Digital Signal Processing 75 (2018) 153–167

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Adaptive filtering for the identification of bilinear forms

Constantin Paleologu [a,*], Jacob Benesty [b], Silviu Ciochină [a]

[a] University Politehnica of Bucharest, 1-3, Iuliu Maniu Blvd., 061071, Bucharest, Romania
[b] INRS-EMT, University of Quebec, Montreal, QC, H5A 1K6, Canada

ARTICLE INFO

Keywords:
Adaptive filter
Bilinear forms
Multiple-input/single-output (MISO) system
System identification
Wiener filter

ABSTRACT

Bilinear systems are involved in many interesting applications, especially related to the approximation of nonlinear systems. In this context, the bilinear term is usually defined in terms of an input–output relation (i.e., with respect to the data). Recently, a different approach has been introduced, by defining the bilinear term with respect to the impulse responses of a spatiotemporal model, which resembles a multiple-input/single-output (MISO) system. Also, in this framework, the Wiener filter has been studied to address the identification problem of these bilinear forms. Since the Wiener filter may not be always very efficient or convenient to use in practice, we propose in this paper an adaptive filtering approach. Consequently, we develop and analyze some basic algorithms tailored for the identification of bilinear forms, i.e., least-mean-square (LMS), normalized LMS (NLMS), and recursive-least-squares (RLS). Simulations performed in the context of system identification (based on the MISO system approach) support the theoretical findings and indicate the appealing performance of these algorithms.

1. Introduction

Bilinear systems have been found to be popular in a wide range of domains [1], being addressed in the literature in different ways and contexts. Most often, they are related to the approximation of nonlinear systems. It is known that a bilinear model can approximate a large class of nonlinear systems via a finite sum of the Volterra series expansion between the inputs and outputs of the system. Hence, a bilinear system can be considered among the simplest recursive nonlinear systems. On the other hand, bilinear systems behave similarly (to some extent) to linear models, which further simplify the analysis [2]. Therefore, they provide a good compromise between the accuracy of nonlinear systems and the tractability of linear systems.

Due to their practical features, the bilinear systems have been involved in many interesting applications, e.g., [3–17] and the references therein. Among these, we can mention system identification [4], [5], [14], digital filter synthesis [6], prediction problems [7], channel equalization [8], echo cancellation [9], chaotic communications [12], active noise control [13], [17], neural networks [16], etc. Many of these works were concerned with developing adaptive filtering algorithms for nonlinear systems modeled as bilinear systems. Nevertheless, in all these frameworks, the bilinear term is defined with respect to the data, i.e., in terms of an input–output relation.

In this work, we focus on a different approach by defining the bilinear term with respect to the impulse responses of a spatiotemporal model, in the context of multiple-input/single-output (MISO) systems. Recently, in [18], this problem has been addressed from a system identification perspective and two forms of the Wiener filter (namely direct and iterative) were developed in this context. However, the Wiener filter may not be always convenient to use in practice, due to some well-known limitations (e.g., matrix inversion, estimation of the statistics, etc.). Consequently, the next natural step is to analyze this framework in terms of an adaptive filtering approach, which represents the main motivation behind this paper. In this context, we focus on the most popular adaptive algorithms, i.e., least-mean-square (LMS), normalized LMS (NLMS), and recursive-least-squares (RLS).

Similar frameworks can be found in [19–24], in the context of particular applications, e.g., channel equalization [19], target detection [23], and nonlinear acoustic echo cancellation [20–22], [24]. However, most of these works were not associated or analyzed in conjunction with bilinear forms. Usually, they were referred as cascaded systems (e.g., similar to the Hammerstein model [25]) or joint adaptation processes, while the resulted algorithms were derived mostly in an ad-hoc manner, without analyzing their convergence features in a more general framework. The main goal of this paper is to perform a first step toward the development and analysis of adaptive filters tailored for the identification of bilinear forms, under the general framework of a MISO system identifica-

* Corresponding author.
E-mail addresses: pale@comm.pub.ro (C. Paleologu), benesty@emt.inrs.ca (J. Benesty), silviu@comm.pub.ro (S. Ciochină).

https://doi.org/10.1016/j.dsp.2018.01.010
1051-2004/

tion problem. The overall approach can be interpreted (to some extent) as a multidimensional adaptive filtering technique.

The framework and the algorithms proposed in this paper could be used in the context of different applications related to the identification of such bilinear forms. For example, nonlinear acoustic echo cancellation represents an appealing choice, since the general configuration related to this application can be interpreted as a particular case of our proposed model; basically, in this context, the general scheme reduces to a Hammerstein model [20–22], [24]. Also, we can link our contribution to multichannel interference cancellation, e.g., in the context of adaptive noise cancellation (where the noise signal is picked up by a microphone array [26]) or multichannel dereverberation. In addition, the particular form of the MISO system could be exploited for the identification of block-sparse systems [27]. For the sake of generality, we do not focus in this paper on one particular application, but on the capabilities and features of the proposed algorithms in the general framework of system identification.

The rest of the paper is organized as follows. In Section 2, the proposed signal model with bilinear forms is introduced. Next, the LMS algorithm tailored for bilinear forms is presented in Section 3, together with its convergence analysis. Section 4 is dedicated to the NLMS algorithm for bilinear forms; in this context, a variable step-size version is also presented, aiming to achieve a proper compromise between the main performance criteria (i.e., convergence rate versus misadjustment). In Section 5, the RLS algorithm for bilinear forms is introduced, targeting a faster convergence rate as compared to its LMS-based counterparts. Simulation results are presented in Section 6, in the context of system identification (from a MISO system perspective). Finally, Section 7 concludes this paper and outlines some perspectives for future works.

2. Signal model with bilinear forms

In the proposed approach, the bilinear term is defined with respect to the impulse responses of a spatiotemporal model, in the context of MISO systems. Consequently, the signal model is

$$\begin{aligned} d(n) &= \mathbf{h}^T \mathbf{X}(n)\mathbf{g} + w(n) \\ &= y(n) + w(n), \end{aligned} \tag{1}$$

where $d(n)$ is the zero-mean desired (or reference) signal at the discrete-time index n, $\mathbf{h}$ and $\mathbf{g}$ are the two impulse responses of the system of lengths L and M, respectively, the superscript ^{T} is the transpose operator,

$$\mathbf{X}(n) = \left[\mathbf{x}_1(n) \quad \mathbf{x}_2(n) \quad \cdots \quad \mathbf{x}_M(n) \right] \tag{2}$$

is the zero-mean multiple-input signal matrix of size $L \times M$,

$$\mathbf{x}_m(n) = \left[x_m(n) \quad x_m(n-1) \quad \cdots \quad x_m(n-L+1) \right]^T \tag{3}$$

is a vector containing the L most recent samples of the mth $(m = 1, 2, \ldots, M)$ input signal, $y(n) = \mathbf{h}^T \mathbf{X}(n)\mathbf{g}$ is the bilinear form, and $w(n)$ is the zero-mean additive noise. It is assumed that all the signals are real valued, and $\mathbf{X}(n)$ and $w(n)$ are uncorrelated. A simple block diagram of this model is illustrated in Fig. 1(a).

The two impulse responses, i.e., $\mathbf{h}$ and $\mathbf{g}$, correspond to the temporal and spatial parts of the system, respectively. It is easy to verify that for every fixed $\mathbf{h}$, $y(n)$ is a linear function of $\mathbf{g}$, and for every fixed $\mathbf{g}$, it is a linear function of $\mathbf{h}$. Therefore, $y(n)$ is bilinear in $\mathbf{h}$ and $\mathbf{g}$ [28].

It can be noticed that the reference signal from (1) can be expressed as

$$\begin{aligned} d(n) &= \sum_{m=1}^{M} g_m \mathbf{h}^T \mathbf{x}_m(n) + w(n) \\ &= \sum_{m=1}^{M} y_m(n) + w(n), \end{aligned} \tag{4}$$

which illustrates the processing line of each input signal, as shown in Fig. 1(b). On the other hand, based on the vectorization operation (i.e., conversion of a matrix into a vector [28]), the matrix $\mathbf{X}(n)$ of size $L \times M$ can be rewritten as a vector of length ML:

$$\begin{aligned} \mathrm{vec}\left[\mathbf{X}(n)\right] &= \left[\mathbf{x}_1^T(n) \quad \mathbf{x}_2^T(n) \quad \cdots \quad \mathbf{x}_M^T(n) \right]^T \\ &= \widetilde{\mathbf{x}}(n). \end{aligned} \tag{5}$$

Consequently, the output signal $y(n)$ can be expressed as

$$\begin{aligned} y(n) &= \mathbf{h}^T \mathbf{X}(n)\mathbf{g} \\ &= \mathrm{tr}\left[\left(\mathbf{h}\mathbf{g}^T \right)^T \mathbf{X}(n) \right] \\ &= \mathrm{vec}^T \left(\mathbf{h}\mathbf{g}^T \right) \mathrm{vec}\left[\mathbf{X}(n)\right] \\ &= (\mathbf{g} \otimes \mathbf{h})^T \widetilde{\mathbf{x}}(n) \\ &= \mathbf{f}^T \widetilde{\mathbf{x}}(n), \end{aligned} \tag{6}$$

where $\mathrm{tr}[\cdot]$ denotes the trace of a square matrix, $\otimes$ is the Kronecker product, and $\mathbf{f} = \mathbf{g} \otimes \mathbf{h}$ is the spatiotemporal impulse response (of length ML), which is simply the Kronecker product between the two individual impulse responses $\mathbf{g}$ and $\mathbf{h}$. Hence, the signal model in (1) results in

$$d(n) = \mathbf{f}^T \widetilde{\mathbf{x}}(n) + w(n), \tag{7}$$

which can be seen as a particular form of a MISO system, as depicted in Fig. 1(c). In the general case of a MISO system, $\mathbf{f}$ has ML different elements. On the other hand, in this bilinear context, $\mathbf{f} = \mathbf{g} \otimes \mathbf{h}$ is formed with $M + L$ different elements only even though it is of length ML.

A particular case of this system is the Hammerstein model [25], which is illustrated in Fig. 1(d). In this context, there is a single input signal, $x(n)$, which passes through a cascade of two systems, i.e., a nonlinear block and a linear system. In this case, the reference signal is similar to (1), but the mth $(m = 1, 2, \ldots, M)$ column of the input signal matrix $\mathbf{X}(n)$ has a particular form, i.e., $\left[x^m(n) \quad x^m(n-1) \quad \cdots \quad x^m(n-L+1) \right]^T$. Based on this model, the cascaded adaptive filters were developed in the context of different applications, like nonlinear acoustic echo cancellation [20–22], [24]. However, they were not associated with bilinear forms or analyzed in a more general framework, like we target in this paper.

Based on the equivalent model in (7), the variance of $d(n)$ is

$$\begin{aligned} \sigma_d^2 &= E\left[d^2(n) \right] \\ &= (\mathbf{g} \otimes \mathbf{h})^T \mathbf{R} (\mathbf{g} \otimes \mathbf{h}) + \sigma_w^2, \end{aligned} \tag{8}$$

where $E[\cdot]$ denotes mathematical expectation, $\mathbf{R} = E\left[\widetilde{\mathbf{x}}(n) \widetilde{\mathbf{x}}^T(n) \right]$ is the covariance matrix of $\widetilde{\mathbf{x}}(n)$, and $\sigma_w^2 = E\left[w^2(n) \right]$ is the variance of $w(n)$. As a result, the signal-to-noise ratio (SNR) of the MISO system is

$$\mathrm{SNR} = \frac{(\mathbf{g} \otimes \mathbf{h})^T \mathbf{R} (\mathbf{g} \otimes \mathbf{h})}{\sigma_w^2}. \tag{9}$$

The covariance matrix $\mathbf{R}$ consists of M^2 submatrices of size $L \times L$, i.e.,

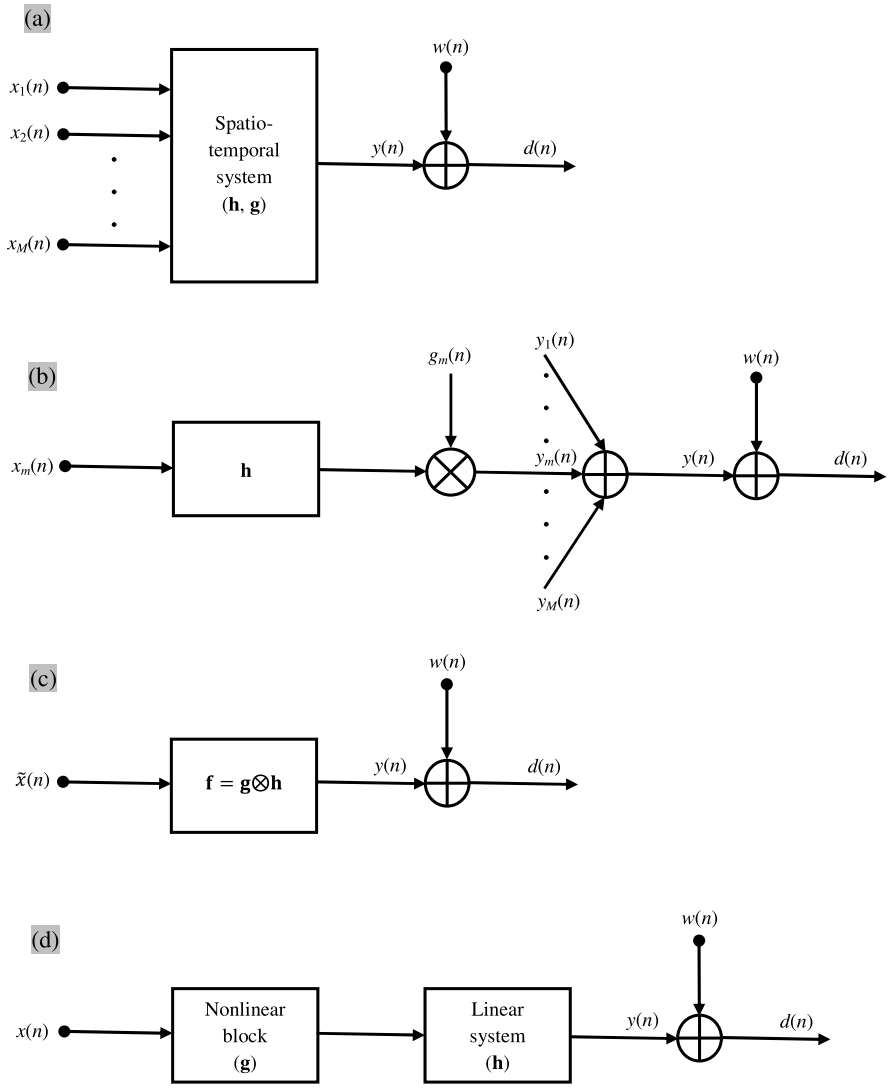

Fig. 1. Block diagrams associated to the signal model: (a) the general form of the signal model [from (1)]; (b) the processing line of the mth ($m = 1, 2, \ldots, M$) input signal [according to (4)]; (c) the equivalent form of the signal model [from (7)]; and (d) a particular case (cascaded systems): the Hammerstein model.

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1M} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{M1} & \mathbf{R}_{M2} & \cdots & \mathbf{R}_{MM} \end{bmatrix}, \tag{10}$$

where each submatrice $\mathbf{R}_{ij} = E\left[\mathbf{x}_i(n)\mathbf{x}_j^T(n)\right]$ (with $i, j = 1, 2, \ldots, M$) contains the rows from $(i-1)L+1$ to iL and the columns from $(j-1)L+1$ to jL of $\mathbf{R}$. Clearly, $\mathbf{R}_{mm}$, $m = 1, 2, \ldots, M$ are the covariance matrices of $\mathbf{x}_m(n)$, $m = 1, 2, \ldots, M$. It can be noticed that $\mathbf{R}$ is a symmetric block Toeplitz matrix. If the input signals $\mathbf{x}_m(n)$, $m = 1, 2, \ldots, M$ are independent, then $\mathbf{R}_{ij} = \mathbf{0}_{L\times L}$, for $i \neq j$, where $\mathbf{0}_{L\times L}$ denotes a matrix of size $L \times L$ with all the elements equal to zero. In this case, $\mathbf{R}$ becomes a block diagonal matrix, whose main diagonal blocks square matrices are $\mathbf{R}_{mm}$, $m = 1, 2, \ldots, M$.

In the framework of this MISO system, the main objective is to identify the temporal and spatial impulse responses $\mathbf{h}$ and $\mathbf{g}$ with two filters $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{g}}$ of lengths L and M, respectively, and the spatiotemporal impulse response $\mathbf{f} = \mathbf{g} \otimes \mathbf{h}$ with the long filter $\widehat{\mathbf{f}} = \widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}$ of length ML. Let $\eta \neq 0$ be a real-valued number. It is clear from (1) that

$$\begin{aligned} y(n) &= \mathbf{h}^T\mathbf{X}(n)\mathbf{g} \\ &= \left(\frac{1}{\eta}\mathbf{h}\right)^T \mathbf{X}(n)\left(\eta\mathbf{g}\right), \end{aligned} \tag{11}$$

so that the pair $\mathbf{h}/\eta$ and $\eta\mathbf{g}$ is equivalent to the pair $\mathbf{h}$ and $\mathbf{g}$ in the bilinear form. This implies that we can only identify $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{g}}$ up to a scaling factor. A similar discussion can be found in [19] and [24] in the context of blind identification/equalization and nonlinear acoustic echo cancellation, respectively. However, since

$$\begin{aligned} \mathbf{f} &= \mathbf{g} \otimes \mathbf{h} \\ &= (\eta\mathbf{g}) \otimes \left(\frac{1}{\eta}\mathbf{h}\right), \end{aligned} \tag{12}$$

the spatiotemporal impulse response will be identified with no scaling ambiguity. Therefore, to evaluate the identification of the temporal and spatial filters, we should use the normalized projection misalignment (NPM) [29]:

$$\mathrm{NPM}\left(\mathbf{h}, \widehat{\mathbf{h}}\right) = 1 - \left(\frac{\mathbf{h}^T\widehat{\mathbf{h}}}{\|\mathbf{h}\|\left\|\widehat{\mathbf{h}}\right\|}\right)^2, \tag{13}$$

$$\mathrm{NPM}\left(\mathbf{g},\widehat{\mathbf{g}}\right)=1-\left(\frac{\mathbf{g}^T\widehat{\mathbf{g}}}{\|\mathbf{g}\|\,\|\widehat{\mathbf{g}}\|}\right)^2, \tag{14}$$

and the identification of the spatiotemporal filter should be evaluated with the usual normalized misalignment (NM):

$$\mathrm{NM}\left(\mathbf{f},\widehat{\mathbf{f}}\right)=\frac{\left\|\mathbf{f}-\widehat{\mathbf{f}}\right\|^2}{\|\mathbf{f}\|^2}, \tag{15}$$

where $\|\cdot\|$ denotes the Euclidean norm.

3. LMS algorithm and its convergence

In [18], the previous system identification problem has been addressed based on the Wiener filter. In this context, two forms of the Wiener filter were developed, namely direct and iterative. Among them, the iterative solution is much more attractive as it leads to good estimates of the impulse responses, even when a small amount of data is used for the estimation of the statistics. However, the iterative Wiener filter presented in [18] may not be very efficient or convenient to use in practice (especially in real-time applications), due to the well-known limitations of the Wiener filter, i.e., matrix inversion operations, estimation of the statistics, etc. Consequently, we may approach the identification problem in terms of adaptive filtering. This is explored in this section, in the context of the LMS algorithm.

Let us consider two adaptive filters $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$, and the estimated signal $\widehat{y}(n)=\widehat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\widehat{\mathbf{g}}(n-1)$. As a result, we can define the error signal between the desired and estimated signals as

$$\begin{aligned}e(n)&=d(n)-\widehat{y}(n)\\&=d(n)-\widehat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\widehat{\mathbf{g}}(n-1)\\&=d(n)-\left[\widehat{\mathbf{g}}(n-1)\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n)\\&=d(n)-\widehat{\mathbf{f}}^T(n-1)\widetilde{\mathbf{x}}(n).\end{aligned} \tag{16}$$

Alternatively, we may define

$$e_{\widehat{\mathbf{g}}}(n)=d(n)-\widehat{\mathbf{h}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n), \tag{17}$$

$$e_{\widehat{\mathbf{h}}}(n)=d(n)-\widehat{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n), \tag{18}$$

with

$$\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)=\left[\widehat{\mathbf{g}}(n-1)\otimes\mathbf{I}_L\right]^T\widetilde{\mathbf{x}}(n), \tag{19}$$

$$\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)=\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n), \tag{20}$$

where $\mathbf{I}_L$ and $\mathbf{I}_M$ are the identity matrices of sizes $L\times L$ and $M\times M$, respectively. It can be verified that $e_{\widehat{\mathbf{g}}}(n)=e_{\widehat{\mathbf{h}}}(n)=e(n)$. However, for the clarity of the coming developments, we prefer to keep the notation from (17) and (18).

In this context, the LMS algorithm for bilinear forms (namely LMS-BF) is then

$$\begin{aligned}\widehat{\mathbf{h}}(n)&=\widehat{\mathbf{h}}(n-1)-\frac{\mu_{\widehat{\mathbf{h}}}}{2}\times\frac{\partial e_{\widehat{\mathbf{g}}}^2(n)}{\partial\widehat{\mathbf{h}}(n-1)}\\&=\widehat{\mathbf{h}}(n-1)+\mu_{\widehat{\mathbf{h}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n)\end{aligned} \tag{21}$$

and

$$\begin{aligned}\widehat{\mathbf{g}}(n)&=\widehat{\mathbf{g}}(n-1)-\frac{\mu_{\widehat{\mathbf{g}}}}{2}\times\frac{\partial e_{\widehat{\mathbf{h}}}^2(n)}{\partial\widehat{\mathbf{g}}(n-1)}\\&=\widehat{\mathbf{g}}(n-1)+\mu_{\widehat{\mathbf{g}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n),\end{aligned} \tag{22}$$

where $\mu_{\widehat{\mathbf{h}}}>0$ and $\mu_{\widehat{\mathbf{g}}}>0$ are the so-called step-size parameters. For the initialization, we may choose

$$\widehat{\mathbf{h}}(0)=\begin{bmatrix}1&0&\cdots&0\end{bmatrix}^T,$$

$$\widehat{\mathbf{g}}(0)=\frac{1}{M}\begin{bmatrix}1&1&\cdots&1\end{bmatrix}^T.$$

In order to analyze the convergence of the LMS-BF algorithm, let us define the a posteriori misalignments related to the temporal and spatial impulse responses:

$$\mathbf{c}_{\mathbf{h}}(n)=\frac{1}{\eta}\mathbf{h}-\widehat{\mathbf{h}}(n), \tag{23}$$

$$\mathbf{c}_{\mathbf{g}}(n)=\eta\mathbf{g}-\widehat{\mathbf{g}}(n), \tag{24}$$

where η is the arbitrary scaling factor. As mentioned in Section 2, we can only identify the impulse responses up to this scaling factor; however, the pair $\mathbf{h}/\eta$ and $\eta\mathbf{g}$ is equivalent to the pair $\mathbf{h}$ and $\mathbf{g}$ in the bilinear form.

First, let us perform the analysis in terms of the spatial filter $\mathbf{g}$. Based on (22), we can further develop (24) as

$$\begin{aligned}\mathbf{c}_{\mathbf{g}}(n)&=\eta\mathbf{g}-\widehat{\mathbf{g}}(n-1)-\mu_{\widehat{\mathbf{g}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)\\&=\mathbf{c}_{\mathbf{g}}(n-1)-\mu_{\widehat{\mathbf{g}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n).\end{aligned} \tag{25}$$

Taking the Euclidean norm in (25), then the expectation on both sides, and denoting $m_{\mathbf{g}}(n)=E\left[\left\|\mathbf{c}_{\mathbf{g}}(n)\right\|^2\right]$, we obtain

$$\begin{aligned}m_{\mathbf{g}}(n)=&m_{\mathbf{g}}(n-1)-2\mu_{\widehat{\mathbf{g}}}E\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)\right]\\&+\mu_{\widehat{\mathbf{g}}}^2E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}^2(n)\right].\end{aligned} \tag{26}$$

Based on (18) and (20), the second term from the right-hand side of (26) results in

$$\begin{aligned}&E\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)\right]\\&=E\left\{\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\left\{w(n)+\left[\eta\mathbf{g}^T\otimes\mathbf{c}_{\mathbf{h}}^T(n-1)\right]\widetilde{\mathbf{x}}(n)\right.\right.\\&\left.\left.+\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\right\}\right\}\\&=E\left\{\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\right.\\&\left.+\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\right\}\\&=p_{\mathbf{g}}(n)+E\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\right],\end{aligned} \tag{27}$$

where

$$p_{\mathbf{g}}(n)=E\left\{\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\right\}. \tag{28}$$

Furthermore, taking into account that $\mathbf{c}_{\mathbf{g}}(n-1)$ and $\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)$ are uncorrelated, we can develop (27) as follows:

$$\begin{aligned}&E\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)\right]\\&=p_{\mathbf{g}}(n)+\mathrm{tr}\left\{E\left[\mathbf{c}_{\mathbf{g}}(n-1)\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\right]\right\}\\&=p_{\mathbf{g}}(n)+\mathrm{tr}\left\{E\left[\mathbf{c}_{\mathbf{g}}(n-1)\mathbf{c}_{\mathbf{g}}^T(n-1)\right]E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\right]\right\}.\end{aligned} \tag{29}$$

Next, we have to focus on the last expectation term in (29), which can be expressed as

$$\begin{aligned}&E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\right]\\&=E\left\{\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\right\}.\end{aligned} \tag{30}$$

The terms on the main diagonal of this matrix are $E\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\right]$, $m=1,2,\ldots,M$. At this point, let us assume that the covariance matrices of the inputs are

close to a diagonal one, i.e., $\mathbf{R}_{mm} = E\left[\mathbf{x}_m(n)\mathbf{x}_m^T(n)\right] \approx \sigma_{x_m}^2 \mathbf{I}_L$, $m = 1, 2, \ldots, M$. This is a fairly restrictive assumption on the input signals; however, it has been widely used to simplify the convergence analysis of the classical LMS algorithm [30–32]. Also, let us consider that the input signals are independent and have the same power, i.e., $\sigma_{x_m}^2 \approx \sigma_x^2$, $m = 1, 2, \ldots, M$. Consequently,

$$\begin{aligned} &E\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\right] \\ &= \mathrm{tr}\left\{E\left[\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\widehat{\mathbf{h}}^T(n-1)\right]\right\} \\ &= \mathrm{tr}\left\{E\left[\mathbf{x}_m(n)\mathbf{x}_m^T(n)\right]E\left[\widehat{\mathbf{h}}(n-1)\widehat{\mathbf{h}}^T(n-1)\right]\right\} \\ &= \sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1), \end{aligned} \tag{31}$$

where we used the notation $n_{\widehat{\mathbf{h}}}(n) = E\left[\left\|\widehat{\mathbf{h}}(n)\right\|^2\right]$. Finally, using (30) and (31) in (29), we obtain

$$E\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)\right] = p_{\mathbf{g}}(n) + m_{\mathbf{g}}(n-1)\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1). \tag{32}$$

Also, under similar considerations, the first term from the right-hand side of (32) results in

$$\begin{aligned} p_{\mathbf{g}}(n) &= E\left\{\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\right\} \\ &= \mathrm{tr}\left\{E\left\{\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\mathbf{c}_{\mathbf{g}}^T(n-1)\right.\right. \\ &\quad\left.\left.\times\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\right\}\right\} \\ &= \sigma_x^2\mathrm{tr}\left\{E\left\{\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\otimes\widehat{\mathbf{h}}^T(n-1)\right]\right\}\right\} \\ &= \sigma_x^2\mathrm{tr}\left\{E\left\{\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\eta\mathbf{g}\right]\otimes\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{c}_{\mathbf{h}}(n-1)\right]\right\}\right\} \\ &= \sigma_x^2\eta\mathbf{g}^T E\left[\mathbf{c}_{\mathbf{g}}(n-1)\right]E\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{c}_{\mathbf{h}}(n-1)\right]. \end{aligned} \tag{33}$$

Second, we detail the evaluation of the third term from the right-hand side of (26), i.e., $E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}^2(n)\right]$. To begin, let us focus on the product $\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)$. Following the same considerations and assumptions related to (30) and (31), it results in

$$\begin{aligned} \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) &= \widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n) \\ &= \mathrm{tr}\left\{\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\right\} \\ &\approx M\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1). \end{aligned} \tag{34}$$

Hence, for some degree of stationarity of the input signals, we can treat the term from (34) as a deterministic quantity, so that

$$E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}^2(n)\right] \approx M\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)E\left[e_{\widehat{\mathbf{h}}}^2(n)\right]. \tag{35}$$

In order to evaluate $E\left[e_{\widehat{\mathbf{h}}}^2(n)\right]$, let us use (7), (18), (23), and (24), to rewrite the error $e_{\widehat{\mathbf{h}}}(n)$ as

$$e_{\widehat{\mathbf{h}}}(n) = w(n) + \left[\eta\mathbf{g}^T\otimes\mathbf{c}_{\mathbf{h}}^T(n-1)\right]\widetilde{\mathbf{x}}(n) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1). \tag{36}$$

Therefore, developing based on (36), removing the uncorrelated terms, and using the same considerations as before, it results that

$$\begin{aligned} &E\left[e_{\widehat{\mathbf{h}}}^2(n)\right] \\ &= \sigma_w^2 + E\left\{\left[\eta\mathbf{g}^T\otimes\mathbf{c}_{\mathbf{h}}^T(n-1)\right]\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\right\} \\ &+ E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\mathbf{c}_{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\right] \\ &+ 2E\left\{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\widetilde{\mathbf{x}}^T(n)\left[\eta\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)\right]\right\} \\ &= \sigma_w^2 + \sigma_x^2\eta^2\left\|\mathbf{g}\right\|^2 m_{\mathbf{h}}(n-1) + \sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)m_{\mathbf{g}}(n-1) + 2p_{\mathbf{g}}(n), \end{aligned} \tag{37}$$

where we used the notation $m_{\mathbf{h}}(n) = E\left[\left\|\mathbf{c}_{\mathbf{h}}(n)\right\|^2\right]$. Finally, based on (35) and (37), we obtain

$$\begin{aligned} &E\left[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}^2(n)\right] \\ &= M\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)\left[\sigma_w^2 + \sigma_x^2\eta^2\left\|\mathbf{g}\right\|^2 m_{\mathbf{h}}(n-1)\right. \\ &\left.+ \sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)m_{\mathbf{g}}(n-1) + 2p_{\mathbf{g}}(n)\right]. \end{aligned} \tag{38}$$

Next, introducing (32) and (38) in (26), then developing and grouping the terms, it results in

$$m_{\mathbf{g}}(n) = A_{\mathbf{g}}(n)m_{\mathbf{g}}(n-1) + B_{\mathbf{g}}(n), \tag{39}$$

where

$$A_{\mathbf{g}}(n) = 1 - 2\mu_{\widehat{\mathbf{g}}}\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1) + \mu_{\widehat{\mathbf{g}}}^2 M\sigma_x^4 n_{\widehat{\mathbf{h}}}^2(n-1), \tag{40}$$

$$\begin{aligned} B_{\mathbf{g}}(n) &= -2\mu_{\widehat{\mathbf{g}}}p_{\mathbf{g}}(n) + \mu_{\widehat{\mathbf{g}}}^2 M\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1) \\ &\quad\times\left[\sigma_w^2 + \sigma_x^2\eta^2\left\|\mathbf{g}\right\|^2 m_{\mathbf{h}}(n-1) + 2p_{\mathbf{g}}(n)\right]. \end{aligned} \tag{41}$$

The term from (40) is related to the convergence rate and stability. As expected, it depends on the step-size value, the filter length, and the input signals power. It is interesting to notice that it is also related to the spatial filter estimate (in terms of its norm). On the other hand, it does not depend on the scaling factor or the system noise power. Also, it can be noticed that

$$A_{\mathbf{g}}(n) = \left[1 - \mu_{\widehat{\mathbf{g}}}\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)\right]^2 + \mu_{\widehat{\mathbf{g}}}^2(M-1)\sigma_x^4 n_{\widehat{\mathbf{h}}}^2(n-1) > 0. \tag{42}$$

Consequently, the stability condition is $A_{\mathbf{g}}(n) < 1$, which leads to

$$\mu_{\widehat{\mathbf{g}}} < \frac{2}{M\sigma_x^2 n_{\widehat{\mathbf{h}}}(n-1)}. \tag{43}$$

The term from (41) is related to the misadjustment. As expected, it depends on the step-size value, the filter length, the input signals power, and the system noise power. But it is also related to the term $p_{\mathbf{g}}(n)$ from (33) and the scaling factor, which are difficult to evaluate or estimate.

Similarly, following the convergence analysis of the LMS-BF algorithm in terms of the temporal filter $\mathbf{h}$ (under similar circumstances as in the case of the spatial filter $\mathbf{g}$), we obtain

$$m_{\mathbf{h}}(n) = A_{\mathbf{h}}(n)m_{\mathbf{h}}(n-1) + B_{\mathbf{h}}(n), \tag{44}$$

with

$$A_{\mathbf{h}}(n) = 1 - 2\mu_{\widehat{\mathbf{h}}}\sigma_x^2 n_{\widehat{\mathbf{g}}}(n-1) + \mu_{\widehat{\mathbf{h}}}^2 L\sigma_x^4 n_{\widehat{\mathbf{g}}}^2(n-1), \tag{45}$$

$$\begin{aligned} B_{\mathbf{h}}(n) &= -2\mu_{\widehat{\mathbf{h}}}p_{\mathbf{h}}(n) + \mu_{\widehat{\mathbf{h}}}^2 L\sigma_x^2 n_{\widehat{\mathbf{g}}}(n-1) \\ &\quad\times\left[\sigma_w^2 + \sigma_x^2\frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2 m_{\mathbf{g}}(n-1) + 2p_{\mathbf{h}}(n)\right], \end{aligned} \tag{46}$$

where we used the notation $n_{\widehat{\mathbf{g}}}(n) = E\left[\left\|\widehat{\mathbf{g}}(n)\right\|^2\right]$ and

$$\begin{aligned} p_{\mathbf{h}}(n) &= E\left\{\mathbf{c}_{\mathbf{h}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}^T(n)\left[\mathbf{c}_{\mathbf{g}}(n-1)\otimes\frac{1}{\eta}\mathbf{h}\right]\right\} \\ &= \sigma_x^2\frac{1}{\eta}\mathbf{h}^T E\left[\mathbf{c}_{\mathbf{h}}(n-1)\right]E\left[\widehat{\mathbf{g}}^T(n-1)\mathbf{c}_{\mathbf{g}}(n-1)\right]. \end{aligned} \tag{47}$$

Also, similar to (34), we can show that

$$\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^{T}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) \approx L\sigma_x^2 n_{\widehat{\mathbf{g}}}(n-1). \tag{48}$$

The stability condition is $A_{\mathbf{h}}(n) < 1$, which leads to

$$\mu_{\widehat{\mathbf{h}}} < \frac{2}{L\sigma_x^2 n_{\widehat{\mathbf{g}}}(n-1)}. \tag{49}$$

Denoting $\lim_{t\to\infty} m_{\mathbf{g}}(n) = m_{\mathbf{g}}(\infty)$ and $\lim_{t\to\infty} m_{\mathbf{h}}(n) = m_{\mathbf{h}}(\infty)$, taking into account that $\lim_{t\to\infty} p_{\mathbf{g}}(n) = 0$ and $\lim_{t\to\infty} p_{\mathbf{h}}(n) = 0$ [based on (33) and (47)], using

$$\lim_{t\to\infty} n_{\widehat{\mathbf{h}}}(n-1) = \frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2, \tag{50}$$

$$\lim_{t\to\infty} n_{\widehat{\mathbf{g}}}(n-1) = \eta^2\left\|\mathbf{g}\right\|^2, \tag{51}$$

and developing in (39) and (44), respectively, it results that

$$m_{\mathbf{h}}(\infty) = \frac{\mu_{\widehat{\mathbf{h}}} L\left[\sigma_w^2 + \sigma_x^2\frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2 m_{\mathbf{g}}(\infty)\right]}{2-\mu_{\widehat{\mathbf{h}}} L\sigma_x^2\eta^2\left\|\mathbf{g}\right\|^2}, \tag{52}$$

$$m_{\mathbf{g}}(\infty) = \frac{\mu_{\widehat{\mathbf{g}}} M\left[\sigma_w^2 + \sigma_x^2\eta^2\left\|\mathbf{g}\right\|^2 m_{\mathbf{h}}(\infty)\right]}{2-\mu_{\widehat{\mathbf{g}}} M\sigma_x^2\frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2}. \tag{53}$$

Finally, rearranging and grouping the terms in (52) and (53), we obtain

$$m_{\mathbf{h}}(\infty) = \frac{\mu_{\widehat{\mathbf{h}}} L\sigma_w^2}{\Delta}, \tag{54}$$

$$m_{\mathbf{g}}(\infty) = \frac{\mu_{\widehat{\mathbf{g}}} M\sigma_w^2}{\Delta}, \tag{55}$$

where

$$\Delta = 2 - \sigma_x^2\left(\mu_{\widehat{\mathbf{h}}} L\eta^2\left\|\mathbf{g}\right\|^2 + \mu_{\widehat{\mathbf{g}}} M\frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2\right). \tag{56}$$

It can be noticed that in the absence of noise (i.e., $\sigma_w^2 = 0$) we have $m_{\mathbf{h}}(\infty) = 0$ and $m_{\mathbf{g}}(\infty) = 0$. Clearly, the term from (56) should be positive, which leads to

$$\mu_{\widehat{\mathbf{h}}} L\eta^2\left\|\mathbf{g}\right\|^2 + \mu_{\widehat{\mathbf{g}}} M\frac{1}{\eta^2}\left\|\mathbf{h}\right\|^2 < \frac{2}{\sigma_x^2}. \tag{57}$$

This "connection" between $\mu_{\widehat{\mathbf{g}}}$ and $\mu_{\widehat{\mathbf{h}}}$ limits the theoretical range of these parameters [from (43) and (49), respectively]. Consequently, conditions (43), (49), and (57) should be considered together when setting the step-sizes of the LMS-BF algorithm.

In order to interpret the results in terms of the NPMs [as defined in (13) and (14)], let us consider the normalized versions of (23) and (24) in terms of their square norms, i.e.,

$$\frac{\left\|\mathbf{c}_{\mathbf{h}}(n)\right\|^2}{\left\|\frac{1}{\eta}\mathbf{h}\right\|^2} = \frac{\left\|\frac{1}{\eta}\mathbf{h} - \widehat{\mathbf{h}}(n)\right\|^2}{\left\|\frac{1}{\eta}\mathbf{h}\right\|^2}, \tag{58}$$

$$\frac{\left\|\mathbf{c}_{\mathbf{g}}(n)\right\|^2}{\left\|\eta\mathbf{g}\right\|^2} = \frac{\left\|\eta\mathbf{g} - \widehat{\mathbf{g}}(n)\right\|^2}{\left\|\eta\mathbf{g}\right\|^2}. \tag{59}$$

According to [29], these measures are equivalent to the NPMs from (13) and (14), which "absorb" the scaling factor. In this context, the NPMs should converge to the normalized limits:

$$\underline{m}_{\mathbf{h}}(\infty) = \frac{m_{\mathbf{h}}(\infty)}{\left\|\frac{1}{\eta}\mathbf{h}\right\|^2}, \tag{60}$$

$$\underline{m}_{\mathbf{g}}(\infty) = \frac{m_{\mathbf{g}}(\infty)}{\left\|\eta\mathbf{g}\right\|^2}, \tag{61}$$

as will be supported in the experiments from Section 6.

4. NLMS-type algorithms

The results from (43) and (49) suggest a time-dependent form of the step-size parameters. Consequently, a first natural choice is to reformulate the approach in terms of the normalized NLMS algorithm, which may be more advantageous to use when we deal with nonstationary signals. In this context, the update relations become

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n), \tag{62}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n), \tag{63}$$

where $\mu_{\widehat{\mathbf{h}}}(n) > 0$ and $\mu_{\widehat{\mathbf{g}}}(n) > 0$ are the time-dependent step-size parameters.

A reasonable way to derive the step-size parameters, taking into account the stability conditions, is to cancel the a posteriori error signals [33], [34]. In our framework, let us define

$$\varepsilon_{\widehat{\mathbf{g}}}(n) = d(n) - \widehat{\mathbf{h}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n), \tag{64}$$

$$\varepsilon_{\widehat{\mathbf{h}}}(n) = d(n) - \widehat{\mathbf{g}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n). \tag{65}$$

Replacing (62) in (64), and (63) in (65), with the requirements $\varepsilon_{\widehat{\mathbf{g}}}(n) = 0$ and $\varepsilon_{\widehat{\mathbf{h}}}(n) = 0$, respectively, we obtain

$$e_{\widehat{\mathbf{g}}}(n)\left[1 - \mu_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)\right] = 0, \tag{66}$$

$$e_{\widehat{\mathbf{h}}}(n)\left[1 - \mu_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\right] = 0. \tag{67}$$

Next, assuming that the a priori error is not equal to zero, we find $\mu_{\widehat{\mathbf{h}}}(n) = 1/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)]$ and $\mu_{\widehat{\mathbf{g}}}(n) = 1/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)]$, which represent the theoretical step-sizes of the NLMS algorithm for bilinear forms, namely NLMS-BF.

In practice, two positive constants $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$ (usually smaller than 1), multiply these step-sizes to achieve a proper compromise between the convergence rate and the misadjustment [30]; these constants are known as the normalized step-sizes. Also, the NLMS-BF algorithm needs to be regularized by adding two positive constants, $\delta_{\widehat{\mathbf{h}}}$ and $\delta_{\widehat{\mathbf{g}}}$, to the denominators of the step-size parameters. In general, these regularization parameters are chosen proportional to the input signal variance [35]. Therefore, the NLMS-BF algorithm is defined by the updates:

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\alpha_{\widehat{\mathbf{h}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) + \delta_{\widehat{\mathbf{h}}}}, \tag{68}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \frac{\alpha_{\widehat{\mathbf{g}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) + \delta_{\widehat{\mathbf{g}}}}, \tag{69}$$

while the spatiotemporal filter can be obtained based on the Kronecker product, i.e., $\widehat{\mathbf{f}}(n) = \widehat{\mathbf{g}}(n) \otimes \widehat{\mathbf{h}}(n)$.

Based on the convergence analysis of the LMS-BF algorithm from Section 3, let us focus on (54)–(56), where we aim to replace $\mu_{\widehat{\mathbf{h}}}$ and $\mu_{\widehat{\mathbf{g}}}$ by $\alpha_{\widehat{\mathbf{h}}}/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)]$ and $\alpha_{\widehat{\mathbf{g}}}/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)]$, respectively (also neglecting the regularization parameters). To this purpose, taking (34) and (48) into account, considering the limits from (50) and (51), and knowing that $\left\|\mathbf{f}\right\|^2 = \left\|\mathbf{g}\otimes\mathbf{h}\right\|^2 = \left\|\mathbf{g}\right\|^2\left\|\mathbf{h}\right\|^2$, the normalized limits from (60) and (61) become in the case of the NLMS-BF algorithm:

$$\underline{m}'_{\mathbf{h}}(\infty) = \frac{\alpha_{\widehat{\mathbf{h}}}\sigma_w^2}{\sigma_x^2\left\|\mathbf{f}\right\|^2\left[2 - (\alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}})\right]}, \tag{70}$$

$$\underline{m}'_{\mathbf{g}}(\infty) = \frac{\alpha_{\widehat{\mathbf{g}}}\sigma_w^2}{\sigma_x^2\left\|\mathbf{f}\right\|^2\left[2 - (\alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}})\right]}. \tag{71}$$

The sum of these terms, i.e., $\underline{m}'_{\widehat{\mathbf{h}}}(\infty)+\underline{m}'_{\widehat{\mathbf{g}}}(\infty)$, can be interpreted as a measure of the global misalignment. Clearly, it is proportional to the term $(\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}})/\left[2-(\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}})\right]$. Since the two filters work in "tandem" (i.e., the behavior of one of them depends on the other), it is recommended to control the global misalignment by imposing the constraint

$$\frac{\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}}}{2-(\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}})}\leq 1, \tag{72}$$

which leads to $\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}}\leq 1$. As we can notice, for $\alpha_{\widehat{\mathbf{h}}}=\alpha_{\widehat{\mathbf{g}}}$, the limits from (70) and (71) are equal, i.e., the two filters reach the same misalignment level. Also, for different choices of $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$, such that $\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}}$ is a constant, the misalignment level of the global filter $\widehat{\mathbf{f}}(n)$ should be the same.

Alternatively, we may use the regular NLMS algorithm to identify the spatiotemporal impulse response, $\mathbf{f}$, i.e.,

$$\widehat{\mathbf{f}}(n)=\widehat{\mathbf{f}}(n-1)+\frac{\alpha\widetilde{\mathbf{x}}(n)e(n)}{\widetilde{\mathbf{x}}^T(n)\widetilde{\mathbf{x}}(n)+\delta}, \tag{73}$$

where α and δ denote the normalized step-size and the regularization parameter, respectively, while the error signal $e(n)$ is defined in (16). Then, the components $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$ can be obtained similarly to the case of the direct Wiener filter, as explained in [18].

In case of the spatiotemporal impulse response $\mathbf{f}$, we can define the a posteriori misalignment as

$$\mathbf{c}_{\mathbf{f}}(n)=\mathbf{f}-\widehat{\mathbf{f}}(n) \tag{74}$$

and introduce the notation $m_{\mathbf{f}}(n)=E\left[\|\mathbf{c}_{\mathbf{f}}(n)\|^2\right]$. Following a similar convergence analysis in case of the regular NLMS algorithm (which is not detailed here for the sake of brevity), we obtain

$$m_{\mathbf{f}}(\infty)=\frac{\alpha\sigma_w^2}{\sigma_x^2\,(2-\alpha)} \tag{75}$$

and the normalized limit $\underline{m}'_{\mathbf{f}}(\infty)=m_{\mathbf{f}}(\infty)/\|\mathbf{f}\|^2$, which is proportional to the term $\alpha/(2-\alpha)$. Taking (70) and (71) into account, if we choose $\alpha=\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}}$, we obtain $\underline{m}'_{\mathbf{f}}(\infty)=\underline{m}'_{\mathbf{h}}(\infty)+\underline{m}'_{\mathbf{g}}(\infty)$. In other words, the normalized misalignments of the regular NLMS and NLMS-BF algorithms converge to the same limit (when identifying the spatiotemporal impulse response $\mathbf{f}$). Nevertheless, we should note that the solution based on the regular NLMS algorithm [from (73)] involves an adaptive filter of length ML, while the NLMS-BF algorithm [defined by the updates (68) and (69)] uses two shorter filters of lengths L and M, respectively. Consequently, a faster converge rate is expected for the NLMS-BF algorithm as compared to the conventional approach (as will be also supported by the experiments provided in Section 6). Of course, a similar conclusion applies in case of the LMS-BF algorithm from Section 3 as compared to its regular counterpart. However, the NLMS-based algorithms are more reliable in practice, in terms of selecting the step-size parameters.

In order to achieve a proper compromise between the convergence rate and misadjustment of the NLMS-BF algorithm, we can also develop a variable step-size (VSS) version. For example, we can follow the line of the nonparametric VSS-NLMS (NPVSS-NLMS) algorithm from [36], which is one of the most popular (and practical) VSS algorithms.

In this case, the conditions to be imposed are $E\left[\varepsilon_{\widehat{\mathbf{g}}}^2(n)\right]=\sigma_w^2$ and $E\left[\varepsilon_{\widehat{\mathbf{h}}}^2(n)\right]=\sigma_w^2$. The basic idea is to recover the system noise from the error of the adaptive filter after this one converges to the true solution, which is the main goal in system identification problems. Developing (64) and (65) with the previous conditions and knowing that the step-sizes are deterministic in nature, we obtain two quadratic equations for which the obvious solutions are

$$\overline{\mu}_{\widehat{\mathbf{h}}}(n)=\frac{1}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)}\left[1-\frac{\sigma_w}{\sigma_e(n)}\right], \tag{76}$$

$$\overline{\mu}_{\widehat{\mathbf{g}}}(n)=\frac{1}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)}\left[1-\frac{\sigma_w}{\sigma_e(n)}\right], \tag{77}$$

where $\sigma_e^2(n)=E\left[e^2(n)\right]$ is the power of the a priori error signal [as shown in the beginning of Section 3, $e_{\widehat{\mathbf{g}}}(n)=e_{\widehat{\mathbf{h}}}(n)=e(n)$, but we keep the notation for convenience].

As we can notice from (76) and (77), the variable normalized step-sizes are equal, i.e., $\alpha_{\widehat{\mathbf{h}}}(n)=\alpha_{\widehat{\mathbf{g}}}(n)=\alpha(n)$. On the other hand, since $0\leq 1-\sigma_w/\sigma_e(n)\leq 1$ and taking into account the condition that results from (72), we should use

$$\alpha(n)=\frac{1}{2}\left[1-\frac{\sigma_w}{\sigma_e(n)}\right], \tag{78}$$

so that the proposed VSS-NLMS-BF algorithm is defined by the updates:

$$\widehat{\mathbf{h}}(n)=\widehat{\mathbf{h}}(n-1)+\frac{\alpha(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)+\delta_{\widehat{\mathbf{h}}}}, \tag{79}$$

$$\widehat{\mathbf{g}}(n)=\widehat{\mathbf{g}}(n-1)+\frac{\alpha(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)+\delta_{\widehat{\mathbf{g}}}}. \tag{80}$$

Before the algorithm converges or when there is an abrupt change of the system, $\sigma_e(n)$ is large as compared to σ_w, so that $\alpha(n)$ goes to its upper bound, thus providing fast convergence and tracking. Moreover, in the steady-state, $\sigma_e(n)\approx\sigma_w$ and $\alpha(n)$ goes to its minimum, leading to low misadjustment.

In practice, the power of the a priori error signal can be recursively estimated as [36]

$$\widehat{\sigma}_e^2(n)=\beta\widehat{\sigma}_e^2(n-1)+(1-\beta)e^2(n), \tag{81}$$

with $0\ll\beta<1$. Also, the power of the system noise can be estimated in different ways, e.g., [37], [38]; in this context, we should note that different other estimators can be used for the noise power, but the analysis of their influence on the algorithms' performance is beyond the scope of this paper. Finally, since power estimates are used in the evaluation of $\alpha(n)$, it is recommended to take the absolute in (78) to avoid any biased behavior (i.e., some negative values that might result). Also, a very small positive constant should be added to the denominator in (78), in order to avoid division by zero.

5. RLS algorithm

Another important class of adaptive filters is obtained from the least-squares (LS) error criterion [30]. In this case, the cost functions are

$$J_{\widehat{\mathbf{g}}}\left[\widehat{\mathbf{h}}(n)\right]=\sum_{i=1}^{n}\lambda_{\widehat{\mathbf{h}}}^{n-i}\left[d(i)-\widehat{\mathbf{h}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(i)\right]^2, \tag{82}$$

$$J_{\widehat{\mathbf{h}}}\left[\widehat{\mathbf{g}}(n)\right]=\sum_{i=1}^{n}\lambda_{\widehat{\mathbf{g}}}^{n-i}\left[d(i)-\widehat{\mathbf{g}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(i)\right]^2, \tag{83}$$

where $\lambda_{\widehat{\mathbf{h}}}$ $(0\ll\lambda_{\widehat{\mathbf{h}}}<1)$ and $\lambda_{\widehat{\mathbf{g}}}$ $(0\ll\lambda_{\widehat{\mathbf{g}}}<1)$ are the forgetting factors and $\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(i)=[\widehat{\mathbf{g}}(i-1)\otimes\mathbf{I}_L]^T\widetilde{\mathbf{x}}(i)$ and $\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(i)=[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(i-1)]^T\widetilde{\mathbf{x}}(i)$. The minimization of $J_{\widehat{\mathbf{g}}}\left[\widehat{\mathbf{h}}(n)\right]$ and $J_{\widehat{\mathbf{h}}}\left[\widehat{\mathbf{g}}(n)\right]$ with respect to $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$, respectively, lead to the well-known normal equations [30]:

$$\mathbf{R}_{\widehat{\mathbf{g}}}(n)\widehat{\mathbf{h}}(n)=\mathbf{p}_{\widehat{\mathbf{g}}}(n), \tag{84}$$

$$\mathbf{R}_{\widehat{\mathbf{h}}}(n)\widehat{\mathbf{g}}(n)=\mathbf{p}_{\widehat{\mathbf{h}}}(n), \tag{85}$$

where

$$\mathbf{R}_{\widehat{\mathbf{g}}}(n) = \sum_{i=1}^{n} \lambda_{\widehat{\mathbf{h}}}^{n-i} \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(i) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^{T}(i) = \lambda_{\widehat{\mathbf{h}}} \mathbf{R}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^{T}(n), \tag{86}$$

$$\mathbf{p}_{\widehat{\mathbf{g}}}(n) = \sum_{i=1}^{n} \lambda_{\widehat{\mathbf{h}}}^{n-i} \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(i) d(i) = \lambda_{\widehat{\mathbf{h}}} \mathbf{p}_{\widehat{\mathbf{g}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) d(n), \tag{87}$$

$$\mathbf{R}_{\widehat{\mathbf{h}}}(n) = \sum_{i=1}^{n} \lambda_{\widehat{\mathbf{g}}}^{n-i} \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(i) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^{T}(i) = \lambda_{\widehat{\mathbf{g}}} \mathbf{R}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^{T}(n), \tag{88}$$

$$\mathbf{p}_{\widehat{\mathbf{h}}}(n) = \sum_{i=1}^{n} \lambda_{\widehat{\mathbf{g}}}^{n-i} \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(i) d(i) = \lambda_{\widehat{\mathbf{g}}} \mathbf{p}_{\widehat{\mathbf{h}}}(n-1) + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) d(n). \tag{89}$$

From the previous equations, it is straightforward to derive the RLS algorithm for bilinear forms, namely RLS-BF. This algorithm is defined by the updates:

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) e_{\widehat{\mathbf{g}}}(n) = \widehat{\mathbf{h}}(n-1) + \mathbf{k}_{\widehat{\mathbf{g}}}(n) e_{\widehat{\mathbf{g}}}(n), \tag{90}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) e_{\widehat{\mathbf{h}}}(n) = \widehat{\mathbf{g}}(n-1) + \mathbf{k}_{\widehat{\mathbf{h}}}(n) e_{\widehat{\mathbf{h}}}(n), \tag{91}$$

where $e_{\widehat{\mathbf{g}}}(n)$ and $e_{\widehat{\mathbf{h}}}(n)$ are defined in (17) and (18), and $\mathbf{k}_{\widehat{\mathbf{g}}}(n) = \mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)$ and $\mathbf{k}_{\widehat{\mathbf{h}}}(n) = \mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)$ can be computed as follows:

$$\mathbf{k}_{\widehat{\mathbf{g}}}(n) = \frac{\mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n-1) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)}{\lambda_{\widehat{\mathbf{h}}} + \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^{T}(n) \mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n-1) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)}, \tag{92}$$

$$\mathbf{k}_{\widehat{\mathbf{h}}}(n) = \frac{\mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n-1) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)}{\lambda_{\widehat{\mathbf{g}}} + \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^{T}(n) \mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n-1) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)}. \tag{93}$$

Also, based on the matrix inversion lemma [30], the updates of $\mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n)$ and $\mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n)$ result in

$$\mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n) = \lambda_{\widehat{\mathbf{h}}}^{-1} \mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n-1) - \lambda_{\widehat{\mathbf{h}}}^{-1} \mathbf{k}_{\widehat{\mathbf{g}}}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^{T}(n) \mathbf{R}_{\widehat{\mathbf{g}}}^{-1}(n-1), \tag{94}$$

$$\mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n) = \lambda_{\widehat{\mathbf{g}}}^{-1} \mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n-1) - \lambda_{\widehat{\mathbf{g}}}^{-1} \mathbf{k}_{\widehat{\mathbf{h}}}(n) \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^{T}(n) \mathbf{R}_{\widehat{\mathbf{h}}}^{-1}(n-1). \tag{95}$$

A convergence analysis of the RLS-BF algorithm, though highly interesting, is beyond the scope of this paper. The RLS-BF algorithm should outperform the LMS-based algorithms, especially in terms of convergence rate (as will be supported by the experiments provided at the end of Section 6). The price to pay for this advantage is a huge increase in the computational complexity.

6. Simulation results

Simulations are performed from a system identification perspective, in the framework of the MISO system described in Section 2. The temporal impulse response, $\mathbf{h}$, is randomly generated (with Gaussian distribution) and its length is set to $L = 64$. The coefficients of the spatial impulse response, $\mathbf{g}$, are also randomly generated and its length is set to $M = 8$ in most of the simulations (except the one reported in Fig. 20, where $M = 4$). The input signals $x_m(n)$, $m = 1, 2, \ldots, M$ are either white Gaussian noises, AR(1) processes [each one of them is generated by filtering a white Gaussian noise through a first-order system $1/\left(1 - 0.8z^{-1}\right)$], or speech sequences. The additive noise, $w(n)$, is white and Gaussian; its variance is set to $\sigma_w^2 = 0.01$ in most of the experiments (except those reported in Figs. 3 and 5). The performance measures are either the NPM (in dB) [evaluated based on (13) and (14)], or the NM (in dB) [which is evaluated based on (15)]. As discussed at the end of Section 2, the first performance measure is involved when evaluating the identification of the individual impulse responses, $\mathbf{h}$ and $\mathbf{g}$, while the second one is used to evaluate the identification of the global impulse response, $\mathbf{f}$.

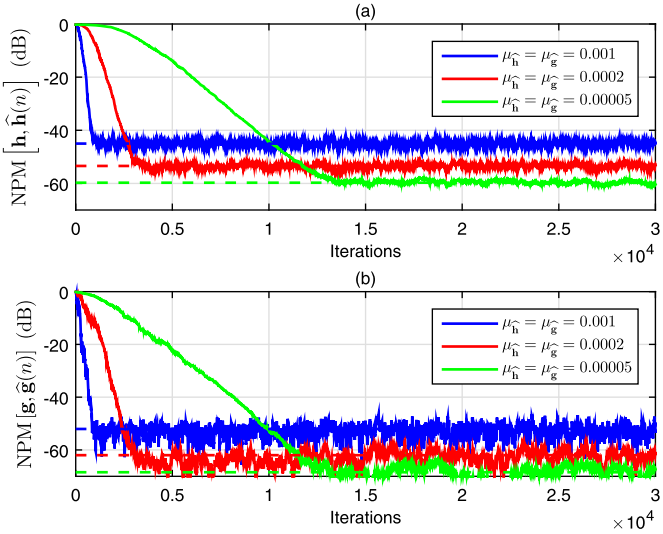

Fig. 2. Performance of the LMS-BF algorithm in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the step-size parameters. The normalized limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are white Gaussian noises, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

In the first set of simulations, the theoretical findings from Section 3 are validated in case of the LMS-BF algorithm. In Fig. 2, the performance measures NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ are depicted for different values of the step-size parameters. These are compared to the theoretical limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$ from (60) and (61), respectively. The scaling factor η is estimated by running the algorithm and evaluating (at the end) the term $\frac{1}{M} \sum_{i=1}^{M} \widehat{g}_i(n)/g_i$; alternatively, the factor $1/\eta$ can be estimated based on the term $\frac{1}{L} \sum_{i=1}^{L} \widehat{h}_i(n)/h_i$. From this figure, it can be noticed that the NPMs converge to their theoretical limits.

In Fig. 3, the performance measures NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$, together with the theoretical limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$, are depicted for different values of the system noise power, using $\mu_{\widehat{\mathbf{h}}} = \mu_{\widehat{\mathbf{g}}} = 0.0002$. First, we can observe that the NPMs converge to their theoretical limits. Also, it can be noticed that the system noise does not influence the initial convergence rate, as outlined in the discussion related to (40).

The previous experiments are repeated in Figs. 4 and 5, but using AR(1) processes as input signals. As we can see, the same conclusions apply in this case, i.e., the NPMs converge to their theoretical limits (as shown in Fig. 4) and the initial convergence rate of the algorithm is not influenced by the system noise power (according to Fig. 5). Also, as expected, the convergence rate of the LMS-BF algorithm is slower as compared to the previous case reported in Figs. 2 and 3 (where the input signals are white Gaussian noises).

In Fig. 6, the performance of the LMS-BF algorithm (using different values of the step-size parameters) is evaluated in terms of the NM, for the identification of the global impulse response, $\mathbf{f}$, of length $ML = 8 \times 64 = 512$; the input signals are AR(1) processes. It is clear that larger values of the step-sizes improve the con-

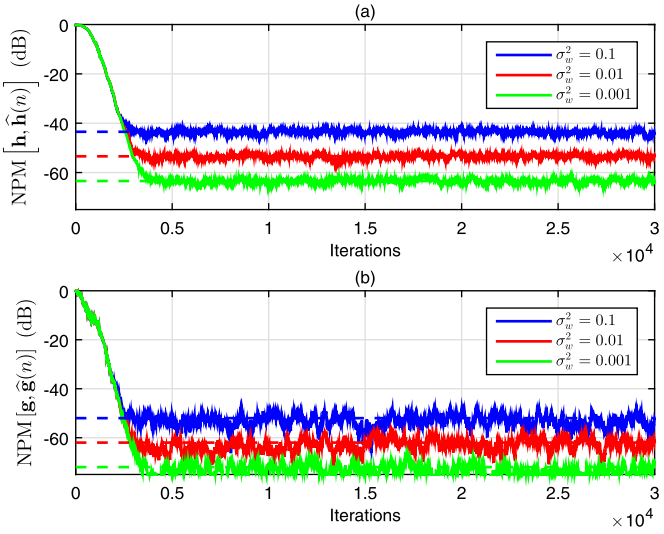

Fig. 3. Performance of the LMS-BF algorithm in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the system noise power. The normalized limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are white Gaussian noises, $\mu_{\widehat{\mathbf{h}}} = \mu_{\widehat{\mathbf{g}}} = 0.0002$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

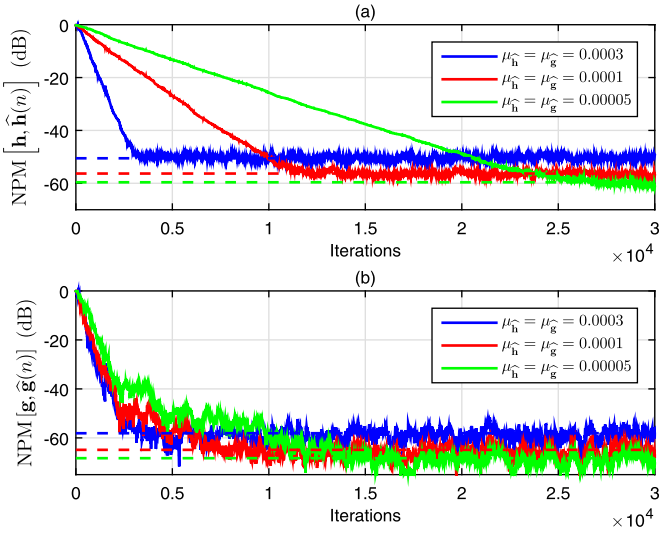

Fig. 4. Performance of the LMS-BF algorithm in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the step-size parameters. The normalized limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

vergence rate, but increase the misalignment. In this context, we should note that the regular LMS algorithm can also be used to identify the spatiotemporal impulse response. This algorithm easily results based on desired signal expressed as in (7) and the error signal from the last line of (16); consequently, its update is given by

$$\widehat{\mathbf{f}}(n) = \widehat{\mathbf{f}}(n-1) + \mu \widetilde{\mathbf{x}}(n) e(n), \tag{96}$$

where μ denotes the step-size parameter of the regular LMS algorithm. In Fig. 7, the performances of the regular LMS and LMS-BF algorithms are evaluated in terms of identifying the spatiotemporal (i.e., global) impulse response, $\mathbf{f}$; the input signals are AR(1) processes. Two different values of the step-size μ are used in the case of the regular LMS algorithm. As we can notice in Fig. 7, the

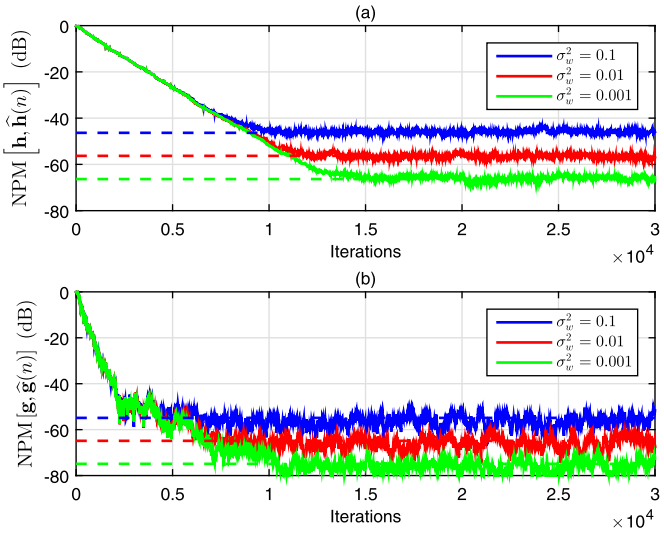

Fig. 5. Performance of the LMS-BF algorithm in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the system noise power. The normalized limits $\underline{m}_{\mathbf{h}}(\infty)$ and $\underline{m}_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are AR(1) processes, $\mu_{\widehat{\mathbf{h}}} = \mu_{\widehat{\mathbf{g}}} = 0.0001$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

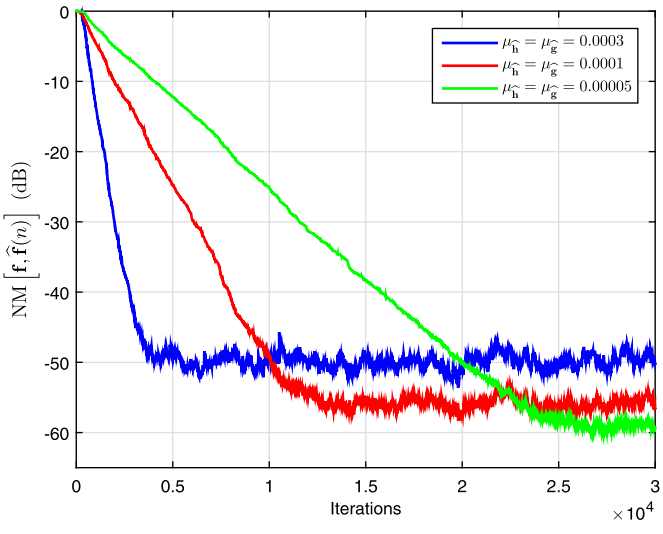

Fig. 6. Performance of the LMS-BF algorithm in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$ for different values of the step-size parameters. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

LMS-BF algorithm using $\mu_{\widehat{\mathbf{h}}} = \mu_{\widehat{\mathbf{g}}} = 0.0001$ converges to the same misalignment level as the regular LMS algorithm using $\mu = 0.0002$, but achieves a much faster convergence rate. The reason is that the solution based on the regular LMS algorithm involves an adaptive filter of length ML, while the proposed LMS-BF algorithm uses two shorter filters of lengths L and M, respectively. Of course, the convergence rate of the regular LMS algorithm can be improved by using a larger value of the step-size parameter (e.g., $\mu = 0.001$ in Fig. 7), but paying with an increase of the misalignment level. In this case, the LMS-BF algorithm outperforms the regular LMS algorithm in terms of both convergence rate and misalignment level.

In the next set of experiments, the performance of the NLMS-BF algorithm (presented in Section 4) is evaluated. In order to validate the results from (70) and (71), the experiments reported in Figs. 2 and 4 are repeated here in case of the NLMS-BF algorithm, using different values of the normalized step-size parameters, $\alpha_{\widehat{\mathbf{h}}}$

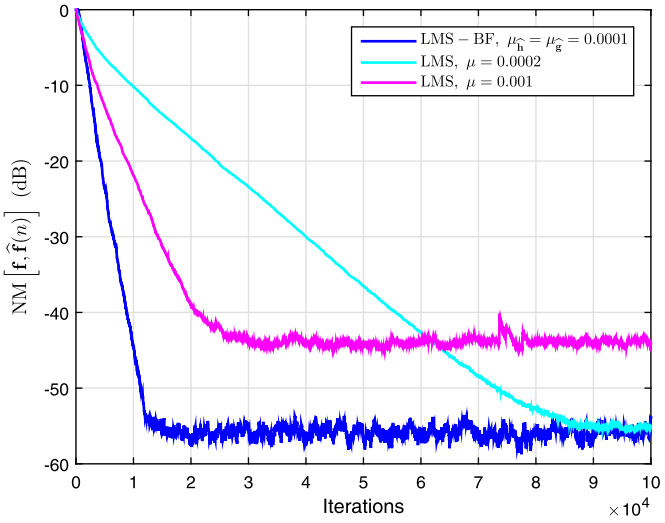

Fig. 7. Performance of the LMS and LMS-BF algorithms in terms of NM$\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$ for different values of the step-size parameters. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

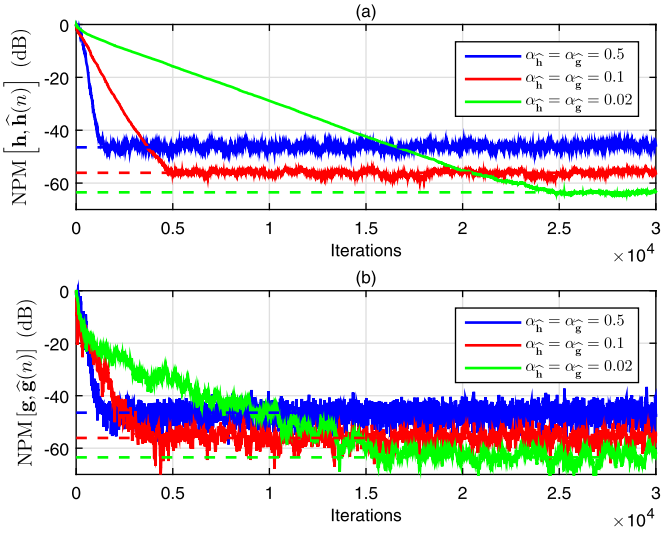

Fig. 8. Performance of the NLMS-BF algorithm in terms of (a) NPM$\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the normalized step-size parameters. The normalized limits $\underline{m}'_{\mathbf{h}}(\infty)$ and $\underline{m}'_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are white Gaussian noises, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

and $\alpha_{\widehat{\mathbf{g}}}$. As we can notice in Figs. 8 and 9, the NPMs converge to their theoretical limits $\underline{m}'_{\mathbf{h}}(\infty)$ and $\underline{m}'_{\mathbf{g}}(\infty)$ from (70) and (71), respectively. Also, since $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}}$, the two filters converge to the same misalignment level. Of course, as expected, the convergence rate of the NLMS-BF algorithm is slower in case of AR(1) inputs (Fig. 9), as compared to the case when the input signals are white Gaussian noises (Fig. 8).

In the previous simulations, the normalized step-size parameters of the NLMS-BF algorithm were chosen such that $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}}$. Of course, different strategies can be considered when setting the values of these parameters. In Fig. 10, the performance of the NLMS-BF algorithm using different values of $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$ (such that $\alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}} = 1$) is evaluated in terms of the NPMs. Also, in Fig. 11, the results are evaluated in terms of the NM (for the identification of the global impulse response $\mathbf{f}$). The input signals are white Gaussian noises. Clearly, the overall behavior is mainly controlled

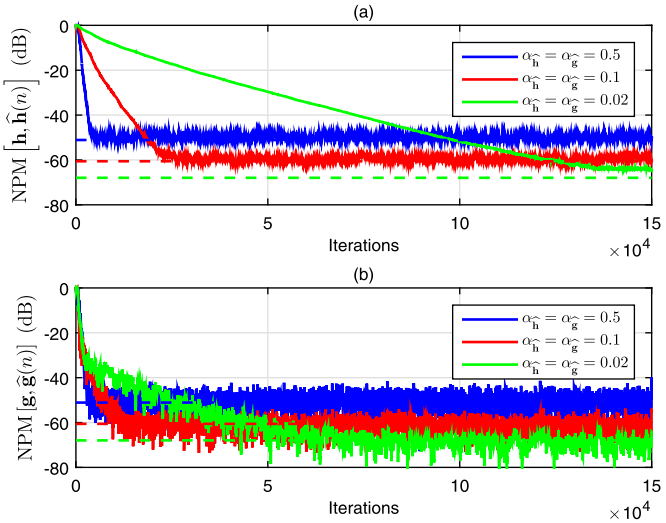

Fig. 9. Performance of the NLMS-BF algorithm in terms of (a) NPM$\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the normalized step-size parameters. The normalized limits $\underline{m}'_{\mathbf{h}}(\infty)$ and $\underline{m}'_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

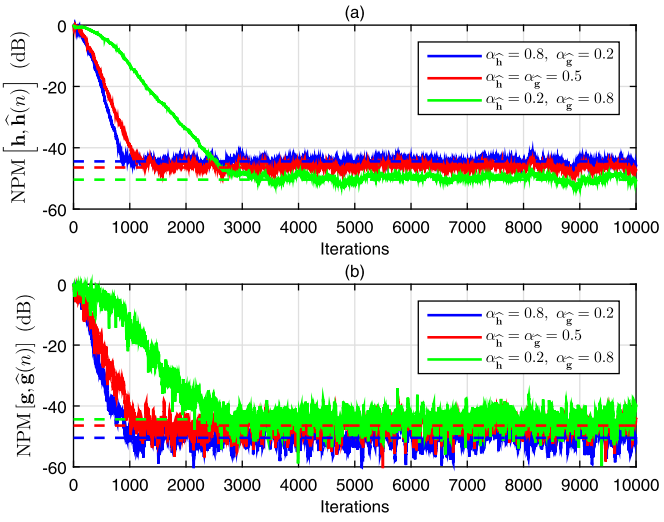

Fig. 10. Performance of the NLMS-BF algorithm in terms of (a) NPM$\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g}, \widehat{\mathbf{g}}(n)]$ for different values of the normalized step-size parameters, with $\alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}} = 1$. The normalized limits $\underline{m}'_{\mathbf{h}}(\infty)$ and $\underline{m}'_{\mathbf{g}}(\infty)$ are also included (dashed lines). The input signals are white Gaussian noises, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

by the temporal filter $\widehat{\mathbf{h}}(n)$, which is the longer one in our scenario (since $L > M$). As shown in Section 4 [following the discussion after (72)], for different choices of $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$, such that $\alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}}$ is a constant, the global filter $\widehat{\mathbf{f}}(n)$ converges to the same misalignment level (as indicated in Fig. 11).

Next, the performances of the regular NLMS and NLMS-BF algorithms are evaluated in terms of identifying the spatiotemporal (i.e., global) impulse response, $\mathbf{f}$, of length $ML = 8 \times 64 = 512$. The results presented in Figs. 12 and 13 support the discussion that follows (75). As we can notice, the NLMS-BF algorithm using $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$ converge to the same misalignment level as the regular NLMS algorithm using $\alpha = \alpha_{\widehat{\mathbf{h}}} + \alpha_{\widehat{\mathbf{g}}}$. However, the NLMS-BF algorithm outperforms its regular counterpart in terms of the convergence rate; the gain is more apparent when using AR(1) pro-

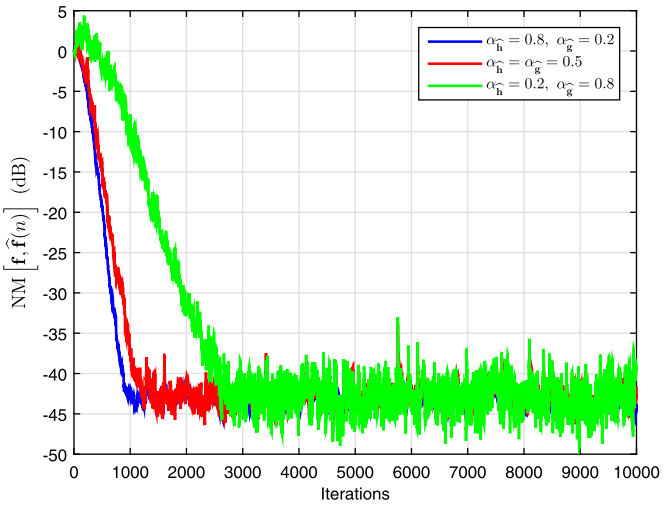

Fig. 11. Performance of the NLMS-BF algorithm in terms of NM$\left[\mathbf{f},\widehat{\mathbf{f}}(n)\right]$ for different values of the normalized step-size parameters, with $\alpha_{\widehat{\mathbf{h}}}+\alpha_{\widehat{\mathbf{g}}}=1$. The input signals are white Gaussian noises, $\sigma_w^2=0.01$, and $ML=512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

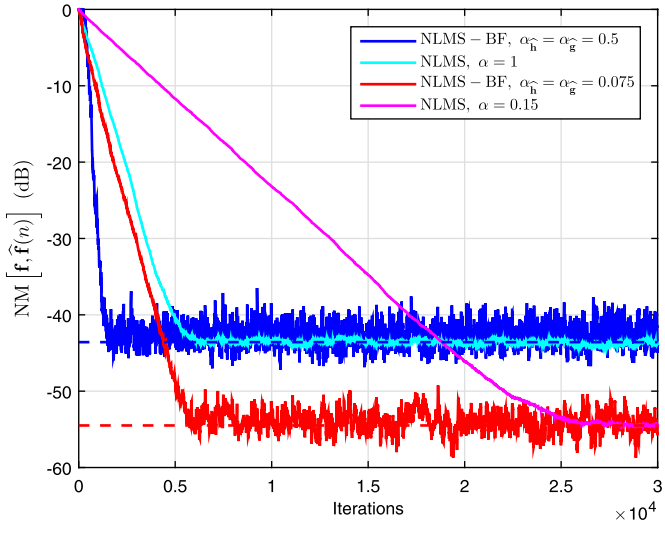

Fig. 12. Performance of the NLMS and NLMS-BF algorithms in terms of NM$\left[\mathbf{f},\widehat{\mathbf{f}}(n)\right]$ for different values of the normalized step-size parameters. The normalized limit $\underline{m}'_{\mathbf{h}}(\infty)+\underline{m}'_{\mathbf{g}}(\infty)$ is also included (dashed lines). The input signals are white Gaussian noises, $\sigma_w^2=0.01$, and $ML=512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

cesses as input signals (Fig. 13). As mentioned in Section 4, the reason is that the regular NLMS algorithm updates an adaptive filter of length ML, while the NLMS-BF algorithm uses two shorter filters of lengths L and M, respectively.

The VSS-NLMS-BF algorithm developed at the end of Section 4 aims to achieve a proper compromise between the main performance criteria, i.e., convergence rate versus misadjustment. In Figs. 14 and 15, the NLMS-BF algorithm using constant values of the normalized step-size parameters is compared to the VSS-NLMS-BF algorithm, in terms of the NPMs and NM, respectively; the input signals are white Gaussian noises. It is assumed that the value of σ_w^2 is available within the VSS-NLMS-BF algorithm. As we can notice from these figures, the VSS-NLMS-BF algorithm has a fast convergence rate (specific to the NLMS-BF algorithm using larger normalized step-sizes), while reaching a low misalignment level (similar to the NLMS-BF algorithm with smaller normalized step-sizes).

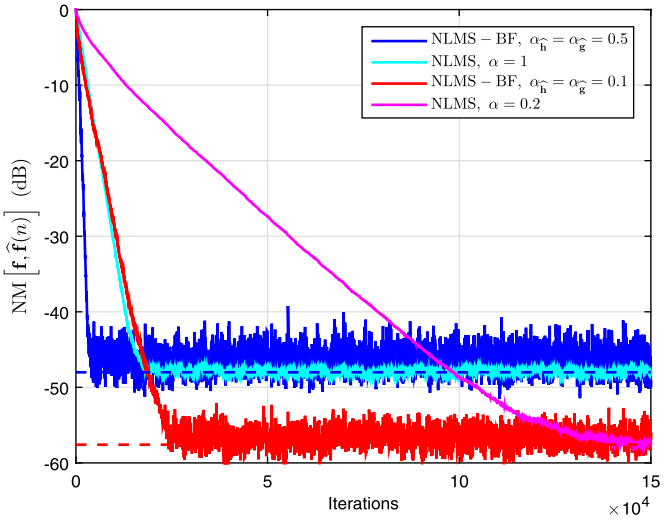

Fig. 13. Performance of the NLMS and NLMS-BF algorithms in terms of NM$\left[\mathbf{f},\widehat{\mathbf{f}}(n)\right]$ for different values of the normalized step-size parameters. The normalized limit $\underline{m}'_{\mathbf{h}}(\infty)+\underline{m}'_{\mathbf{g}}(\infty)$ is also included (dashed lines). The input signals are AR(1) processes, $\sigma_w^2=0.01$, and $ML=512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

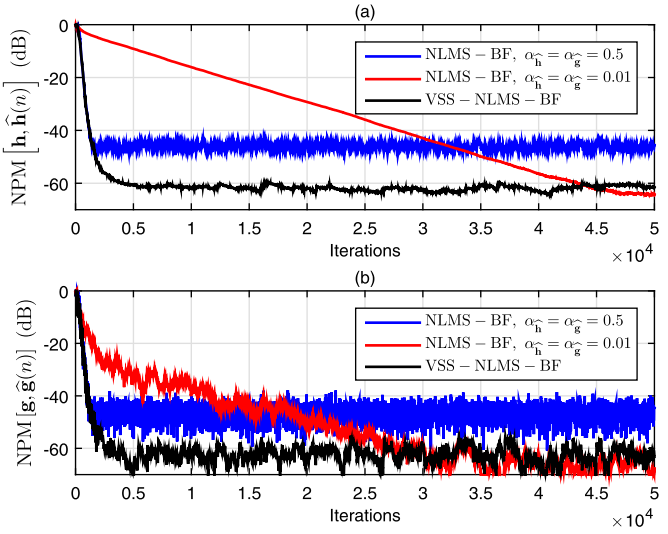

Fig. 14. Performance of the NLMS-BF (using different values of the normalized step-size parameters) and VSS-NLMS-BF algorithms in terms of (a) NPM$\left[\mathbf{h},\widehat{\mathbf{h}}(n)\right]$ and (b) NPM$[\mathbf{g},\widehat{\mathbf{g}}(n)]$. The input signals are white Gaussian noises, $\sigma_w^2=0.01$, $L=64$, and $M=8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

The previous experiment is repeated in Figs. 16 and 17, using AR(1) processes as input signals. The main conclusion is basically the same, i.e., the VSS-NLMS-BF algorithm leads to a proper compromise in terms of the performance criteria, achieving a fast convergence rate but also reaching a low misalignment level. As expected, the convergence rates of the algorithms are slower in this case, as compared to the previous case reported in Figs. 14 and 15 (where the input signals are white Gaussian noises).

Now, we evaluate the RLS-BF algorithm by simulations. In Figs. 18 and 19, the RLS-BF algorithm using different values of the forgetting factors ($\lambda_{\widehat{\mathbf{h}}}$ and $\lambda_{\widehat{\mathbf{g}}}$) is compared to the iterative Wiener filter recently proposed in [18]; the input signals are AR(1) processes. We should note that the Wiener filter provides the optimal solution of the system identification problem, but it requires the estimation of the statistics (e.g., the covariance matrix $\mathbf{R}$ of the multi-input signal; see Section 2) and also involves matrix inver-

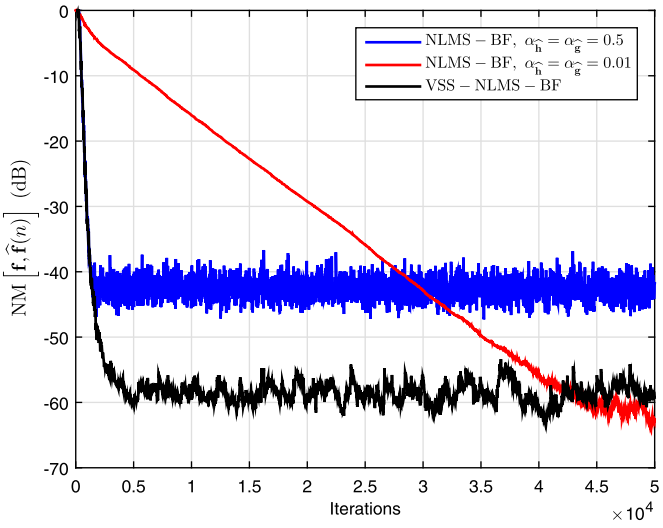

Fig. 15. Performance of the NLMS-BF (using different values of the normalized step-size parameters) and VSS-NLMS-BF algorithms in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$. The input signals are white Gaussian noises, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

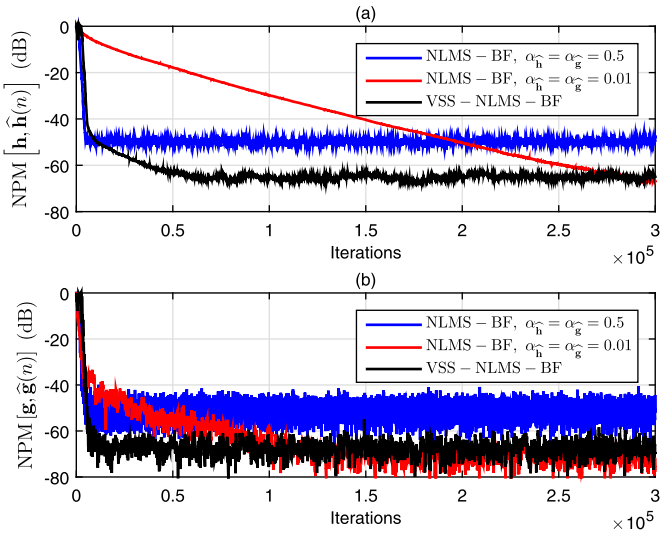

Fig. 16. Performance of the NLMS-BF (using different values of the normalized step-size parameters) and VSS-NLMS-BF algorithms in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

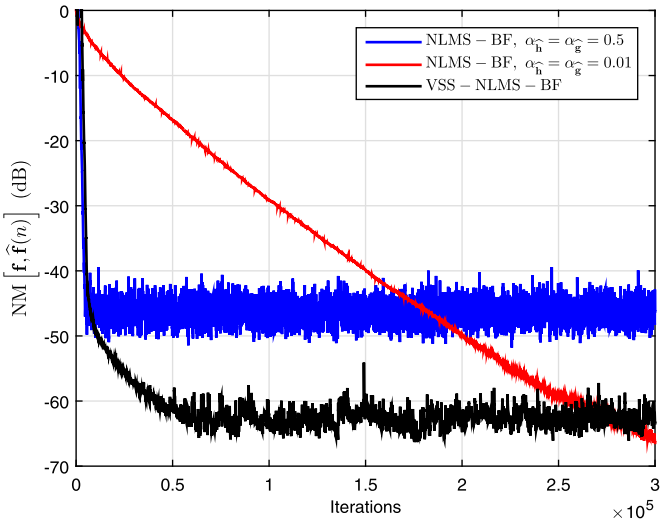

Fig. 17. Performance of the NLMS-BF (using different values of the normalized step-size parameters) and VSS-NLMS-BF algorithms in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

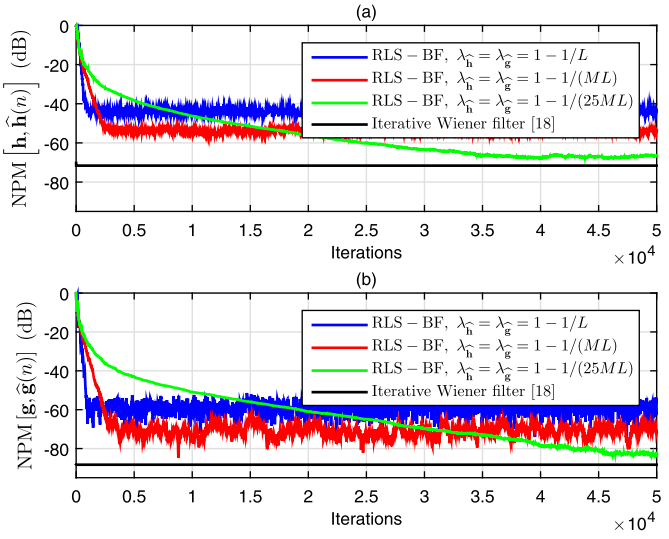

Fig. 18. Performance of the RLS-BF algorithm (using different values of the forgetting factors) and iterative Wiener filter [18] in terms of (a) NPM $\left[\mathbf{h}, \widehat{\mathbf{h}}(n)\right]$ and (b) NPM $[\mathbf{g}, \widehat{\mathbf{g}}(n)]$. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, $L = 64$, and $M = 8$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

sion operations [18]. Due to these reasons, it is prohibitive for real-time applications. As we can notice in Figs. 18 and 19, the convergence rate of the RLS-BF algorithm increases for lower values of the forgetting factors. On the other hand, as the values of the forgetting factors increase, the RLS-BF algorithm achieves a lower misalignment level, which is very close to the optimal solution provided by the Wiener filter. In other words, it can be noticed that larger values of the forgetting factors lead to a lower misalignment but also to a slower convergence rate.

Next, let us consider that the input signals are dependent, similar to the context of the nonlinear acoustic echo cancellation application described in [20–22] and [24], which resembles a Hammerstein model [25]. This is a particular case of our framework (which is related to cascaded systems), as explained in Section 2 and illustrated in Fig. 1(d). Consequently, we choose $x_m(n) = x^m(n)$, $m = 1, 2, \ldots, M$, where $x(n)$ is an AR(1) process; for this simulation, we set $M = 4$ (while $L = 64$). The results are presented in Fig. 20, where the RLS-BF algorithm (using different forgetting factors) is compared to the NLMS-BF algorithm, in terms of the NM (for the identification of the global impulse response, $\mathbf{f}$, of length $ML = 4 \times 64 = 256$). The step-sizes of the NLMS-BF algorithm are set to $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.5$, thus targeting a fast convergence rate. As we can notice, the advantage of the RLS-BF algorithm becomes more apparent in this case, outperforming by far the NLMS-BF algorithm in terms of convergence rate.

In the next experiment, reported in Fig. 21, the performances of the LMS-BF, NLMS-BF, and RLS-BF algorithms are compared in terms of the NM (for the identification of the global impulse response, $\mathbf{f}$, of length $ML = 8 \times 64 = 512$). In this simulation, the input signals are speech sequences. The adaptation parameters are chosen so that the algorithms reach the same misalignment level.

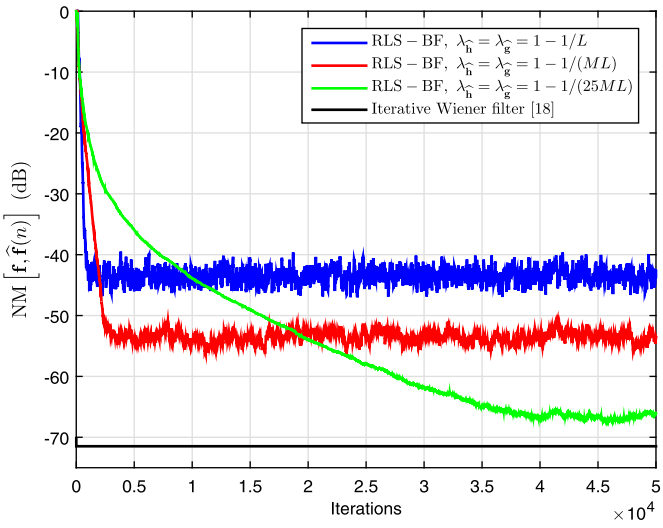

Fig. 19. Performance of the RLS-BF algorithm (using different values of the forgetting factors) and iterative Wiener filter [18] in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$. The input signals are AR(1) processes, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

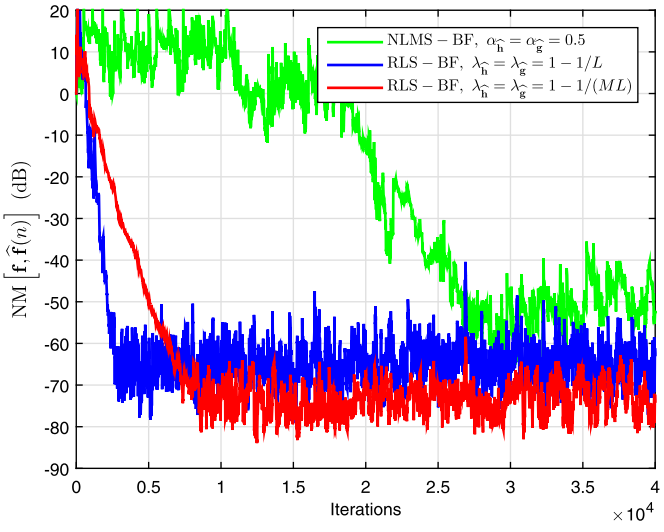

Fig. 20. Performance of the NLMS-BF (using $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.5$) and RLS-BF algorithm (using different values of the forgetting factors) in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$. The input signals are $x_m(n) = x^m(n)$, with $m = 1, 2, \ldots, M$, where $x(n)$ is an AR(1) process; the other parameters are $\sigma_w^2 = 0.01$ and $ML = 256$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

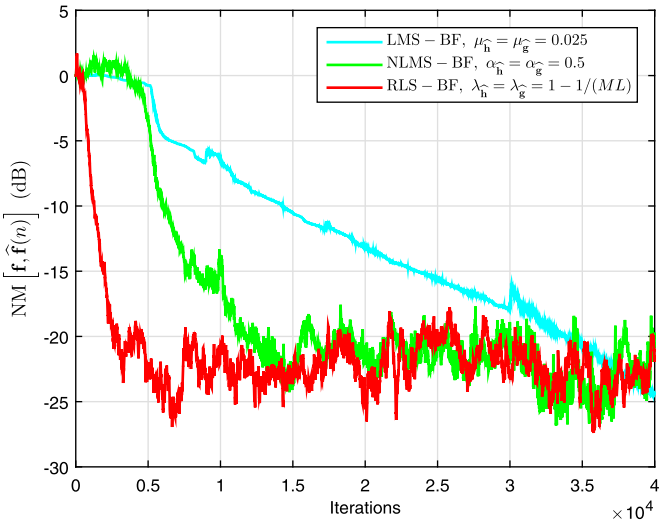

Fig. 21. Performance of the LMS-BF, NLMS-BF, and RLS-BF algorithms in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$. The adaptation parameters are $\mu_{\widehat{\mathbf{h}}} = \mu_{\widehat{\mathbf{g}}} = 0.025$, $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.5$, and $\lambda_{\widehat{\mathbf{h}}} = \lambda_{\widehat{\mathbf{g}}} = 1 - 1/(ML)$. The input signals are speech sequences, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

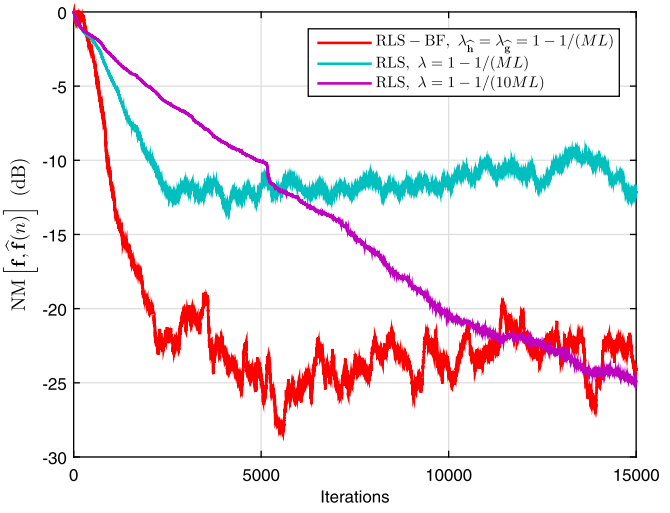

Fig. 22. Performance of the RLS and RLS-BF algorithms in terms of NM $\left[\mathbf{f}, \widehat{\mathbf{f}}(n)\right]$ for different values of the forgetting factors. The input signals are speech sequences, $\sigma_w^2 = 0.01$, and $ML = 512$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

As we can notice, the NLMS-BF algorithm performs better (in terms of the convergence rate) for non-stationary inputs like speech, as compared to the LMS-BF algorithm. Also, as expected, the RLS-BF algorithm outperforms its LMS-based counterparts, by achieving the fastest convergence rate.

Similar to the regular LMS and NLMS algorithms, which can be used for the identification of the global impulse response $\mathbf{f}$, it is straightforward to design a regular RLS algorithm for this purpose. Based on desired signal expressed in (7) and the error signal from the last line of (16), the update of this algorithm results in

$$\widehat{\mathbf{f}}(n) = \widehat{\mathbf{f}}(n-1) + \mathbf{k}(n)e(n), \tag{97}$$

where $\mathbf{k}(n)$ is the Kalman gain vector that can be evaluated as

$$\mathbf{k}(n) = \frac{\mathbf{R}^{-1}(n-1)\widetilde{\mathbf{x}}(n)}{\lambda + \widetilde{\mathbf{x}}^T(n)\mathbf{R}^{-1}(n-1)\widetilde{\mathbf{x}}(n)}, \tag{98}$$

with $\mathbf{R}(n)$ denoting the estimated covariance matrix of the multi-input signal $\widetilde{\mathbf{x}}(n)$ (its inverse can be recursively computed based on the matrix inversion lemma [30]), while λ is the forgetting factor of the regular RLS algorithm. Nevertheless, we should note that the computational complexity of the regular RLS algorithm is much higher as compared to the proposed RLS-BF algorithm, since the matrix $\mathbf{R}(n)$ is very large (of size $ML \times ML$), while the matrices $\mathbf{R}_{\widehat{\mathbf{g}}}(n)$ and $\mathbf{R}_{\widehat{\mathbf{h}}}(n)$ are much smaller (i.e., $L \times L$ and $M \times M$, respectively).

In the last experiment (reported in Fig. 22), the performances of the regular RLS and RLS-BF algorithms are evaluated in terms of identifying the spatiotemporal (i.e., global) impulse response, $\mathbf{f}$, of length $ML = 8 \times 64 = 512$; the input signals are speech sequences. Two different values of the forgetting factor λ are used in the case of the regular RLS algorithm. As we can notice in Fig. 22, a fast convergence rate is obtained by the regular RLS and RLS-BF algorithms when using the same values of the for-

getting factors, i.e., $\lambda = \lambda_{\widehat{\mathbf{h}}} = \lambda_{\widehat{\mathbf{g}}} = 1 - 1/(ML)$. However, the proposed RLS-BF algorithm reaches a lower misalignment level, similar to the regular RLS algorithm using a larger forgetting factor, i.e., $\lambda = 1 - 1/(10ML)$; on the other hand, the convergence rate of the regular RLS algorithm is slower in this case. Overall, the proposed RLS-BF algorithm outperforms its regular counterpart, while also having a lower computational complexity.

7. Conclusions and perspectives

In this paper, we have presented an adaptive filtering approach for the identification of bilinear forms, in the context of MISO systems. As compared to most previous works related to bilinear systems (where the bilinear form is usually defined in terms of an input–output relation), we have focused on a different model, by defining the bilinear term with respect to the impulse responses of the spatiotemporal model. The idea was previously addressed in [18], in terms of the Wiener filter.

The algorithms developed in this paper, namely LMS-BF, NLMS-BF (together with its variable step-size version, i.e., VSS-NLMS-BF), and RLS-BF are tailored for such bilinear forms. In this framework, a convergence analysis of the LMS-BF algorithm has been provided and the results have been also extended to the NLMS-BF algorithm. The VSS-NLMS-BF algorithm has been designed to achieve a better compromise between the main performance criteria, i.e., convergence rate versus misadjustment. As expected, the RLS-BF algorithm outperforms its LMS-based counterparts in terms of convergence rate. Overall, simulation results support the theoretical analysis and indicate the good performance of these algorithms. Consequently, they could represent appealing solutions for bilinear system identification problems.

Future works will focus on developing other efficient algorithms based on this approach. Moreover, it would be interesting to explore and analyze the case when the two adaptive filters work based on different types of algorithms (in this paper, we have considered only the case when they are updated based on the same type of algorithm). Also, it would be highly useful to identify more potential applications of this multidimensional adaptive filtering technique.

Acknowledgments

This work was supported by European Space Agency under Grant no. 4000121222/17/NL/CBi and University Politehnica Bucharest under Grant no. 18/05.10.2017.

The authors would like to thank the Handling Editor and the reviewers for the valuable comments and suggestions.

References

[1] R.R. Mohler, W.J. Kolodziej, An overview of bilinear system theory and applications, IEEE Trans. Syst. Man Cybern. SMC-10 (Oct. 1980) 683–688.
[2] C. Bruni, G. DiPillo, G. Koch, Bilinear systems: an appealing class of “nearly linear” systems in theory and applications, IEEE Trans. Autom. Control AC-19 (Aug. 1974) 334–348.
[3] T.U. Halawani, R.R. Mohler, W.J. Kolodziej, A two-step bilinear filtering approximation, IEEE Trans. Acoust. Speech Signal Process. ASSP-32 (Apr. 1984) 344–352.
[4] M. Inagaki, H. Mochizuki, Bilinear system identification by Volterra kernels estimation, IEEE Trans. Autom. Control AC-29 (Aug. 1984) 746–749.
[5] H.K. Baik, V.J. Mathews, Adaptive lattice bilinear filters, IEEE Trans. Signal Process. 41 (June 1993) 2033–2046.
[6] U. Forssén, Adaptive bilinear digital filters, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. 40 (Nov. 1993) 729–735.
[7] G.-K. Ma, J. Lee, V.J. Mathews, A RLS bilinear filter for channel equalization, in: Proc. IEEE ICASSP, 1994, III-257–III-260.
[8] J. Lee, V.J. Mathews, Adaptive bilinear predictors, in: Proc. IEEE ICASSP, III, 1994, pp. 489–492.
[9] R. Hu, H.M. Hassan, Echo cancellation in high speed data transmission systems using adaptive layered bilinear filters, IEEE Trans. Commun. 42 (Feb./Mar./Apr. 1994) 655–663.
[10] T. Bose, M.-Q. Chen, Conjugate gradient method in adaptive bilinear filtering, IEEE Trans. Signal Process. 43 (June 1995) 1503–1508.
[11] J. Lee, V.J. Mathews, Output-error LMS bilinear filters with stability monitoring, in: Proc. IEEE ICASSP, 1995, pp. 965–968.
[12] Z. Zhu, H. Leung, Adaptive identification of nonlinear systems with application to chaotic communications, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. 47 (July 2000) 1072–1080.
[13] S.M. Kuo, H.-T. Wu, Nonlinear adaptive bilinear filters for active noise control systems, IEEE Trans. Circuits Syst. I, Regul. Pap. 52 (Mar. 2005) 617–624.
[14] P. Lopes dos Santos, J.A. Ramos, J.L. Martins de Carvalho, Identification of bilinear systems with white noise inputs: an iterative deterministic–stochastic subspace approach, IEEE Trans. Control Syst. Technol. 17 (Sept. 2009) 1145–1153.
[15] J. Zhang, H. Zhao, A novel adaptive bilinear filter based on pipelined architecture, Digit. Signal Process. 20 (Jan. 2010) 23–38.
[16] H. Zhao, X. Zeng, Z. He, Low-complexity nonlinear adaptive filter based on a pipelined bilinear recurrent neural network, IEEE Trans. Neural Netw. 22 (Sept. 2011) 1494–1507.
[17] L. Tan, J. Jiang, Nonlinear active noise control using diagonal-channel LMS and RLS bilinear filters, in: Proc. IEEE MWSCAS, 2014, pp. 789–792.
[18] J. Benesty, C. Paleologu, S. Ciochină, On the identification of bilinear forms with the Wiener filter, IEEE Signal Process. Lett. 24 (May 2017) 653–657.
[19] D. Gesbert, P. Duhamel, Robust blind joint data/channel estimation based on bilinear optimization, in: Proc. IEEE WSSAP, 1996, pp. 168–171.
[20] A. Stenger, W. Kellermann, R. Rabenstein, Adaptation of acoustic echo cancellers incorporating a memoryless nonlinearity, in: Proc. IEEE IWAENC, 1999, 4 pp.
[21] A. Stenger, W. Kellermann, Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling, Signal Process. 80 (Sept. 2000) 1747–1760.
[22] A. Stenger, W. Kellermann, RLS-adapted polynomial for nonlinear acoustic echo cancelling, in: Proc. EUSIPCO, 2000, 4 pp.
[23] R. Abrahamsson, S.M. Kay, P. Stoica, Estimation of the parameters of a bilinear model with applications to submarine detection and system identification, Digit. Signal Process. 17 (July 2007) 756–773.
[24] Y. Huang, J. Skoglund, A. Luebs, Practically efficient nonlinear acoustic echo cancellers using cascaded block RLS and FLMS adaptive filters, in: Proc. IEEE ICASSP, 2017, pp. 596–600.
[25] E.-W. Bai, D. Li, Convergence of the iterative Hammerstein system identification algorithm, IEEE Trans. Autom. Control 49 (Nov. 2004) 1929–1940.
[26] J. Benesty, I. Cohen, J. Chen, Fundamentals of Signal Enhancement and Array Signal Processing, Wiley-IEEE Press, Singapore, 2018.
[27] J. Liu, S.L. Grant, Proportionate adaptive filtering for block-sparse system identification, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (Apr. 2016) 623–630.
[28] D.A. Harville, Matrix Algebra from a Statistician’s Perspective, Springer-Verlag, New York, 1997.
[29] D.R. Morgan, J. Benesty, M.M. Sondhi, On the evaluation of estimated impulse responses, IEEE Signal Process. Lett. 5 (July 1998) 174–176.
[30] S. Haykin, Adaptive Filter Theory, fourth edition, Prentice-Hall, Upper Saddle River, NJ, 2002.
[31] A.I. Sulyman, A. Zerguine, Convergence and steady-state analysis of a variable step-size NLMS algorithm, Signal Process. 83 (June 2003) 1255–1273.
[32] S. Ciochină, C. Paleologu, J. Benesty, Analysis of an LMS algorithm for bilinear forms, in: Proc. IEEE DSP, 2017, 5 pp.
[33] D.R. Morgan, S.G. Kratzer, On a class of computationally efficient, rapidly converging, generalized NLMS algorithms, IEEE Signal Process. Lett. 3 (Aug. 1996) 245–247.
[34] C. Paleologu, J. Benesty, S. Ciochină, An NLMS algorithm for the identification of bilinear forms, in: Proc. EUSIPCO, 2017, pp. 2689–2693.
[35] J. Benesty, C. Paleologu, S. Ciochină, On regularization in adaptive filtering, IEEE Trans. Audio Speech Lang. Process. 19 (Aug. 2011) 1734–1742.
[36] J. Benesty, H. Rey, L. Rey Vega, S. Tressens, A non-parametric VSS NLMS algorithm, IEEE Signal Process. Lett. 13 (Oct. 2006) 581–584.
[37] C. Paleologu, S. Ciochină, J. Benesty, Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation, IEEE Signal Process. Lett. 15 (2008) 5–8.
[38] M.A. Iqbal, S.L. Grant, Novel variable step size NLMS algorithms for echo cancellation, in: Proc. IEEE ICASSP, 2008, pp. 241–244.

Constantin Paleologu was born in Romania in 1975. In 1998, he received the Masters degree in telecommunications networks from the Faculty of Electronics and Telecommunications, University Politehnica of Bucharest, Romania. He also received a Masters degree in digital signal processing in 1999 and a Ph.D. degree in adaptive signal processing in 2003, both from the same institution.

During his Ph.D. program (from December 1999 to July 2003), he worked on adaptive filters and echo cancellation. Since October 1998 he has been with the Telecommunications Department, University Politehnica

of Bucharest, where he is currently a Professor. His research interests include adaptive filtering algorithms and acoustic signal processing. He co-authored the books *Sparse Adaptive Filters for Echo Cancellation* (Morgan and Claypool, 2010) and *A Perspective on Stereophonic Acoustic Echo Cancellation* (Springer-Verlag, 2011).

He received the "IN HOC SIGNO VINCES" award from the Romanian National Research Council in 2009. In 2010, he received the "IN TEMPORE OPPORTUNO" award from University Politehnica of Bucharest and the "Gheorghe Cartianu Award" from the Romanian Academy.

Silviu Ciochină is a Professor with the University Politehnica of Bucharest and was the Head of the Telecommunications Department, Faculty of Electronics, Telecommunications, and Information Technology (between 2004 and 2016). His main areas of interest are digital signal processing, adaptive algorithms, and wireless communication technologies. He is the author of about 200 papers published in international journals and conference proceedings, seven of them being distinguished with "paper award."

Prof. Ciochina is author or co-author of 10 books, three of them published by international book publishers. He was awarded by Romanian Academy (Traian Vuia-1981 and Gheorghe Cartianu-1999 prizes), Education Ministry and Defense Ministry (for works in the Radar field). He has been involved in many national and international R&D projects in the areas of communications, signal processing, and radar. He was the coordinator of the Romanian teams implied in the FP6 projects ATHENA (STREP-507312), ENTHRONE 1 (IP-507637), and ENTHRONE 2 (IP-038463).

Jacob Benesty received a Master degree in microwaves from Pierre & Marie Curie University, France, in 1987, and a Ph.D. degree in control and signal processing from Orsay University, France, in April 1991. During his Ph.D. (from Nov. 1989 to Apr. 1991), he worked on adaptive filters and fast algorithms at the Centre National d'Etudes des Telecomunications (CNET), Paris, France. From January 1994 to July 1995, he worked at Telecom Paris University on multichannel adaptive filters and acoustic echo cancellation. From October 1995 to May 2003, he was first a Consultant and then a Member of the Technical Staff at Bell Laboratories, Murray Hill, NJ, USA. In May 2003, he joined the University of Quebec, INRS-EMT, in Montreal, Quebec, Canada, as a Professor. He is also a Visiting Professor at the Technion, in Haifa, Israel, and an Adjunct Professor at Aalborg University, in Denmark.

His research interests are in signal processing, acoustic signal processing, and multimedia communications. He is the inventor of many important technologies. In particular, he was the lead researcher at Bell Labs who conceived and designed the world-first real-time hands-free full-duplex stereophonic teleconferencing system. Also, he conceived and designed the world-first PC-based multi-party hands-free full-duplex stereo conferencing system over IP networks.

He is the editor of the book series Springer Topics in Signal Processing. He was the general chair and technical chair of many international conferences. He was a member of several IEEE technical committees. Four of his journal papers were awarded by the IEEE Signal processing Society. In 2010, he received the Gheorghe Cartianu Award from the Romanian Academy. He has co-authored and co-edited/co-authored numerous books in the area of acoustic signal processing.