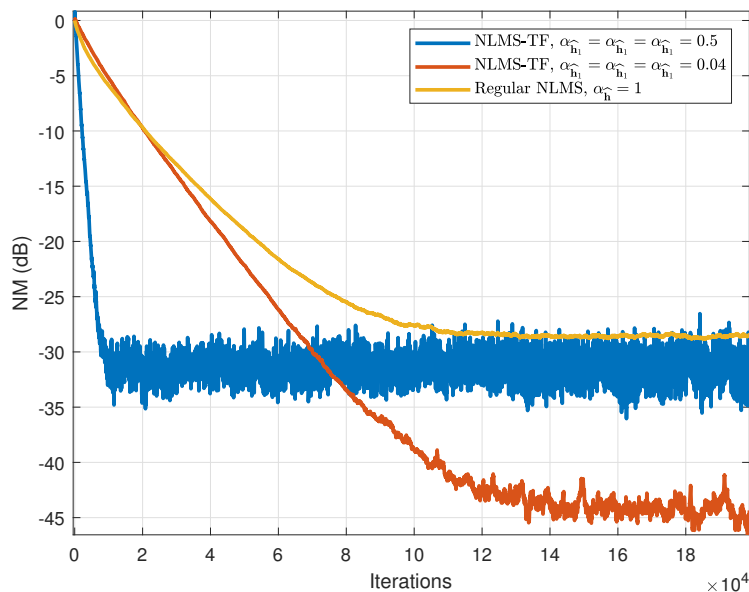
The next step is to study the behavior of the NLMS-TF filter. First, the performance of the NLMS-TF algorithm is depicted in Figure 7, for different values of the normalized step-sizes. The regularization constants are  $\delta_{\hat{h}_1} = \delta_{\hat{h}_2} = \delta_{\hat{h}_3} = 0.001$ . The same conclusion as for the LMS-TF case is valid, namely that the decrease in the normalized step-sizes leads to a smaller value of the final NM but at the cost of a slower convergence rate. When the step-size values decrease 10 times, the number

of iterations needed to reach convergence increases almost 10 times, while the steady-state NM value decreases by a bit more than 10 dB.



**Figure 7.** NM of the NLMS-TF algorithm using different values of the step-size parameters.

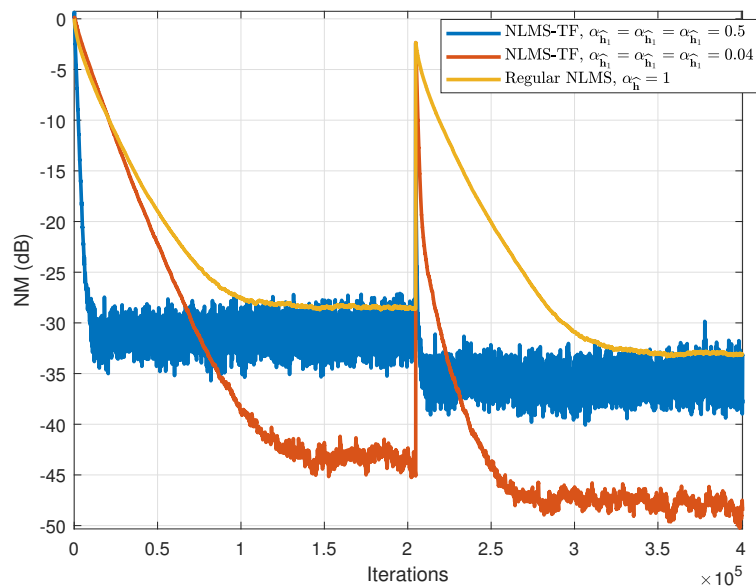
We then compare the NLMS-TF algorithm with its regular counterpart (applied on the global filter) in Figure 8. Again, we observe that the NLMS-TF behaves better than the regular NLMS algorithm, from the perspective of both convergence rate and final NM value.



**Figure 8.** NM of the NLMS-TF and regular NLMS algorithms.

Finally, the tracking capability of the NLMS-TF algorithm is of interest, that is, the capability of the algorithm to react to abrupt changes to the impulse responses. In order to study this characteristic, we simulated a sudden change of the random impulse response  $\mathbf{h}_2$  in the middle of the experiment. The results are presented in Figure 9. The improvement brought by the NLMS-TF algorithm is clear. The algorithm tracks faster after the change of the system, while the value of the NM is smaller, as compared to the regular NLMS filter. This proves that even if the environment changes, the proposed approach exhibits good behavior.





**Figure 9.** NM of the NLMS-TF and regular NLMS algorithms. The impulse response  $\mathbf{h}_2$  changes in the middle of the experiment.

## 6. Conclusions

In this paper, we addressed the problem of multilinear system identification, focusing in particular on trilinear forms in the framework of MISO systems. Trilinear forms are defined with respect to the impulse responses of the system and are treated using third-order tensors. In this context, we derived the corresponding Wiener filter, as well as the LMS and NLMS adaptive algorithms, tailored for such trilinear forms (LMS-TF and NLMS-TF). We have shown through simulations that the proposed algorithms lead to better solutions as compared to their regular counterparts, due to the reformulation of the system identification problem of high dimension in lower dimension problems. Experimental results support the theoretical analysis and highlight the good performance of the proposed solutions for the problem of system identification. Future work can focus on extending the approach to higher-order multilinear systems.

**Author Contributions:** Conceptualization, L.-M.D.; Formal analysis, S.C.; Methodology, J.B.; Software, C.P.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Roy, R.; Sherman, J. A learning technique for Volterra series representation. *IEEE Trans. Autom. Control* **1967**, *12*, 761–764. [[CrossRef](#)]
- Barker, H.A.; Obidegwu, S.N.; Pradisthayon, T. Performance of antisymmetric pseudorandom signals in the measurement of 2nd-order Volterra kernels by crosscorrelation. *Proc. IEEE* **1972**, *119*, 353–362.
- Annabestani, M.; Naghavi, N.; Nejad, M.M. Nonautoregressive nonlinear identification of IPMC in large deformation situations using generalized Volterra-based approach. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 2866–2872. [[CrossRef](#)]
- Zhang, Z.; Ma, Y. Modeling of rate-dependent hysteresis using a GPO-based adaptive filter. *Sensors* **2016**, *16*, 205. [[CrossRef](#)]
- Rugh, W.J. *Nonlinear System Theory: The Volterra/Wiener Approach*; Johns Hopkins University Press: Baltimore, MD, USA, 1981.
- Carassale, L.; Kareem, A. Modeling nonlinear systems by Volterra series. *J. Eng. Mech.* **2010**, *136*, 801–818. [[CrossRef](#)]

7. Benesty, J.; Paleologu, C.; Ciochină, S. On the identification of bilinear forms with the Wiener filter. *IEEE Signal Process. Lett.* **2017**, *24*, 653–657. [[CrossRef](#)]
8. Bai, E.-W.; Li, D. Convergence of the iterative Hammerstein system identification algorithm. *IEEE Trans. Autom. Control* **2004**, *49*, 1929–1940. [[CrossRef](#)]
9. Paleologu, C.; Benesty, J.; Ciochină, S. Adaptive filtering for the identification of bilinear forms. *Digit. Signal Process.* **2018**, *75*, 153–167. [[CrossRef](#)]
10. Dogariu, L.; Paleologu, C.; Ciochină, S.; Benesty, J.; Piantanida, P. Identification of bilinear forms with the Kalman filter. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4134–4138.
11. Dogariu, L.-M.; Ciochină, S.; Paleologu, C.; Benesty, J. A connection between the Kalman filter and an optimized LMS algorithm for bilinear forms. *Algorithms* **2018**, *11*, 211. [[CrossRef](#)]
12. Gesbert, D.; Duhamel, P. Robust blind joint data/channel estimation based on bilinear optimization. In Proceedings of the 8th Workshop on Statistical Signal and Array Processing, Corfu, Greece, 24–26 June 1996; pp. 168–171.
13. Stenger, A.; Kellermann, W. Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling. *Signal Process.* **2000**, *80*, 1747–1760. [[CrossRef](#)]
14. Ribeiro, L.N.; Schwarz, S.; Rupp, M.; de Almeida, A.L.F.; Mota, J.C.M. A low-complexity equalizer for massive MIMO systems based on array separability. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 28 August–2 September 2017; pp. 2522–2526.
15. da Costa, M.N.; Favier, G.; Romano, J.M.T. Tensor modelling of MIMO communication systems with performance analysis and Kronecker receivers. *Signal Process.* **2018**, *145*, 304–316. [[CrossRef](#)]
16. Ljung, L. *System Identification: Theory for the User*, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 1999.
17. Gay, S.L.; Benesty, J. (Eds.) *Acoustic Signal Processing for Telecommunication*; Kluwer Academic Publisher: Boston, MA, USA, 2000.
18. Benesty, J.; Gaensler, T.; Morgan, D.R.; Sondhi, M.M.; Gay, S.L. *Advances in Network and Acoustic Echo Cancellation*; Springer: Berlin, Germany, 2001.
19. Rupp, M.; Schwarz, S. A tensor LMS algorithm. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 3347–3351.
20. Rupp, M.; Schwarz, S. Gradient-based approaches to learn tensor products. In Proceedings of the 2015 23rd European Signal Processing Conference (EUSIPCO), Nice, France, 31 August–4 September 2015; pp. 2486–2490.
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. 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](#)]
23. Boussé, 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](#)]
24. Lathauwer, L.D. *Signal Processing Based on Multilinear Algebra*. Ph.D. Thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1997.
25. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
26. Comon, P. Tensors: A brief introduction. *IEEE Signal Process. Mag.* **2014**, *31*, 44–53. [[CrossRef](#)]
27. Cichocki, A.; Mandic, D.P.; Phan, A.H.; Caiafa, C.F.; Zhou, G.; Zhao, Q.; Lathauwer, L.D. Tensor decompositions for signal processing applications. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163. [[CrossRef](#)]
28. Loan, C.F.V. The ubiquitous Kronecker product. *J. Comput. Appl. Math.* **2000**, *123*, 85–100. [[CrossRef](#)]
29. 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](#)]
30. Elisei-Iliescu, C.; Paleologu, C.; Benesty, J.; Stanciu, C.; Anghel, C.; Ciochină, S. Recursive least-squares algorithms for the identification of low-rank systems. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 903–918. [[CrossRef](#)]
31. Kiers, H.A.L. Towards a standardized notation and terminology in multiway analysis. *J. Chemom.* **2000**, *14*, 105–122. [[CrossRef](#)]
32. Kroonenberg, P. *Applied Multiway Data Analysis*; Wiley: Hoboken, NJ, USA, 2008.

33. Ribeiro, L.N.; de Almeida, A.L.F.; Mota, J.C.M. Identification of separable systems using trilinear filtering. In Proceedings of the 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Cancun, Mexico, 13–16 December 2015; pp. 189–192.
34. Morgan, D.R.; Kratzer, S.G. On a class of computationally efficient, rapidly converging, generalized NLMS algorithms. *IEEE Signal Process. Lett.* **1996**, *3*, 245–247. [[CrossRef](#)]
35. Haykin, S. *Adaptive Filter Theory*, 4th ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2002.
36. Benesty, J.; Paleologu, C.; Ciochină, S. On regularization in adaptive filtering. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 1734–1742. [[CrossRef](#)]
37. *Digital Network Echo Cancellers*; ITU-T Recommendations G.168; 2002. Available online: <https://www.itu.int/rec/T-REC-G.168/en> (accessed on 16 April 2019).
38. Morgan, D.R.; Benesty, J.; Sondhi, M.M. On the evaluation of estimated impulse responses. *IEEE Signal Process. Lett.* **1998**, *5*, 174–176. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).