“POLITEHNICA” UNIVERSITY OF BUCHAREST

ETTI-B DOCTORAL SCHOOL

Decision No. 414 from 24.07.2019

Adaptive Algorithms for Multilinear in Parameters Structures

Algoritmi adaptivi pentru structuri multiliniare în parametri

by **Laura-Maria IANCU (DOGARIU)**

DISSERTATION COMMITTEE

President	**Prof. Dr. Ing. Gheorghe BREZEANU**	from	**University Politehnica of Bucharest**
Advisor	**Prof. Dr. Ing. Silviu CIOCHINĂ**	from	**University Politehnica of Bucharest**
Reviewer	**Prof. Dr. Ing. Constantin PALEOLOGU**	from	**University Politehnica of Bucharest**
Reviewer	**Prof. Dr. Ing. Doru-Florin CHIPER**	from	**Gheorghe Asachi Technical University of Iaşi**
Reviewer	**Prof. Dr. Ing. Dinu COLŢUC**	from	**Valahia University of Târgovişte**

Bucharest 2019

Acknowledgements

First of all, I owe my deepest gratitude to my PhD advisor, Prof. Dr. Eng. Silviu Ciochină, for his constant guidance, scientific support and valuable advice throughout the past three years. I am also extremely grateful to Prof. Dr. Eng. Constantin Paleologu for his contribution to my work, for being by my side on every step of the road and for standing as a model for me.

I would like to thank Prof. Dr. Eng. Doru-Florin Chiper and Prof. Dr. Eng. Dinu Colţuc for agreeing to be my thesis reviewers and for their useful suggestions and comments.

I would also like to thank my colleagues in the Telecommunications Department for their kindness and for making me feel part of the group.

I would like to express my gratitude to my parents for their unconditional love and for encouraging me to always pursue my dreams.

I owe special thanks to my husband Mihai, who has been a constant source of love and support throughout the years; without him, I would have much less happiness in my life.

Finally, I would like to thank my brother for reminding me the importance of occasionally interrupting the working cycle so you can enjoy all the other beautiful moments in life.

Contents

List of Tables

List of Figures

Abbreviations and Acronyms

APA	affine projection algorithm
AR	autoregressive
BF	bilinear form(s)
IPAPA	improved proportionate affine projection algorithm
IPNLMS	improved proportionate normalized least-mean-square
JO-NLMS	joint-optimized normalized least-mean-square
KF	Kalman filter
LMS	least-mean-square
MIMO	multiple-input/multiple-output
MISO	multiple-input/single-output
MMSE	minimum mean-squared error
MSE	mean-squared error
NKP	nearest Kronecker product
NLMS	normalized least-mean-square
NM	normalized misalignment
NPM	normalized projection misalignment
OLMS	optimized least-mean-square
RLS	recursive least-squares
SKF	simplified Kalman filter
SMCM	system mismatch covariance matrix

SNR	signal-to-noise ratio
SVD	singular value decomposition
TF	trilinear form(s)
WGN	white Gaussian noise

Chapter 1

Introduction

1.1 Context and Description of the Research Field

Adaptive algorithms have been successfully used in system identification for a long time (see [1, 2, 3] and many others). The well-known Wiener filter [4] constitutes the theoretical benchmark and starting point for many of these algorithms. Some of the most popular such algorithms are the least-mean-square (LMS) algorithm [5, 6, 7], the recursive least-squares (RLS) algorithm [8], and the affine projection algorithm (APA) [1, 9, 10]. The Kalman filter (KF) [11] can also be used successfully in system identification problems. Different variants of these algorithms have been developed to solve specific problems; the most related applications to the work presented in this thesis are in the acoustic echo cancellation area, where the previously mentioned algorithms are extremely popular [1, 2, 9, 10, 12, 13, 14, 15, 16].

Nonlinear systems have also been studied for a long time, and the theory of nonlinear algorithms is used nowadays in many important fields. However, a global technique for treating nonlinear problems has not been found yet, and the approaches to address this problem are different and depend on the application and on the types of nonlinearities. A particular type of nonlinear systems are the multilinear in parameters systems. This specific category can be regarded as an ensemble of (approximately) linear structures. Basically, the multilinear term is understood in the context of a global nonlinear separable system, which is composed of shorter linear systems. In this way, if all the individual impulse responses are kept constant, except one, the global filter would be a linear function of that system. Some popular examples of this type are nonlinear electronic devices, for example acoustic echo cancellers, which can be used in applications for two-party or multi-party voice communications (e.g., videoconference solutions). These can be modeled as a cascade of nonlinear and linear systems, similar to the Hammerstein model [17, 18]. Other structures that may be treated as separable multilinear systems are the multiple-input/single-output (MISO) systems. The high dimension of the parameter space in such problems can be approached with techniques which use tensor decompositions and modelling.

1.2 Purpose of the Thesis

This thesis focuses on analyzing and developing adaptive algorithms for nonlinear systems. More specifically, the second and third chapters aim to develop such algorithms tailored for several types of multilinear in parameters systems. In this framework, the cases of bilinear and trilinear structures are studied, obtaining appealing solutions that can be applied with better performance than the existing approaches. In addition, a mathematical analysis of the properties of the system mismatch covariance matrix (SMCM) of the LMS algorithm is conducted, with interesting results. The parameters involved in the adaptive algorithms are also studied, in order to reach the estimations that can ensure good identification performances.

Moreover, in the fourth chapter, we treat the general case of systems which are not perfectly separable, and in this context we propose a KF that uses the nearest Kronecker product (NKP) decomposition technique, together with low-rank approximation, to identify this kind of spatiotemporal systems. Then, in the fifth chapter, we focus on a slightly different case of supposedly linear systems that present small nonlinearities and we provide an adaptive solution for the identification of such systems.

For all the theoretical studies that are presented in this thesis, experiments are also conducted to show the applicability of the approaches. The performance of the proposed algorithms is always illustrated in comparison with the already existent options that can be used in the same scenarios. This is done in order to have a correct perspective on the advantages and drawbacks of the provided solutions.

1.3 Thesis Contents

The thesis is structured into four main chapters, followed by a final one dedicated to the summary of the work and presenting future directions. Each chapter contains separate subchapters for every new subject treated in the general context of the chapter. Every original theoretical contribution is followed by experimental data to support the theoretical findings. The contents of each chapter are shortly described below.

Chapter 2 starts by describing the general framework in which the subsequent developments are made. In this context, bilinear forms (BF) are defined in the context of a MISO system, with respect to the impulse responses of a spatiotemporal model. As will be explained, this approach is different from the usual definition of BF used in previous works, namely in terms of an input-output relation. Once the framework was defined, the chapter continues by shortly reviewing a Wiener filter, along with an LMS adaptive algorithm and an NLMS, tailored for BF (LMS-BF, respectively NLMS-BF). These previous works' results are necessary for a better understanding of the subsequent developments. The next step is the detailed presentation our contributions in the field of BF. To begin with, an APA and an improved proportionate version of it are developed in the context of BF (APA-BF, respectively IPAPA-BF). Then, an optimized LMS algorithm adapted to the case of BF (OLMS-BF) is introduced in subchapter 2.5, which was obtained through the optimization of the step-size parameter in the LMS-BF algorithm. This is followed by a mathematical analysis of the properties of the system mismatch covariance matrix (SMCM) in the LMS adaptive filter, in 2.6. Then, in 2.7, a KF for BF is derived (KF-BF), together with a computationally simplified version of it (SKF-BF). Finally, an interesting parallel between the SKF-BF and the OLMS-BF algorithm is outlined. The last part of the chapter draws some conclusions on the presented matters.

In Chapter 3, we begin with an introduction in the study of tensors, which are the main mathematical tools used when working with trilinear forms (TF), followed by laying out the system model for the context of TF. We continue by deriving a Wiener filter tailored for the particular case of TF, comprising both a direct and an iterative version, and then we present the LMS algorithm for TF (LMS-TF). By changing the step-size from constant to time-variant, we reach the NLMS algorithm used for TF (NLMS-TF). The performance of the developed solutions is emphasized through simulations and then a few general final remarks are set out.

Chapter 4 commences by overviewing an efficient approach for the identification of low-rank systems, via the NKP decomposition, followed by a low-rank approximation. The core of the chapter is represented by the development (using the already mentioned techniques) of a KF capable of working in system identification problems with long length impulse responses to be identified (KF-NKP). The potentially problematic choice of some of the parameters of this algorithm is discussed, and solutions are provided and then tested through extensive simulations. The advantages and drawbacks of the proposed approach are outlined in the final part of the chapter.

Next, in Chapter 5, an interesting case study is outlined, namely an adaptive solution for the identification of supposedly linear systems that contain small nonlinearities. The problem setup is introduced, then the proposed method is presented, and the simplicity and the efficiency are highlighted through experiments. Lastly, a discussion on the possible future improvements is conducted.

The last chapter is dedicated to a summary of the contributions described in the thesis and to perspectives for future work.

Chapter 2

Bilinear Forms (BF)

This chapter presents the development of various well-known algorithms in the context of bilinear forms (BF). We start by introducing the system model and explaining how BF are understood in this framework (in subchapter 2.1). Then, in subchapters 2.2 and 2.3, a brief presentation of the Wiener filter and the least-mean-square (LMS) type adaptive algorithms is done. Despite not being actual contributions of this work, they are required, as they provide a basis for the subsequent developments which constitute the novel ideas of the thesis. We continue by detailing the derivation of an affine projection algorithm for BF (APA-BF), together with an improved proportionate version of it, followed by an optimized LMS (OLMS) algorithm tailored for the identification of BF, in subchapter 2.5, completed by experimental results which prove the performance of the proposed solution. Next, in subchapter 2.6, we provide a theoretical analysis of the system mismatch covariance matrix (SMCM) in the LMS adaptive algorithm, together with simulations to support the presented ideas. The Kalman filter for BF (KF-BF) is then introduced in 2.7, together with a computationally simplified version (SKF-BF). Again, we support our statements by showing experimental results. An interesting parallel between the OLMS and the KF is then drawn in subchapter 2.8, and a comparison of their performance is done, both theoretically and through simulations. We end the chapter by drawing the conclusions with respect to the work presented.

The results presented in the following have been published in [19, 20, 21, 22, 23].

2.1 System Model for Bilinear Forms

The information presented in this subchapter was published in [21].

Bilinear systems were previously treated in various contexts [24], such as the nonlinear systems approximation. The applications derived from there are numerous, among which we can mention system identification [25, 26, 27, 28], design of digital filters [29], echo cancellation [30], chaotic communications [31], active noise control [32], neural networks [33], etc. In all these papers, the bilinear term is understood in terms of an input-output relation (with respect to the data).

More recently, a new approach was studied in [34], where the bilinear term was defined in the context of a multiple-input/single-output (MISO) system, with respect to the impulse responses of a spatiotemporal model. In [34], the Wiener filter solution for the identification of such BF was proposed, and then, in [35, 36], some adaptive solutions based on different basic algorithms were provided. Similar frameworks can be found in [37, 38, 39, 17], in conjunction with specific applications, such as channel equalization and nonlinear acoustic echo cancellation; however, these works were not associated with the identification of BF or analyzed in this context.

Throughout this chapter, we will consider the interpretation of BF in the framework of a MISO system, as being composed of the impulse responses of a spatiotemporal model. The signal model that will be used for the system identification problem with BF is given by [34]:

$$d(n) = \mathbf{h}^T(n)\mathbf{X}(n)\mathbf{g}(n) + v(n) = y(n) + v(n), \tag{2.1}$$

where $d(n)$ is the zero-mean desired (or reference) signal at the discrete-time index n, $\mathbf{h}(n)$ and $\mathbf{g}(n)$ are the two impulse responses of the system of lengths L and M, respectively, the superscript ^{T} is the transpose operator, $\mathbf{X}(n) = [\mathbf{x}_1(n) \quad \mathbf{x}_2(n) \quad \dots \quad \mathbf{x}_M(n)]$ is the zero-mean multiple-input signal matrix, $\mathbf{x}_m(n) = [x_m(n) \quad x_m(n-1) \quad \dots \quad x_m(t-L+1)]^T$ is a vector containing the L most recent time samples of the mth ($m = 1, 2, \dots, M$) input signal, $y(n) = \mathbf{h}^T(n)\mathbf{X}(n)\mathbf{g}(n)$ is the output signal and it represents a bilinear form, and $v(n)$ is a zero-mean additive noise (with the variance σ_v^2). It is assumed that all signals are real-valued, and $\mathbf{X}(n)$ and $v(n)$ are independent. As we can notice, $y(n)$ is a bilinear function of $\mathbf{h}(n)$ and $\mathbf{g}(n)$, because for every fixed $\mathbf{h}(n)$, $y(n)$ is a linear function of $\mathbf{g}(n)$, and for every fixed $\mathbf{g}(n)$, it is a linear function of $\mathbf{h}(n)$. In this context, the impulse responses $\mathbf{h}(n)$ and $\mathbf{g}(n)$ can be regarded as the temporal, respectively the spatial parts of the system, respectively.

We can rewrite the matrix $\mathbf{X}(n)$, of size $L \times M$, as a vector of length ML, by using the vectorization operation:

$$\mathrm{vec}[\mathbf{X}(n)] = [\mathbf{x}_1^T(n) \quad \mathbf{x}_2^T(n) \quad \dots \quad \mathbf{x}_M^T(n)]^T = \widetilde{\mathbf{x}}(n). \tag{2.2}$$

Therefore, the output signal can be expressed as

$$\begin{aligned} y(n) = \mathbf{h}^T(n)\mathbf{X}(n)\mathbf{g}(n) &= \mathrm{tr}\left[\left(\mathbf{h}(n)\mathbf{g}^T(n)\right)^T \mathbf{X}(n)\right] \\ &= \mathrm{vec}^T\left(\mathbf{h}(n)\mathbf{g}^T(n)\right)\mathrm{vec}[\mathbf{X}(n)] = [\mathbf{g}(n) \otimes \mathbf{h}(n)]^T \widetilde{\mathbf{x}}(n) \\ &= \mathbf{f}^T(n)\widetilde{\mathbf{x}}(n), \end{aligned} \tag{2.3}$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace of a matrix, $\otimes$ denotes the Kronecker product between the individual impulse responses, while the vector $\mathbf{f}(n) = \mathbf{g}(n) \otimes \mathbf{h}(n)$, of length ML, represents the spatiotemporal (i.e., global) impulse response of the system. Consequently, the signal model in (2.1) becomes

$$d(n) = [\mathbf{g}(n) \otimes \mathbf{h}(n)]^T \widetilde{\mathbf{x}}(n) + v(n) = \mathbf{f}^T(n)\widetilde{\mathbf{x}}(n) + v(n). \tag{2.4}$$

The major difference compared to a general MISO system is yielded by the fact that in this bilinear context $\mathbf{f}(n)$ is formed with only $M+L$ different elements, despite its length ML.

The goal is the identification of the two impulse responses $\mathbf{h}(n)$ and $\mathbf{g}(n)$, and, in this way, of the spatiotemporal impulse response $\mathbf{f}(n)$. For this aim, we can use two adaptive filters, $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$; hence, the global impulse response can be evaluated as

$$\widehat{\mathbf{f}}(n) = \widehat{\mathbf{g}}(n) \otimes \widehat{\mathbf{h}}(n). \tag{2.5}$$

Let $\eta \neq 0$ be a real-valued constant. We can see from (2.1) that

$$\left[\frac{1}{\eta}\mathbf{h}^T(n)\right] \mathbf{X}(n) \left[\eta\mathbf{g}(n)\right] = \mathbf{h}^T(n)\mathbf{X}(n)\mathbf{g}(n) = y(n), \tag{2.6}$$

meaning that the pairs $[\mathbf{h}(n)/\eta, \eta\mathbf{g}(n)]$ and $[\mathbf{h}(n), \mathbf{g}(n)]$ are equivalent in the bilinear form. This implies that we can only identify $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$ up to a scaling factor. A similar discussion can be found in [17, 37] in the framework of blind identification/equalization and nonlinear acoustic echo cancellation, respectively. Nevertheless, because

$$\mathbf{f}(n) = \mathbf{g}(n) \otimes \mathbf{h}(n) = [\eta\mathbf{g}(n)] \otimes \left[\frac{1}{\eta}\mathbf{h}(n)\right], \tag{2.7}$$

the global impulse response can be identified with no scaling ambiguity. Consequently, for the performance evaluation of the identification of the temporal and spatial filters, we can use the normalized projection misalignment (NPM), defined in [40]:

$$\text{NPM}\left[\mathbf{h}(n), \widehat{\mathbf{h}}(n)\right] = 1 - \left[\frac{\mathbf{h}^T(n)\widehat{\mathbf{h}}(n)}{\|\mathbf{h}(n)\|\|\widehat{\mathbf{h}}(n)\|}\right]^2, \tag{2.8}$$

$$\text{NPM}\left[\mathbf{g}(n), \widehat{\mathbf{g}}(n)\right] = 1 - \left[\frac{\mathbf{g}^T(n)\widehat{\mathbf{g}}(n)}{\|\mathbf{g}(n)\|\|\widehat{\mathbf{g}}(n)\|}\right]^2, \tag{2.9}$$

where $\| \cdot \|$ denotes the Euclidean norm. On the other hand, for the identification of the global filter, $\mathbf{f}(n)$, we should use the normalized misalignment (NM):

$$\text{NM}\left[\mathbf{f}(n), \widehat{\mathbf{f}}(n)\right] = \frac{\|\mathbf{f}(n) - \widehat{\mathbf{f}}(n)\|^2}{\|\mathbf{f}(n)\|^2}. \tag{2.10}$$

When studying the system identification problem in terms of the Wiener filter, the assumption is that the two impulse responses that need to be identified are time-invariant (which represents a basic assumption in this context). In practice, however, when approaching the system identification problem using adaptive algorithms, we must take into account the fact that the systems which need to be identified vary in time. Thus, we assume that $\mathbf{h}(n)$ and $\mathbf{g}(n)$ are zero-mean random vectors, following a simplified first-order Markov model, i.e.,

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{w}_{\mathbf{h}}(n), \tag{2.11}$$

$$\mathbf{g}(n) = \mathbf{g}(n-1) + \mathbf{w}_{\mathbf{g}}(n), \tag{2.12}$$

where $\mathbf{w}_{\mathbf{h}}(n)$ and $\mathbf{w}_{\mathbf{g}}(n)$ are zero-mean white Gaussian noise (WGN) vectors, with correlation matrices $\mathbf{R}_{\mathbf{w}_{\mathbf{h}}}(n) = \sigma^2_{w_{\mathbf{h}}}\mathbf{I}_L$ and $\mathbf{R}_{\mathbf{w}_{\mathbf{g}}}(n) = \sigma^2_{w_{\mathbf{g}}}\mathbf{I}_M$, respectively (with $\mathbf{I}_L$ and $\mathbf{I}_M$ being the identity matrices of sizes $L \times L$ and $M \times M$, respectively). We assume that $\mathbf{w}_{\mathbf{h}}(n)$ is uncorrelated with $\mathbf{h}(n-1)$ and $v(n)$, while $\mathbf{w}_{\mathbf{g}}(n)$ is uncorrelated with $\mathbf{g}(n-1)$ and $v(n)$. The variances $\sigma^2_{w_{\mathbf{h}}}$ and $\sigma^2_{w_{\mathbf{g}}}$ capture the nonstationarity in $\mathbf{h}(n)$ and $\mathbf{g}(n)$, respectively.

2.2 Wiener Filter for Bilinear Forms

The Wiener filter for BF was introduced in [34]. The short presentation of this algorithm which is made in this subchapter is inspired from [34]. This overview constitutes the basis for the subsequent developments (contributions of the present work) and is therefore required at this point.

As specified before, when deriving the Wiener filter, the usual assumption is that the impulse responses are time invariant. Consequently, given the two estimated filters $\widehat{\mathbf{h}}$ and $\widehat{\mathbf{g}}$, the estimated signal is given by:

$$\widehat{y}(n) = \widehat{\mathbf{h}}^T \mathbf{X}(n) \widehat{\mathbf{g}}. \tag{2.13}$$

As a result, the error signal between the desired and estimated signals is defined as

$$e(n) = d(n) - \widehat{y}(n) = d(n) - \widehat{\mathbf{h}}^T \mathbf{X}(n) \widehat{\mathbf{g}} = d(n) - \left[\widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}\right]^T \widetilde{\mathbf{x}}(n) = d(n) - \widehat{\mathbf{f}}^T \widetilde{\mathbf{x}}(n). \tag{2.14}$$

2.2.1 Direct Wiener Filter

The first step when computing the spatiotemporal filter $\widehat{\mathbf{f}} = \widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}$ is to minimize a cost function, which is based on the error signal in (2.14). The most suitable and popular choice for this criterion is the mean-squared error (MSE):

$$\begin{aligned} J\left(\widehat{\mathbf{h}}, \widehat{\mathbf{g}}\right) = \mathbb{E}\left[e^2(n)\right] &= \sigma_d^2 - 2\left(\widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}\right)^T \mathbf{p} + \left(\widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}\right)^T \mathbf{R} \left(\widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}}\right) \\ &= \sigma_d^2 - 2\widehat{\mathbf{f}}^T \mathbf{p} + \widehat{\mathbf{f}}^T \mathbf{R} \widehat{\mathbf{f}} = J\left(\widehat{\mathbf{f}}\right), \end{aligned} \tag{2.15}$$

where $\mathbb{E}[\cdot]$ denotes mathematical expectation, $\mathbf{p} = \mathbb{E}\left[\mathbf{x}(n) d(n)\right]$ (the cross-correlation vector between the input and the reference signals), and $\mathbf{R} = \mathbb{E}\left[\mathbf{x}(n)\mathbf{x}^T(n)\right]$ (the covariance matrix of the input signal). The quantity $\sigma_d^2 = \mathbb{E}\left[d^2(n)\right]$ is the variance of the reference signal, which can be computed based on (2.4) as:

$$\sigma_d^2 = \mathbb{E}\left[d^2(n)\right] = (\mathbf{g} \otimes \mathbf{h})^T \mathbf{R} (\mathbf{g} \otimes \mathbf{h}) + \sigma_v^2. \tag{2.16}$$

As a consequence, the signal-to-noise ratio (SNR) can be expressed as:

$$\text{SNR} = \frac{(\mathbf{g} \otimes \mathbf{h})^T \mathbf{R} (\mathbf{g} \otimes \mathbf{h})}{\sigma_v^2}. \tag{2.17}$$

The minimization of (2.15) leads to the well-known solution of the Wiener filter:

$$\widehat{\mathbf{f}} = \mathbf{R}^{-1}\mathbf{p}. \tag{2.18}$$

The estimated global impulse response can be rewritten as

$$\widehat{\mathbf{f}} = \widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}} = \begin{bmatrix} \widehat{\mathbf{h}}^T & \widehat{g}_2\widehat{\mathbf{h}}^T & \cdots & \widehat{g}_M\widehat{\mathbf{h}}^T \end{bmatrix}^T = \begin{bmatrix} \widehat{\mathbf{f}}_1^T & \widehat{\mathbf{f}}_2^T & \cdots & \widehat{\mathbf{f}}_M^T \end{bmatrix}^T, \tag{2.19}$$

where $\widehat{g}_m, m = 1, \cdots, M$ are the components of $\widehat{\mathbf{g}}$. Since (2.19) represents a system of ML equations with $M + L$ unknowns, there is no unique solution. Consequently, a scaling ambiguity arises, which we absorb in $\widehat{\mathbf{g}}$ by assuming (without loss of generality) that $\widehat{g}_1 = 1$. It is clear from (2.19) that the first L elements of $\widehat{\mathbf{f}}$ correspond to $\widehat{\mathbf{h}}$. In order to compute the elements $\widehat{g}_i, i = 2, \cdots, M$, we need to minimize $\|\widehat{\mathbf{f}}_i - \widehat{g}_i\widehat{\mathbf{h}}\|^2$. The following expression is obtained:

$$\widehat{g}_i = \frac{\widehat{\mathbf{f}}_i^T\widehat{\mathbf{h}}}{\widehat{\mathbf{h}}^T\widehat{\mathbf{h}}}. \tag{2.20}$$

The matrix $\mathbf{R}$, from relation (2.18), needs to be estimated. This matrix, of size $ML \times ML$, consists of M^2 submatrices of size $L \times L$ each ($\mathbf{R}_{ij} = E\left[\mathbf{x}_i(n)\mathbf{x}_j^T(n)\right]$, $i, j = 1, 2, \cdots, M$); each $\mathbf{R}_{ij}$ contains the rows from $(i-1)L+1$ to iL and the columns from $(j-1)L+1$ to jL of $\mathbf{R}$. $\mathbf{R}_{mm}$, $m = 1, 2, \cdots, M$ are the covariance matrices of $\mathbf{x}_m(n)$. It can be observed that $\mathbf{R}$ is a symmetric block Toeplitz matrix.

Because of the large size of the matrix $\mathbf{R}$, we need to have available a large amount of data in order to obtain a good estimate of it. Otherwise, the estimate of $\mathbf{R}$ will be very ill conditioned or will not even be full rank, leading to a very inaccurate value of $\widehat{\mathbf{f}}$. In this closed-form method, ML coefficients need to be identified, but we actually have only $M+L$ coefficients to identify. Moreover, in the case when there is a mismatch between the true model and the assumed one, this technique will likely yield bad results [34].

2.2.2 Iterative Wiener Filter

The second possibility is to derive an iterative version of the Wiener filter [34], which exploits the idea that only $M+L$ coefficients need to be identified. This procedure can give good results with a smaller amount of data as compared to the direct Wiener filter. This method is similar to the coordinate descent approach [41]. As mentioned at the beginning of this chapter, the following developments belong to [34], and we try to present them as briefly as possible. The following relations are used in the developments [42]:

$$\widehat{\mathbf{g}} \otimes \widehat{\mathbf{h}} = \left(\widehat{\mathbf{g}} \otimes \mathbf{I}_L\right) \widehat{\mathbf{h}} \tag{2.21}$$

$$= \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}\right) \widehat{\mathbf{g}}, \tag{2.22}$$

where $\mathbf{I}_L$ and $\mathbf{I}_M$ are the identity matrices having sizes $L \times L$ and $M \times M$, respectively. Using relations (2.21) and (2.22), we can express the MSE criterion as

$$J\left(\widehat{\mathbf{h}}, \widehat{\mathbf{g}}\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}^T \mathbf{p}_{\widehat{\mathbf{g}}} + \widehat{\mathbf{h}}^T \mathbf{R}_{\widehat{\mathbf{g}}} \widehat{\mathbf{h}} \tag{2.23}$$

$$= \sigma_d^2 - 2\widehat{\mathbf{g}}^T \mathbf{p}_{\widehat{\mathbf{h}}} + \widehat{\mathbf{g}}^T \mathbf{R}_{\widehat{\mathbf{h}}} \widehat{\mathbf{g}}, \tag{2.24}$$

where

$$\mathbf{p}_{\widehat{\mathbf{g}}} = \left(\widehat{\mathbf{g}} \otimes \mathbf{I}_L\right)^T \mathbf{p}, \tag{2.25}$$

$$\mathbf{R}_{\widehat{\mathbf{g}}} = \left(\widehat{\mathbf{g}} \otimes \mathbf{I}_L\right)^T \mathbf{R} \left(\widehat{\mathbf{g}} \otimes \mathbf{I}_L\right), \tag{2.26}$$

and

$$\mathbf{p}_{\widehat{\mathbf{h}}} = \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}\right)^T \mathbf{p}, \tag{2.27}$$

$$\mathbf{R}_{\widehat{\mathbf{h}}} = \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}\right)^T \mathbf{R} \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}\right). \tag{2.28}$$

In the case when $\widehat{\mathbf{g}}$ is fixed, we express (2.23) as

$$J_{\widehat{\mathbf{g}}}\left(\widehat{\mathbf{h}}\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}^T \mathbf{p}_{\widehat{\mathbf{g}}} + \widehat{\mathbf{h}}^T \mathbf{R}_{\widehat{\mathbf{g}}} \widehat{\mathbf{h}}, \tag{2.29}$$

and when $\widehat{\mathbf{h}}$ is fixed, we write (2.24) as

$$J_{\widehat{\mathbf{h}}}\left(\widehat{\mathbf{g}}\right) = \sigma_d^2 - 2\widehat{\mathbf{g}}^T \mathbf{p}_{\widehat{\mathbf{h}}} + \widehat{\mathbf{g}}^T \mathbf{R}_{\widehat{\mathbf{h}}} \widehat{\mathbf{g}}.[34] \tag{2.30}$$

In the following, the iterative algorithm is derived. First, at iteration 0, we can initialize $\widehat{\mathbf{g}}^{(0)} = (1/M) \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$. By substituting $\widehat{\mathbf{g}}^{(0)}$ in (2.25)–(2.26), we obtain

$$\mathbf{p}_{\widehat{\mathbf{g}}}^{(0)} = \left(\widehat{\mathbf{g}}^{(0)} \otimes \mathbf{I}_L\right)^T \mathbf{p},$$

$$\mathbf{R}_{\widehat{\mathbf{g}}}^{(0)} = \left(\widehat{\mathbf{g}}^{(0)} \otimes \mathbf{I}_L\right)^T \mathbf{R} \left(\widehat{\mathbf{g}}^{(0)} \otimes \mathbf{I}_L\right).$$

Replacing these quantities into the MSE in (2.29), we get at iteration 1:

$$J_{\widehat{\mathbf{g}}}\left(\widehat{\mathbf{h}}^{(1)}\right) = \sigma_d^2 - 2\left(\widehat{\mathbf{h}}^{(1)}\right)^T \mathbf{p}_{\widehat{\mathbf{g}}}^{(0)} + \left(\widehat{\mathbf{h}}^{(1)}\right)^T \mathbf{R}_{\widehat{\mathbf{g}}}^{(0)} \widehat{\mathbf{h}}^{(1)},$$

which can be minimized with respect to $\widehat{\mathbf{h}}^{(1)}$ and it yields

$$\widehat{\mathbf{h}}^{(1)} = \left(\mathbf{R}_{\widehat{\mathbf{g}}}^{(0)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{g}}}^{(0)}.$$

Using $\widehat{\mathbf{h}}^{(1)}$ in (2.27)–(2.28), we get

$$\begin{aligned}\mathbf{p}_{\widehat{\mathbf{h}}}^{(1)} &= \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(1)}\right)^T \mathbf{p},\\ \mathbf{R}_{\widehat{\mathbf{h}}}^{(1)} &= \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(1)}\right)^T \mathbf{R} \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(1)}\right),\end{aligned}$$

and the corresponding minimum MSE (MMSE) is

$$J_{\widehat{\mathbf{g}}}\left(\widehat{\mathbf{h}}^{(1)}\right) = \sigma_d^2 - \left(\mathbf{p}_{\widehat{\mathbf{g}}}^{(0)}\right)^T \left(\mathbf{R}_{\widehat{\mathbf{g}}}^{(0)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{g}}}^{(0)}.$$

Using $\mathbf{p}_{\widehat{\mathbf{h}}}^{(1)}$ and $\mathbf{R}_{\widehat{\mathbf{h}}}^{(1)}$, we can calculate the MSE in (2.30) as

$$J_{\widehat{\mathbf{h}}}\left(\widehat{\mathbf{g}}^{(1)}\right) = \sigma_d^2 - 2\left(\widehat{\mathbf{g}}^{(1)}\right)^T \mathbf{p}_{\widehat{\mathbf{h}}}^{(1)} + \left(\widehat{\mathbf{g}}^{(1)}\right)^T \mathbf{R}_{\widehat{\mathbf{h}}}^{(1)} \widehat{\mathbf{g}}^{(1)}.$$

By minimizing the previous expression with respect to $\widehat{\mathbf{g}}^{(1)}$, we obtain

$$\widehat{\mathbf{g}}^{(1)} = \left(\mathbf{R}_{\widehat{\mathbf{h}}}^{(1)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{h}}}^{(1)},$$

with the corresponding MMSE:

$$J_{\widehat{\mathbf{h}}}\left(\widehat{\mathbf{g}}^{(1)}\right) = \sigma_d^2 - \left(\mathbf{p}_{\widehat{\mathbf{h}}}^{(1)}\right)^T \left(\mathbf{R}_{\widehat{\mathbf{h}}}^{(1)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{h}}}^{(1)}.$$

By further iterating up to n, we obtain the temporal impulse response's estimate:

$$\widehat{\mathbf{h}}^{(n)} = \left(\mathbf{R}_{\widehat{\mathbf{g}}}^{(n-1)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{g}}}^{(n-1)}, \tag{2.31}$$

where

$$\mathbf{R}_{\widehat{\mathbf{g}}}^{(n-1)} = \left(\widehat{\mathbf{g}}^{(n-1)} \otimes \mathbf{I}_L\right)^T \mathbf{R} \left(\widehat{\mathbf{g}}^{(n-1)} \otimes \mathbf{I}_L\right), \tag{2.32}$$

$$\mathbf{p}_{\widehat{\mathbf{g}}}^{(n-1)} = \left(\widehat{\mathbf{g}}^{(n-1)} \otimes \mathbf{I}_L\right)^T \mathbf{p}, \tag{2.33}$$

and the spatial impulse response's estimate:

$$\widehat{\mathbf{g}}^{(n)} = \left(\mathbf{R}_{\widehat{\mathbf{h}}}^{(n)}\right)^{-1} \mathbf{p}_{\widehat{\mathbf{h}}}^{(n)}, \tag{2.34}$$

where

$$\mathbf{R}_{\widehat{\mathbf{h}}}^{(n)} = \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(n)}\right)^T \mathbf{R} \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(n)}\right), \tag{2.35}$$

$$\mathbf{p}_{\widehat{\mathbf{h}}}^{(n)} = \left(\mathbf{I}_M \otimes \widehat{\mathbf{h}}^{(n)}\right)^T \mathbf{p}. \tag{2.36}$$

Finally, the Wiener global filter at iteration n is

$$\widehat{\widehat{\mathbf{f}}}^{(n)} = \widehat{\mathbf{g}}^{(n)} \otimes \widehat{\mathbf{h}}^{(n)}, \tag{2.37}$$

where $\widehat{\mathbf{g}}^{(n)}$ and $\widehat{\mathbf{h}}^{(n)}$ are given by (2.34) and (2.31), respectively. This approach is called the iterative Wiener filter [34]. The convergence proof for the iterative Wiener filter in a similar scenario can be found in [43]. A very similar iterative method was also studied in [44]. The decrease of the NM below a certain established threshold value can be set as a condition for stopping the iterative process.

2.3 Least-Mean-Square Type Adaptive Algorithms for the Identification of Bilinear Forms

In this part, we start by shortly introducing the LMS adaptive algorithm tailored for the special case of BF, namely LMS-BF, together with its normalized version (NLMS-BF), which were derived in [35]. Although they are not contributions of this work, a brief overview of these algorithms is required in order to provide a better understanding of the subsequent parts (especially the OLMS-BF algorithm, which constitutes the object of subchapter 2.5).

2.3.1 Least-Mean-Square Algorithm for Bilinear Forms (LMS-BF)

This part was presented in [35]. In the previous subchapter, the system identification problem was treated based on the Wiener filter, and two types of the Wiener filter were derived: the direct form and the iterative form. It was seen that the iterative solution is much better, since it leads to good estimates of the impulse responses, even with a small amount of data available for the statistics estimation. Despite this, for some practical cases, such as real-time applications, for example, the iterative Wiener filter presented in [34] may not be very convenient, because of the well-known limitations of the Wiener filter (matrix inversion operations, estimation of the statistics, etc). Therefore, we may treat the identification problem in terms of adaptive filtering. The LMS algorithm solution is shortly described in the following.

First, we consider the two adaptive filters $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$, and the corresponding estimated signal:

$$\widehat{y}(n) = \widehat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\widehat{\mathbf{g}}(n-1). \tag{2.38}$$

Consequently, we may define the error signal between the desired and estimated signals as

$$\begin{aligned} e(n) &= d(n) - \widehat{y}(n) = d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\widehat{\mathbf{g}}(n-1) \\ &= d(n) - \Big[\widehat{\mathbf{g}}(n-1) \otimes \widehat{\mathbf{h}}(n-1)\Big]^T \widetilde{\mathbf{x}}(n) = d(n) - \widehat{\mathbf{f}}^T(n-1)\widetilde{\mathbf{x}}(n). \end{aligned} \tag{2.39}$$

We may define it alternatively as

$$e_{\widehat{\mathbf{g}}}(n) = d(n) - \widehat{\mathbf{h}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n), \tag{2.40}$$

$$e_{\widehat{\mathbf{h}}}(n) = d(n) - \widehat{\mathbf{g}}^T(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n), \tag{2.41}$$

with

$$\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) = \left[\widehat{\mathbf{g}}(n-1) \otimes \mathbf{I}_L\right]^T \widetilde{\mathbf{x}}(n), \tag{2.42}$$

$$\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) = \Big[\mathbf{I}_M \otimes \widehat{\mathbf{h}}(n-1)\Big]^T \widetilde{\mathbf{x}}(n), \tag{2.43}$$

where $\mathbf{I}_L$ and $\mathbf{I}_M$ are the identity matrices of sizes $L \times L$ and $M \times M$, respectively. It can easily be checked that $e_{\widehat{\mathbf{g}}}(n) = e_{\widehat{\mathbf{h}}}(n) = e(n)$. However, the notation from (2.40) and (2.41) shall be kept for the clarity of the developments.

In this context, the LMS algorithm for BF (namely LMS-BF) can be written as

$$\begin{aligned} \widehat{\mathbf{h}}(n) &= \widehat{\mathbf{h}}(n-1) - \frac{\mu_{\widehat{\mathbf{h}}}}{2} \times \frac{\partial e^2_{\widehat{\mathbf{g}}}(n)}{\partial \widehat{\mathbf{h}}(n-1)} \\ &= \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}}} \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) e_{\widehat{\mathbf{g}}}(n) \end{aligned} \tag{2.44}$$

and

$$\begin{aligned} \widehat{\mathbf{g}}(n) &= \widehat{\mathbf{g}}(n-1) - \frac{\mu_{\widehat{\mathbf{g}}}}{2} \times \frac{\partial e^2_{\widehat{\mathbf{h}}}(n)}{\partial \widehat{\mathbf{g}}(n-1)} \\ &= \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}}} \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) e_{\widehat{\mathbf{h}}}(n), \end{aligned} \tag{2.45}$$

where $\mu_{\widehat{\mathbf{h}}} > 0$ and $\mu_{\widehat{\mathbf{g}}} > 0$ are the step-size parameters. As initialization, we can choose [35]

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

2.3.2 Normalized LMS Algorithm for Bilinear Forms (NLMS-BF) and the Improved Proportionate NLMS-BF (IPNLMS-BF)

The step-size parameters from relations (2.44) and (2.45) are constant values, chosen such that they ensure the stability of the algorithm (for more details, see [35]). However, when dealing with nonstationary signals, it may be more advantageous to use an adaptive, time-dependent step-size parameter. In this context, a first idea is to reformulate the solution in terms of the normalized NLMS algorithm. Consequently, the update relations become

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n), \tag{2.46}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n), \tag{2.47}$$

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

A good way of computing the step-size parameters, taking into consideration the stability conditions, is by cancelling the a posteriori error signals [45]. In our case, we can define

$$\varepsilon_{\widehat{\mathbf{g}}}(n) = d(n) - \widehat{\mathbf{h}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n), \tag{2.48}$$

$$\varepsilon_{\widehat{\mathbf{h}}}(n) = d(n) - \widehat{\mathbf{g}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n). \tag{2.49}$$

Substituting (2.46) into (2.48), and (2.47) into (2.49), with the conditions $\varepsilon_{\widehat{\mathbf{g}}}(n) = 0$ and $\varepsilon_{\widehat{\mathbf{h}}}(n) = 0$, respectively, we get

$$e_{\widehat{\mathbf{g}}}(n)\left[1 - \mu_{\widehat{\mathbf{h}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)\right] = 0, \tag{2.50}$$

$$e_{\widehat{\mathbf{h}}}(n)\left[1 - \mu_{\widehat{\mathbf{g}}}(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)\right] = 0. \tag{2.51}$$

Next, considering that the a priori error is not null, we obtain that $\mu_{\widehat{\mathbf{h}}}(n) = 1/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)]$ and $\mu_{\widehat{\mathbf{g}}}(n) = 1/[\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)]$, which are the theoretical step-size parameters of the NLMS algorithm for BF, namely NLMS-BF.

In practice, in order to achieve a good compromise between convergence rate and misadjustment [1], we may employ two positive constants $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$, usually smaller than 1, which multiply these step-sizes. These constants are called the normalized step-sizes. The NLMS-BF algorithm also needs to be regularized by adding two positive constants, $\delta_{\widehat{\mathbf{h}}}$ and $\delta_{\widehat{\mathbf{g}}}$, to the denominators of the step-sizes. These regularization parameters are usually chosen to be proportional to the variance of the input signal [46]. Consequently, the NLMS-BF algorithm is described by the updates:

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\alpha_{\widehat{\mathbf{h}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) + \delta_{\widehat{\mathbf{h}}}}, \tag{2.52}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \frac{\alpha_{\widehat{\mathbf{g}}}\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) + \delta_{\widehat{\mathbf{g}}}}, \tag{2.53}$$

whereas the global filter can be obtained using the Kronecker product, i.e., $\widehat{\mathbf{f}}(n) = \widehat{\mathbf{g}}(n) \otimes \widehat{\mathbf{h}}(n)$ [35].

The following developments of this subchapter were published before in [23].

In many system identification problems, the systems to be identified are sparse in nature, i.e., only a small percentage of the impulse response components have a significant magnitude, while the rest are zero or small. Consequently, sparse adaptive filters are frequently involved in such system identification problems. In this framework, most of these algorithms were developed in the context of echo cancellation [47, 15]. Usually, they are referred to as proportionate algorithms [48, 49, 50, 51]. The basic idea is to "proportionately" redistribute the adaptation gain among all the coefficients, emphasizing the large ones in order to speed up their convergence and, consequently, to increase the overall convergence rate.

Starting from the the basic idea of proportionate-type algorithms, relation (2.52) can be reformulated as

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\alpha_{\widehat{\mathbf{h}}}\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n)e_{\widehat{\mathbf{g}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}^T(n)\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) + \widetilde{\delta}_{\widehat{\mathbf{h}}}}, \tag{2.54}$$

where

$$\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1) = \text{diag}\left[\begin{array}{ccc} q_{\widehat{\mathbf{h}},1}(n-1) & \cdots & q_{\widehat{\mathbf{h}},L}(n-1) \end{array}\right] \tag{2.55}$$

denotes an $L \times L$ diagonal matrix, containing the proportionate factors $q_{\widehat{\mathbf{h}},l}(n-1) \geq 0$ $(l = 1, 2, \ldots, L)$ which depend on the coefficients of $\widehat{\mathbf{h}}(n-1)$, and $\widetilde{\delta}_{\widehat{\mathbf{h}}} = \delta_{\widehat{\mathbf{h}}}/L$ denotes the regularization constant [46]. In a similar way, a proportionate term can also be introduced in (2.53), resulting in

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \frac{\alpha_{\widehat{\mathbf{g}}}\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n)e_{\widehat{\mathbf{h}}}(n)}{\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}^T(n)\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1)\widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) + \widetilde{\delta}_{\widehat{\mathbf{g}}}}, \tag{2.56}$$

where

$$\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1) = \text{diag}\left[\begin{array}{ccc} q_{\widehat{\mathbf{g}},1}(n-1) & \cdots & q_{\widehat{\mathbf{g}},M}(n-1) \end{array}\right] \tag{2.57}$$

denotes an $M \times M$ diagonal matrix, containing the proportionate factors $q_{\widehat{\mathbf{g}},m}(n-1) \geq 0$ $(m = 1, 2, \ldots, M)$ related to the coefficients of $\widehat{\mathbf{g}}(n-1)$; also, $\widetilde{\delta}_{\widehat{\mathbf{g}}} = \delta_{\widehat{\mathbf{g}}}/M$ denotes a regularization constant [46].

For the identification of sparse impulse responses, the improved proportionate NLMS (IPNLMS) algorithm proposed in [48] represents one of the most reliable choices. This algorithm uses the ℓ_1 norm to exploit the sparsity of the impulse response that we need to identify. Next, by following the line of the IPNLMS algorithm [48], we obtain that

$$q_{\widehat{\mathbf{h}},l}(n-1) = \frac{1-\kappa_{\widehat{\mathbf{h}}}}{2L} + (1+\kappa_{\widehat{\mathbf{h}}})\frac{\left|\widehat{h}_l(n-1)\right|}{2\left\|\widehat{\mathbf{h}}(n-1)\right\|_1},\ 1 \leq l \leq L, \tag{2.58}$$

$$q_{\widehat{\mathbf{g}},m}(n-1) = \frac{1-\kappa_{\widehat{\mathbf{g}}}}{2M} + (1+\kappa_{\widehat{\mathbf{g}}})\frac{\left|\widehat{g}_m(n-1)\right|}{2\left\|\widehat{\mathbf{g}}(n-1)\right\|_1},\ 1 \leq m \leq M, \tag{2.59}$$

where $\kappa_{\widehat{\mathbf{h}}}$ $(-1 \leq \kappa_{\widehat{\mathbf{h}}} < 1)$ and $\kappa_{\widehat{\mathbf{g}}}$ $(-1 \leq \kappa_{\widehat{\mathbf{g}}} < 1)$ are parameters that can control the amount of proportionality, while $\left\|\cdot\right\|_1$ is the ℓ_1 norm. Using (2.54) and (2.56) instead of (2.52) and (2.53), respectively, the IPNLMS algorithm tailored for the case of BF, namely the IPNLMS-BF [52] results.

Finally, it can be mentioned that the regular IPNLMS algorithm [48] could also be used for the identification of the global system impulse response, $\mathbf{f}$. This algorithm is obtained based on the reference signal expressed as in (2.4), together with the error signal defined in the last part of expression (2.39), and including in its update the $ML \times ML$ proportionate matrix

$$\mathbf{Q}_{\widehat{\mathbf{f}}}(n-1) = \text{diag}\left[\begin{array}{ccc} q_{\widehat{\mathbf{f}},1}(n-1) & \cdots & q_{\widehat{\mathbf{f}},ML}(n-1) \end{array}\right], \tag{2.60}$$

which is a diagonal matrix containing the proportionate factors $q_{\widehat{\mathbf{f}},k}(n-1) \geq 0$ $(k = 1, 2, \ldots, ML)$ depending on the coefficients of $\widehat{\mathbf{f}}(n-1)$. These can be evaluated as

$$q_{\widehat{\mathbf{f}},k}(n-1) = \frac{1-\kappa}{2ML} + (1+\kappa)\frac{\left|\widehat{f}_k(n-1)\right|}{2\left\|\widehat{\mathbf{f}}(n-1)\right\|_1},\ 1 \leq k \leq ML, \tag{2.61}$$

where κ $(-1 \leq \kappa < 1)$ controls the amount of proportionality.

However, another observation should be made, namely that the solution obtained with the regular IPNLMS algorithm contains an adaptive filter of length ML, whereas the IPNLMS-BF algorithm [using (2.54) and (2.56)] involves two short filters having lengths L and M, respectively. Hence, the IPNLMS-BF can attain a faster convergence speed in comparison to the IPNLMS algorithm [52].

2.4 Improved Proportionate Affine Projection Algorithm for Bilinear Forms (IPAPA-BF)

The publication which contains the developments from this subchapter is [23].

In this chapter, we extend the approach introduced in the previous part to the APA [53] and develop a proportionate version tailored for the identification of BF. The main advantage of the APA over the NLMS algorithm consists of a superior convergence rate, especially for correlated inputs. Simulation results indicate that the proposed proportionate APA for BF outperforms the IPNLMS-BF algorithm in terms of convergence rate.

Following the development of the NLMS-BF algorithm which was briefly overviewed in subchapter 2.3.2, it is straightforward to derive the APA in the bilinear context. First, let us introduce the notation:

$$\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}(n) = \begin{bmatrix} \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n) & \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n-1) & \cdots & \widetilde{\mathbf{x}}_{\widehat{\mathbf{g}}}(n-P+1) \end{bmatrix}, \tag{2.62}$$

$$\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}(n) = \begin{bmatrix} \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n) & \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n-1) & \cdots & \widetilde{\mathbf{x}}_{\widehat{\mathbf{h}}}(n-P+1) \end{bmatrix}, \tag{2.63}$$

$$\mathbf{d}(n) = \begin{bmatrix} d(n) & d(n-1) & \cdots & d(n-P+1) \end{bmatrix}^T, \tag{2.64}$$

where P denotes the projection order. Next, we define the error vectors:

$$\mathbf{e}_{\widehat{\mathbf{g}}}(n) = \mathbf{d}(n) - \widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}^{T}(n)\widehat{\mathbf{h}}(n-1), \tag{2.65}$$

$$\mathbf{e}_{\widehat{\mathbf{h}}}(n) = \mathbf{d}(n) - \widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}^{T}(n)\widehat{\mathbf{g}}(n-1), \tag{2.66}$$

so that the updates of the APA for BF (namely APA-BF) result in

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \alpha_{\widehat{\mathbf{h}}}\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}(n)\left[\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}^{T}(n)\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}(n) + \delta_{\widehat{\mathbf{h}}}\mathbf{I}_P\right]^{-1}\mathbf{e}_{\widehat{\mathbf{g}}}(n), \tag{2.67}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \alpha_{\widehat{\mathbf{g}}}\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}(n)\left[\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}^{T}(n)\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}(n) + \delta_{\widehat{\mathbf{g}}}\mathbf{I}_P\right]^{-1}\mathbf{e}_{\widehat{\mathbf{h}}}(n), \tag{2.68}$$

where $\mathbf{I}_P$ is the identity matrix of size $P \times P$. Clearly, for $P = 1$, the NLMS-BF algorithm [35] is obtained.

The proportionate approach can also be applied to the APA-BF. Following the development of the IPNLMS-BF algorithm and taking (2.67)–(2.68) into account, the updates of the improved proportionate APA for BF (namely IPAPA-BF) result in

$$\begin{aligned}\widehat{\mathbf{h}}(n) &= \widehat{\mathbf{h}}(n-1) + \alpha_{\widehat{\mathbf{h}}}\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1)\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}(n)\\ &\quad\times\left[\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}^{T}(n)\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1)\widetilde{\mathbf{X}}_{\widehat{\mathbf{g}}}(n) + \widetilde{\delta}_{\widehat{\mathbf{h}}}\mathbf{I}_P\right]^{-1}\mathbf{e}_{\widehat{\mathbf{g}}}(n), \qquad (2.69)\\ \widehat{\mathbf{g}}(n) &= \widehat{\mathbf{g}}(n-1) + \alpha_{\widehat{\mathbf{g}}}\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1)\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}(n)\\ &\quad\times\left[\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}^{T}(n)\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1)\widetilde{\mathbf{X}}_{\widehat{\mathbf{h}}}(n) + \widetilde{\delta}_{\widehat{\mathbf{g}}}\mathbf{I}_P\right]^{-1}\mathbf{e}_{\widehat{\mathbf{h}}}(n), \qquad (2.70)\end{aligned}$$

where $\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1)$ and $\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1)$ are defined in (2.55) and (2.57), respectively, using the proportionate factors from (2.58)–(2.59). The regularization parameters $\widetilde{\delta}_{\widehat{\mathbf{h}}}$ and $\widetilde{\delta}_{\widehat{\mathbf{g}}}$ are chosen as in the case of the IPNLMS-BF algorithm. As we can notice from (2.69)–(2.70), the IPNLMS-BF algorithm is obtained for $P = 1$, while the APA-BF results if $\mathbf{Q}_{\widehat{\mathbf{h}}}(n-1) = \mathbf{I}_L$ and $\mathbf{Q}_{\widehat{\mathbf{g}}}(n-1) = \mathbf{I}_M$.

Similar to the discussion from the end of 2.3.2, we should outline that the regular APA [53] and its proportionate version IPAPA [54] could also be used to identify the global impulse response, $\mathbf{f}$. Based on the desired signal expressed in (2.4), the update of the conventional IPAPA is given by

$$\widehat{\mathbf{f}}(n) = \widehat{\mathbf{f}}(n-1) + \alpha \mathbf{Q}_{\widehat{\mathbf{f}}}(n-1)\widetilde{\mathbf{X}}(n) \left[\widetilde{\mathbf{X}}^T(n)\mathbf{Q}_{\widehat{\mathbf{f}}}(n-1)\widetilde{\mathbf{X}}(n) + \delta\right]^{-1} \mathbf{e}(n), \qquad (2.71)$$

where the proportionate matrix $\mathbf{Q}_{\widehat{\mathbf{f}}}(n-1)$ is obtained based on (2.60)–(2.61), the data matrix $\widetilde{\mathbf{X}}(n)$ of size $ML \times P$ is defined as $\widetilde{\mathbf{X}}(n) = \left[\begin{array}{cccc} \widetilde{\mathbf{x}}(n) & \widetilde{\mathbf{x}}(n-1) & \cdots & \widetilde{\mathbf{x}}(n-P+1) \end{array}\right]$, the error vector results in $\mathbf{e}(n) = \mathbf{d}(n) - \widetilde{\mathbf{X}}^T(n)\widehat{\mathbf{f}}(n-1)$, while α and δ denote the normalized step-size and the regularization parameter, respectively. The update of the regular APA results from (2.71) when $\mathbf{Q}_{\widehat{\mathbf{f}}}(n-1) = \mathbf{I}_{ML}$, where $\mathbf{I}_{ML}$ is the identity matrix of size $ML \times ML$.

However, the solutions based on the conventional APA and IPAPA involve long adaptive filters of length ML, while the APA-BF and IPAPA-BF use two shorter filters of lengths L and M. Therefore, the APA-BF and IPAPA-BF should own a faster convergence rate as compared to the regular APA and IPAPA, respectively.

2.4.1 Experimental Results

Simulations are performed in the framework of the MISO system described before, from a system identification perspective. The temporal impulse response $\mathbf{h}$ is the first impulse response from G168 Recommendation [55], padded with zeros until the length $L = 512$. The spatial impulse response $\mathbf{g}$ is obtained by generating an exponential decay, having coefficients $g_m = 0.5^m$, with $m = 1, 2, \ldots, M$; in all experiments, we choose $M = 4$. Then, both the temporal and the spatial impulse responses are normalized, such that $\|\mathbf{h}\| = \|\mathbf{g}\| = 1$. The spatiotemporal system impulse response is obtained as $\mathbf{f} = \mathbf{g} \otimes \mathbf{h}$, with the length $ML = 2048$.

The input signals $x_m(n),\ m = 1, 2, \ldots, M$ are generated either as WGNs or as autoregressive signals of first order [AR(1)] processes; each AR(1) process is obtained by filtering a WGN through a first-order system $1/\left(1 - 0.8z^{-1}\right)$. The noise $v(n)$ is

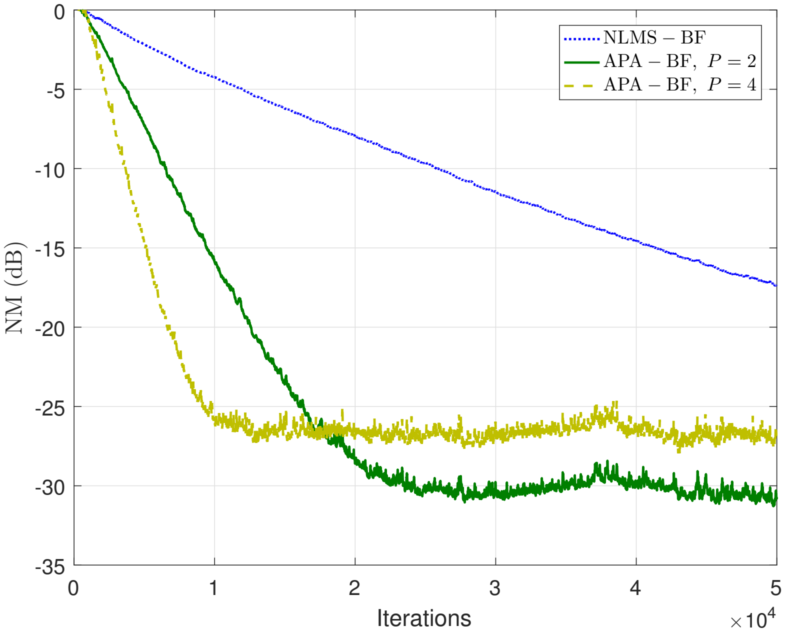


Figure 2.1: Performance of the NLMS-BF and APA-BF in terms of NM. The input signals are AR(1) processes and $ML = 2048$.

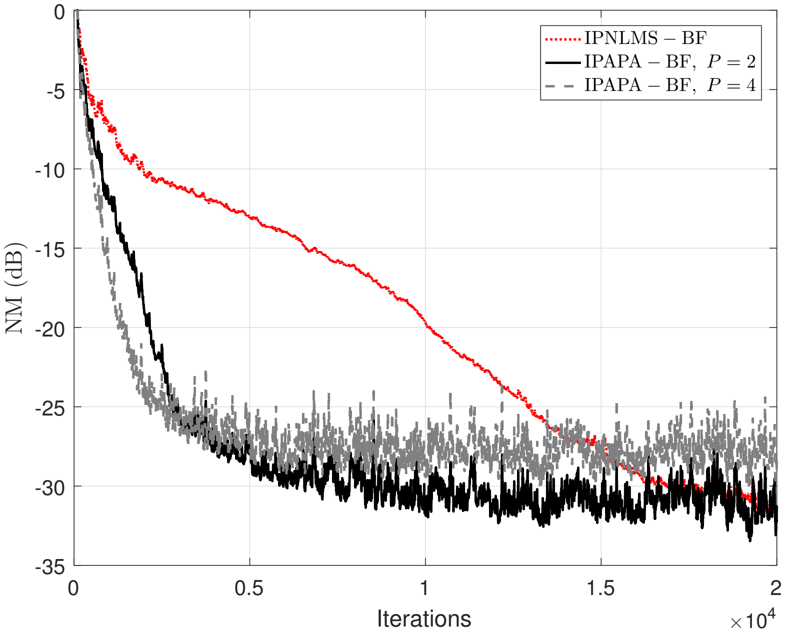


Figure 2.2: Performance of the IPNLMS-BF and IPAPA-BF in terms of NM. The input signals are AR(1) processes and $ML = 2048$.

an additive WGN, having the variance $\sigma_v^2 = 0.01$. The measure of the performance is the NM from (2.10), in dB, which evaluates the identification of the global impulse response $\mathbf{f}$.

First, the performance of the APA-BF is investigated. As compared to the NLMS-BF algorithm, the APA-BF should provide an improved convergence rate, especially for correlated inputs. This aspect is supported in Fig. 2.1, where the input signals are AR(1) processes, the step-sizes are set to $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.1$, and the regularization parameters are $\delta_{\widehat{\mathbf{h}}} = \delta_{\widehat{\mathbf{g}}} = 20\sigma_{\widetilde{x}}^2$, where $\sigma_{\widetilde{x}}^2$ is the variance of $\widetilde{\mathbf{x}}(n)$. The APA-BF uses different values of the projection order, i.e., $P = 2$ and 4. As shown in Section 2.4, the NLMS-BF algorithm is equivalent to APA-BF using $P = 1$. As we can notice in these figures, the APA-BF clearly outperforms the NLMS-BF algorithm in terms of convergence rate. The gain is significant for the APA-BF using $P = 2$, as compared to the NLMS-BF algorithm. Even if the APA-BF using $P = 4$ leads to a faster convergence rate, it also pays with an increased misalignment level. Due to this reason (and also considering the complexity issues), the APA-BF using $P = 2$ could be preferable in practice, since it achieves a proper compromise between the convergence rate and misalignment.

The same conclusions are valid in case of the IPAPA-BF as compared to the IPNLMS-BF algorithm. The results provided in Fig. 2.2 support this aspect. The step-sizes are the same as in the previous experiment, but the regularization terms are set to $\widetilde{\delta}_{\widehat{\mathbf{h}}} = 20\sigma_{\widetilde{x}}^2/L$ and $\widetilde{\delta}_{\widehat{\mathbf{g}}} = 20\sigma_{\widetilde{x}}^2/M$, and the specific parameters are $\kappa_{\widehat{\mathbf{h}}} = \kappa_{\widehat{\mathbf{g}}} = 0$. Again, the projection order $P = 2$ leads to the best compromise between the performance criteria.

In Fig. 2.3, the performance of the regular APA is compared to its counterparts tailored for BF, i.e., the APA-BF and IPAPA-BF. The input signals are WGNs and the projection order is $P = 2$ for all the algorithms. First, we can notice that the APA-BF outperforms the conventional APA in terms of convergence rate (as outlined

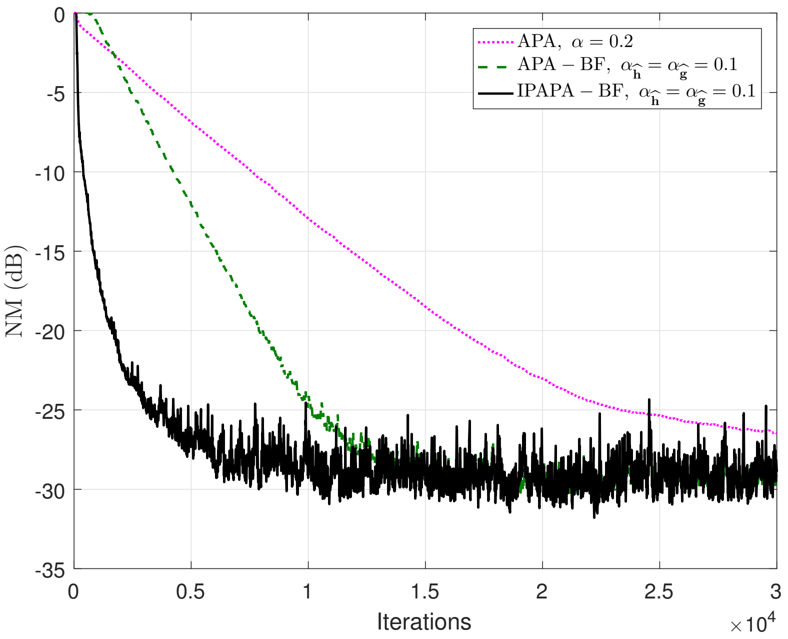


Figure 2.3: Performance of the APA, APA-BF, and IPAPA-BF in terms of NM. The input signals are white Gaussian noises and $ML = 2048$.

in Section 2.4). Second, the IPAPA-BF converges faster as compared to the APA-BF in case of sparse impulse responses.

Finally, the performance of the conventional IPAPA and proposed IPAPA-BF are investigated in Fig. 2.4. The input signals are AR(1) processes, the projection order is set to $P = 2$, and different values of the step-sizes are used. The regular IPAPA uses $\delta = 20\sigma_{\widetilde{x}}^2/(ML)$ and $\kappa = 0$. As we can notice, the IPAPA using $\alpha = 1$ achieves the fastest converge rate but a higher misalignment level. The IPAPA with $\alpha = 0.2$ improves the misalignment level, paying with a slightly slower converge rate. However, the IPAPA-BF using smaller step-sizes outperforms the conventional algorithm, achieving a convergence rate similar to the IPAPA with $\alpha = 1$, but reaching a much lower misalignment level. This performance gain supports the discussion from the end of Section 2.4.

Despite being intended for sparse impulse responses, the IPAPA-BF offers a good performance also for less sparse echo paths. The different sparseness degrees can be

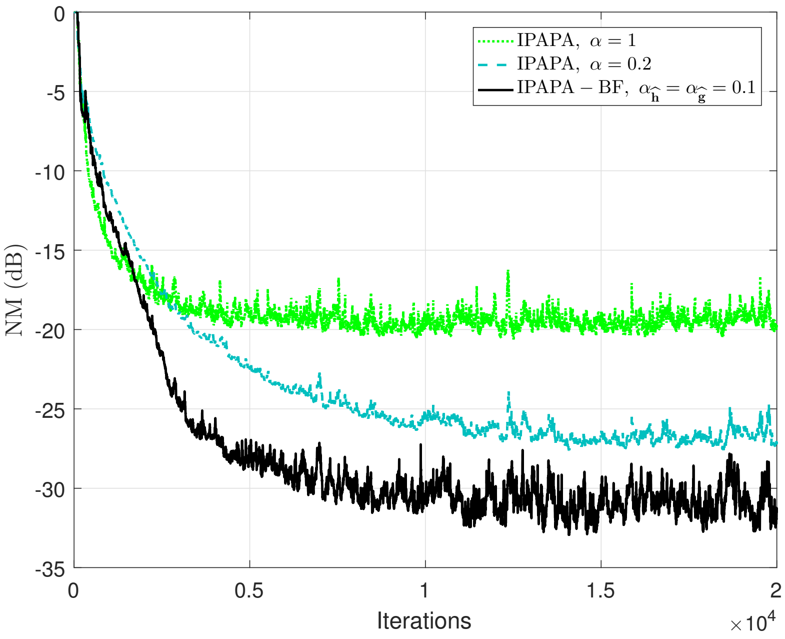


Figure 2.4: Performance of the IPAPA and IPAPA-BF in terms of NM for different values of the normalized step-size parameters α, $\alpha_{\widehat{\mathbf{h}}}$, and $\alpha_{\widehat{\mathbf{g}}}$. The input signals are AR(1) processes and $ML = 2048$.

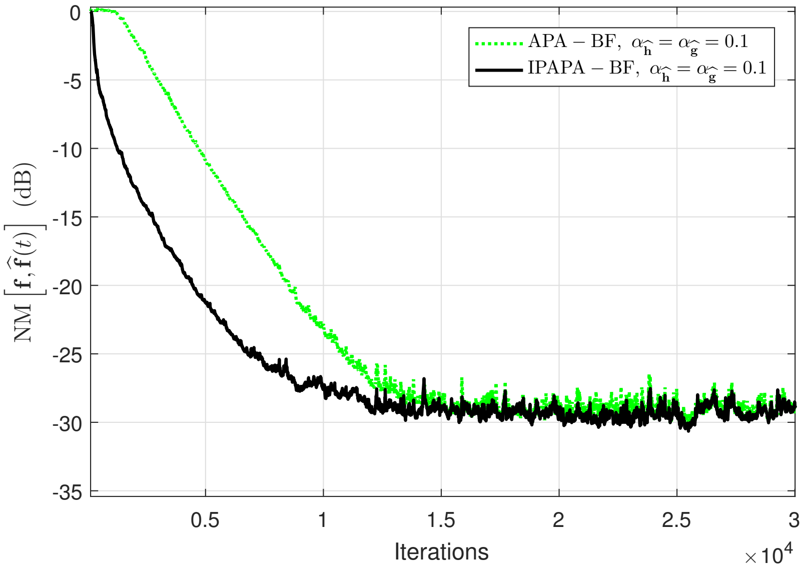


Figure 2.5: Performance of the APA-BF and IPAPA-BF in terms of NM. The input signals are white Gaussian noises and $ML = 2048$.

evaluated using a sparseness measure defined as [15]:

$$\xi_{12}(\mathbf{h}) = \frac{L}{L-\sqrt{L}}\left(1 - \frac{\|\mathbf{h}\|_1}{\sqrt{L}\|\mathbf{h}\|_2}\right), \tag{2.72}$$

where $\|\mathbf{h}\|_1 = \sum_{l=1}^{L} |h_l|$ is the ℓ_1 norm of the vector $\mathbf{h} = \begin{bmatrix} h_1 & h_2 & \dots & h_L \end{bmatrix}^T$. If the measure from (2.72) is close to 1, the impulse response is sparse. On the other hand, if the measure is close to 0, it means that the impulse response is less sparse.

In Fig. 2.5, the NM provided by IPAPA-BF and APA-BF is shown when the temporal impulse response is an echo path having the sparseness measure $\xi_{12}(\mathbf{h}) = 0.6131$. By comparison, the first impulse response from G168 recommendation [55], which was used for all previous experiments, has the sparseness measure $\xi_{12}(\mathbf{h}) = 0.897$. It can be seen that the proportionate version IPAPA-BF outperforms APA-BF in terms of convergence speed, leading to the conclusion that it may be appropriate to use this version also for the less sparse impulse responses.

2.5 An Optimized LMS Algorithm for Bilinear Forms (OLMS-BF)

In this subchapter, we approach the system identification problem based on the LMS algorithm, aiming to optimize its step-size parameter in order to address the compromise between the main performance criteria, i.e., convergence rate versus misadjustment [56]. In the following, the proposed OLMS-BF algorithm is derived based on the same system model given in Section 2.1. As will be explained in Section 2.8, this algorithm has striking resemblances with the SKF-BF, even if their derivations follow different patterns. The development of the OLMS-BF algorithm presented in the following was published in [21].

Let us consider the two estimated impulse responses $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$, such that the estimated output signal is given by (2.38). As a consequence, the a priori error signal between the desired signal and the estimated one can be defined following (2.39), i.e.,

$$\begin{aligned} e(n) &= d(n) - \widehat{y}(n) \\ &= [\mathbf{g}(n) \otimes \mathbf{h}(n)]^T \widetilde{\mathbf{x}}(n) + v(n) - \left[\widehat{\mathbf{g}}(n-1) \otimes \widehat{\mathbf{h}}(n-1)\right]^T \widetilde{\mathbf{x}}(n) \\ &= \mathbf{h}^T(n)\mathbf{x}_{\mathbf{g}}(n) + v(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{g}}}(n) && (2.73) \\ &= \mathbf{g}^T(n)\mathbf{x}_{\mathbf{h}}(n) + v(n) - \widehat{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n), && (2.74) \end{aligned}$$

where $\mathbf{x}_{\mathbf{g}}(n) = [\mathbf{g}(n) \otimes \mathbf{I}_L]^T\widetilde{\mathbf{x}}(n)$ and $\mathbf{x}_{\mathbf{h}}(n) = [\mathbf{I}_M \otimes \mathbf{h}(n)]^T\widetilde{\mathbf{x}}(n)$, while $\mathbf{x}_{\widehat{\mathbf{g}}}(n)$ and $\mathbf{x}_{\widehat{\mathbf{h}}}(n)$ are defined in (2.42) and (2.43), respectively.

Next, we can define the a posteriori misalignments (which represent the state estimation errors) related to the temporal and spatial impulse responses:

$$\mathbf{c}_{\mathbf{h}}(n) = \frac{1}{\eta}\mathbf{h}(n) - \widehat{\mathbf{h}}(n), \tag{2.75}$$

$$\mathbf{c}_{\mathbf{g}}(n) = \eta\mathbf{g}(n) - \widehat{\mathbf{g}}(n), \tag{2.76}$$

for which their correlation matrices are $\mathbf{R}_{\mathbf{c}_{\mathbf{h}}}(n) = \mathbb{E}[\mathbf{c}_{\mathbf{h}}(n)\mathbf{c}_{\mathbf{h}}^T(n)]$ and $\mathbf{R}_{\mathbf{c}_{\mathbf{g}}}(n) = \mathbb{E}[\mathbf{c}_{\mathbf{g}}(n)\mathbf{c}_{\mathbf{g}}^T(n)]$, respectively. As mentioned in subchapter 2.1, we can only identify the impulse responses up to this arbitrary scaling factor η; however, the pair $\mathbf{h}(n)/\eta$ and $\eta\mathbf{g}(n)$ is equivalent to the pair $\mathbf{h}$ and $\mathbf{g}$ in the bilinear form. Let us also define the a priori misalignments related to the two impulse responses:

$$\mathbf{c}_{\mathbf{h}_\mathrm{a}}(n) = \frac{1}{\eta}\mathbf{h}(n) - \widehat{\mathbf{h}}(n-1) = \mathbf{c}_{\mathbf{h}}(n-1) + \frac{1}{\eta}\mathbf{w}_{\mathbf{h}}(n), \tag{2.77}$$

$$\mathbf{c}_{\mathbf{g}_\mathrm{a}}(n) = \eta\mathbf{g}(n) - \widehat{\mathbf{g}}(n-1) = \mathbf{c}_{\mathbf{g}}(n-1) + \eta\mathbf{w}_{\mathbf{g}}(n), \tag{2.78}$$

whose correlation matrices are $\mathbf{R}_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n) = \mathbb{E}\left[\mathbf{c}_{\mathbf{h}_\mathrm{a}}(n)\mathbf{c}_{\mathbf{h}_\mathrm{a}}^T(n)\right]$ and $\mathbf{R}_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n) =$

$\mathbb{E}\left[\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}(n)\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}^{T}(n)\right]$, respectively. For the sake of simplicity of the upcoming developments, let us introduce the notation:

$$\overline{\mathbf{w}}_{\mathbf{h}}(n) = \frac{1}{\eta}\mathbf{w}_{\mathbf{h}}(n), \tag{2.79}$$

$$\overline{\mathbf{w}}_{\mathbf{g}}(n) = \eta\mathbf{w}_{\mathbf{g}}(n). \tag{2.80}$$

These new terms are also zero-mean WGN vectors, having the correlation matrices $\mathbf{R}_{\overline{\mathbf{w}}_{\mathbf{h}}}(n) = \sigma_{\overline{w}_{\mathbf{h}}}^{2}\mathbf{I}_{L}$ and $\mathbf{R}_{\overline{\mathbf{w}}_{\mathbf{g}}}(n) = \sigma_{\overline{w}_{\mathbf{g}}}^{2}\mathbf{I}_{M}$, respectively. Clearly, we have $\sigma_{\overline{w}_{\mathbf{h}}}^{2} = \sigma_{w_{\mathbf{h}}}^{2}/\eta^{2}$ and $\sigma_{\overline{w}_{\mathbf{g}}}^{2} = \eta^{2}\sigma_{w_{\mathbf{g}}}^{2}$.

The desired signal $d(n)$ may also be expressed as

$$\begin{aligned} d(n) &= \mathbf{g}^{T}(n)\mathbf{x}_{\mathbf{h}}(n) + \mathbf{g}^{T}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) - \mathbf{g}^{T}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + v(n) \\ &= \mathbf{g}^{T}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + v(n) + v_{\mathbf{g}}(n), \end{aligned} \tag{2.81}$$

where the term:

$$v_{\mathbf{g}}(n) = \mathbf{g}^{T}(n)\left[\mathbf{x}_{\mathbf{h}}(n) - \mathbf{x}_{\widehat{\mathbf{h}}}(n)\right] = \mathbf{c}_{\mathbf{h}_{\mathrm{a}}}^{T}(n)\mathbf{x}_{\mathbf{g}}(n) = \left[\mathbf{c}_{\mathbf{h}}(n-1) + \overline{\mathbf{w}}_{\mathbf{h}}(n)\right]^{T}\mathbf{x}_{\mathbf{g}}(n) \tag{2.82}$$

can be seen as an additional "noise" term, of variance $\sigma_{v_{\mathbf{g}}}^{2}(n)$, introduced by the system $\mathbf{g}$. In a similar way, for the second system we have

$$\begin{aligned} d(n) &= \mathbf{h}^{T}(n)\mathbf{x}_{\mathbf{g}}(n) + \mathbf{h}^{T}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) - \mathbf{h}^{T}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) + v(n) \\ &= \mathbf{h}^{T}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) + v(n) + v_{\mathbf{h}}(n), \end{aligned} \tag{2.83}$$

where

$$v_{\mathbf{h}}(n) = \mathbf{h}^{T}(n)\left[\mathbf{x}_{\mathbf{g}}(n) - \mathbf{x}_{\widehat{\mathbf{g}}}(n)\right] = \mathbf{c}_{\mathbf{g}_{\mathrm{a}}}^{T}(n)\mathbf{x}_{\mathbf{h}}(n) = \left[\mathbf{c}_{\mathbf{g}}(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}(n)\right]^{T}\mathbf{x}_{\mathbf{h}}(n) \tag{2.84}$$

can be interpreted as an additional "noise" term, of variance $\sigma_{v_{\mathbf{h}}}^{2}(n)$, related to the

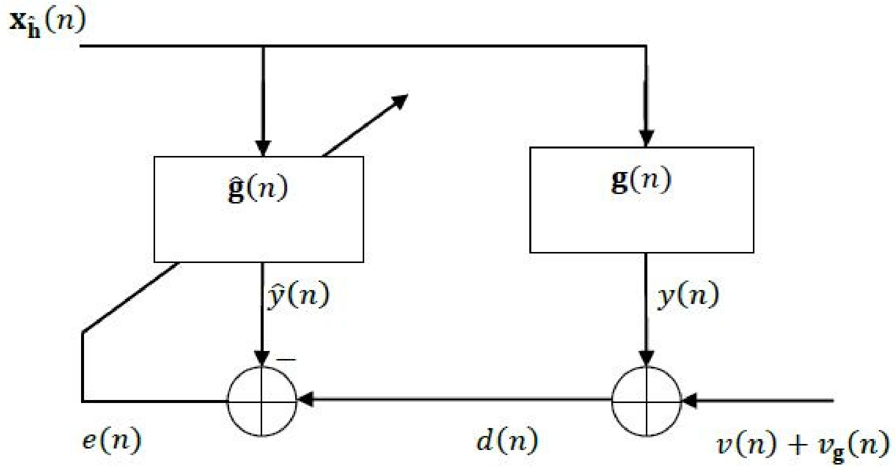


Figure 2.6: Equivalent system identification scheme when considering the system $\mathbf{g}(n)$ and the input $\mathbf{x}_{\widehat{\mathbf{h}}}(n)$.

system $\mathbf{h}$. Here, the a posteriori and a priori misalignments corresponding to the system $\mathbf{g}$ were defined in (2.76) and (2.78), respectively. In Figs. 2.6 and 2.7, the equivalent system identification scheme is represented in terms of the two components, $\mathbf{g}(n)$ and $\mathbf{h}(n)$, respectively; it can be observed that each system influences the other one through the additional "noise" term. In the framework of the LMS-BF algorithm [35], the updates are the following:

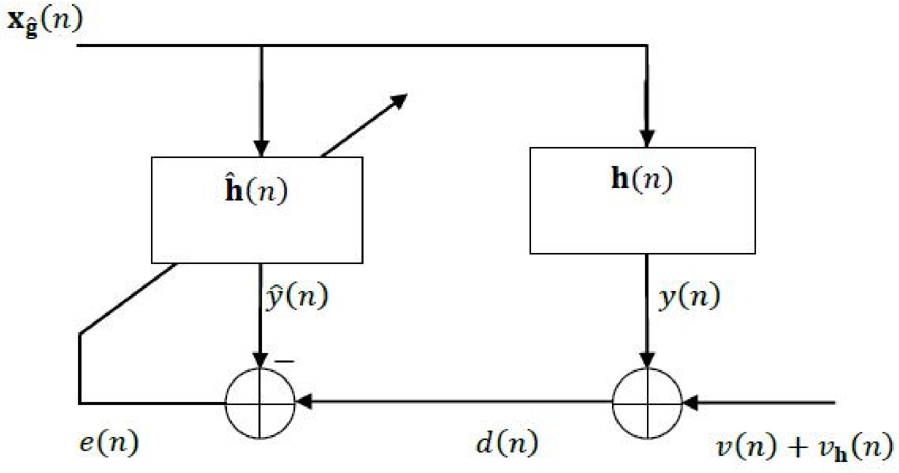


Figure 2.7: Equivalent system identification scheme when considering the system $\mathbf{h}(n)$ and the input $\mathbf{x}_{\widehat{\mathbf{g}}}(n)$.

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}}}\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n), \tag{2.85}$$

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}}}\mathbf{x}_{\widehat{\mathbf{g}}}(n)e(n), \tag{2.86}$$

where $\mu_{\widehat{\mathbf{g}}}$ and $\mu_{\widehat{\mathbf{h}}}$ are the step-size parameters. In this context, the vectors corresponding to the a posteriori misalignments become

$$\begin{aligned}\mathbf{c}_{\mathbf{g}}(n) &= \mathbf{g}(n) - \widehat{\mathbf{g}}(n-1) - \mu_{\widehat{\mathbf{g}}}\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n)\\ &= \mathbf{c}_{\mathbf{g}}(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}(n) - \mu_{\widehat{\mathbf{g}}}\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n), \qquad (2.87)\\ \mathbf{c}_{\mathbf{h}}(n) &= \mathbf{h}(n) - \widehat{\mathbf{h}}(n-1) - \mu_{\widehat{\mathbf{h}}}\mathbf{x}_{\widehat{\mathbf{g}}}(n)e(n)\\ &= \mathbf{c}_{\mathbf{h}}(n-1) + \overline{\mathbf{w}}_{\mathbf{h}}(n) - \mu_{\widehat{\mathbf{h}}}\mathbf{x}_{\widehat{\mathbf{g}}}(n)e(n). \qquad (2.88)\end{aligned}$$

At this point, let us introduce the notation $m_{\mathbf{g}}(n) = \mathbb{E}\left[\|\mathbf{c}_{\mathbf{g}}(n)\|^2\right]$ and $m_{\mathbf{h}}(n) = \mathbb{E}\left[\|\mathbf{c}_{\mathbf{h}}(n)\|^2\right]$. Taking the square ℓ_2 norms in both sides of (2.87) and (2.88), respectively, we can recursively evaluate

$$\begin{aligned}m_{\mathbf{g}}(n) &= m_{\mathbf{g}}(n-1) - 2\mu_{\widehat{\mathbf{g}}}\mathbb{E}\big\{\big[\mathbf{c}_{\mathbf{g}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\big]\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n)\big\}\\ &\quad + \mu_{\widehat{\mathbf{g}}}^2\mathbb{E}\big[\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)e^2(n)\big] + M\sigma_{\overline{w}_{\mathbf{g}}}^2\\ &= m_{\mathbf{g}}(n-1) - 2A_{\mathbf{g}}\mu_{\widehat{\mathbf{g}}} + \mu_{\widehat{\mathbf{g}}}^2 B_{\mathbf{g}} + M\sigma_{\overline{w}_{\mathbf{g}}}^2, \qquad (2.89)\\ m_{\mathbf{h}}(n) &= m_{\mathbf{h}}(n-1) - 2\mu_{\widehat{\mathbf{h}}}\mathbb{E}\big\{\big[\mathbf{c}_{\mathbf{h}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{h}}^T(n)\big]\mathbf{x}_{\widehat{\mathbf{g}}}(n)e(n)\big\}\\ &\quad + \mu_{\widehat{\mathbf{h}}}^2\mathbb{E}\big[\mathbf{x}_{\widehat{\mathbf{g}}}^T(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)e^2(n)\big] + L\sigma_{\overline{w}_{\mathbf{h}}}^2\\ &= m_{\mathbf{h}}(n-1) - 2A_{\mathbf{h}}\mu_{\widehat{\mathbf{h}}} + \mu_{\widehat{\mathbf{h}}}^2 B_{\mathbf{h}} + L\sigma_{\overline{w}_{\mathbf{h}}}^2, \qquad (2.90)\end{aligned}$$

where

$$A_{\mathbf{g}} = \mathbb{E}\big\{ \big[\mathbf{c}_{\mathbf{g}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\big] \mathbf{x}_{\widehat{\mathbf{h}}}(n) e(n) \big\}, \tag{2.91}$$

$$B_{\mathbf{g}} = \mathbb{E}\big[\mathbf{x}_{\widehat{\mathbf{h}}}^T(n) \mathbf{x}_{\widehat{\mathbf{h}}}(n) e^2(n) \big], \tag{2.92}$$

$$A_{\mathbf{h}} = \mathbb{E}\big\{ \big[\mathbf{c}_{\mathbf{h}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{h}}^T(n)\big] \mathbf{x}_{\widehat{\mathbf{g}}}(n) e(n) \big\}, \tag{2.93}$$

$$B_{\mathbf{h}} = \mathbb{E}\big[\mathbf{x}_{\widehat{\mathbf{g}}}^T(n) \mathbf{x}_{\widehat{\mathbf{g}}}(n) e^2(n) \big]. \tag{2.94}$$

It is very difficult to further process the expectation terms from (2.91)–(2.94) (and, consequently, (2.89) and (2.90)) without any supporting assumptions on the character of the input signals. Hence, let us consider that the covariance matrices of the inputs are close to a diagonal one. This is a fairly restrictive assumption on the input signals, which has been widely used to simplify the convergence analysis of many adaptive algorithms [56, 57]. Also, let us consider that the input signals are independent and have the same power. In this context, the computations of the expectation terms from (2.91)–(2.94), which are required in the development of (2.89) and (2.90), are provided in the following.

First, based on (2.81), we can express the error as

$$\begin{aligned} e(n) &= \mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + v(n) + v_{\mathbf{g}}(n) \\ &= \mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + v(n) + \mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\mathbf{g}}(n) + \overline{\mathbf{w}}_{\mathbf{h}}^T(n)\mathbf{x}_{\mathbf{g}}(n). \end{aligned} \tag{2.95}$$

Using this relation, together with (2.39) and (2.43), and taking into account that $\mathbf{c}_{\mathbf{g}}(n-1)$ and $\mathbf{x}_{\widehat{\mathbf{h}}}(n)$ are uncorrelated, the term from (2.91) results in

$$\begin{aligned}
A_{\mathbf{g}} &= \mathbb{E}\big\{\big[\mathbf{c}_{\mathbf{g}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\big]\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n)\big\} \\
&= \mathbb{E}\left\{\left[\mathbf{c}_{\mathbf{g}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\right]\mathbf{x}_{\widehat{\mathbf{h}}}(n)\left[v(n) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + v_{\mathbf{g}}(n) + \mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\right]\right\} \\
&= p_{\mathbf{g}}(n) + \mathbb{E}\left[\mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\overline{\mathbf{w}}_{\mathbf{g}}(n)\right] \\
&= p_{\mathbf{g}}(n) + \mathrm{tr}\left\{\mathbb{E}\left[\mathbf{c}_{\mathbf{g}}(n-1)\mathbf{c}_{\mathbf{g}}^T(n-1) + \overline{\mathbf{w}}_{\mathbf{g}}(n)\overline{\mathbf{w}}_{\mathbf{g}}^T(n)\right]\mathbb{E}\left[\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\right]\right\} \\
&= p_{\mathbf{g}}(n) + \Big[m_{\mathbf{g}}(n-1) + M\sigma_{\overline{w}_{\mathbf{g}}}^2\Big]\mathbb{E}\left[\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\right], \qquad (2.96)
\end{aligned}$$

where the term denoted by $p_{\mathbf{g}}(n)$ is evaluated as

$$\begin{aligned}
p_{\mathbf{g}}(n) &= \mathbb{E}\big\{\mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\tilde{\mathbf{x}}^T(n)[\mathbf{g}\otimes\mathbf{c}_{\mathbf{h}}(n-1)]\big\} \\
&= \mathbb{E}\big\{\mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\mathbf{g}}(n)\big\}, \qquad (2.97)
\end{aligned}$$

and we took into account that $\mathbf{c}_{\mathbf{g}}(n-1)$ and $\mathbf{x}_{\widehat{\mathbf{h}}}(n)$ are uncorrelated.

Next, we should concentrate on the last expectation term in (2.96), which can be expressed as

$$\mathbb{E}\left[\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\right] = \mathbb{E}\left\{\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\right\}. \qquad (2.98)$$

The main diagonal terms of this matrix are $\mathbb{E}\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\right]$, $m = 1, 2, \ldots, M$. In the following, we consider the assumption that the input signals are independent and have the same power, while their covariance matrices are close to a diagonal matrix [56, 57]. Consequently,where $\sigma_{\mathbf{x}}^2 = \mathbb{E}\left[\|\widetilde{\mathbf{x}}(n)\|^2\right]$.

$$\begin{aligned}
\mathbb{E}\left[\widehat{\mathbf{h}}^T(n-1)\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\right] &= \mathrm{tr}\left\{\mathbb{E}\left[\mathbf{x}_m(n)\mathbf{x}_m^T(n)\widehat{\mathbf{h}}(n-1)\widehat{\mathbf{h}}^T(n-1)\right]\right\} \\
&= \mathrm{tr}\left\{\mathbb{E}\left[\mathbf{x}_m(n)\mathbf{x}_m^T(n)\right]\mathbb{E}\left[\widehat{\mathbf{h}}(n-1)\widehat{\mathbf{h}}^T(n-1)\right]\right\} \\
&= \sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]. \qquad (2.99)
\end{aligned}$$

Finally, using (2.98) and (2.99) in (2.96), we obtain

$$A_{\mathbf{g}} = p_{\mathbf{g}}(n) + \sigma_{\mathbf{x}}^2 \mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\left[m_{\mathbf{g}}(n-1) + M\sigma_{\overline{w}_{\mathbf{g}}}^2\right]. \tag{2.100}$$

In a similar manner, the corresponding term from (2.93) is derived as

$$A_{\mathbf{h}} = p_{\mathbf{h}}(n) + \sigma_{\mathbf{x}}^2 \mathbb{E}\left[\left\|\widehat{\mathbf{g}}(n-1)\right\|^2\right]\left[m_{\mathbf{h}}(n-1) + L\sigma_{\overline{w}_{\mathbf{h}}}^2\right], \tag{2.101}$$

where the term $p_{\mathbf{h}}(n)$ is given by:

$$\begin{aligned} p_{\mathbf{h}}(n) &= \mathbb{E}\big\{\mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{g}}}(n)\tilde{\mathbf{x}}^T(n)[\mathbf{c}_{\mathbf{g}}(n-1)\otimes\mathbf{h}]\big\} \\ &= \mathbb{E}\big\{\mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{g}}}(n)\mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\mathbf{h}}(n)\big\}. \end{aligned} \tag{2.102}$$

Further, we detail the evaluation of the expectation term from (2.92). To begin, let us focus on the product $\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)$. Relying on the same considerations and assumptions from (2.98) and (2.99), we obtain

$$\begin{aligned} \mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) &= \widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n) \\ &= \mathrm{tr}\left\{\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]^T\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)\left[\mathbf{I}_M\otimes\widehat{\mathbf{h}}(n-1)\right]\right\} \\ &\approx M\sigma_{\mathbf{x}}^2\left\|\widehat{\mathbf{h}}(n-1)\right\|^2. \end{aligned} \tag{2.103}$$

Similarly,

$$\mathbf{x}_{\widehat{\mathbf{g}}}^T(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) \approx L\sigma_{\mathbf{x}}^2\left\|\widehat{\mathbf{g}}(n-1)\right\|^2. \tag{2.104}$$

Hence, considering some degree of stationarity of the input signals, (2.103) can be seen as a deterministic quantity, yielding

$$B_{\mathbf{g}} = \mathbb{E}\left[\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)e^2(n)\right] \approx M\sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\mathbb{E}\left[e^2(n)\right]. \tag{2.105}$$

Let us focus on the computation of the expectation term $\mathbb{E}\left[e^2(n)\right]$. Using (2.95), we obtain

$$\begin{aligned}\mathbb{E}\left[e^2(n)\right] &= \sigma_v^2 + \mathbb{E}\left[\overline{\mathbf{w}}_{\mathbf{g}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\overline{\mathbf{w}}_{\mathbf{g}}(n)\right] \\ &\quad + \mathbb{E}\left\{\left[v_{\mathbf{g}}(n) + \mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{c}_{\mathbf{g}}(n-1)\right]\left[v_{\mathbf{g}}(n) + \mathbf{c}_{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\right]\right\}. \end{aligned} \tag{2.106}$$

Summarizing, the terms result in

$$A_{\mathbf{g}} = p_{\mathbf{g}}(n) + \sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\left[m_{\mathbf{g}}(n-1) + M\sigma_{\overline{w}_{\mathbf{g}}}^2\right], \tag{2.107}$$

$$A_{\mathbf{h}} = p_{\mathbf{h}}(n) + \sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{g}}(n-1)\right\|^2\right]\left[m_{\mathbf{h}}(n-1) + L\sigma_{\overline{w}_{\mathbf{h}}}^2\right], \tag{2.108}$$

$$\begin{aligned} B_{\mathbf{g}} &= M\sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\left\{\sigma_v^2 + 2p_{\mathbf{g}}(n) + \frac{1}{L}\sigma_{v_{\mathbf{g}}}^2(n) + \sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\right. \\ &\quad \left. \times\left[m_{\mathbf{g}}(n-1) + M\sigma_{\overline{w}_{\mathbf{g}}}^2\right]\right\}, \end{aligned} \tag{2.109}$$

$$\begin{aligned} B_{\mathbf{h}} &= L\sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{g}}(n-1)\right\|^2\right]\left\{\sigma_v^2 + 2p_{\mathbf{h}}(n) + \frac{1}{M}\sigma_{v_{\mathbf{h}}}^2(n) + \sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\widehat{\mathbf{g}}(n-1)\right\|^2\right]\right. \\ &\quad \left. \times\left[m_{\mathbf{h}}(n-1) + L\sigma_{\overline{w}_{\mathbf{h}}}^2\right]\right\}. \end{aligned} \tag{2.110}$$

At this point, we need to evaluate the variance of $v_{\mathbf{g}}$, which can be developed as

$$\begin{aligned} \sigma_{v_{\mathbf{g}}}^2(n) &= \mathbb{E}\left[v_{\mathbf{g}}^2(n)\right] \\ &= \mathbb{E}\left\{\left[\mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\mathbf{g}}(n) + \overline{\mathbf{w}}_{\mathbf{h}}^T(n)\mathbf{x}_{\mathbf{g}}(n)\right]\left[\mathbf{x}_{\mathbf{g}}^T(n)\mathbf{c}_{\mathbf{h}}(n-1) + \mathbf{x}_{\mathbf{g}}^T(n)\overline{\mathbf{w}}_{\mathbf{h}}(n)\right]\right\} \\ &= \mathbb{E}\left[\mathbf{c}_{\mathbf{h}}^T(n-1)\mathbf{x}_{\mathbf{g}}(n)\mathbf{x}_{\mathbf{g}}^T(n)\mathbf{c}_{\mathbf{h}}(n-1) + \overline{\mathbf{w}}_{\mathbf{h}}^T(n)\mathbf{x}_{\mathbf{g}}(n)\mathbf{x}_{\mathbf{g}}^T(n)\overline{\mathbf{w}}_{\mathbf{h}}(n)\right] \\ &= \operatorname{tr}\left\{\mathbb{E}\left[\mathbf{c}_{\mathbf{h}}(n-1)\mathbf{c}_{\mathbf{h}}^T(n-1)\right]\mathbb{E}\left[\mathbf{x}_{\mathbf{g}}(n)\mathbf{x}_{\mathbf{g}}^T(n)\right] + \sigma_{\overline{w}_{\mathbf{h}}}^2\mathbb{E}\left[\mathbf{x}_{\mathbf{g}}(n)\mathbf{x}_{\mathbf{g}}^T(n)\right]\right\} \\ &= \mathbb{E}\left[\left\|\mathbf{x}_{\mathbf{g}}(n)\right\|^2\right]\left\{L\sigma_{\overline{w}_{\mathbf{h}}}^2 + \mathbb{E}\left[\left\|\mathbf{c}_{\mathbf{h}}(n-1)\right\|^2\right]\right\} \\ &= L\sigma_{\mathbf{x}}^2\mathbb{E}\left[\left\|\mathbf{g}(n)\right\|^2\right]\left[L\sigma_{\overline{w}_{\mathbf{h}}}^2 + m_{\mathbf{h}}(n-1)\right]. \end{aligned} \tag{2.111}$$

Therefore, (2.106) results in

$$\mathbb{E}\left[e^2(n)\right] = \sigma_v^2 + \sigma_{\mathbf{x}}^2 \left[M\sigma_{\overline{w}_{\mathbf{g}}}^2 + m_{\mathbf{g}}(n-1)\right] \mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right] + \frac{1}{L}\sigma_{v_{\mathbf{g}}}^2(n) + 2p_{\mathbf{g}}(n), \tag{2.112}$$

thus obtaining

$$\begin{aligned} B_{\mathbf{g}} = {} & M\sigma_{\mathbf{x}}^2 \mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right] \left\{\sigma_v^2 + 2p_{\mathbf{g}}(n) + \frac{1}{L}\sigma_{v_{\mathbf{g}}}^2(n) + \sigma_{\mathbf{x}}^2 \mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right]\right. \\ & \left. \times \left[m_{\mathbf{g}}(n-1) + M\sigma_{\overline{w}_{\mathbf{g}}}^2\right]\right\}. \end{aligned} \tag{2.113}$$

For the corresponding term from (2.94), we use the dual expression for $e(n)$, which leads to

$$\mathbb{E}\left[e^2(n)\right] = \sigma_v^2 + \sigma_{\mathbf{x}}^2 \left[L\sigma_{\overline{w}_{\mathbf{h}}}^2 + m_{\mathbf{h}}(n-1)\right] \mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right] + \frac{1}{M}\sigma_{v_{\mathbf{h}}}^2(n) + 2p_{\mathbf{h}}(n), \tag{2.114}$$

where [similar to (2.111)]

$$\sigma_{v_{\mathbf{h}}}^2(n) = \mathbb{E}\big[v_{\mathbf{h}}^2(n)\big] = M\sigma_{\mathbf{x}}^2 \mathbb{E}\big[\|\mathbf{h}(n)\|^2\big] \left[M\sigma_{\overline{w}_{\mathbf{g}}}^2 + m_{\mathbf{g}}(n-1)\right]. \tag{2.115}$$

Thus, we finally obtain

$$\begin{aligned} B_{\mathbf{h}} = {} & L\sigma_{\mathbf{x}}^2 \mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right] \left\{\sigma_v^2 + 2p_{\mathbf{h}}(n) + \frac{1}{M}\sigma_{v_{\mathbf{h}}}^2(n) + \sigma_{\mathbf{x}}^2 \mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right]\right. \\ & \left. \times \left[m_{\mathbf{h}}(n-1) + L\sigma_{\overline{w}_{\mathbf{h}}}^2\right]\right\}. \end{aligned} \tag{2.116}$$

Summarizing, we can use (2.100), (2.101), (2.113), and (2.116) in (2.89)–(2.90), in order to obtain the recursive relations from (2.117)–(2.118), which are further used in the development of the OLMS-BF algorithm.

Consequently, using (2.107)–(2.110) in (2.89) and (2.90), we obtain

$$\begin{aligned} m_{\mathbf{g}}(n) &= m_{\mathbf{g}}(n-1)\left\{1-2\mu_{\widehat{\mathbf{g}}}\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{h}}(n-1)\|^2\big]+\mu_{\widehat{\mathbf{g}}}^2\sigma_{\mathbf{x}}^4 M\big\{\mathbb{E}\big[\|\widehat{\mathbf{h}}(n-1)\|^2\big]\big\}^2\right\} \\ &\quad -2\mu_{\widehat{\mathbf{g}}}p_{\mathbf{g}}(n)+\mu_{\widehat{\mathbf{g}}}^2 M\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{h}}(n-1)\|^2\big]\left\{\sigma_v^2+\frac{1}{L}\sigma_{v_{\mathbf{g}}}^2(n)\right. \\ &\quad \left.+M\sigma_{\overline{w}_{\mathbf{g}}}^2\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{h}}(n-1)\|^2\big]+2p_{\mathbf{g}}(n)\right\}M\sigma_{\overline{w}_{\mathbf{g}}}^2\left\{1-2\mu_{\widehat{\mathbf{g}}}\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{h}}(n-1)\|^2\big]\right\}, \end{aligned} \tag{2.117}$$

$$\begin{aligned} m_{\mathbf{h}}(n) &= m_{\mathbf{h}}(n-1)\left\{1-2\mu_{\widehat{\mathbf{h}}}\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{g}}(n-1)\|^2\big]+\mu_{\widehat{\mathbf{h}}}^2\sigma_{\mathbf{x}}^4 L\big\{\mathbb{E}\big[\|\widehat{\mathbf{g}}(n-1)\|^2\big]\big\}^2\right\} \\ &\quad -2\mu_{\widehat{\mathbf{h}}}p_{\mathbf{h}}(n)+\mu_{\widehat{\mathbf{h}}}^2 L\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{g}}(n-1)\|^2\big]\left\{\sigma_v^2+\frac{1}{M}\sigma_{v_{\mathbf{h}}}^2(n)\right. \\ &\quad \left.+L\sigma_{\overline{w}_{\mathbf{h}}}^2\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{g}}(n-1)\|^2\big]+2p_{\mathbf{h}}(n)\right\}+L\sigma_{\overline{w}_{\mathbf{h}}}^2\left\{1-2\mu_{\widehat{\mathbf{h}}}\sigma_{\mathbf{x}}^2\mathbb{E}\big[\|\widehat{\mathbf{g}}(n-1)\|^2\big]\right\}. \end{aligned} \tag{2.118}$$

In the context of system identification problems, the main goal is to reduce the system misalignment, which basically represents the difference between the true impulse response and the estimated one. Therefore, in our framework, the optimal step-size parameters (denoted in the following by $\mu_{\widehat{\mathbf{g}},\text{o}}$ and $\mu_{\widehat{\mathbf{h}},\text{o}}$) can be found by minimizing (2.117) and (2.118). This is done by canceling the derivatives of (2.117) and (2.118) with respect to the step-sizes, which result in:

$$\frac{\partial m_{\mathbf{g}}(n)}{\partial \mu_{\widehat{\mathbf{g}}}} = -2A_{\mathbf{g}}+2B_{\mathbf{g}}\mu_{\widehat{\mathbf{g}}}=0 \Rightarrow \mu_{\widehat{\mathbf{g}},\text{o}}=\frac{A_{\mathbf{g}}}{B_{\mathbf{g}}}, \tag{2.119}$$

$$\frac{\partial m_{\mathbf{h}}(n)}{\partial \mu_{\widehat{\mathbf{h}}}} = -2A_{\mathbf{h}}+2B_{\mathbf{h}}\mu_{\widehat{\mathbf{h}}}=0 \Rightarrow \mu_{\widehat{\mathbf{h}},\text{o}}=\frac{A_{\mathbf{h}}}{B_{\mathbf{h}}}. \tag{2.120}$$

By replacing $A_{\mathbf{g}}, B_{\mathbf{g}}, A_{\mathbf{h}}$, and $B_{\mathbf{h}}$ with their expressions (see (2.107)–(2.110)), the step-size parameters of the proposed OLMS-BF algorithm are found. Finally, introducing these parameters in (2.85) and (2.86), the updates of the OLMS-BF algorithm become

$$\begin{aligned}\widehat{\mathbf{g}}(n) &= \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}},\mathrm{o}}(n) x_{\widehat{\mathbf{h}}}(n) e(n) \\ &= \widehat{\mathbf{g}}(n-1) + \frac{x_{\widehat{\mathbf{h}}}(n) e(n)}{M\sigma_{\mathbf{x}}^2 \mathbb{E}\left[\|\widehat{\mathbf{h}}(n-1)\|^2\right]\left\{1 + \frac{p_{\mathbf{g}}(n)+\sigma_v^2+\frac{1}{L}\sigma_{v_{\mathbf{g}}}^2(n)}{p_{\mathbf{g}}(n)+\sigma_{\mathbf{x}}^2\mathbb{E}\left[\|\widehat{\mathbf{h}}(n-1)\|^2\right]\left[m_{\mathbf{g}}(n-1)+M\sigma_{w_{\mathbf{g}}}^2\right]}\right\}},\end{aligned} \tag{2.121}$$

$$\begin{aligned}\widehat{\mathbf{h}}(n) &= \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}},\mathrm{o}}(n) x_{\widehat{\mathbf{g}}}(n) e(n) \\ &= \widehat{\mathbf{h}}(n-1) + \frac{x_{\widehat{\mathbf{g}}}(n) e(n)}{L\sigma_{\mathbf{x}}^2 \mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right]\left\{1 + \frac{p_{\mathbf{h}}(n)+\sigma_v^2+\frac{1}{M}\sigma_{v_{\mathbf{h}}}^2(n)}{p_{\mathbf{h}}(n)+\sigma_{\mathbf{x}}^2\mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right]\left[m_{\mathbf{h}}(n-1)+L\sigma_{w_{\mathbf{h}}}^2\right]}\right\}}.\end{aligned} \tag{2.122}$$

The most problematic terms in (2.121) and (2.122) are $p_{\mathbf{g}}(n)$ and $p_{\mathbf{h}}(n)$ (from (2.97) and (2.102), respectively), which depend on the true impulse responses. However, as it will be shown in subchapter 2.8, these terms could be omitted in practice.

2.5.1 Experimental Results

Simulations were conducted in the context of system identification, in order to outline the performance of the proposed OLMS-BF algorithm.

In the first set of experiments, the behavior of the OLMS-BF algorithm is analyzed, as compared to the NLMS-BF algorithm [35]. The NLMS-BF algorithm uses different values of its step-size parameters, $\alpha_{\widehat{\mathbf{h}}}$ and $\alpha_{\widehat{\mathbf{g}}}$. The performances are now evaluated in terms of both NPMs and NM, using both types of input signals as before [WGNs and AR(1) processes]. The results are presented in Figs. 2.8 and 2.9, using WGNs as inputs, and in Figs. 2.10 and 2.11, where the input signals are AR(1) processes. It can be noticed that the proposed solution achieves similar convergence rate but a much lower misalignment level than the NLMS-BF algorithm with $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.5$ (which provides the fastest convergence rate [35]). On the other hand, if we target a lower misalignment and set the step-sizes of the NLMS-BF to smaller values (i.e., $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.1$ and $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.01$), the convergence rate also decreases. However, the

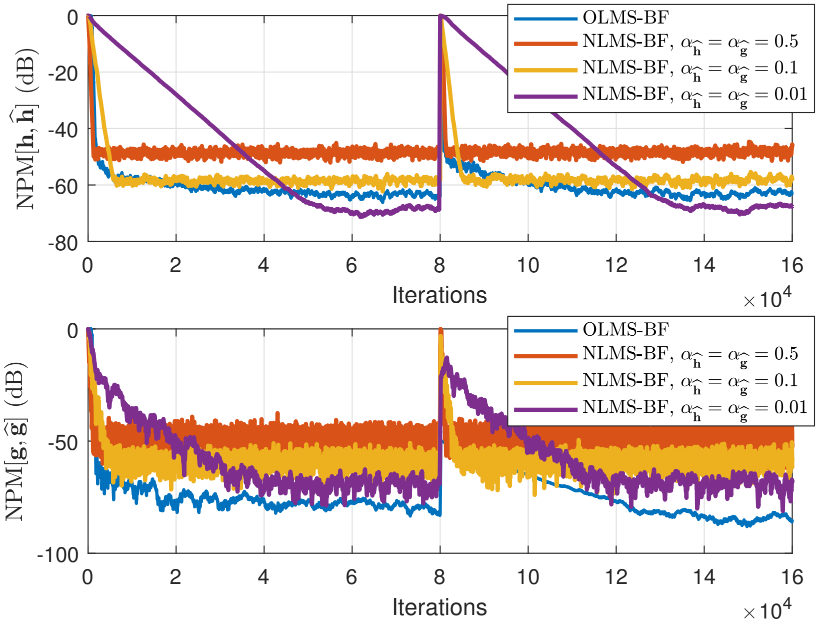


Figure 2.8: Normalized projection misalignment of the OLMS-BF and NLMS-BF (using different step-size parameters): (*top*) identification of the temporal impulse response $\mathbf{h}(n)$, (*bottom*) identification of the spatial impulse response $\mathbf{g}(n)$. The input signals are WGNs, $L = 64$, and $M = 8$.

OLMS-BF algorithm leads to a misalignment level similar to the NLMS-BF algorithm using the smallest step-sizes. In addition, when the input signals are AR(1) processes, the improvement offered by the OLMS-BF algorithm is even more apparent.

Next, the performance of the OLMS-BF algorithm is evaluated along with the joint-optimized NLMS (JO-NLMS) algorithm [58], which is applied for the identification of the global impulse response of length $ML = 512$. The results are presented in Figs. 2.12 and 2.13, using WGNs and AR(1) input signals, respectively. As specified in Section 2.8.2, the JO-NLMS algorithm is the regular counterpart of the OLMS-BF in a classical (one-dimensional) system identification scenario. We can see that the proposed solution (tailored for BF, i.e., exploiting the two-dimensional decomposition) offers both faster convergence and tracking, as well as a lower misalignment, as compared to the JO-NLMS algorithm. The performance improvement is even more important in case of AR(1) input signals.

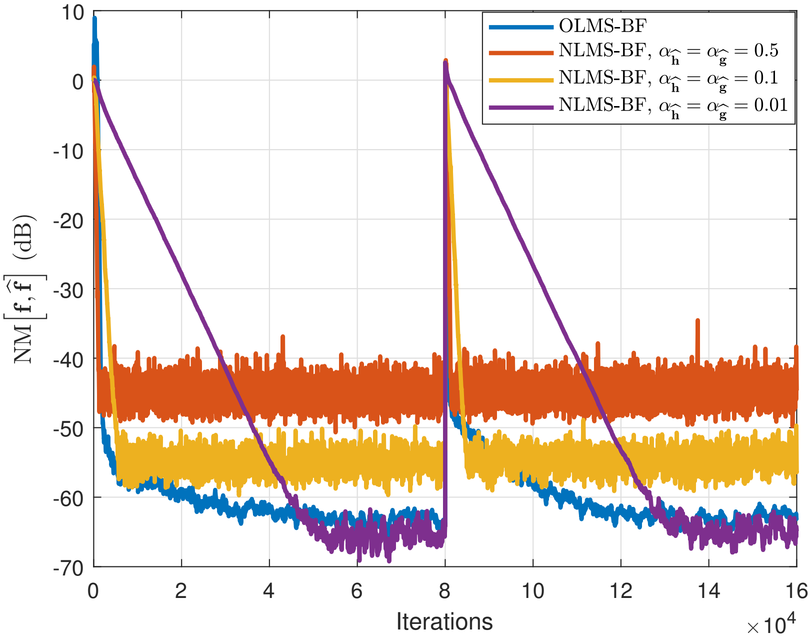


Figure 2.9: Normalized misalignment of the OLMS-BF and NLMS-BF (using different step-size parameters). The input signals are WGNs and $ML = 512$.

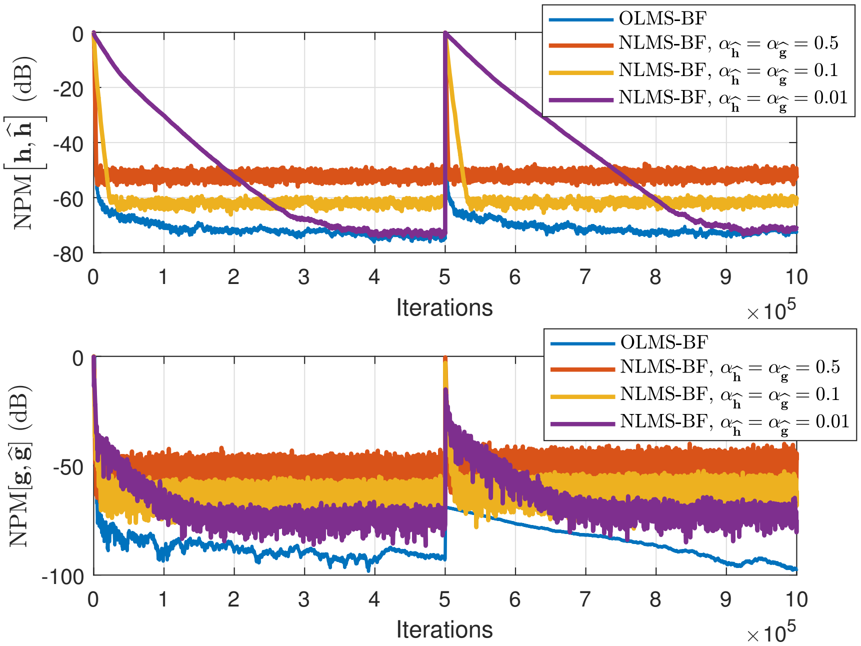


Figure 2.10: Normalized projection misalignment of the OLMS-BF and NLMS-BF (using different step-size parameters): (*top*) identification of the temporal impulse response $\mathbf{h}(n)$, (*bottom*) identification of the spatial impulse response $\mathbf{g}(n)$. The input signals are AR(1) processes, $L = 64$, and $M = 8$.

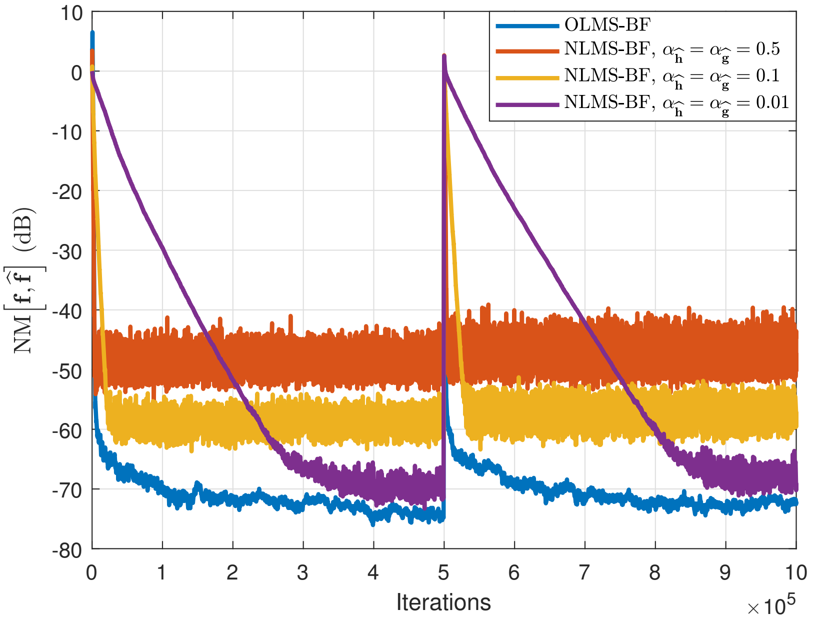


Figure 2.11: Normalized misalignment of the OLMS-BF and NLMS-BF algorithms (using different step-size parameters). The input signals are AR(1) processes and $ML = 512$.

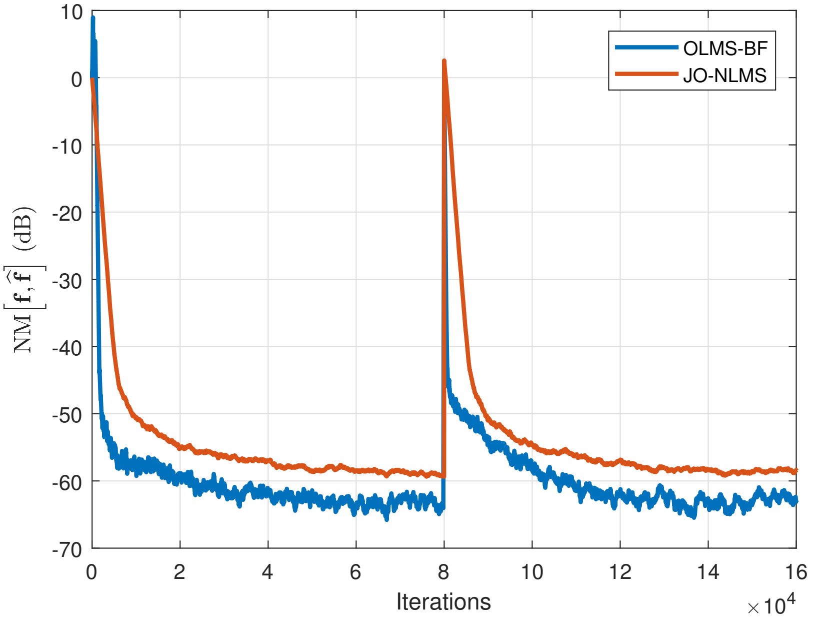


Figure 2.12: Normalized misalignment of the OLMS-BF and regular JO-NLMS algorithms. The input signals are WGNs and $ML = 512$.

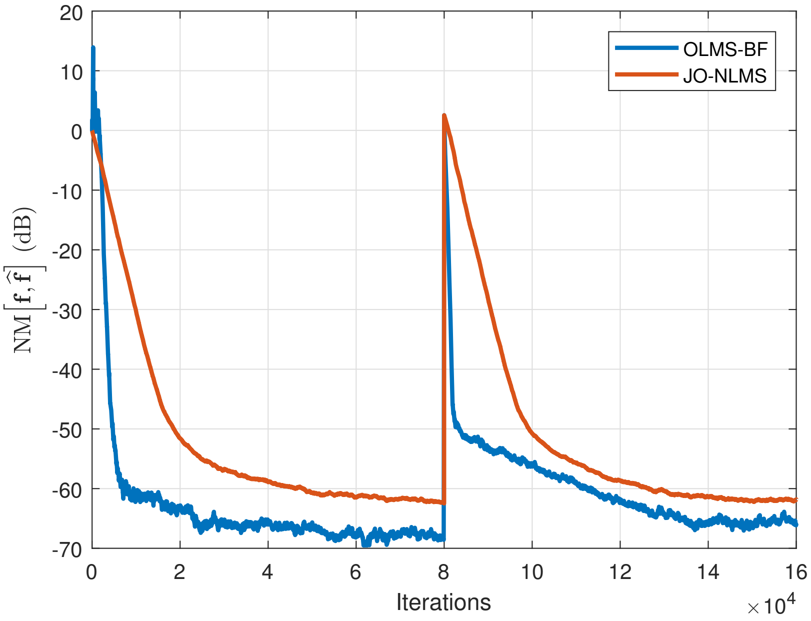


Figure 2.13: Normalized misalignment for the OLMS-BF and regular JO-NLMS algorithms. The input signals are AR(1) processes and $ML = 512$.

Finally, to validate our approach, we assess the performance of the OLMS-BF algorithm when applying it in a context which is closer to a real scenario. The temporal impulse response $\mathbf{h}(n)$ is a real-world echo path of length $L = 512$. The spatial impulse response $\mathbf{g}(n)$, of length $M = 4$, is generated using an exponential decay with the elements $g_m = 0.5^m$, $m = 1, \ldots, M$. Both impulse responses are then normalized such that $\|\mathbf{h}(n)\| = \|\mathbf{g}(n)\| = 1$. The input signal is an AR(1) process and we compare the behaviors of the OLMS-BF and NLMS-BF algorithms. The performance are illustrated in Figs. 2.14 and 2.15. We can notice that the proposed solution slightly outperforms the fastest convergence rate of NLMS-BF, given by $\alpha_{\widehat{\mathbf{h}}} = \alpha_{\widehat{\mathbf{g}}} = 0.5$, but at the same time offering a much lower value of the misalignment. If, however, we use the NLMS-BF algorithm with the smaller step-sizes (in order to obtain a better misalignment), the resulting convergence rate is much lower than the one of the OLMS-BF algorithm.

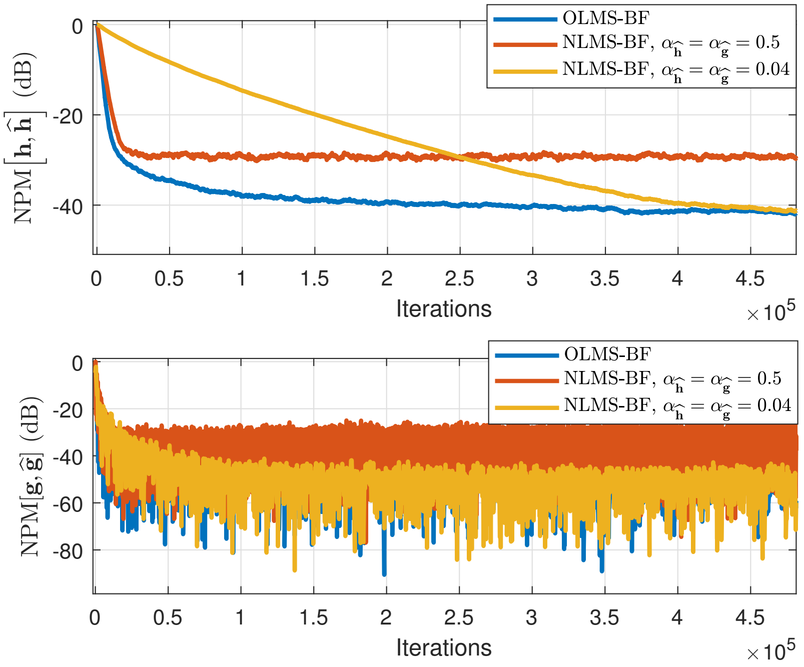


Figure 2.14: Normalized projection misalignment of the OLMS-BF and NLMS-BF (using different step-size parameters): (*top*) identification of the temporal impulse response $\mathbf{h}(n)$, (*bottom*) identification of the spatial impulse response $\mathbf{g}(n)$. The input signals are AR(1) processes, $L = 512$, and $M = 4$.

2.6 On the Properties of the System Mismatch Covariance Matrix (SMCM) in the LMS Adaptive Algorithm

The developments layed out in this subchapter represent original contributions, which were published in [22].

The SMCM is frequently encountered in the convergence analysis of gradient descent adaptive algorithms [56, 57, 3, 59, 60, 61]. It also appears in some studies concerning the optimization of variable step size LMS algorithms [58] or adaptive algorithms based on the KF [16] and in the convergence analysis of such algorithms. Some variable step-size algorithms are also based on the minimization of the mean square system mismatch. Such studies require information about this specific matrix,

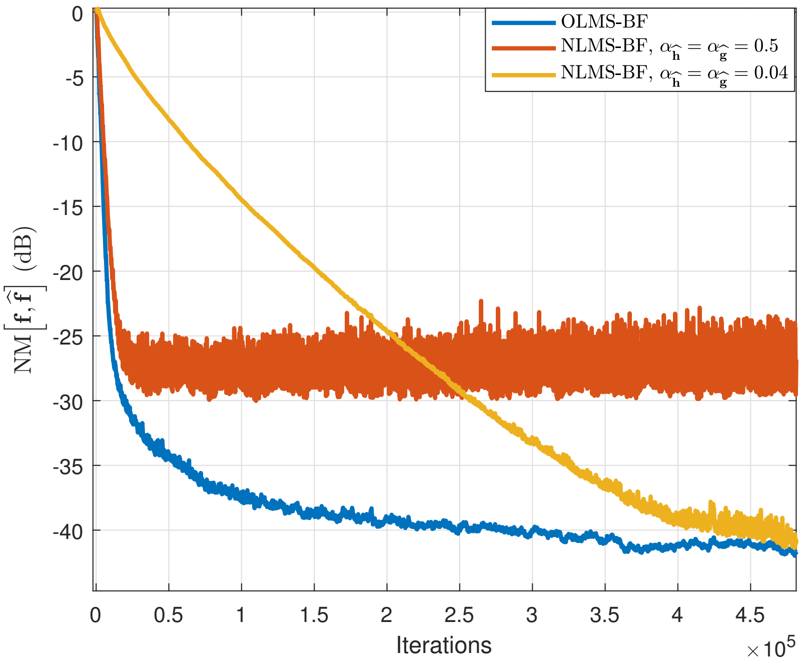


Figure 2.15: Normalized projection misalignment for the OLMS-BF and NLMS-BF algorithms. The input signals are AR(1) processes and $ML = 2048$.

and it is usually stated that after the algorithm has started to converge, the SMCM tends to a scaled unit matrix. In this subchapter, we analyze the validity conditions of this assumption and provide simulation results to support the theoretical findings.

2.6.1 A Recursive Equation for the SMCM

Let us start the analysis by considering an adaptive configuration aimed to estimate the impulse response of an unknown system $\mathbf{g}$ of length N, having as input a wide-sense-stationary signal $\mathbf{x}(n) = [x(n), \cdots, x(n-N+1)]^T$. The desired signal is:

$$d(n) = \mathbf{g}^T \mathbf{x}(n) + v(n), \tag{2.123}$$

where $v(n)$ is the system noise. In a LMS-type adaptive algorithm with a step-size parameter μ, the coefficients are updated according to [56]:

$$\widehat{\mathbf{g}}\left(n\right)=\widehat{\mathbf{g}}\left(n-1\right)+\mu\mathbf{x}\left(n\right)e_a\left(n\right), \tag{2.124}$$

where

$$e_a\left(n\right)=d\left(n\right)-\widehat{\mathbf{g}}^T\left(n-1\right)\mathbf{x}\left(n\right) \tag{2.125}$$

is the a priori error. The system mismatch is:

$$\mathbf{c}\left(n\right)=\mathbf{g}-\widehat{\mathbf{g}}\left(n\right)=\mathbf{c}\left(n-1\right)-\mu\mathbf{x}\left(n\right)e_a\left(n\right). \tag{2.126}$$

Hence, the a priori error can be also be expressed as:

$$e_a\left(n\right)=\mathbf{g}^T\mathbf{x}\left(n\right)+v\left(n\right)-\widehat{\mathbf{g}}^T\left(n-1\right)\mathbf{x}\left(n\right)=\mathbf{c}^T\left(n-1\right)\mathbf{x}\left(n\right)+v\left(n\right). \tag{2.127}$$

A recursive equation for the system mismatch results in:

$$\begin{aligned}\mathbf{c}\left(n\right)&=\mathbf{c}\left(n-1\right)-\mu\mathbf{x}\left(n\right)\left[\mathbf{x}^T\left(n\right)\mathbf{c}\left(n-1\right)+v\left(n\right)\right]\\&=\left[\mathbf{I}-\mu\mathbf{x}\left(n\right)\mathbf{x}^T\left(n\right)\right]\mathbf{c}\left(n-1\right)-\mu\mathbf{x}\left(n\right)v\left(n\right),\end{aligned} \tag{2.128}$$

where **I** is the identity matrix.

By definition, the SMCM is:

$$\mathbf{R_c}(n)\triangleq\mathbb{E}\left\{\mathbf{c}\left(n\right)\mathbf{c}^T\left(n\right)\right\}. \tag{2.129}$$

This is a symmetric, positive semidefinite matrix. From this definition and relation (2.128) it results that:

$$\begin{aligned}\mathbf{R_c}(n)=\mathbb{E}\big\{&\left[(\mathbf{I}-\mu\mathbf{x}\left(n\right)\mathbf{x}^T\left(n\right))\mathbf{c}\left(n-1\right)-\mu\mathbf{x}\left(n\right)v\left(n\right)\right]\\&\times\left[\mathbf{c}^T\left(n-1\right)\left(\mathbf{I}-\mu\mathbf{x}\left(n\right)\mathbf{x}^T\left(n\right)\mathbf{x}\left(n\right)\right)-\mu\mathbf{x}^T\left(n\right)v\left(n\right)\right]\big\}.\end{aligned} \tag{2.130}$$

Next, let us consider the following set of assumptions:

- $v(n)$ - white noise, uncorrelated to $\mathbf{x}(n)$ and $\mathbf{c}(n-1)$. Consequently:

$$\begin{aligned}
\mathbf{R_c}(n) = \mathbb{E}\Big\{ &\left[\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n)\right]\mathbf{c}(n-1)\mathbf{c}^T(n-1)\left[\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n)\right] \\
&+ \mu^2\mathbf{x}(n)\mathbf{x}^T(n)v^2(n)\Big\} \\
&= \mathbf{R_c}(n-1) - \mathbb{E}\big\{\mu\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{c}(n-1)\mathbf{c}^T(n-1) \\
&+ \mu\mathbf{c}(n-1)\mathbf{c}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n) \\
&- \mu^2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{c}(n-1)\mathbf{c}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\big\} + \mu^2\sigma_v^2\mathbf{R_x},
\end{aligned} \tag{2.131}$$

where $\mathbf{R_x}$ is the correlation matrix of the input signal and σ_v^2 is the noise variance. It is well-known that $\mathbf{R_x}$ is a symmetric, positive semidefinite matrix, containing on the main diagonal the mean power of the process, denoted by σ_x^2.

- $\mathbf{x}(n)$ and $\mathbf{c}(n-1)$ uncorrelated. Consequently:

$$\mathbb{E}\left\{\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{c}(n-1)\mathbf{c}^T(n-1)\right\} = \mathbf{R_x}\mathbf{R_c}(n-1). \tag{2.132}$$

- $\mathbf{x}(n)$ - zero mean Gaussian process. As a result (see [62]):

$$\begin{aligned}
&\mathbb{E}\left\{\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{c}(n-1)\mathbf{c}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\right\} \\
&= 2\mathbf{R_x}\mathbf{R_c}(n-1)\mathbf{R_x} + \mathbf{R_x}\mathrm{tr}\{\mathbf{R_c}(n-1)\mathbf{R_x}\}.
\end{aligned} \tag{2.133}$$

In this way, $\mathbf{R_c}(n)$ becomes:

$$\begin{aligned}
\mathbf{R_c}(n) = \mathbf{R_c}(n-1) - \mu[\mathbf{R_x}\mathbf{R_c}(n-1) + \mathbf{R_c}(n-1)\mathbf{R_x}] + \mu^2[2\mathbf{R_x}\mathbf{R_c}(n-1)\mathbf{R_x} \\
+ \mathbf{R_x}\mathrm{tr}\{\mathbf{R_c}(n-1)\mathbf{R_x}\}] + \mu^2\sigma_v^2\mathbf{R_x}.
\end{aligned} \tag{2.134}$$

The above equation reveals the dependence of the SMCM on the correlation matrix of the input signal, on the value of the step-size, and on the noise power.

Then, the update equations for the matrix elements $r_{c_{i,j}}$ can be easily derived:

$$r_{c_{i,j}}(n) = r_{c_{i,j}}(n-1) - \mu \sum_{k=0}^{N-1} [r_{i,k} r_{c_{k,j}}(n-1) + r_{j,k} r_{c_{i,k}}(n-1)] \\ + 2\mu^2 \sum_{k,l=0}^{N-1} r_{i,k} r_{c_{k,l}}(n-1) r_{j,l} + \mu^2 r_{i,j}[S(n-1) + \sigma_v^2], \tag{2.135}$$

where $r_{i,j}$ represent the elements of the matrix $\mathbf{R_x}$ and

$$S\left(n-1\right) \triangleq \sum_{\mathbf{k=0}}^{N-1} \sum_{i=0}^{N-1} \left[r_{i,k} r_{ci,k}\left(n-1\right)\right] = \text{vecdiag}^T \left\{\mathbf{R_c}\left(n-1\right)\right\} \text{vecdiag}\{\mathbf{R_x}\}. \tag{2.136}$$

In the relation above, the operator vecdiag$\{\mathbf{M}\}$ yields the vector formed with the elements on the main diagonal of the matrix $\mathbf{M}$.

An important observation here is that $S\left(n-1\right)$ is not a function of i or j.

In the end, we obtain:

$$r_{c_{i,j}}(n) = [1 - 2\mu\sigma_x^2 + 2\mu^2(\sigma_x^4 + r_{i,j}^2)] r_{c_{i,j}}(n-1) - \mu \left[\sum_{\substack{k=0 \\ k \neq i}}^{N-1} r_{i,k} r_{c_{k,j}}\left(n-1\right) \right. \\ \left. + \sum_{\substack{k=0 \\ k \neq j}}^{N-1} r_{j,k} r_{c_{i,k}}\left(n-1\right) \right] + 2\mu^2 \sum_{\substack{k,l=0 \\ k,l \neq i,j}}^{N-1} r_{i,k} r_{c_{k,l}}\left(n-1\right) r_{j,l} \; + \mu^2 r_{i,j} \left[S\left(n-1\right) + \sigma_v^2\right]. \tag{2.137}$$

2.6.2 Asymptotic Behavior

To characterize the SMCM after the convergence has been established, the limit when $n \to \infty$ is of interest. In this case, supposing the algorithm is convergent, equation (2.137) leads to:

$$2[\sigma_x^2-\mu(\sigma_x^4+r_{i,j}^2)]r_{c_{i,j}}(\infty)=-\Bigg[\sum_{\substack{k=0\\k\neq i}}^{N-1}r_{i,k}r_{c_{k,j}}(\infty)+\sum_{\substack{k=0\\k\neq j}}^{N-1}r_{j,k}r_{c_{i,k}}(\infty)\Bigg]$$
$$+2\mu\sum_{\substack{k,l=0\\k,l\neq i,j\\k,l\neq j,i}}^{N-1}r_{i,k}r_{c_{k,l}}(\infty)r_{j,l}+\mu r_{i,j}[S(\infty)\ +\sigma_v^2]. \quad (2.138)$$

From here, it results that:

$$r_{c_{i,j}}(\infty)=\frac{1}{2\left[\sigma_x^2-\mu\left({\sigma_x}^4+r_{i,j}^2\right)\right]}\Bigg[2\mu\sum_{\substack{k,l=0\\k,l\neq i,j\\k,l\neq j,i}}^{N-1}r_{i,k}r_{c_{k,l}}(\infty)r_{j,l}-\sum_{\substack{k=0\\k\neq i}}^{N-1}\left[r_{i,k}r_{c_{k,j}}(\infty)\right]$$
$$-\sum_{\substack{k=0\\k\neq j}}^{N-1}[r_{j,k}r_{c_{i,k}}(\infty)]+\mu r_{i,j}(S(\infty)+\sigma_v^2)\Bigg]. \quad (2.139)$$

Two statements have to be tested here:

1. Is $r_{c_{i,j}}(\infty)=0,\ \ \forall i\neq j$, for any nonzero values of μ and σ_v^2?

2. Do all $r_{c_{i,i}}(\infty)$ have the same value, $\forall i$?

The answer to the first one is obvious. By observing the last term of the sum from (2.139), it can be seen that $r_{c_{i,j}}(\infty)$ could be zero for any nonzero σ_v^2 only if μ or $r_{i,j}, i\neq j$ are zero.

For the second statement, it would be necessary for

$$r_{c_{i,i}}(\infty)=\frac{1}{2\sigma_x^2\left(1-2\mu\sigma_x^2\right)}\Bigg[-2\sum_{\substack{k=0\\k\neq i}}^{N-1}r_{i,k}r_{c_{k,i}}(\infty)+2\mu\sum_{\substack{k,l=0\\k,l\neq i,i}}^{N-1}r_{i,k}r_{c_{k,l}}(\infty)r_{i,l}$$
$$+\mu\sigma_x^2\left(S(\infty)+\sigma_v^2\right)\Bigg] \quad (2.140)$$

not to depend on i. In fact, the sum of the first two terms from the square bracket should not depend on i. This condition is not fulfilled in general.

2.6.3 Case Studies: White Noise and Autoregressive Input Signals

The next step is to characterize the SMCM for two of the most encountered types of stationary signals used for system identification problems: the white noise and the AR input signals.

White noise

In the case when the input signal is white noise, its correlation matrix is of the form:

$$\mathbf{R}_{\mathbf{x}}=\sigma_x^2\mathbf{I}, \tag{2.141}$$

hence:

$$\begin{aligned}\mathbf{R}_{\mathbf{c}}(n) = \mathbf{R}_{\mathbf{c}}(n-1) - 2\mu\sigma_x^2\mathbf{R}_{\mathbf{c}}(n-1) + \mu^2\sigma_x^4[2\mathbf{R}_{\mathbf{c}}\left(n-1\right) + \mathbf{I}\ \mathrm{tr}\{\mathbf{R}_{\mathbf{c}}\left(n-1\right)\}] \\ + \mu^2\sigma_v^2\sigma_x^2\mathbf{I}.\end{aligned} \tag{2.142}$$

After reaching convergence:

$$\mathbf{R}_{\mathbf{c}}\left(\infty\right) = \mathbf{R}_{\mathbf{c}}\left(\infty\right) - 2\mu\sigma_x^2\mathbf{R}_{\mathbf{c}}\left(\infty\right) + \mu^2\sigma_x^4[2\mathbf{R}_{\mathbf{c}}\left(\infty\right) + \mathbf{I}\ \mathrm{tr}\left\{\mathbf{R}_{\mathbf{c}}\left(\infty\right)\right\}] + \mu^2\sigma_v^2\sigma_x^2\mathbf{I}. \tag{2.143}$$

It follows that:

$$\mathbf{R}_{\mathbf{c}}\left(\infty\right) = \frac{\mu\left[\sigma_x^2\ \mathrm{tr}\left\{\mathbf{R}_{\mathbf{c}}\left(\infty\right)\right\} + \sigma_v^2\right]}{2\left[1-\mu\sigma_x^2\right]}\mathbf{I} \tag{2.144}$$

and consequently,

$$\mathrm{tr}\left\{\mathbf{R}_{\mathbf{c}}\left(\infty\right)\right\} = \frac{\mu\left[\sigma_x^2\mathrm{tr}\left\{\mathbf{R}_{\mathbf{c}}\left(\infty\right)\right\} + \sigma_v^2\right]}{2[1-\mu\sigma_x^2]}N = m\left(\infty\right) = \frac{\mu N\sigma_v^2}{2-\left(N+2\right)\mu\sigma_x^2}. \tag{2.145}$$

Therefore, in this case, the SMCM tends to a scaled unit matrix.

Autoregressive Process

Next, let us consider as input a first-order AR process, described by:

$$x\left(n\right)=\alpha x\left(n-1\right)+w\left(n\right),\quad 0<\alpha\leq 1, \tag{2.146}$$

where $w\left(n\right)$ is a zero mean WGN signal with variance σ_w^2. In this case, it can be easily shown that:

$$r_{i,j}=\sigma_w^2\frac{\alpha^{|i-j|}}{1-\alpha^2}={\sigma_x}^2\alpha^{|i-j|}, \tag{2.147}$$

where $\sigma_x^2=\frac{\sigma_w^2}{1-\alpha^2}$. Therefore, after reaching convergence:

$$r_{c_{i,j}}\left(\infty\right)=\frac{1}{2[\sigma_x^2-\mu\sigma_x^4\left(1+\alpha^{2|i-j|}\right)]}\Bigg\{2\mu\sigma_x^4\sum_{\substack{k,l=0\\k,l\neq i,j\\k,l\neq j,i}}^{N-1}\alpha^{|i-k|+|j-l|}r_{c_{k,l}}\left(\infty\right)$$
$$-\sigma_x^2\sum_{\substack{k=0\\k\neq i}}^{\mathbf{N-1}}\alpha^{|i-k|}r_{c_{k,j}}\left(\infty\right)-\sigma_x^2\sum_{\substack{k=0\\k\neq j}}^{\mathbf{N-1}}\left[\alpha^{|j-k|}r_{c_{i,k}}\left(\infty\right)\right]+\mu\sigma_x^2\alpha^{|i-j|}\left[S\left(\infty\right)+\sigma_v^2\right]\Bigg\}. \tag{2.148}$$

In this way:

$$S\left(\infty\right)\triangleq\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left[r_{l,k}r_{c_{l,k}}\left(\infty\right)\right]=\sigma_x^2\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left[\alpha^{|l-k|}r_{c_{l,k}}\left(\infty\right)\right]. \tag{2.149}$$

As shown in subchapter 2.6.2, $r_{c_{i,j}}(\infty)$ for any $i\neq j$ are not generally zero for a nonzero value of the step size and external noise. Moreover, the elements of the main diagonal:

$$r_{c_{i,i}}(\infty)=\frac{1}{2\sigma^2(1-\sigma_x^2\mu)}\Bigg\{2\mu\sigma_x^4\sum_{\substack{k,l=0\\k,l\neq i,j}}^{N-1}\alpha^{|i-k|+|i-l|}r_{c_{k,l}}(\infty)-2\sigma_x^2\sum_{\substack{k=0\\k\neq i}}^{N-1}[\alpha^{|i-k|}r_{c_{i,k}}(\infty)]$$
$$+\mu\sigma_x^2[S(\infty)+\sigma_v^2]\Bigg\} \tag{2.150}$$

are in general distinct, because of the first two sums.

2.6.4 Experimental Results

Finally, we aim to provide the outcomes of simulations, which prove once again the results reached through analytical computations in the previous sections.

First, the correlation matrix of the input signal $\mathbf{R_x}$ is plotted for three cases of possible inputs: $\alpha = 0$ (white noise), $\alpha = 0.95$ (AR signal), and $\alpha = 0.99$ (strongly correlated AR process), respectively. The variance of the process is $\sigma_x^2 = 0.1$ and the length of the filter is $N = 100$.

As it is well-known, the spread of nonzero elements outside the main diagonal increases with the degree of correlation (as α increases from 0 to 0.99). This fact is illustrated for the matrices of dimension $N \times N$ in Fig. 2.16.

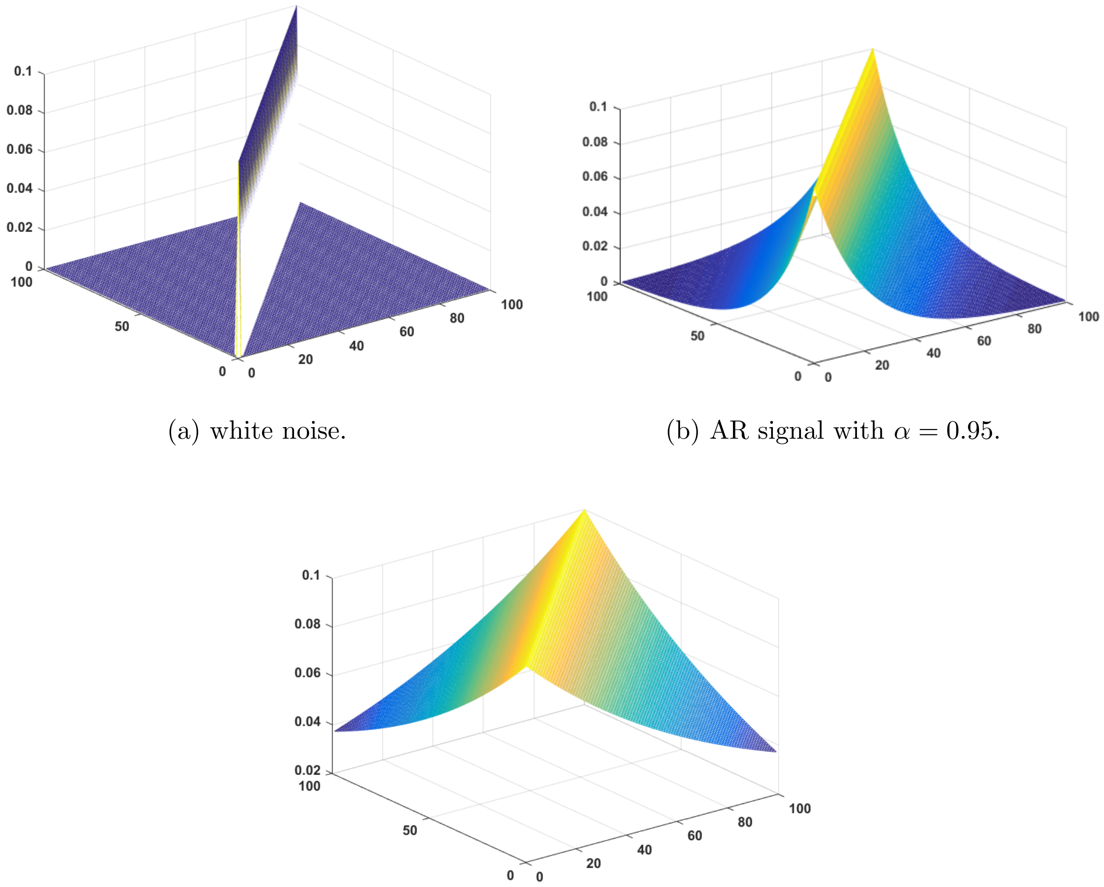
In Fig. 2.17, the evolution of the trace of the SMCM, normalized to the first value, is represented in logarithmic scale for the first 10^5 iterations of the algorithm, for the same three types of input signals as before, allowing the evaluation of the convergence process. The step-size value is $\mu = 0.01$, the noise power is $\sigma_v^2 = 0.02$, and all the other parameters are kept the same. Of course, the convergence process is slower for the highly correlated input signal.

Next, the SMCM is represented in Fig. 2.18 for the same types of input signals after 10^4 iterations of the algorithm (such that the algorithm is close to convergence, but it has not converged yet), and respectively in Fig. 2.19, after 10^5 iterations (such that it reached convergence), under the same previous conditions.

It can be noticed from Figs. 2.18a and 2.19a that:

- the elements outside the main diagonal are all equal,
- they have a significantly smaller value than the ones on the main diagonal,

confirming the theoretical results from (2.144).



(a) white noise.

(b) AR signal with $\alpha = 0.95$.

(c) AR signal with $\alpha = 0.99$.

Figure 2.16: The correlation matrix, $\mathbf{R}_{\mathbf{x}}$, for three types of input signals.

These two observations, however, are exactly valid only for the white noise input. As the correlation degree of the input signal increases, the value of the non-zero elements outside the main diagonal becomes higher. In the case of correlated signals, the two assumptions can be made only in the case of low noise and for a small step-size. However, this is more visible during the convergence phase, when the steady-state is not completely reached (see Fig. 2.18). The differences with respect to the scaled unity matrix model decrease with the number of iterations (see Fig. 2.19), without being completely canceled.

The elements on the main diagonal of the SMCM, $r_{c_{i,i}}$, are also of interest. Their values in the convergence regime are represented in Fig. 2.20, where all the param-

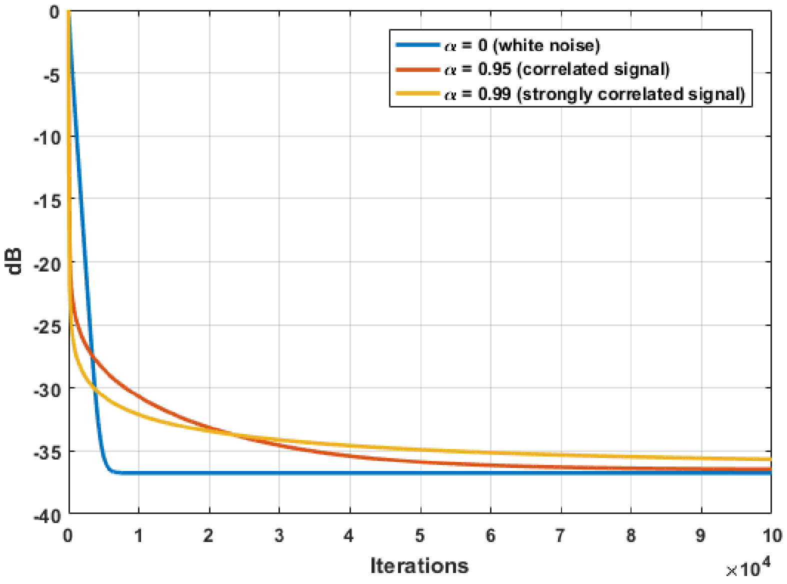
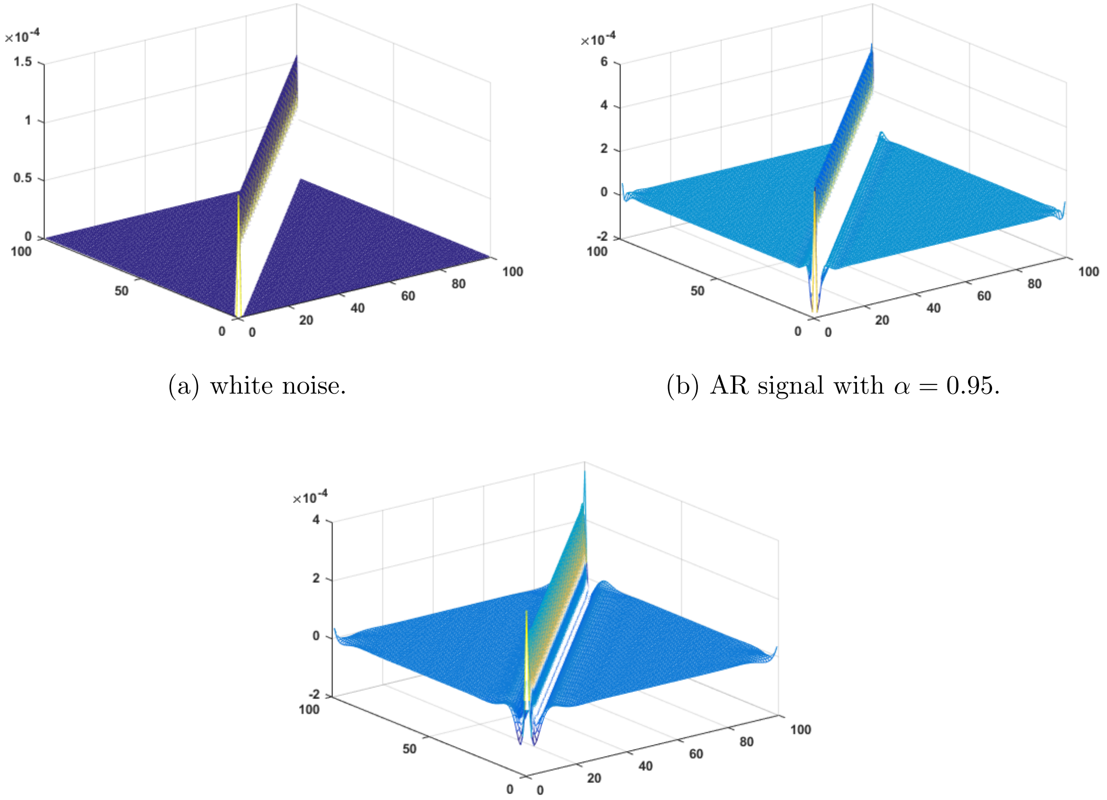


Figure 2.17: The evolution of the trace of the SMCM, normalized to the first value, in dB, for three types of input signals

eters of the simulation are the same as for Fig. 2.19. As expected, in the case of white noise, the graphical representation is constant, because, as it was proved in subchapter 2.6.3, these elements are all equal. On the other hand, we can see that for the AR signals, despite the symmetry of the representations, these elements are not all equal, but instead they present some changes in their values, which was also shown in subchapter 2.6.3.

In Fig. 2.21, the same elements are represented, but this time for a larger step-size, $\mu = 0.02$, thus revealing their increasing behavior with the step size.

In Figs. 2.22 and 2.23 we can see the evolution of the elements on the first line of the SMCM after 10^4, respectively 10^5 iterations of the algorithm. The simulation parameters are the same as for Figs. 2.18 and 2.19.



(a) white noise.

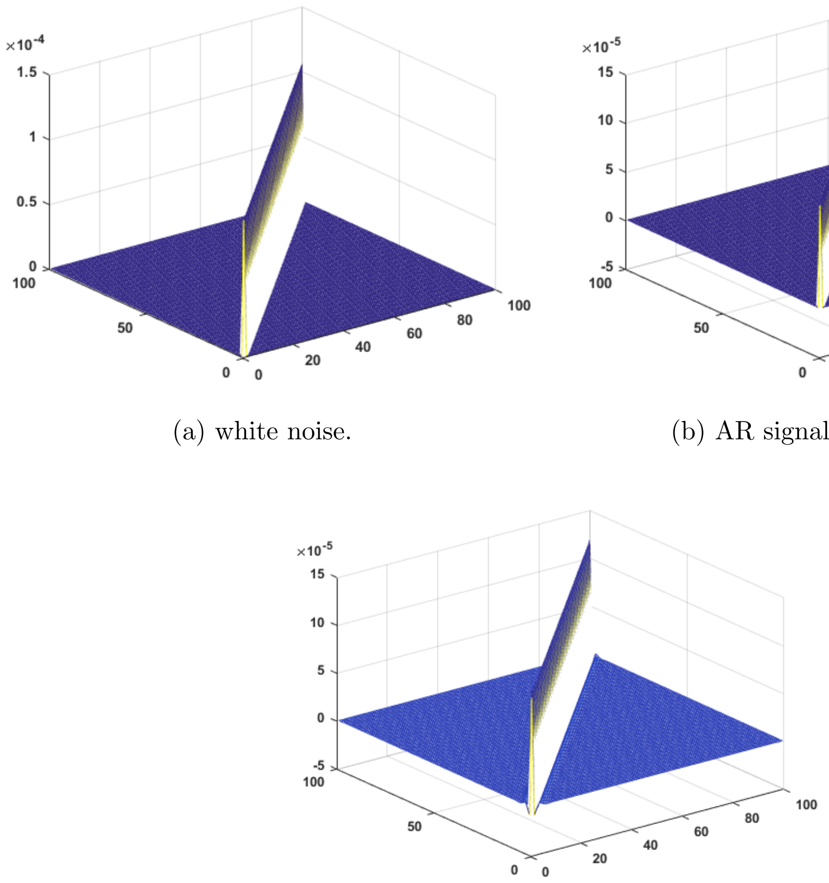
(b) AR signal with $\alpha = 0.95$.

(c) AR signal with $\alpha = 0.99$.

Figure 2.18: The SMCM for three types of input signals after 10^4 iterations of the algorithm

2.7 Kalman Filter for the Identification of Bilinear Forms (KF-BF)

In this section, we address the BF system identification problem using the KF. We start by developing the general form of this algorithm in subchapter 2.7.1, and then continue with a computationally simplified version in subchapter 2.7.2, highlighting the gain offered in terms of complexity, but also the compromise in terms of performance. In the end, we provide a comparison between the KF and the OLMS algorithm for BF, highlighting some interesting aspects which connect the two algorithms. The developments presented here were published in [19, 22].



(a) white noise.

(b) AR signal with $\alpha = 0.95$.

(c) AR signal with $\alpha = 0.99$.

Figure 2.19: The SMCM for three types of input signals after 10^5 iterations of the algorithm

2.7.1 Kalman Filter for Bilinear Forms (KF-BF)

In the framework of the KF, the signal model from (2.1) may be interpreted as the observation equation, while the system impulse responses can be considered as state equations. Given the two adaptive filters $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{g}}(n)$, the estimated signal has the expression derived in relation (2.38). As a result, the a priori error signal between the desired and estimated signals can be defined as

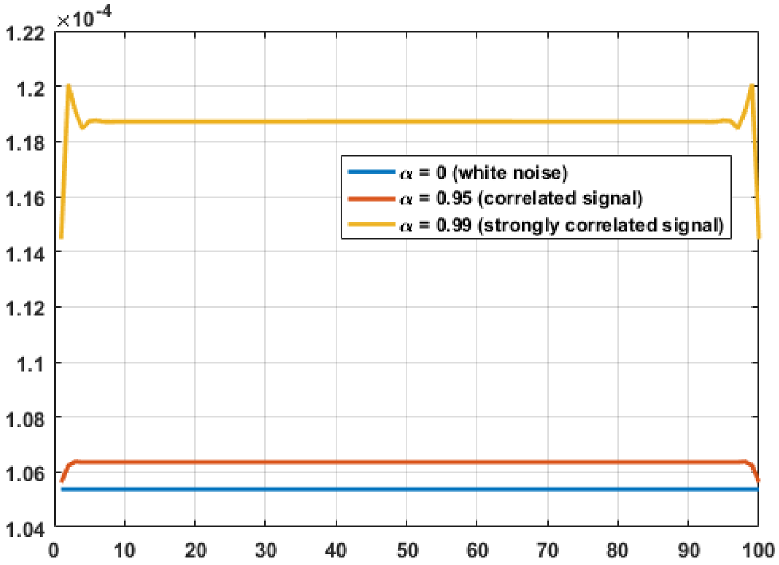


Figure 2.20: The elements on the main diagonal of the SMCM for three types of input signals, when $\mu = 0.01$

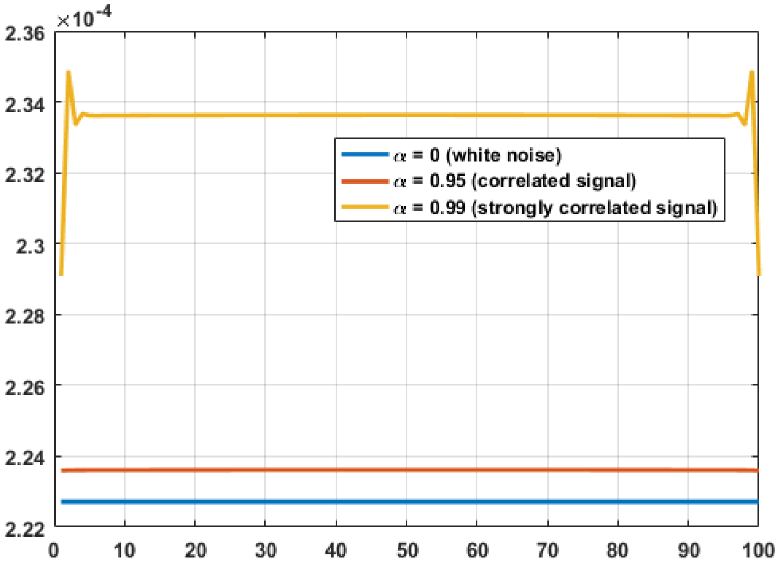


Figure 2.21: The elements on the main diagonal of the SMCM for three types of input signals, when $\mu = 0.02$

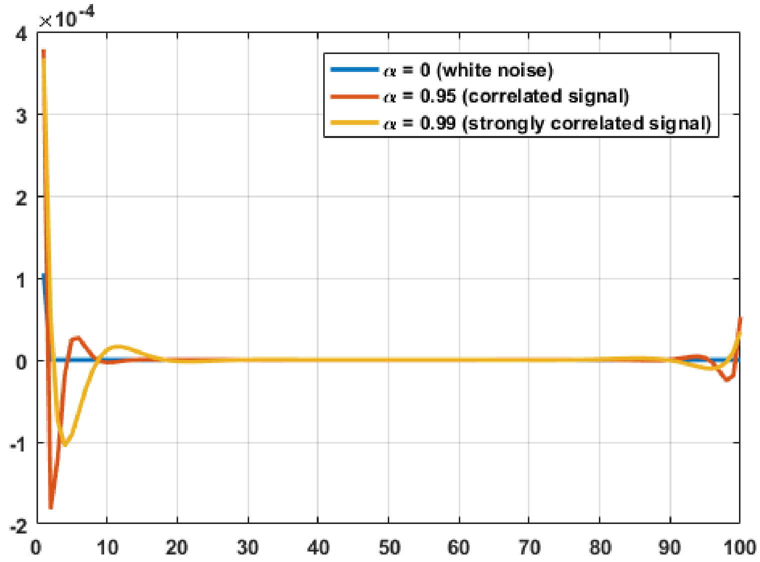


Figure 2.22: The elements on the first line of the SMCM for three types of input signals after 10^4 iterations of the algorithm

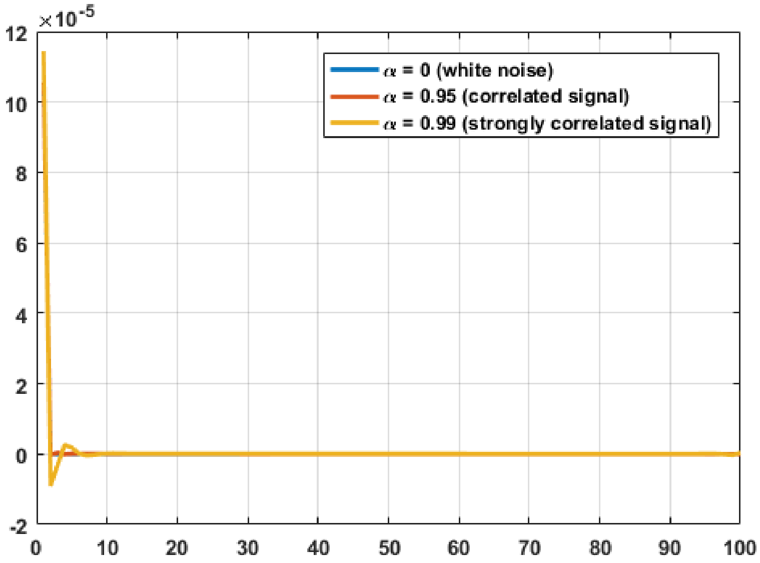


Figure 2.23: The elements on the first line of the SMCM for three types of input signals after 10^5 iterations of the algorithm

$$\begin{aligned} e(n) &= d(n) - \widehat{y}(n) \\ &= d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\widehat{\mathbf{g}}(n-1) \\ &= d(n) - \left[\widehat{\mathbf{g}}(n-1) \otimes \widehat{\mathbf{h}}(n-1)\right]^T \widetilde{\mathbf{x}}(n) \\ &= d(n) - \widehat{\mathbf{f}}^T(n-1)\widetilde{\mathbf{x}}(n) \\ &= d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{g}}}(n) \\ &= d(n) - \widehat{\mathbf{g}}^T(n-1)\mathbf{x}_{\widehat{\mathbf{h}}}(n), \end{aligned} \tag{2.151}$$

where

$$\mathbf{x}_{\widehat{\mathbf{g}}}(n) = [\widehat{\mathbf{g}}(n-1) \otimes \mathbf{I}_L]^T\widetilde{\mathbf{x}}(n), \tag{2.152}$$

$$\mathbf{x}_{\widehat{\mathbf{h}}}(n) = [\mathbf{I}_M \otimes \widehat{\mathbf{h}}(n-1)]^T\widetilde{\mathbf{x}}(n). \tag{2.153}$$

In the context of the linear sequential Bayesian approach, the optimal estimates of the state vectors have the forms [63]:

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mathbf{k}_{\mathbf{h}}(n)e(n), \tag{2.154}$$

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mathbf{k}_{\mathbf{g}}(n)e(n), \tag{2.155}$$

where $\mathbf{k}_{\mathbf{h}}(n)$ and $\mathbf{k}_{\mathbf{g}}(n)$ are the Kalman gain vectors.

In the following, we will use the a posteriori misalignments corresponding to the two system impulse responses, which were defined in subchapter 2.5 [relations (2.75), (2.76)]. We will also need to use the a priori misalignment terms, defined by (2.77), (2.78). Consequently, we obtain

$$\mathbf{R}_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n) = \mathbf{R}_{\mathbf{c}_\mathbf{h}}(n-1) + \sigma^2_{\overline{w}_\mathbf{h}}\mathbf{I}_L, \tag{2.156}$$

$$\mathbf{R}_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n) = \mathbf{R}_{\mathbf{c}_\mathbf{g}}(n-1) + \sigma^2_{\overline{w}_\mathbf{g}}\mathbf{I}_M. \tag{2.157}$$

In this context, the Kalman gain vectors are computed from the minimization of the criteria:

$$J_{\mathbf{h}}(n) = \frac{1}{L}\mathrm{tr}\left[\mathbf{R}_{\mathbf{c}_{\mathbf{h}}}(n)\right], \tag{2.158}$$

$$J_{\mathbf{g}}(n) = \frac{1}{M}\mathrm{tr}\left[\mathbf{R}_{\mathbf{c}_{\mathbf{g}}}(n)\right], \tag{2.159}$$

with respect to $\mathbf{k}_{\mathbf{h}}(n)$ and $\mathbf{k}_{\mathbf{g}}(n)$, respectively. From these minimizations, we find that

$$\mathbf{k}_{\mathbf{h}}(n) = \frac{\mathbf{R}_{\mathbf{c}_{\mathbf{h}_{\mathrm{a}}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)}{\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\mathbf{R}_{\mathbf{c}_{\mathbf{h}_{\mathrm{a}}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) + \sigma_v^2}, \tag{2.160}$$

$$\mathbf{k}_{\mathbf{g}}(n) = \frac{\mathbf{R}_{\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)}{\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\mathbf{R}_{\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + \sigma_v^2}, \tag{2.161}$$

and

$$\mathbf{R}_{\mathbf{c}_{\mathbf{h}}}(n) = \left[\mathbf{I}_L - \mathbf{k}_{\mathbf{h}}(n)\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\right]\mathbf{R}_{\mathbf{c}_{\mathbf{h}_{\mathrm{a}}}}(n), \tag{2.162}$$

$$\mathbf{R}_{\mathbf{c}_{\mathbf{g}}}(n) = \left[\mathbf{I}_M - \mathbf{k}_{\mathbf{g}}(n)\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\right]\mathbf{R}_{\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}}(n). \tag{2.163}$$

Summarizing, the KF for BF (namely KF-BF) is defined by equations (2.154), (2.155), (2.156), (2.157), and (2.160)–(2.163). As we can notice, the computational complexity of this algorithm is proportional to $\mathcal{O}(L^2 + M^2)$.

2.7.2 Simplified Kalman Filter for Bilinear Forms (SKF-BF)

Next, in order to reduce the computational complexity of the KF-BF, a simplified version of this algorithm is derived. The idea of this low complexity algorithm is inspired by the work developed in [16], in the context of echo cancellation. To begin, let us assume that the KF-BF has converged to its steady-state. Consequently, $\mathbf{R}_{\mathbf{c}_{\mathbf{h}_{\mathrm{a}}}}(n)$ and $\mathbf{R}_{\mathbf{c}_{\mathbf{g}_{\mathrm{a}}}}(n)$ tend to become diagonal matrices, which have all the elements on the main diagonal equal to small positive numbers, $\sigma^2_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n)$ and $\sigma^2_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n)$, respectively. Therefore, we can use the approximations:

$$\mathbf{R}_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n) \approx \sigma^2_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n)\mathbf{I}_L, \tag{2.164}$$

$$\mathbf{R}_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n) \approx \sigma^2_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n)\mathbf{I}_M. \tag{2.165}$$

Hence, the Kalman gain vectors for the temporal and spatial impulse responses become

$$\mathbf{k}_{\mathbf{h}}(n) = \frac{\mathbf{x}_{\widehat{\mathbf{g}}}(n)}{\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n) + \delta_{\mathbf{h}}(n)}, \tag{2.166}$$

$$\mathbf{k}_{\mathbf{g}}(n) = \frac{\mathbf{x}_{\widehat{\mathbf{h}}}(n)}{\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n) + \delta_{\mathbf{g}}(n)}, \tag{2.167}$$

where $\delta_{\mathbf{h}}(n) = \sigma_v^2/\sigma^2_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n)$ and $\delta_{\mathbf{g}}(n) = \sigma_v^2/\sigma^2_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n)$ can be seen as variable regularization parameters. Then, we use the Kalman vectors from (2.166) and (2.167) in the updates (2.154) and (2.155), respectively.

Next, a new simplification can be made, by considering that the matrices appearing in the updates of $\mathbf{R}_{\mathbf{c}_{\mathbf{h}}}(n)$ and $\mathbf{R}_{\mathbf{c}_{\mathbf{g}}}(n)$ can be approximated as

$$\mathbf{I}_L - \mathbf{k}_{\mathbf{h}}(n)\mathbf{x}^T_{\widehat{\mathbf{g}}}(n) \approx \left[1 - \frac{1}{L}\mathbf{k}^T_{\mathbf{h}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)\right]\mathbf{I}_L, \tag{2.168}$$

$$\mathbf{I}_M - \mathbf{k}_{\mathbf{g}}(n)\mathbf{x}^T_{\widehat{\mathbf{h}}}(n) \approx \left[1 - \frac{1}{M}\mathbf{k}^T_{\mathbf{g}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\right]\mathbf{I}_M. \tag{2.169}$$

We can perform these approximations because, as the filters start to converge, the misalignments of the individual coefficients tend to become uncorrelated; due to this fact, the matrices $\mathbf{R}_{\mathbf{c}_{\mathbf{h}}}(n)$ and $\mathbf{R}_{\mathbf{c}_{\mathbf{g}}}(n)$ tend to become diagonal. Using the notation:

$$\mathbf{R}_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n) \approx \sigma^2_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n)\mathbf{I}_L = r_{\mathbf{c}_{\mathbf{h}_\mathrm{a}}}(n)\mathbf{I}_L,$$

$$\mathbf{R}_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n) \approx \sigma^2_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n)\mathbf{I}_M = r_{\mathbf{c}_{\mathbf{g}_\mathrm{a}}}(n)\mathbf{I}_M, \tag{2.170}$$

together with

$$\begin{aligned} \mathbf{R}_{\mathbf{c_h}}(n) &\approx \sigma^2_{\mathbf{c_h}}(n)\mathbf{I}_L = r_{\mathbf{c_h}}(n)\mathbf{I}_L, \\ \mathbf{R}_{\mathbf{c_g}}(n) &\approx \sigma^2_{\mathbf{c_g}}(n)\mathbf{I}_M = r_{\mathbf{c_g}}(n)\mathbf{I}_M, \end{aligned} \tag{2.171}$$

we can summarize the SKF-BF in Table 2.1. As we can notice, its computational complexity is proportional to $\mathcal{O}(L+M)$, which represents an important gain as compared to KF-BF.

Table 2.1: Simplified Kalman filter for bilinear forms (SKF-BF).

Initialization:

$\widehat{\mathbf{h}}(0) = [1 \quad 0 \quad \ldots \quad 0]^T,\ \widehat{\mathbf{g}}(0) = \frac{1}{M}[1 \quad 1 \quad \ldots \quad 1]^T$

$r_{\mathbf{c_h}}(0) = \epsilon_{\mathbf{h}},\ r_{\mathbf{c_g}}(0) = \epsilon_{\mathbf{g}}$ (positive constants)

Parameters: $\sigma^2_{w_{\mathbf{h}}},\ \sigma^2_{w_{\mathbf{g}}},\ \sigma^2_v$ known or estimated

Algorithm:

$r_{\mathbf{c_{h_a}}}(n) = r_{\mathbf{c_h}}(n-1) + \sigma^2_{w_{\mathbf{h}}}(n)$

$r_{\mathbf{c_{g_a}}}(n) = r_{\mathbf{c_g}}(n-1) + \sigma^2_{w_{\mathbf{g}}}(n)$

$\delta_{\mathbf{h}}(n) = \frac{\sigma^2_v}{r_{\mathbf{c_{h_a}}}(n)}$

$\delta_{\mathbf{g}}(n) = \frac{\sigma^2_v}{r_{\mathbf{c_{g_a}}}(n)}$

$e(n) = d(n) - \mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\widehat{\mathbf{h}}(n-1) = d(n) - \mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\widehat{\mathbf{g}}(n-1)$

$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \frac{\mathbf{x}_{\widehat{\mathbf{g}}}(n)e(n)}{\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)+\delta_{\mathbf{h}}(n)}$

$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \frac{\mathbf{x}_{\widehat{\mathbf{h}}}(n)e(n)}{\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)+\delta_{\mathbf{g}}(n)}$

$r_{\mathbf{c_h}}(n) = \left\{1 - \frac{\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)}{L\left[\mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)+\delta_{\mathbf{h}}(n)\right]}\right\} r_{\mathbf{c_{h_a}}}(n)$

$r_{\mathbf{c_g}}(n) = \left\{1 - \frac{\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)}{M\left[\mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)+\delta_{\mathbf{g}}(n)\right]}\right\} r_{\mathbf{c_{g_a}}}(n)$

2.7.3 Experimental Results

Experiments are performed in the context of system identification, in order to highlight the performance of the Kalman-based algorithms for BF (KF-BF and SKF-BF), in comparison with their regular counterparts (KF and SKF). The temporal and the spatial impulse responses are randomly generated from a Gaussian distribution, having the lengths equal to $L = 64$ and $M = 8$, respectively. This leads to a length of the spatiotemporal impulse response equal to $ML = 8 \times 64 = 512$. It is also useful to evaluate the tracking capabilities of the algorithms; to this purpose, a sudden change in the temporal impulse response is applied in the middle of simulations, by generating a new random vector of length $L = 64$, also from a Gaussian distribution. The input signals $x_m(n),\ m = 1, 2, \ldots, M$ are either WGNs or AR(1) processes [which were obtained after passing a WGN through a first-order system with the transfer function $1/\left(1 - 0.8z^{-1}\right)$]. The additive noise $v(n)$ is white and Gaussian, having the variance $\sigma_v^2 = 0.01$; we assume that this parameter is available in the experiments. The performance measure is the NM (in dB) [see (2.10)], to evaluate the identification of the global impulse response.

In Figs. 2.24 and 2.25, the KF-BF is compared to the regular KF for WGN and AR(1) input signals, respectively. The specific parameters of the algorithms are set to $\sigma_{\overline{w}_{\mathbf{h}}}^2 = \sigma_{\overline{w}_{\mathbf{g}}}^2 = \sigma_w^2 = 10^{-9}$. It can be noticed from both figures that the KF-BF achieves a faster convergence rate as compared to the regular KF, for both types of input signals, providing also a better tracking capability. The gain is even more apparent in case of AR(1) inputs.

The previous experiment is repeated (for the same two types of inputs) in Figs. 2.26 and 2.27, this time comparing the SKF-BF with the regular SKF [16]. As it can be observed, the simplified versions (SKF-BF and SKF) yield a slower convergence rate [especially in case of AR(1) inputs] as compared to the full versions (KF-BF and KF, respectively); however, the computational complexities for these simplified versions are much lower. As it was expected, the SKF-BF outperforms the regular SKF in terms of the convergence rate; the improvement is much more visible in the case of AR(1) inputs.

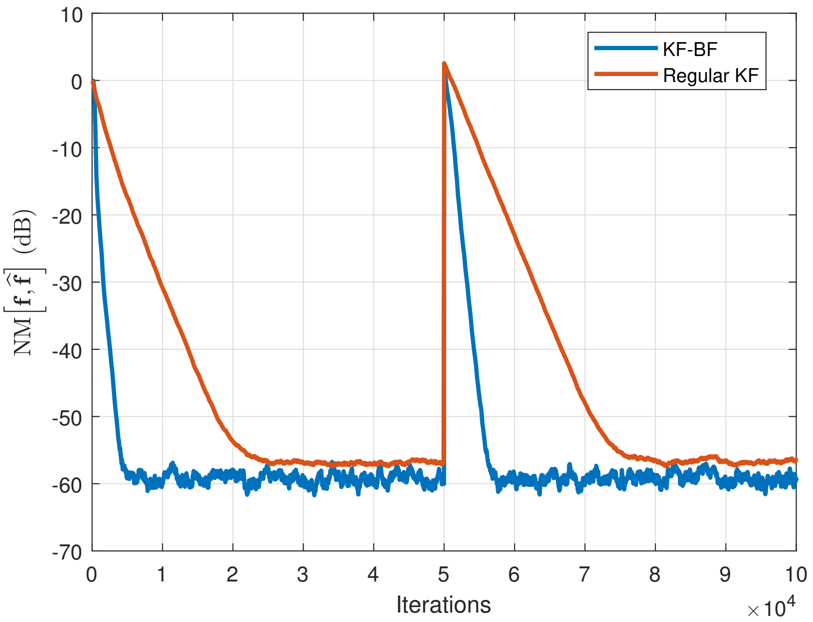


Figure 2.24: Normalized misalignment of the KF-BF and regular KF using WGNs as input signals. The length of the global impulse response is $ML = 512$. The specific parameters are set to $\sigma^2_{w_{\mathbf{h}}} = \sigma^2_{w_{\mathbf{g}}} = \sigma^2_{w} = 10^{-9}$.

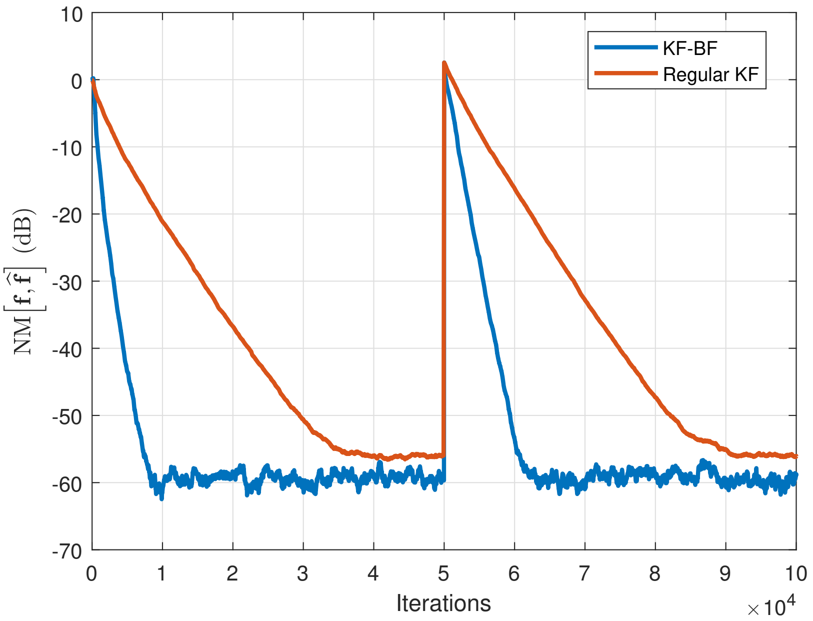


Figure 2.25: Normalized misalignment of the KF-BF and regular KF using AR(1) processes as input signals. The length of the global impulse response is $ML = 512$. The specific parameters are set to $\sigma^2_{w_{\mathbf{h}}} = \sigma^2_{w_{\mathbf{g}}} = \sigma^2_{w} = 10^{-9}$.

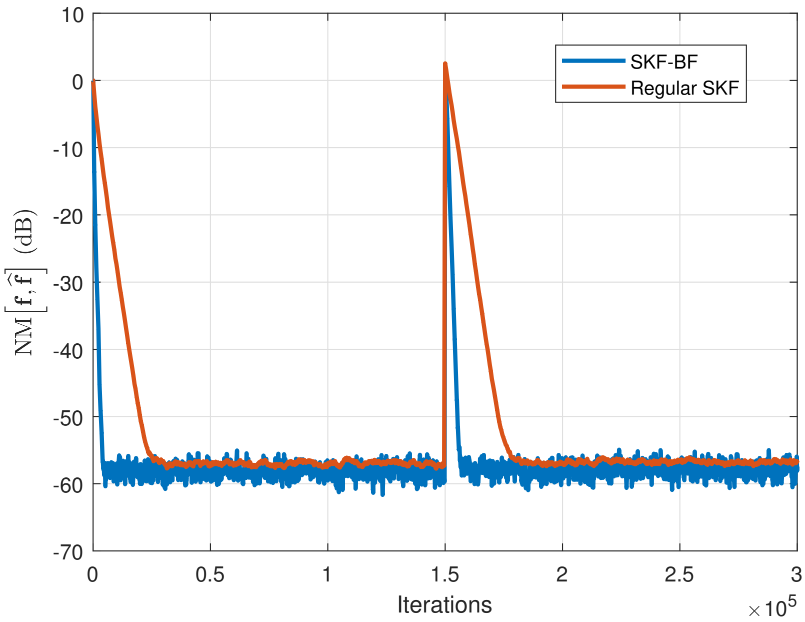


Figure 2.26: Normalized misalignment of the SKF-BF and regular SKF using WGNs as input signals. The length of the global impulse response is $ML = 512$. The specific parameters are set to $\sigma^2_{\overline{w}_\mathbf{h}} = \sigma^2_{\overline{w}_\mathbf{g}} = \sigma^2_w = 10^{-9}$.

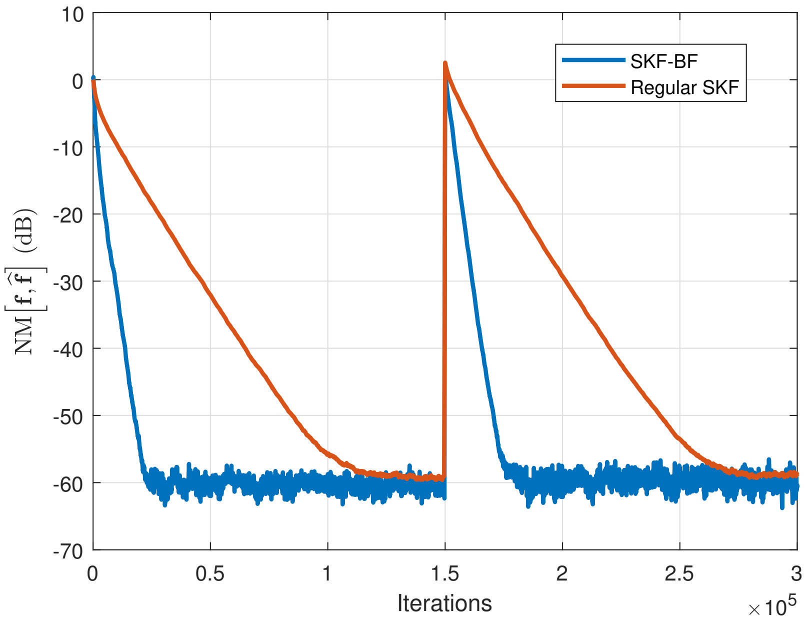


Figure 2.27: Normalized misalignment of the SKF-BF and regular SKF using AR(1) processes as input signals. The length of the global impulse response is $ML = 512$. The specific parameters are set to $\sigma^2_{\overline{w}_\mathbf{h}} = \sigma^2_{\overline{w}_\mathbf{g}} = \sigma^2_w = 10^{-9}$.

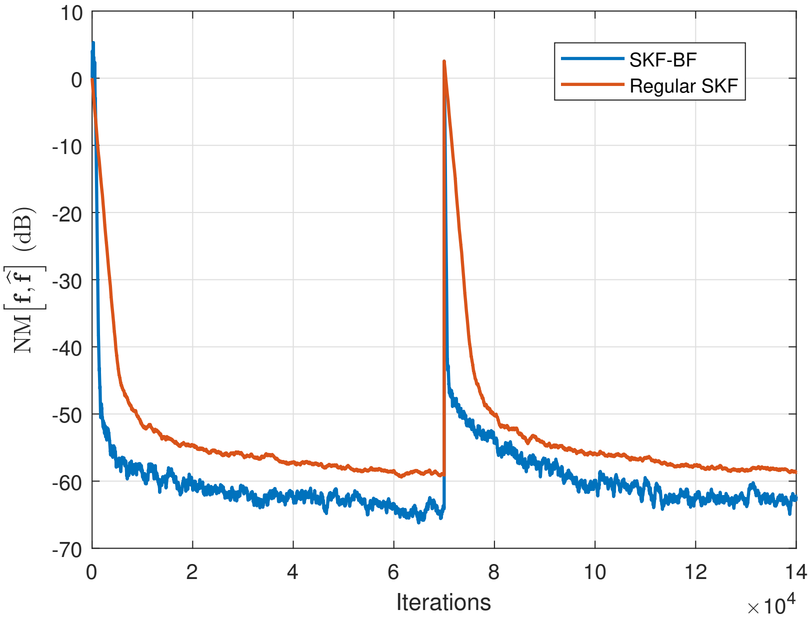


Figure 2.28: Normalized misalignment of the SKF-BF and regular SKF (for WGNs input signals), using the recursive estimates $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ and $\widehat{\sigma}^2_{w}(n)$, respectively; the SKF-BF uses $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. The length of the global impulse responses is $ML = 512$.

Next, the performance of the SKF-BF is evaluated in Figs. 2.28 and 2.29, but instead of using a constant value for $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$, as in the previous experiments, we employ a recursive estimation of this parameter, namely $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n) = \frac{1}{L}\left\|\widehat{\mathbf{h}}(n) - \widehat{\mathbf{h}}(n-1)\right\|^2$. The motivation for this choice will be provided in subchapter 2.8.1, which is dedicated to a discussion on the estimation of this parameter. The spatial impulse response is assumed to be time invariant, so that we can set $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. The regular SKF is considered for comparison, using a similar way to estimate its specific parameter, i.e., $\widehat{\sigma}^2_{w}(n) = 1/(ML)\left\|\widehat{\mathbf{f}}(n) - \widehat{\mathbf{f}}(n-1)\right\|^2$ [16]. Because of the nature of the estimators (as it will be explained in 2.8.2), the algorithms behave like variable step-size adaptive filters, achieving both low misalignment and fast convergence/tracking. Moreover, as we can notice from these two figures, the proposed SKF-BF still outperforms the regular SKF in terms of both performance criteria.

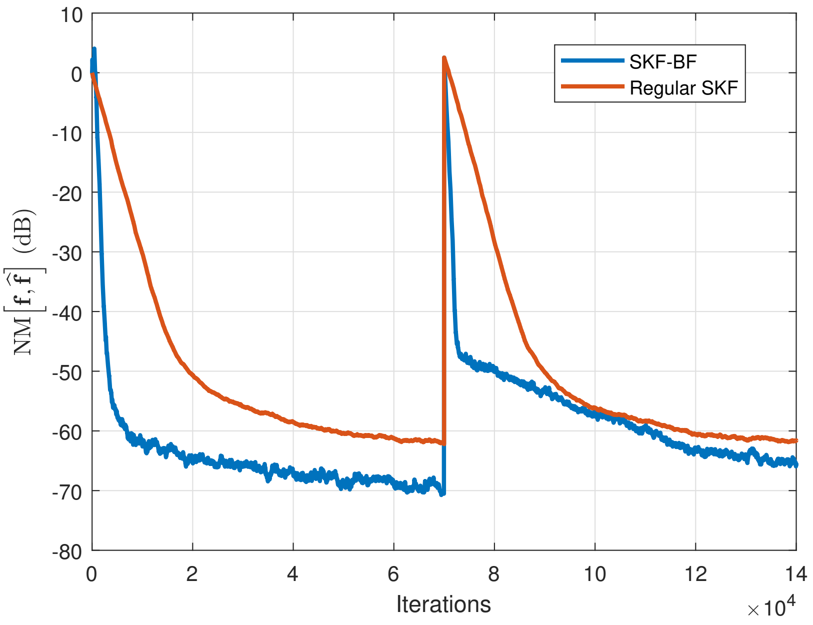


Figure 2.29: Normalized misalignment of the SKF-BF and regular SKF (for AR(1) input signals), using the recursive estimates $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ and $\widehat{\sigma}^2_{w}(n)$, respectively; the SKF-BF uses $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. The length of the global impulse responses is $ML = 512$.

2.8 A Connection Between the Kalman Filter and the Optimized LMS Algorithm for Bilinear Forms

The SKF-BF and OLMS-BF algorithms were developed following different theoretical patterns. However, there are strong similarities between these two algorithms, as will be explained in this section, which contains information published in [22].

The update equations of the SKF-BF are given by (2.154) and (2.155), where the Kalman gain vectors have the expressions in (2.166) and (2.167). It can be noticed that the updates of the SKF-BF can be expressed as

$$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}},\mathrm{K}}(n) x_{\widehat{\mathbf{h}}}(n) e(n), \tag{2.172}$$

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}},\mathrm{K}}(n) x_{\widehat{\mathbf{g}}}(n) e(n), \tag{2.173}$$

where the Kalman step-size parameters are

$$\mu_{\widehat{\mathbf{g}},\mathrm{K}}(n) = \frac{1}{\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)\left\{1 + \frac{M\sigma_v^2}{\left[m_{\mathbf{g}}(n-1)+M\sigma_{\overline{w}_{\mathbf{g}}}^2\right]\mathbf{x}_{\widehat{\mathbf{h}}}^T(n)\mathbf{x}_{\widehat{\mathbf{h}}}(n)}\right\}}, \tag{2.174}$$

$$\mu_{\widehat{\mathbf{h}},\mathrm{K}}(n) = \frac{1}{\mathbf{x}_{\widehat{\mathbf{g}}}^T(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)\left\{1 + \frac{L\sigma_v^2}{\left[m_{\mathbf{h}}(n-1)+L\sigma_{\overline{w}_{\mathbf{h}}}^2\right]\mathbf{x}_{\widehat{\mathbf{g}}}^T(n)\mathbf{x}_{\widehat{\mathbf{g}}}(n)}\right\}}. \tag{2.175}$$

Comparing these parameters with the optimal step-sizes from (2.121) and (2.122) (also taking (2.103) and (2.104) into account), we can notice striking resemblances between SKF-BF and OLMS-BF. In fact, these two algorithms are very similar when

$$p_{\mathbf{g}} = p_{\mathbf{h}} = 0. \tag{2.176}$$

On the other hand, as it was indicated in [35], this could represent a reasonable assumption, since

$$\lim_{n\to\infty} p_{\mathbf{g}}(n) = \lim_{n\to\infty} p_{\mathbf{h}}(n) = 0, \tag{2.177}$$

suggesting that in the steady-state of the algorithm, the influence of the terms $p_{\mathbf{h}}$ and $p_{\mathbf{g}}$ on the step-size parameters diminishes. As will be supported in simulations, (2.176) can be fairly imposed within the OLMS-BF algorithm, while still leading to a very good compromise between the performance criteria (e.g., convergence rate versus misadjustment). Under these considerations, the OLMS-BF algorithm is summarized in Table 2.2 (in a practical form that facilitates its implementation).

Table 2.2: Optimized LMS algorithm for bilinear forms (OLMS-BF) (practical version, i.e., using (2.176)).

Initialization:

$\widehat{\mathbf{h}}(0) = [1 \quad 0 \quad \dots \quad 0]^T, \ \widehat{\mathbf{g}}(0) = \frac{1}{M}[1 \quad 1 \quad \dots \quad 1]^T$

$m_{\mathbf{h}}(0) = \epsilon_{\mathbf{h}}, \ m_{\mathbf{g}}(0) = \epsilon_{\mathbf{g}}$ (positive constants)

Parameters: $\mathbb{E}\big[\|\mathbf{h}(n)\|^2\big], \ \mathbb{E}\big[\|\mathbf{g}(n)\|^2\big], \ \sigma^2_{\overline{w}_{\mathbf{h}}}, \ \sigma^2_{\overline{w}_{\mathbf{g}}}, \ \sigma^2_v$ known or estimated

Algorithm:

$\sigma^2_{v_{\mathbf{h}}}(n) = M\sigma^2_{\mathbf{x}}\mathbb{E}\big[\|\mathbf{h}(n)\|^2\big] \left[m_{\mathbf{g}}(n-1) + M\sigma^2_{\overline{w}_{\mathbf{g}}}\right]$

$\sigma^2_{v_{\mathbf{g}}}(n) = L\sigma^2_{\mathbf{x}}\mathbb{E}\left[\|\mathbf{g}(n)\|^2\right] \left[m_{\mathbf{h}}(n-1) + L\sigma^2_{\overline{w}_{\mathbf{h}}}\right]$

$A_{\mathbf{h}} = \sigma^2_{\mathbf{x}}\mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right] \left[m_{\mathbf{h}}(n-1) + L\sigma^2_{\overline{w}_{\mathbf{h}}}\right]$

$B_{\mathbf{h}} = L\sigma^2_{\mathbf{x}}\mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right] \left\{\sigma^2_v + \frac{1}{M}\sigma^2_{v_{\mathbf{h}}}(n) + \sigma^2_{\mathbf{x}}\mathbb{E}\left[\|\widehat{\mathbf{g}}(n-1)\|^2\right] \left[m_{\mathbf{h}}(n-1) + L\sigma^2_{\overline{w}_{\mathbf{h}}}\right]\right\}$

$A_{\mathbf{g}} = \sigma^2_{\mathbf{x}}\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right] \left[m_{\mathbf{g}}(n-1) + M\sigma^2_{\overline{w}_{\mathbf{g}}}\right]$

$B_{\mathbf{g}} = M\sigma^2_{\mathbf{x}}\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right] \left\{\sigma^2_v + \frac{1}{L}\sigma^2_{v_{\mathbf{g}}}(n) + \sigma^2_{\mathbf{x}}\mathbb{E}\left[\left\|\widehat{\mathbf{h}}(n-1)\right\|^2\right] \left[m_{\mathbf{g}}(n-1) + M\sigma^2_{\overline{w}_{\mathbf{g}}}\right]\right\}$

$\mu_{\widehat{\mathbf{h}},\mathrm{o}} = \frac{A_{\mathbf{h}}}{B_{\mathbf{h}}}$

$\mu_{\widehat{\mathbf{g}},\mathrm{o}} = \frac{A_{\mathbf{g}}}{B_{\mathbf{g}}}$

$e(n) = d(n) - \mathbf{x}^T_{\widehat{\mathbf{g}}}(n)\widehat{\mathbf{h}}(n-1) = d(n) - \mathbf{x}^T_{\widehat{\mathbf{h}}}(n)\widehat{\mathbf{g}}(n-1)$

$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu_{\widehat{\mathbf{h}},\mathrm{o}}x_{\widehat{\mathbf{g}}}(n)e(n)$

$\widehat{\mathbf{g}}(n) = \widehat{\mathbf{g}}(n-1) + \mu_{\widehat{\mathbf{g}},\mathrm{o}}x_{\widehat{\mathbf{h}}}(n)e(n)$

$m_{\mathbf{h}}(n) = m_{\mathbf{h}}(n-1) - 2A_{\mathbf{h}}\mu_{\widehat{\mathbf{h}},\mathrm{o}} + \mu^2_{\widehat{\mathbf{h}},\mathrm{o}}B_{\mathbf{h}} + L\sigma^2_{\overline{w}_{\mathbf{h}}}$

$m_{\mathbf{g}}(n) = m_{\mathbf{g}}(n-1) - 2A_{\mathbf{g}}\mu_{\widehat{\mathbf{g}},\mathrm{o}} + \mu^2_{\widehat{\mathbf{g}},\mathrm{o}}B_{\mathbf{g}} + M\sigma^2_{\overline{w}_{\mathbf{g}}}$

2.8.1 Noise Variance Estimation

Next, a few important observations ought to be made regarding the specific parameters that must be set within the algorithms. Here, the noise power σ^2_v is required in order to compute the Kalman gain vectors (for KF-BF and SKF-BF) or the optimal step-sizes (for OLMS-BF). In practice, we can estimate this parameter in different ways; some simple and efficient methods for this purpose are presented in [64, 65].

The parameters related to the uncertainties in the unknown systems also need to be set or estimated, i.e., $\sigma^2_{\overline{w}_{\mathbf{h}}}$ and $\sigma^2_{\overline{w}_{\mathbf{g}}}$. Choosing small values for these parameters yields a small misalignment, but at the same time a poor tracking. On the other hand, large values (meaning that there are high uncertainties in the unknown systems) lead to a good tracking but also a high misalignment. This means that we always need to have a good compromise between fast tracking and low misalignment. In practice, if we have some a priori information about the systems which we need to identify, we can take it into consideration when setting the values of these parameters. For example, if we assume the spatial impulse response to be time-invariant, we could fix $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$ and tune only the parameter related to the temporal impulse response. Thus, based on the state equation related to $\mathbf{h}(n)$, together with the approximation $\left\|\overline{\mathbf{w}}_{\mathbf{h}}(n)\right\|^2 \approx L\sigma^2_{\overline{w}_{\mathbf{h}}}$ (which is valid when $L \gg 1$), and replacing $\mathbf{h}(n)$ and $\mathbf{h}(n-1)$ by their estimates, we can evaluate

$$\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n) = \frac{1}{L}\left\|\widehat{\mathbf{h}}(n) - \widehat{\mathbf{h}}(n-1)\right\|^2. \tag{2.178}$$

It can be noticed that the estimation from (2.178) is designed to achieve a proper compromise between good tracking and low misalignment. When the algorithm starts to converge or when there is an abrupt change of the system, the difference between $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{h}}(n-1)$ is significant, leading to large values of the parameter $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$, therefore providing fast convergence and tracking. On the contrary, when the algorithm is converging to its steady-state, the difference between $\widehat{\mathbf{h}}(n)$ and $\widehat{\mathbf{h}}(n-1)$ reduces, thus leading to the parameter $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ taking small values and, consequently, to a low misalignment.

2.8.2 Computational Complexity

The previously developed algorithms are designed to identify the individual impulse responses of the bilinear form. The global (spatiotemporal) impulse response can be computed based on the Kronecker product between them. An alternative solution is to use the regular KF to identify the spatiotemporal impulse response directly, relying on the observation (2.4) and identifying the state equation:

$$\mathbf{f}(n) = \mathbf{f}(n-1) + \mathbf{w}(n), \tag{2.179}$$

where $\mathbf{w}(n)$ is a zero-mean WGN signal vector. The covariance matrix of $\mathbf{w}(n)$ is $\mathbf{R}_{\mathbf{w}}(n) = \sigma_w^2 \mathbf{I}_{ML}$, where $\mathbf{I}_{ML}$ is the identity matrix of size $ML \times ML$ and the variance σ_w^2 captures the uncertainties in $\mathbf{f}(n)$.

In this way, following the approach from [16], we can easily derive the regular KF (KF) and its simplified version (namely SKF), which can identify the global impulse response using a single adaptive filter $\widehat{\mathbf{f}}(n)$; for further details, please see Sections VI and VII in [16]. However, we need to mention that the solution found using the regular KF and SKF involves an adaptive filter of length ML, whereas their counterparts tailored for BF (i.e., KF-BF and SKF-BF) use two shorter filters of lengths L and M, respectively. As a consequence, besides a lower computational complexity, a much faster converge rate and tracking are expected for the bilinear algorithms with respect to the conventional ones. The same ideas apply for the OLMS-BF algorithm, as compared to its regular counterpart, i.e., the JO-NLMS algorithm [58], which could be used to identify the global impulse response $\widehat{\mathbf{f}}(n)$.

The computational complexity of the previously discussed algorithms is summarized in Table 2.3. It can be easily seen that the SKF-BF offers a great reduction in terms of complexity with respect to KF-BF. Also, the SKF-BF and OLMS-BF differ only by a small number of operations, thus confirming the similarity that was highlighted in subchapter 2.8. Finally, when $ML \gg M + L$ (which is usually the case in practice), we can notice that the algorithms tailored for BF (namely KF-BF, SKF-BF, and OLMS-BF) offer lower computational complexities as compared to their regular counterparts (i.e., KF, SKF, and JO-NLMS, respectively).

Table 2.3: Computational complexity of the algorithms.

Algorithms	$\times$	$+$	$/$
KF-BF [19]	$3(L^2{+}M^2){+}2ML{+}3L{+}4M$	$2(L^2{+}M^2){+}2ML{+}L{+}2M$	2
KF [16]	$3(ML)^2 + 4ML$	$2(ML)^2 + 3ML$	1
SKF-BF (Table 2.1)	$2ML + 2L + 3M + 6$	$2ML + L + 2M + 4$	2
SKF [16]	$3ML + 6$	$3ML + 4$	1
OLMS-BF (Table 2.2)	$2ML + 2L + 3M + 12$	$2ML + L + 2M + 8$	2
JO-NLMS [58]	$3ML + 6$	$3ML + 5$	1

2.8.3 Experimental Results

We performed experiments in order to outline the strong similarities between the SKF-BF and OLMS-BF algorithms. In Figs. 2.30 and 2.31, we compare the performances of these algorithms using two types of input signals, i.e., WGNs and AR(1) processes, respectively. Both algorithms use the recursive estimate $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ [from (2.178)] and $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. As we can notice, the SKF-BF and OLMS-BF algorithms behave quite similar, especially when the input signals are WGNs (Fig. 2.30). When the input signals are AR(1) processes (Fig. 2.31), the SKF-BF outperforms the OLMS-BF in terms of the initial convergence rate; however, it pays with a slower tracking reaction. Nevertheless, the overall performances of these algorithms are very similar, as supported by the comparison provided before in this subchapter.

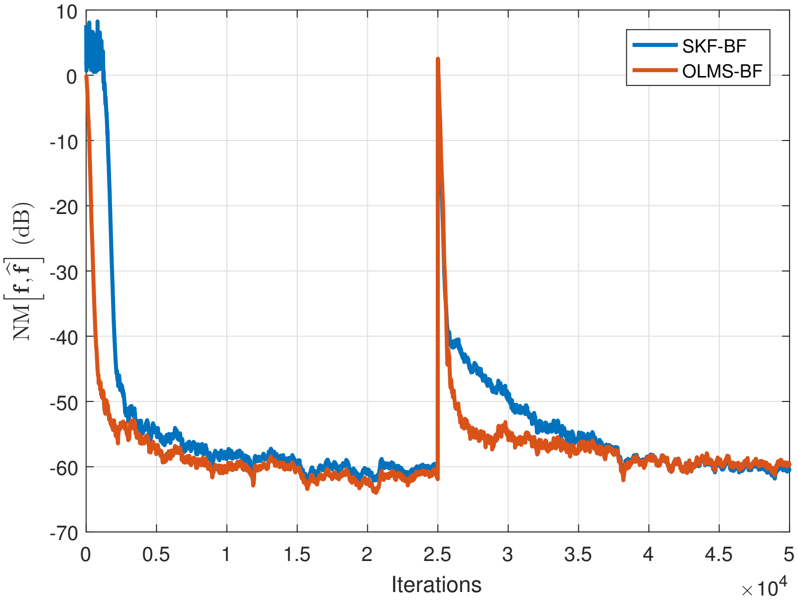


Figure 2.30: Normalized misalignment of the SKF-BF and OLMS-BF algorithms using WGNs as input signals. Both algorithms use the recursive estimate $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ and $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. The length of the global impulse responses is $ML = 512$.

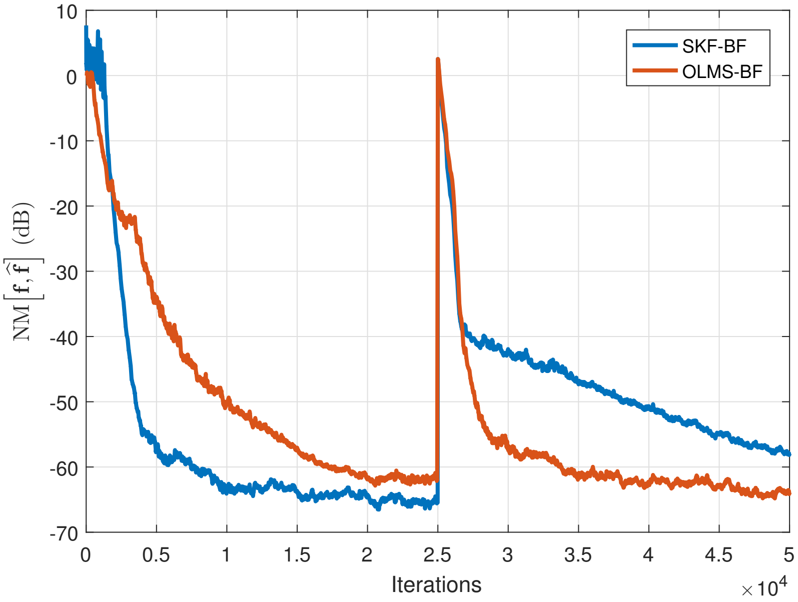


Figure 2.31: Normalized misalignment of the SKF-BF and OLMS-BF algorithms using AR(1) processes as input signals. Both algorithms use the recursive estimate $\widehat{\sigma}^2_{\overline{w}_{\mathbf{h}}}(n)$ and $\sigma^2_{\overline{w}_{\mathbf{g}}} = 0$. The length of the global impulse responses is $ML = 512$.

2.9 Summary and Conclusions

In Chapter 2, we started by reviewing the existing advances in the context of system identification of BF. After thoroughly defining the way in which BF are understood (in the context of a spatiotemporal model) and the system model in 2.1, along with the already existing Wiener filter and LMS-type adaptive algorithms in 2.2 and 2.3, we continued by introducing the original contributions. Subchapter 2.4 presented a proportionate APA tailored for the identification of sparse BF, namely IPAPA-BF. This algorithm follows the line of the celebrated IPNLMS algorithm [48], in terms of computing the proportionate factors. The IPAPA-BF outperforms its non-proportionate counterpart, APA-BF, especially in terms of convergence rate. In addition, the proposed IPAPA-BF outperforms the regular IPAPA, achieving a faster convergence rate but also a lower computational complexity. Simulation results indicated the appealing performance of the proposed algorithm, in the context of bilinear sparse system identification.

The mathematical derivation of the OLMS-BF algorithm was conducted in 2.5, following an optimization of the step-size parameter value. Experimental results indicate that the proposed algorithm outperforms both the NLMS-BF and JO-NLMS algorithms, yielding to the conclusion that it can be successfully used for bilinear system identification problems.

In subchapter 2.6 we proved, both analytically and by simulations, that the statement "after the algorithm has started to converge, the SMCM tends to a scaled unit matrix" is rigorously valid only for a white noise input signal. In the case of a correlated input signal, the elements outside the main diagonal have nonzero values, which increase with the increase of the external noise power and step size. At the same time, a certain variation of the elements on the main diagonal occurs. It can be roughly stated that the lower the autocorrelation of the input signal is, the better is the validity of the above assumption.

Next, in 2.7, we have presented the KF tailored for the identification of BF (namely KF-BF), where the bilinear term has been defined with respect to the impulse responses of the spatiotemporal model. We further derived a simplified version of this algorithm, referred to as SKF-BF, which offers a reduced computational complexity; the price to pay is a slower convergence rate, especially for correlated inputs. Simulation results indicated that the proposed algorithms could represent appealing solutions for such bilinear system identification problems.

In 2.8 it was shown that the SKF-BF and OLMS-BF algorithms perform very similarly. Experimental results also indicate that the algorithms tailored for BF outperform their regular counterparts (in such two-dimensional system identification scenarios), in terms of both convergence rate and tracking, as well as the steady-state misalignment. Adding to that the reduced computational amount provided by the use of two shorter adaptive filters instead of a single (much longer) one, we can conclude that the proposed solutions are useful in the context of system identification with BF.

Chapter 3

Trilinear Forms (TF)

This chapter contains the work developed in the context of system identification for trilinear forms (TF). It begins with a short review on tensors in subchapter 3.1, which are needed for expressing mathematically the TF. Next, the system model for TF is introduced in 3.2, and in this framework, the Wiener filter for TF is developed in 3.3. The LMS algorithm tailored for TF (LMS-TF), along with its normalized version, NLMS-TF, are then developed in 3.4, while in the end of the chapter, in 3.5, we provide a related discussion and conclusions. The results presented in this chapter were published in [66, 67].

In Chapter 2, we made use of a new approach introduced recently in [34], where the bilinear term is considered within the framework of a MISO system, and it is defined with respect to the spatiotemporal model's impulse responses. A particular case of this type of system is the Hammerstein model [18]; in this scenario, a single-input signal passes through a nonlinear block and a linear system, which are cascaded. The bilinear approach is suitable for a particular form of the decomposition, which involves only two terms (i.e., two systems). In some cases, it would be useful to exploit a higher-order decomposition, which could improve the overall performance in terms of both complexity and efficiency.

Motivated by the good performance of the previously mentioned approaches in the study of BF, we further extend this approach to higher-order multilinear in parameters systems. Applications, such as multichannel equalization [37], nonlinear acoustic devices for echo cancellation [39], multiple-input/multiple-output (MIMO) communication systems [68, 69] and others, can be addressed within the framework of multilinear systems. Because many of these applications can be formulated in terms of system identification problems [70], it is of interest to estimate a model based on the available and observed data, which are usually the input and the output of the system.

3.1 Background on Tensors

TF are understood as an extension of the previously studied BF, and they are expressed and studied using tensors. For this reason, we first need to recall the notion of tensor, along with the basic operations which can be done using these mathematical tools. This introductory subchapter was published before in [66, 67].

A tensor is a generalization of a matrix; it is a multidimensional array of data whose entries are referred by using multiple indices [71, 72]. The notation used for a tensor, a matrix, a vector, and a scalar is $\boldsymbol{\mathcal{A}}$, $\mathbf{A}$, $\mathbf{a}$, and a, respectively. In this chapter, we are only interested in the third-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$, meaning that its elements are real-valued and its dimension is $L_1 \times L_2 \times L_3$. In particular, a matrix and a vector are second- and first-order tensors, respectively, with $\mathbf{A} \in \mathbb{R}^{L_1 \times L_2}$ and $\mathbf{a} \in \mathbb{R}^{L_1}$. For a third-order tensor, the first and second indices l_1 and l_2 correspond to the row and column, respectively, as in a matrix, while the third index l_3 corresponds to the tube and describes its depth. These three indices describe the three different modes. The entries of the different order tensors are denoted by

$$(\boldsymbol{\mathcal{A}})_{l_1 l_2 l_3} = a_{l_1 l_2 l_3},$$

$$(\mathbf{A})_{l_1 l_2} = a_{l_1 l_2},$$
$$(\mathbf{a})_{l_1} = a_{l_1},$$

for $l_1 = 1, 2, \ldots, L_1$, $l_2 = 1, 2, \ldots, L_2$, and $l_3 = 1, 2, \ldots, L_3$. The tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ is diagonal if $a_{l_1 l_2 l_3} \neq 0$ only for $l_1 = l_2 = l_3$.

The notion of vectorization, which consists of transforming a matrix into a vector, is very well-known. Matricization does somewhat the same thing but from a third-order tensor into a large matrix. Depending on which index whose all elements are considered first, we have matricization along three different modes [73, 74]:

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

where $\mathbf{A}_{::l_3} \in \mathbb{R}^{L_1 \times L_2}$, $l_3 = 1, 2 \ldots, L_3$ are the frontal slices. In $\mathbf{A}_{[1]}$, we consider the mode row and then vary the columns and the tubes. In $\mathbf{A}_{[2]}$, we take the mode column and then vary the lines and the tubes. Finally, in $\mathbf{A}_{[3]}$, we take the mode tube and then vary the rows and the columns. The mode-1, mode-2, and mode-3 ranks are the ranks of $\mathbf{A}_{[1]}$, $\mathbf{A}_{[2]}$, and $\mathbf{A}_{[3]}$, respectively. The vectorization of a tensor is

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

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

$$\boldsymbol{\mathcal{B}} = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3, \tag{3.1}$$

where $\circ$ is the vector outer product and the elements of $\boldsymbol{\mathcal{B}}$ are given by

$$(\boldsymbol{\mathcal{B}})_{l_1 l_2 l_3} = b_{1l_1} b_{2l_2} b_{3l_3}. \tag{3.2}$$

The frontal slices of $\boldsymbol{\mathcal{B}}$ in (3.1) are $\mathbf{B}_{::l_3} = \mathbf{b}_1 \mathbf{b}_2^T b_{3l_3} \in \mathbb{R}^{L_1 \times L_2}$, $l_3 = 1, 2 \ldots, L_3$. In particular, we have $\mathbf{B} = \mathbf{b}_1 \circ \mathbf{b}_2 = \mathbf{b}_1 \mathbf{b}_2^T$, where ^{T} is the transpose operator. For complex-valued signals, the transpose operator is usually replaced by the complex-conjugate operator, ^{H}, but this is only a convenient convention. Therefore, the rank of a tensor $\boldsymbol{\mathcal{A}}$, denoted $\operatorname{rank}(\boldsymbol{\mathcal{A}})$, is defined as the minimum number of rank-1 tensors that generate $\boldsymbol{\mathcal{A}}$ as their sum. For example, if

$$\boldsymbol{\mathcal{B}} = \sum_{r=1}^{R} \mathbf{b}_{1r} \circ \mathbf{b}_{2r} \circ \mathbf{b}_{3r}, \tag{3.3}$$

then $\operatorname{rank}(\boldsymbol{\mathcal{B}}) = R$ when R is minimal, and (3.3) is called the canonical polyadic decomposition (CPD) of $\boldsymbol{\mathcal{B}}$. There are some fundamental differences between the rank of the matrix $\mathbf{A} \in \mathbb{R}^{L_1 \times L_2}$ and the rank of the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$. The rank of $\mathbf{A}$ can never be larger than $\min\{L_1, L_2\}$ while the rank of $\boldsymbol{\mathcal{A}}$ can be greater than $\min\{L_1, L_2, L_3\}$.

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

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

As a consequence, the Frobenius norm is

$$\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} = \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle} = \left\|\mathbf{A}_{[1]}\right\|_{\mathrm{F}} = \left\|\mathbf{A}_{[2]}\right\|_{\mathrm{F}} = \left\|\mathbf{A}_{[3]}\right\|_{\mathrm{F}}.$$

If $\boldsymbol{\mathcal{A}} = \mathbf{a}_1 \circ \mathbf{a}_2 \circ \mathbf{a}_3$ and $\boldsymbol{\mathcal{B}} = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3$, then

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \mathbf{b}_1^T \mathbf{a}_1 \times \mathbf{b}_2^T \mathbf{a}_2 \times \mathbf{b}_3^T \mathbf{a}_3. \tag{3.4}$$

Therefore,

$$\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} = \|\mathbf{a}_1\| \times \|\mathbf{a}_2\| \times \|\mathbf{a}_3\|, \tag{3.5}$$

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

It is important to be able to multiply a tensor with a matrix [75, 76]. Let the tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ and the matrix $\mathbf{M}_1 \in \mathbb{R}^{M_1 \times L_1}$. The mode-1 product between the tensor $\boldsymbol{\mathcal{A}}$ and the matrix $\mathbf{M}_1$ gives the tensor:

$$\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{A}} \times_1 \mathbf{M}_1,\ \boldsymbol{\mathcal{U}} \in \mathbb{R}^{M_1 \times L_2 \times L_3}, \tag{3.6}$$

whose entries are

$$u_{m_1 l_2 l_3} = \sum_{l_1=1}^{L_1} a_{l_1 l_2 l_3} m_{m_1 l_1}, \tag{3.7}$$

for $m_1 = 1, 2, \ldots, M_1$, and $\mathbf{U}_{[1]} = \mathbf{M}_1 \mathbf{A}_{[1]}$. We have

$$\|\boldsymbol{\mathcal{A}} \times_1 \mathbf{M}_1\|_{\mathrm{F}} \leq \|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} \|\mathbf{M}_1\|_{\mathrm{F}}.$$

In the same way, with the matrix $\mathbf{M}_2 \in \mathbb{R}^{M_2 \times L_2}$, the mode-2 product between the tensor $\boldsymbol{\mathcal{A}}$ and the matrix $\mathbf{M}_2$ gives the tensor:

$$\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{A}} \times_2 \mathbf{M}_2,\ \boldsymbol{\mathcal{U}} \in \mathbb{R}^{L_1 \times M_2 \times L_3}, \tag{3.8}$$

whose entries are

$$u_{l_1 m_2 l_3} = \sum_{l_2=1}^{L_2} a_{l_1 l_2 l_3} m_{m_2 l_2}, \tag{3.9}$$

for $m_2 = 1, 2, \ldots, M_2$, and $\mathbf{U}_{[2]} = \mathbf{M}_2 \mathbf{A}_{[2]}$. We have

$$\|\boldsymbol{\mathcal{A}} \times_2 \mathbf{M}_2\|_{\mathrm{F}} \leq \|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} \|\mathbf{M}_2\|_{\mathrm{F}} .$$

Finally, with the matrix $\mathbf{M}_3 \in \mathbb{R}^{M_3 \times L_3}$, the mode-3 product between the tensor $\boldsymbol{\mathcal{A}}$ and the matrix $\mathbf{M}_3$ gives the tensor:

$$\boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{A}} \times_3 \mathbf{M}_3,\ \boldsymbol{\mathcal{U}} \in \mathbb{R}^{L_1 \times L_2 \times M_3}, \tag{3.10}$$

whose entries are

$$u_{l_1 l_2 m_3} = \sum_{l_3=1}^{L_3} a_{l_1 l_2 l_3} m_{m_3 l_3}, \tag{3.11}$$

for $m_3 = 1, 2, \ldots, M_3$, and $\mathbf{U}_{[3]} = \mathbf{M}_3 \mathbf{A}_{[3]}$. We have

$$\|\boldsymbol{\mathcal{A}} \times_3 \mathbf{M}_3\|_{\mathrm{F}} \leq \|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}} \|\mathbf{M}_3\|_{\mathrm{F}} .$$

We can multiply the tensor $\boldsymbol{\mathcal{A}}$ with the three previous defined matrices $\mathbf{M}_1$, $\mathbf{M}_2$, and $\mathbf{M}_3$. We get the tensor:

$$\begin{aligned} \boldsymbol{\mathcal{U}} &= \boldsymbol{\mathcal{A}} \times_1 \mathbf{M}_1 \times_2 \mathbf{M}_2 \times_3 \mathbf{M}_3 = \boldsymbol{\mathcal{A}} \times_1 \mathbf{M}_1 \times_3 \mathbf{M}_3 \times_2 \mathbf{M}_2 \\ &= \boldsymbol{\mathcal{A}} \times_2 \mathbf{M}_2 \times_1 \mathbf{M}_1 \times_3 \mathbf{M}_3 = \boldsymbol{\mathcal{A}} \times_2 \mathbf{M}_2 \times_3 \mathbf{M}_3 \times_1 \mathbf{M}_1 \\ &= \boldsymbol{\mathcal{A}} \times_3 \mathbf{M}_3 \times_1 \mathbf{M}_1 \times_2 \mathbf{M}_2 = \boldsymbol{\mathcal{A}} \times_3 \mathbf{M}_3 \times_2 \mathbf{M}_2 \times_1 \mathbf{M}_1, \\ \boldsymbol{\mathcal{U}} &\in \mathbb{R}^{M_1 \times M_2 \times M_3}. \end{aligned} \tag{3.12}$$

Let

$$\boldsymbol{\mathcal{B}} = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3, \tag{3.13}$$

where $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ are vectors of length L_1, L_2, and L_3, respectively. It is easy to check that

$$\boldsymbol{\mathcal{B}} \times_1 \mathbf{M}_1 = \mathbf{M}_1\mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3,$$

$$\boldsymbol{\mathcal{B}} \times_2 \mathbf{M}_2 = \mathbf{b}_1 \circ \mathbf{M}_2\mathbf{b}_2 \circ \mathbf{b}_3,$$

$$\boldsymbol{\mathcal{B}} \times_3 \mathbf{M}_3 = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{M}_3\mathbf{b}_3.$$

The tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ can always be decomposed as [77]

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{B}} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \times_3 \mathbf{S}_3, \tag{3.14}$$

where $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ is called the core tensor, and $\mathbf{S}_1 = \begin{bmatrix} \mathbf{s}_{11} & \cdots & \mathbf{s}_{1J_1} \end{bmatrix} \in \mathbb{R}^{L_1 \times J_1}$, $\mathbf{S}_2 = \begin{bmatrix} \mathbf{s}_{21} & \cdots & \mathbf{s}_{2J_2} \end{bmatrix} \in \mathbb{R}^{L_2 \times J_2}$, and $\mathbf{S}_3 = \begin{bmatrix} \mathbf{s}_{31} & \cdots & \mathbf{s}_{3J_3} \end{bmatrix} \in \mathbb{R}^{L_3 \times J_3}$ are the side matrices. Equivalently, (3.14) can be rewritten as

$$\boldsymbol{\mathcal{A}} = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \sum_{j_3=1}^{J_3} b_{j_1 j_2 j_3} \mathbf{s}_{1j_1} \circ \mathbf{s}_{2j_2} \circ \mathbf{s}_{3j_3}. \tag{3.15}$$

In particular, if $\boldsymbol{\mathcal{B}}$ is diagonal, i.e., $b_{j_1 j_2 j_3} = 0$ except when $j_1 = j_2 = j_3$, then

$$\boldsymbol{\mathcal{A}} = \sum_{r=1}^{R} b_{rrr} \mathbf{s}_{1r} \circ \mathbf{s}_{2r} \circ \mathbf{s}_{3r}, \tag{3.16}$$

where $R = \min\{J_1, J_2, J_3\}$. From the decomposition in (3.14), many approximations can be made, which lead to some well-known and popular models.

Let $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ be vectors of length L_1, L_2, and L_3, respectively, the multiplication of $\boldsymbol{\mathcal{A}}$ with the row vectors $\mathbf{b}_1^T$, $\mathbf{b}_2^T$, and $\mathbf{b}_3^T$ gives the scalar:

$$c = \boldsymbol{\mathcal{A}} \times_1 \mathbf{b}_1^T \times_2 \mathbf{b}_2^T \times_3 \mathbf{b}_3^T = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \sum_{l_3=1}^{L_3} a_{l_1 l_2 l_3} b_{1l_1} b_{2l_2} b_{3l_3}. \tag{3.17}$$

In particular, we have

$$c = \mathbf{A} \times_1 \mathbf{b}_1^T \times_2 \mathbf{b}_2^T = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} a_{l_1 l_2} b_{1l_1} b_{2l_2} = \mathbf{b}_1^T \mathbf{A} \mathbf{b}_2 \tag{3.18}$$

and

$$c = \mathbf{a} \times_1 \mathbf{b}_1^T = \sum_{l_1=1}^{L_1} a_{l_1} b_{1l_1} = \mathbf{b}_1^T \mathbf{a}. \tag{3.19}$$

It is easy to check that (3.17), (3.18), and (3.19) are trilinear (with respect to $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$), bilinear (with respect to $\mathbf{b}_1$ and $\mathbf{b}_2$), and linear (with respect to $\mathbf{b}_1$) forms, respectively.

We can express (3.18) as

$$\begin{aligned} c &= \text{tr}\left(\mathbf{b}_2 \mathbf{b}_1^T \mathbf{A}\right) = \text{tr}\left[\left(\mathbf{b}_1 \mathbf{b}_2^T\right)^T \mathbf{A}\right] = \text{vec}^T\left(\mathbf{b}_1 \mathbf{b}_2^T\right) \text{vec}\left(\mathbf{A}\right) \\ &= \left(\mathbf{b}_2 \otimes \mathbf{b}_1\right)^T \text{vec}\left(\mathbf{A}\right), \end{aligned} \tag{3.20}$$

where tr(·) denotes the trace of a square matrix and ⊗ is the Kronecker product.

Expression (3.17) can also be written in a more convenient way. Indeed, we have

$$\begin{aligned} c &= \langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \text{vec}^T\left(\boldsymbol{\mathcal{B}}\right) \text{vec}\left(\boldsymbol{\mathcal{A}}\right) = \text{vec}^T\left(\mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3\right) \text{vec}\left(\boldsymbol{\mathcal{A}}\right) \\ &= \left(\mathbf{b}_3 \otimes \mathbf{b}_2 \otimes \mathbf{b}_1\right)^T \text{vec}\left(\boldsymbol{\mathcal{A}}\right), \end{aligned} \tag{3.21}$$

where $\boldsymbol{\mathcal{B}} = \mathbf{b}_1 \circ \mathbf{b}_2 \circ \mathbf{b}_3$.

3.2 System Model

The system model was published in [66, 67]. Let us consider the output of a MISO system (with real-valued data) at the discrete-time index t defined as

$$y(t) = \boldsymbol{\mathcal{X}}(t) \times_1 \mathbf{h}_1^T \times_2 \mathbf{h}_2^T \times_3 \mathbf{h}_3^T = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \sum_{l_3=1}^{L_3} x_{l_1 l_2 l_3}(t) h_{1l_1} h_{2l_2} h_{3l_3}, \tag{3.22}$$

where the zero-mean input signals can be described in a tensorial form $\boldsymbol{\mathcal{X}}(t) \in$

$\mathbb{R}^{L_1 \times L_2 \times L_3}$, with

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

and the three impulse responses are defined by the vectors $\mathbf{h}_k,\ k = 1, 2, 3$, of lengths L_1, L_2, and L_3, respectively, i.e.,

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

As we can notice, $y(t)$ represents a TF, because it is a linear function of each of the vectors $\mathbf{h}_k,\ k = 1, 2, 3$, if the other two are fixed, and can be considered as an extension of the BF [34].

Using the three impulse responses of the MISO system, we can also introduce a rank-1 tensor of dimension $L_1 \times L_2 \times L_3$:

$$\boldsymbol{\mathcal{H}} = \mathbf{h}_1 \circ \mathbf{h}_2 \circ \mathbf{h}_3, \tag{3.23}$$

so that the elements of $\boldsymbol{\mathcal{H}}$ are

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

Consequently, the output signal results in

$$y(t) = \text{vec}^T(\boldsymbol{\mathcal{H}})\, \text{vec}\left[\boldsymbol{\mathcal{X}}(t)\right], \tag{3.24}$$

where

$$\operatorname{vec}(\boldsymbol{\mathcal{H}}) = \begin{bmatrix} \operatorname{vec}(\mathbf{H}_{::1}) \\ \vdots \\ \operatorname{vec}(\mathbf{H}_{::L_3}) \end{bmatrix} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1 \triangleq \mathbf{h}, \tag{3.25}$$

$$\operatorname{vec}[\boldsymbol{\mathcal{X}}(t)] = \begin{bmatrix} \operatorname{vec}[\mathbf{X}_{::1}(t)] \\ \vdots \\ \operatorname{vec}[\mathbf{X}_{::L_3}(t)] \end{bmatrix} \triangleq \mathbf{x}(t), \tag{3.26}$$

$\mathbf{H}_{::l_3}$ and $\mathbf{X}_{::l_3}(t)$ (with $l_3 = 1, 2 \ldots, L_3$) are the frontal slices of $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{X}}(t)$, respectively, while $\mathbf{h}$ and $\mathbf{x}(t)$ denote two long vectors, each of them having $L_1L_2L_3$ elements. Thus, the output signal can also be expressed as

$$y(t) = \mathbf{h}^T\mathbf{x}(t). \tag{3.27}$$

In this context, our goal is to estimate the global impulse response $\mathbf{h}$, like in a system identification problem. Usually, the output of the MISO system is corrupted by an additive noise, thus resulting the reference signal:

$$d(t) = y(t) + v(t) = \mathbf{h}^T\mathbf{x}(t) + v(t), \tag{3.28}$$

where $v(t)$ is a zero-mean additive noise, which is uncorrelated with the input signals. At this point, we can also define the error signal:

$$e(t) = d(t) - \widehat{y}(t) = d(t) - \widehat{\mathbf{h}}^T\mathbf{x}(t), \tag{3.29}$$

i.e., the difference between the reference signal and the estimated signal, $\widehat{y}(t) = \widehat{\mathbf{h}}^T\mathbf{x}(t)$, which represents the output of the estimated system defined by the impulse response $\widehat{\mathbf{h}}$ of length $L_1L_2L_3$.

As we can notice from (3.25), the global impulse response $\mathbf{h}$ results based on a combination of the shorter impulse responses $\mathbf{h}_k,\ k = 1, 2, 3$, with L_1, L_2, and L_3 coefficients, respectively. In this trilinear context, we only need $L_1 + L_2 + L_3$ different elements to form $\mathbf{h}$, even though this global impulse response is of length $L_1L_2L_3$. Similar to (3.25), the estimate of the global impulse response can be decomposed as

$$\widehat{\mathbf{h}} = \widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1, \tag{3.30}$$

where $\widehat{\mathbf{h}}_k, k = 1, 2, 3$ are three impulse responses of lengths L_1, L_2, and L_3, respectively, which represent the estimates of the individual impulse responses $\mathbf{h}_k,\ k = 1, 2, 3$. However, we should note that there is no unique solution related to the decomposition in (3.30), since for any constants η_1, η_2, and η_3, with $\eta_1\eta_2\eta_3 = 1$, we have $\widehat{\mathbf{h}} = \widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1 = \eta_3\widehat{\mathbf{h}}_3 \otimes \eta_2\widehat{\mathbf{h}}_2 \otimes \eta_1\widehat{\mathbf{h}}_1$. Consequently, $\eta_k\widehat{\mathbf{h}}_k,\ k = 1, 2, 3$ also represent a set of solutions for our problem. Despite this, the global impulse response, $\mathbf{h}$, can be identified with no scaling ambiguity.

3.3 Wiener Filter for Trilinear Forms

This subchapter (which was published in [66, 67]) introduces two types of Wiener filter for the identification of TF: the direct one, as well as an iterative approach.

3.3.1 Direct Wiener Filter

The development of the Wiener filter is based on the minimization of the cost function, represented by the MSE, as optimization criterion. Based on (3.29), let us consider this cost function:

$$J\left(\widehat{\mathbf{h}}\right) = \mathbb{E}\left[e^2(t)\right]. \tag{3.31}$$

Using (3.29) in (3.31), together with the notation $\sigma_d^2 = \mathbb{E}\left[d^2(t)\right]$ (the variance of the reference signal), $\mathbf{p} = \mathbb{E}\left[\mathbf{x}(t)d(t)\right]$ (the cross-correlation vector between the input and the reference signals), and $\mathbf{R} = \mathbb{E}\left[\mathbf{x}(t)\mathbf{x}^T(t)\right]$ (the covariance matrix of the input signal), the cost function can be developed as

$$J\left(\widehat{\mathbf{h}}\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}^T\mathbf{p} + \widehat{\mathbf{h}}^T\mathbf{R}\widehat{\mathbf{h}}. \tag{3.32}$$

The minimization of (3.32) leads to the well-known solution of the Wiener filter:

$$\widehat{\mathbf{h}}_{\mathrm{W}} = \mathbf{R}^{-1}\mathbf{p}. \tag{3.33}$$

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

3.3.2 Iterative Wiener Filter

Next, we propose an iterative alternative to the conventional Wiener filter, following the decomposition from (3.30). First, we can easily verify that

$$\begin{aligned}\widehat{\mathbf{h}} &= \widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1 \\ &= \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)\widehat{\mathbf{h}}_1 && (3.34) \\ &= \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)\widehat{\mathbf{h}}_2 && (3.35) \\ &= \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)\widehat{\mathbf{h}}_3 && (3.36)\end{aligned}$$

where $\mathbf{I}_{L_k},\ k = 1, 2, 3$ are the identity matrices of sizes $L_1 \times L_1$, $L_2 \times L_2$, and $L_3 \times L_3$, respectively. Based on the previous relations, the cost function from (3.32) can be expressed in three different ways. For example, using (3.34), we obtain

$$\begin{aligned} J\left(\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2, \widehat{\mathbf{h}}_3\right) &= \sigma_d^2 - 2\widehat{\mathbf{h}}_1^T \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)^T \mathbf{p} \\ &+ \widehat{\mathbf{h}}_1^T \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)^T \mathbf{R} \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right) \widehat{\mathbf{h}}_1. \end{aligned} \tag{3.37}$$

When $\widehat{\mathbf{h}}_2$ and $\widehat{\mathbf{h}}_3$ are fixed, we can rewrite (3.37) as

$$J_{\widehat{\mathbf{h}}_2, \widehat{\mathbf{h}}_3}\left(\widehat{\mathbf{h}}_1\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}_1^T \mathbf{p}_1 + \widehat{\mathbf{h}}_1^T \mathbf{R}_1 \widehat{\mathbf{h}}_1, \tag{3.38}$$

where

$$\mathbf{p}_1 = \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)^T \mathbf{p}, \tag{3.39}$$

$$\mathbf{R}_1 = \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)^T \mathbf{R} \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right). \tag{3.40}$$

In this case, the partial cost function from (3.38) is a convex one and can be minimized with respect to $\widehat{\mathbf{h}}_1$, by equating the gradient to zero.

Similarly, using (3.35) and (3.36), the cost function from (3.32) becomes

$$\begin{aligned} J\left(\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2, \widehat{\mathbf{h}}_3\right) &= \sigma_d^2 - 2\widehat{\mathbf{h}}_2^T \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{p} \\ &+ \widehat{\mathbf{h}}_2^T \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{R} \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right) \widehat{\mathbf{h}}_2 \end{aligned} \tag{3.41}$$

$$\begin{aligned} &= \sigma_d^2 - 2\widehat{\mathbf{h}}_3^T \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{p} \\ &+ \widehat{\mathbf{h}}_3^T \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{R} \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right) \widehat{\mathbf{h}}_3. \end{aligned} \tag{3.42}$$

Also, when $\widehat{\mathbf{h}}_1$ and $\widehat{\mathbf{h}}_3$ are fixed, (3.41) becomes

$$J_{\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_3}\left(\widehat{\mathbf{h}}_2\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}_2^T \mathbf{p}_2 + \widehat{\mathbf{h}}_2^T \mathbf{R}_2 \widehat{\mathbf{h}}_2, \tag{3.43}$$

where

$$\mathbf{p}_2 = \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{p}, \tag{3.44}$$

$$\mathbf{R}_2 = \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{R} \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right), \tag{3.45}$$

while when $\widehat{\mathbf{h}}_1$ and $\widehat{\mathbf{h}}_2$ are fixed, the cost function from (3.42) results in

$$J_{\widehat{\mathbf{h}}_1,\widehat{\mathbf{h}}_2}\left(\widehat{\mathbf{h}}_3\right) = \sigma_d^2 - 2\widehat{\mathbf{h}}_3^T \mathbf{p}_3 + \widehat{\mathbf{h}}_3^T \mathbf{R}_3 \widehat{\mathbf{h}}_3, \tag{3.46}$$

where

$$\mathbf{p}_3 = \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{p}, \tag{3.47}$$

$$\mathbf{R}_3 = \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)^T \mathbf{R} \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right). \tag{3.48}$$

In both cases, the partial cost functions from (3.43) and (3.46) can be minimized with respect to $\widehat{\mathbf{h}}_2$ and $\widehat{\mathbf{h}}_3$, respectively (by equating their gradients to zero).

The previous procedure suggests an iterative approach. To start the algorithm, a set of initial values should be provided for two of the estimated impulse responses. For example, we can choose

$$\widehat{\mathbf{h}}_2^{(0)} = \frac{1}{L_2} \left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right]^T, \tag{3.49}$$

$$\widehat{\mathbf{h}}_3^{(0)} = \frac{1}{L_3} \left[\begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array}\right]^T. \tag{3.50}$$

Hence, based on (3.39) and (3.40), one may compute

$$\mathbf{p}_1^{(0)} = \left(\widehat{\mathbf{h}}_3^{(0)} \otimes \widehat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1}\right)^T \mathbf{p}, \tag{3.51}$$

$$\mathbf{R}_1^{(0)} = \left(\widehat{\mathbf{h}}_3^{(0)} \otimes \widehat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1}\right)^T \mathbf{R} \left(\widehat{\mathbf{h}}_3^{(0)} \otimes \widehat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1}\right). \tag{3.52}$$

In the first iteration, the first cost function to be minimized results from (3.38) [using (3.51)–(3.52)], i.e.,

$$J_{\widehat{\mathbf{h}}_2,\widehat{\mathbf{h}}_3}\left(\widehat{\mathbf{h}}_1^{(1)}\right) = \sigma_d^2 - 2\left(\widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{p}_1^{(0)} + \left(\widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{R}_1^{(0)}\widehat{\mathbf{h}}_1^{(1)}, \tag{3.53}$$

which leads to the solution:

$$\widehat{\mathbf{h}}_1^{(1)} = \left(\mathbf{R}_1^{(0)}\right)^{-1} \mathbf{p}_1^{(0)}. \tag{3.54}$$

Also, since $\widehat{\mathbf{h}}_1^{(1)}$ and $\widehat{\mathbf{h}}_3^{(0)}$ are now available, we can evaluate [based on (3.44)–(3.45)]

$$\mathbf{p}_2^{(1)} = \left(\widehat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{p}, \tag{3.55}$$

$$\mathbf{R}_2^{(1)} = \left(\widehat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{R}\left(\widehat{\mathbf{h}}_3^{(0)} \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1^{(1)}\right), \tag{3.56}$$

so that the cost function from (3.43) becomes

$$J_{\widehat{\mathbf{h}}_1,\widehat{\mathbf{h}}_3}\left(\widehat{\mathbf{h}}_2^{(1)}\right) = \sigma_d^2 - 2\left(\widehat{\mathbf{h}}_2^{(1)}\right)^T \mathbf{p}_2^{(1)} + \left(\widehat{\mathbf{h}}_2^{(1)}\right)^T \mathbf{R}_2^{(1)}\widehat{\mathbf{h}}_2^{(1)}, \tag{3.57}$$

while its minimization leads to

$$\widehat{\mathbf{h}}_2^{(1)} = \left(\mathbf{R}_2^{(1)}\right)^{-1} \mathbf{p}_2^{(1)}. \tag{3.58}$$

Finally, using the solutions $\widehat{\mathbf{h}}_1^{(1)}$ and $\widehat{\mathbf{h}}_2^{(1)}$, we can find $\widehat{\mathbf{h}}_3^{(1)}$ in a similar manner. First, we evaluate [based on (3.47)–(3.48)]

$$\mathbf{p}_3^{(1)} = \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2^{(1)} \otimes \widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{p}, \tag{3.59}$$

$$\mathbf{R}_3^{(1)} = \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2^{(1)} \otimes \widehat{\mathbf{h}}_1^{(1)}\right)^T \mathbf{R}\left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2^{(1)} \otimes \widehat{\mathbf{h}}_1^{(1)}\right). \tag{3.60}$$

Then, we minimize the cost function [which results from (3.46)]:

$$J_{\widehat{\mathbf{h}}_1,\widehat{\mathbf{h}}_2}\left(\widehat{\mathbf{h}}_3^{(1)}\right) = \sigma_d^2 - 2\left(\widehat{\mathbf{h}}_3^{(1)}\right)^T \mathbf{p}_3^{(1)} + \left(\widehat{\mathbf{h}}_3^{(1)}\right)^T \mathbf{R}_3^{(1)}\widehat{\mathbf{h}}_3^{(1)}, \tag{3.61}$$

thus obtaining the solution:

$$\widehat{\mathbf{h}}_3^{(1)} = \left(\mathbf{R}_3^{(1)}\right)^{-1}\mathbf{p}_3^{(1)}. \tag{3.62}$$

Therefore, continuing a similar iterative procedure, at iteration n, we get the estimates of the impulse responses based on the same main steps, as follows.

- **Step 1**:

$$\widehat{\mathbf{h}}_1^{(n)} = \left(\mathbf{R}_1^{(n-1)}\right)^{-1}\mathbf{p}_1^{(n-1)},$$

where

$$\begin{aligned}
\mathbf{p}_1^{(n-1)} &= \left(\widehat{\mathbf{h}}_3^{(n-1)} \otimes \widehat{\mathbf{h}}_2^{(n-1)} \otimes \mathbf{I}_{L_1}\right)^T \mathbf{p}, \\
\mathbf{R}_1^{(n-1)} &= \left(\widehat{\mathbf{h}}_3^{(n-1)} \otimes \widehat{\mathbf{h}}_2^{(n-1)} \otimes \mathbf{I}_{L_1}\right)^T \mathbf{R}\left(\widehat{\mathbf{h}}_3^{(n-1)} \otimes \widehat{\mathbf{h}}_2^{(n-1)} \otimes \mathbf{I}_{L_1}\right).
\end{aligned}$$

- **Step 2**:

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

- **Step 3**:

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

Using the solutions from Steps 1–3, the global impulse response at iteration n results in

$$\widehat{\mathbf{h}}^{(n)} = \widehat{\mathbf{h}}_3^{(n)} \otimes \widehat{\mathbf{h}}_2^{(n)} \otimes \widehat{\mathbf{h}}_1^{(n)}. \tag{3.63}$$

Summarizing, the proposed iterative Wiener filter for the identification of TF is provided in Table 3.1 (in a more compact form that facilitates its implementation).

It was previously shown in [43] that the iterative Wiener filter can be interpreted in terms of a block coordinate descent approach (or nonlinear Gauss-Seidel method) [78, 41]. The convergence of this type of algorithms was proved in [78] (Sec. 2.7, Proposition 2.7.1). In [44], the convergence proof was provided for a similar iterative problem, while in [43], the convergence of the iterative process was studied for a decomposition problem presented in the framework of a bilinear form.

We can set the decrease of the NM below a certain established threshold value as a condition for stopping the iterative process. However, as it will be shown in subchapter 3.3.3, the convergence of the proposed iterative Wiener filter for TF occurs after only a small number of iterations.

The proposed iterative Wiener filter for TF represents an extension of the solution presented in [34] (in the context of BF). However, when the MISO system identification problem results based on (3.22), it is more advantageous to use the algorithm tailored for TF, instead of reformulating the problem in terms of multiple BF. The

Table 3.1: Iterative Wiener filter for the identification of trilinear forms.

```
Data:
$\mathbf{R}$, $\mathbf{p}$ (estimated statistics based on the available data samples)
Initialization:
$\widehat{\mathbf{h}}_2^{(0)} = \frac{1}{L_2} \left[ \begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array} \right]^T$
$\widehat{\mathbf{h}}_3^{(0)} = \frac{1}{L_3} \left[ \begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array} \right]^T$
$\mathbf{q}_1^{(0)} = \widehat{\mathbf{h}}_3^{(0)} \otimes \widehat{\mathbf{h}}_2^{(0)} \otimes \mathbf{I}_{L_1}$
$\mathbf{p}_1^{(0)} = \left( \mathbf{q}_1^{(0)} \right)^T \mathbf{p}$
$\mathbf{R}_1^{(0)} = \left( \mathbf{q}_1^{(0)} \right)^T \mathbf{R} \mathbf{q}_1^{(0)}$
For $n = 1, 2, \ldots$:
$\widehat{\mathbf{h}}_1^{(n)} = \left( \mathbf{R}_1^{(n-1)} \right)^{-1} \mathbf{p}_1^{(n-1)}$
$\mathbf{q}_2^{(n)} = \widehat{\mathbf{h}}_3^{(n-1)} \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1^{(n)}$
$\mathbf{p}_2^{(n)} = \left( \mathbf{q}_2^{(n)} \right)^T \mathbf{p}$
$\mathbf{R}_2^{(n)} = \left( \mathbf{q}_2^{(n)} \right)^T \mathbf{R} \mathbf{q}_2^{(n)}$
$\widehat{\mathbf{h}}_2^{(n)} = \left( \mathbf{R}_2^{(n)} \right)^{-1} \mathbf{p}_2^{(n)}$
$\mathbf{q}_3^{(n)} = \mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2^{(n)} \otimes \widehat{\mathbf{h}}_1^{(n)}$
$\mathbf{p}_3^{(n)} = \left( \mathbf{q}_3^{(n)} \right)^T \mathbf{p}$
$\mathbf{R}_3^{(n)} = \left( \mathbf{q}_3^{(n)} \right)^T \mathbf{R} \mathbf{q}_3^{(n)}$
$\widehat{\mathbf{h}}_3^{(n)} = \left( \mathbf{R}_3^{(n)} \right)^{-1} \mathbf{p}_3^{(n)}$
$\mathbf{q}_1^{(n)} = \widehat{\mathbf{h}}_3^{(n)} \otimes \widehat{\mathbf{h}}_2^{(n)} \otimes \mathbf{I}_{L_1}$
$\mathbf{p}_1^{(n)} = \left( \mathbf{q}_1^{(n)} \right)^T \mathbf{p}$
$\mathbf{R}_1^{(n)} = \left( \mathbf{q}_1^{(n)} \right)^T \mathbf{R} \mathbf{q}_1^{(n)}$
$\widehat{\mathbf{h}}^{(n)} = \widehat{\mathbf{h}}_3^{(n)} \otimes \widehat{\mathbf{h}}_2^{(n)} \otimes \widehat{\mathbf{h}}_1^{(n)}$
```

trilinear approach has some similarities (to some extent) with the one introduced in [79]. However, the batch trilinear Wiener-Hopf algorithm from [79] is more related to an adaptive approach, since the statistics are estimated within the algorithm. On the other hand, in the case of our iterative Wiener filter, the estimates of the statistics are considered to be a priori available (see also the related discussion in the next section), which is basically in the spirit of the Wiener filter.

3.3.3 Experimental Results

In this section, the performance of the proposed iterative Wiener filter for TF is evaluated in the context of system identification. The input signals that form $\boldsymbol{\mathcal{X}}(t)$ are AR(1) processes, which are obtained by generating WGNs and then filtering them through a first-order system $1/\left(1-0.9z^{-1}\right)$. The additive noise $v(t)$, corrupting the output signal $y(t)$, is white and Gaussian, with the variance set to $\sigma_v^2 = 0.01$. The impulse responses used in simulations are depicted in Fig. 3.1. The impulse response $\mathbf{h}_1$ is the first impulse response from the G168 Recommendation [55], of length $L_1 = 64$ (see Fig. 3.1a). Next, $\mathbf{h}_2$ is a random impulse response (with Gaussian distribution) of length $L_2 = 8$ (as shown in Fig. 3.1b). Finally, the coefficients of the impulse response $\mathbf{h}_3$ (depicted in Fig. 3.1c) are evaluated as $h_{3l_3} = 0.5^{l_3-1},\ l_3 = 1, 2, \ldots, L_3$, using $L_3 = 4$. Therefore, the global impulse response from Fig. 3.1d results in $\mathbf{h} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1$ and its length is $L = L_1L_2L_3 = 2048$. As we can see, this global impulse response is similar (to some extent) to a channel with echoes, similar to an acoustic echo path.

In order to evaluate the identification of the individual filters $\mathbf{h}_k,\ k = 1, 2, 3$, we use the NPM [40], defined as:

$$\text{NPM}\left[\mathbf{h}_k, \widehat{\mathbf{h}}_k\right] = 1 - \left[\frac{\mathbf{h}_k^T\widehat{\mathbf{h}}_k}{\|\mathbf{h}_k\|\|\widehat{\mathbf{h}}_k\|}\right]^2. \tag{3.64}$$

On the other hand, for the identification of the global impulse response, $\mathbf{h}$, we should use the NM:

$$\text{NM}\left[\mathbf{h}, \widehat{\mathbf{h}}\right] = \frac{\|\mathbf{h} - \widehat{\mathbf{h}}\|^2}{\|\mathbf{h}\|^2}. \tag{3.65}$$

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

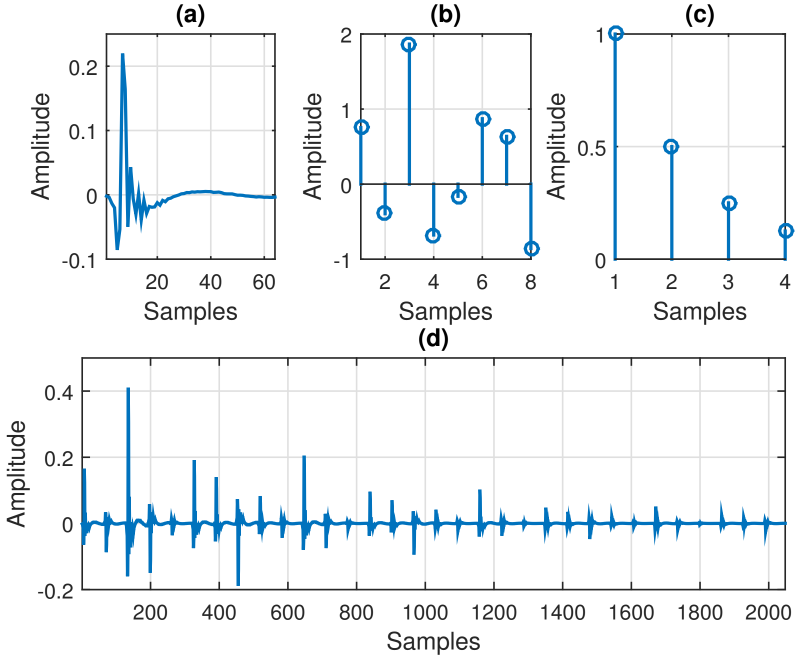


Figure 3.1: Impulse responses used in simulations: (**a**) $\mathbf{h}_1$ of length $L_1 = 64$ (the first impulse response from G168 Recommendation [55]), (**b**) $\mathbf{h}_2$ of length $L_2 = 8$ (random impulse response with Gaussian distribution), (**c**) $\mathbf{h}_3$ of length $L_3 = 4$ (its elements are evaluated as $h_{3l_3} = 0.5^{l_3 - 1}, \; l_3 = 1, \ldots, L_3$), and (**d**) the global impulse response $\mathbf{h} = \mathbf{h}_3 \otimes \mathbf{h}_2 \otimes \mathbf{h}_1$ of length $L = L_1 L_2 L_3 = 2048$.

$$\widehat{\mathbf{R}} = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}(t)\mathbf{x}^T(t), \tag{3.66}$$

$$\widehat{\mathbf{p}} = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}(t)d(t). \tag{3.67}$$

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

The matrix involved in the linear system, to be solved in the case of the conventional Wiener filter from (3.33), is of size $L \times L$; hence, a number of data samples larger than L are needed to estimate the statistics in (3.66) and (3.67), in order to obtain a good solution. This is shown in Fig. 3.2, where different values of N [that is, the available amount of data in (3.66) and (3.67)] are used and the solution provided by the conventional Wiener filter is evaluated for each of these values. Similar

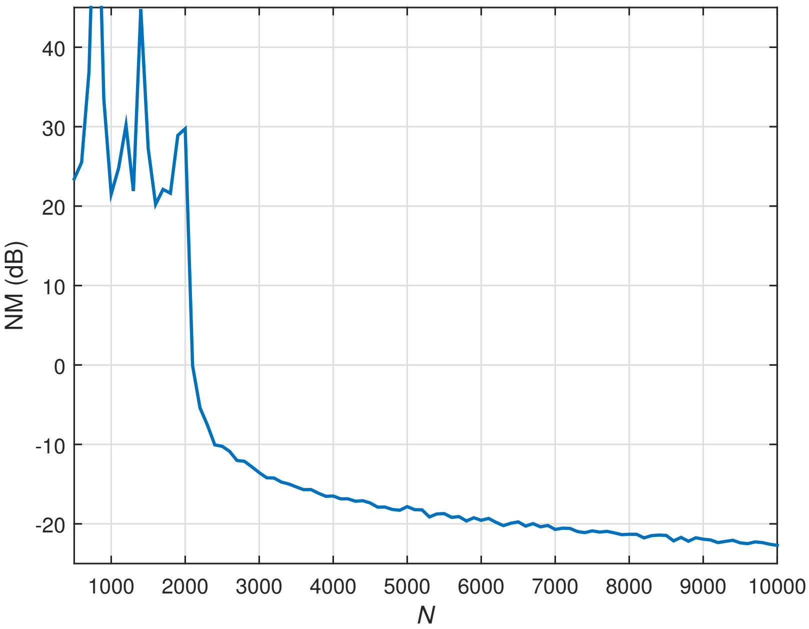


Figure 3.2: NM of the conventional Wiener filter as a function of the number of available data samples used to estimate the statistics (N), for the identification of the global impulse response from Fig. 3.1d. The input signals are AR(1) processes, $L = 2048$, and $\sigma_v^2 = 0.01$.

to (2.10), the performance measure in Fig. 3.2 is the NM (in dB), which is defined as $10\log_{10}\left(\left\|\mathbf{h} - \widehat{\mathbf{h}}_{\mathrm{W}}\right\|^2 / \left\|\mathbf{h}\right\|^2\right)$; in this case, the conventional Wiener solution $\widehat{\mathbf{h}}_{\mathrm{W}}$ results from (3.33) using (3.66) and (3.67). As we can see, the conventional Wiener filter achieves a reasonable decrease in misalignment only when a large amount of data (i.e., $N > L$) are used to estimate the statistics in (3.66) and (3.67).

The main advantage provided by the iterative Wiener filter is that it operates with much shorter filters [due to the decomposition shown in (3.30)] and, consequently, the dimensions of the linear systems of equations to be solved are significantly reduced. Therefore, even with a small amount of data (i.e., $N < L$), the iterative Wiener filter is able to obtain a reliable estimation. This advantage is outlined in Fig. 3.3, where the solution provided by the conventional Wiener filter (based on (3.33) and using (3.66) and (3.67)) is compared to the iterative Wiener filter from (3.63). The performance measure is the NM (in dB), which is evaluated based on (2.10), for the identification of the global system $\mathbf{h}$. Three amounts of data are considered in this experiment, that is, $N = 500, 2500$, and 5000. Clearly, in the first case ($N = 500$), the conventional Wiener filter leads to an inaccurate solution due to the small amount of data (as compared to $L = 2048$). When the amount of data slightly exceeds the value of L (e.g., $N = 2500$), the conventional Wiener filter provides a more reliable solution, that is, the misalignment attenuation is approximately -10 dB. Finally, for a large amount of data ($N = 5000$), this conventional solution is improved in terms of accuracy (e.g., the misalignment is close to -20 dB). On the other hand, in all the previous cases, the proposed iterative Wiener filter achieves a much more accurate solution (with only a few iterations), which outperforms by far the conventional one (even in the case when a small amount of data are available, e.g., $N = 500$). For example, the iterative Wiener filter which uses $N = 500$ yields a lower misalignment level with respect to the conventional Wiener filter with $N = 5000$.

In Fig. 3.4, the performance of the iterative Wiener filter is also illustrated using the NPMs (in dB), based on (3.64), for the identification of the individual impulse responses from Fig. 3.1a–c. Basically, the same conclusion applies, that is, only a few iterations are required by the iterative Wiener filter to achieve a reliable solution (even for a small amount of data).

3.4 Least-Mean-Square Adaptive Algorithms for Trilinear Forms

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

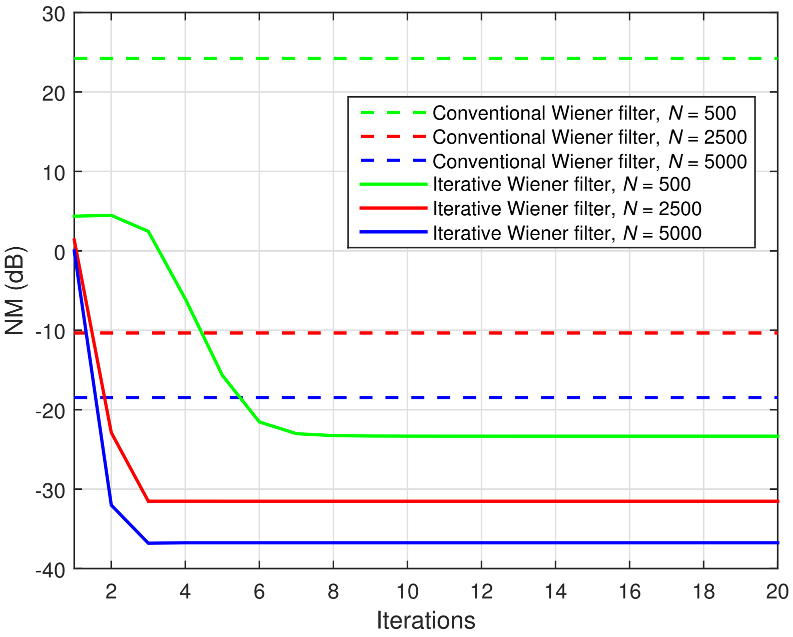


Figure 3.3: NM of the conventional and iterative Wiener filters, for different values of the number of available data samples used to estimate the statistics (N), for the identification of the global impulse response from Fig. 3.1d. The input signals are AR(1) processes, $L = 2048$, and $\sigma_v^2 = 0.01$.

tailored for the new TF approach. Then we continue by deriving a time-variant step-size parameter in 2.3.2, whose expression yields the NLMS filter. Then, in subchapter 3.4.3, we show through experiments that the proposed solutions outperform their regular counterparts.

3.4.1 Least-Mean-Square Algorithm for Trilinear Forms (LMS-TF)

The work presented in this section was published in [66]. First, let us consider the three estimated impulse responses $\widehat{\mathbf{h}}_k,\ k = 1, 2, 3$, and the corresponding a priori error signals:

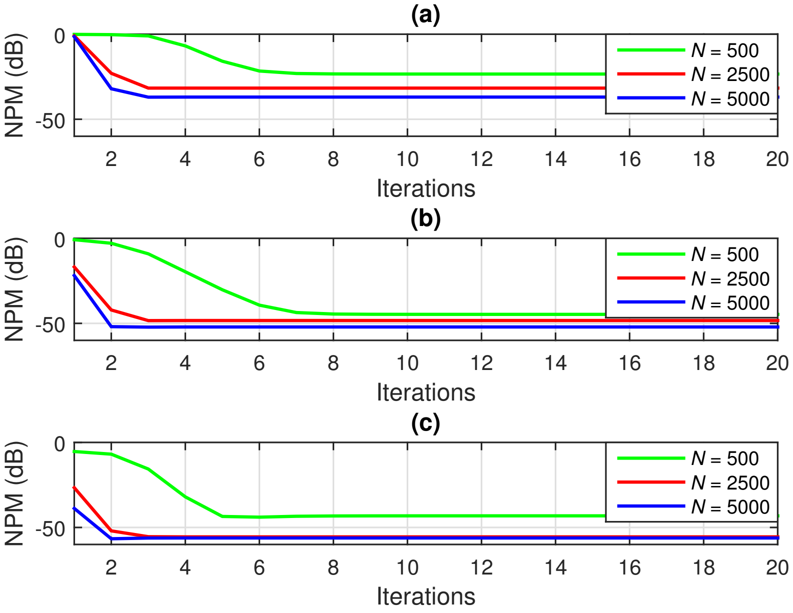


Figure 3.4: NPM of the iterative Wiener filter, for different values of the number of available data samples used to estimate the statistics (N), for the identification of the individual impulse responses from Fig. 3.1a–c: (**a**) NPM $\left(\mathbf{h}_1, \widehat{\mathbf{h}}_1^{(n)}\right)$, (**b**) NPM $\left(\mathbf{h}_2, \widehat{\mathbf{h}}_2^{(n)}\right)$, and (**c**) NPM $\left(\mathbf{h}_3, \widehat{\mathbf{h}}_3^{(n)}\right)$. The input signals are AR(1) processes, $L_1 = 64$, $L_2 = 8$, $L_3 = 4$, and $\sigma_v^2 = 0.01$.

$$e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) = d(t) - \widehat{\mathbf{h}}_1^T(t-1)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t), \tag{3.68}$$

$$e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) = d(t) - \widehat{\mathbf{h}}_2^T(t-1)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t), \tag{3.69}$$

$$e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) = d(t) - \widehat{\mathbf{h}}_3^T(t-1)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t), \tag{3.70}$$

where

$$\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) = \left(\widehat{\mathbf{h}}_3 \otimes \widehat{\mathbf{h}}_2 \otimes \mathbf{I}_{L_1}\right)\mathbf{x}(t), \tag{3.71}$$

$$\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) = \left(\widehat{\mathbf{h}}_3 \otimes \mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_1\right)\mathbf{x}(t), \tag{3.72}$$

$$\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) = \left(\mathbf{I}_{L_3} \otimes \widehat{\mathbf{h}}_2 \otimes \widehat{\mathbf{h}}_1\right)\mathbf{x}(t). \tag{3.73}$$

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

$$\widehat{\mathbf{h}}_1(t) = \widehat{\mathbf{h}}_1(t-1) - \frac{\mu_{\widehat{\mathbf{h}}_1}}{2} \times \frac{\partial {e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}}^2(t)}{\partial \widehat{\mathbf{h}}_1(t-1)} = \widehat{\mathbf{h}}_1(t-1) + \mu_{\widehat{\mathbf{h}}_1} \mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t), \quad (3.74)$$

$$\widehat{\mathbf{h}}_2(t) = \widehat{\mathbf{h}}_2(t-1) - \frac{\mu_{\widehat{\mathbf{h}}_2}}{2} \times \frac{\partial {e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}}^2(t)}{\partial \widehat{\mathbf{h}}_2(t-1)} = \widehat{\mathbf{h}}_2(t-1) + \mu_{\widehat{\mathbf{h}}_2} \mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t), \quad (3.75)$$

$$\widehat{\mathbf{h}}_3(t) = \widehat{\mathbf{h}}_3(t-1) - \frac{\mu_{\widehat{\mathbf{h}}_3}}{2} \times \frac{\partial {e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}}^2(t)}{\partial \widehat{\mathbf{h}}_3(t-1)} = \widehat{\mathbf{h}}_3(t-1) + \mu_{\widehat{\mathbf{h}}_3} \mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t), \quad (3.76)$$

where $\mu_{\widehat{\mathbf{h}}_1} > 0, \mu_{\widehat{\mathbf{h}}_2} > 0, \mu_{\widehat{\mathbf{h}}_3} > 0$ are the step-size parameters. Relations (3.74)–(3.76) define the LMS algorithm for TF, namely LMS-TF.

For the initialization of the estimated impulse responses, we use:

$$\widehat{\mathbf{h}}_1(0) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T, \quad (3.77)$$

$$\widehat{\mathbf{h}}_2(0) = \frac{1}{L_2} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T, \quad (3.78)$$

$$\widehat{\mathbf{h}}_3(0) = \frac{1}{L_3} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T. \quad (3.79)$$

In the end, we can obtain the global filter in the following way:

$$\widehat{\mathbf{h}}(t) = \widehat{\mathbf{h}}_3(t) \otimes \widehat{\mathbf{h}}_2(t) \otimes \widehat{\mathbf{h}}_1(t). \quad (3.80)$$

Alternatively, this global impulse response can be identified directly, using the regular LMS algorithm, by using the following update:

$$\widehat{\mathbf{h}}(t) = \widehat{\mathbf{h}}(t-1) + \mu_{\widehat{\mathbf{h}}} \mathbf{x}(t) e(t), \quad (3.81)$$

where

$$e(t) = d(t) - \widehat{\mathbf{h}}(t-1)\mathbf{x}(t) \quad (3.82)$$

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

However, an observation needs to be made regarding the update in relation (3.81): this involves an adaptive filter of length $L_1L_2L_3$, whereas the LMS-TF algorithm, defined by the updates (3.74)–(3.76), uses three shorter filters of length L_1, L_2, and L_3, respectively. Therefore, the complexity of this new approach is lower and the convergence rate is expected to be faster.

3.4.2 Normalized LMS Algorithm for Trilinear Forms (NLMS-TF)

The step-size parameters in (3.74)–(3.76) take constant values, chosen such that they ensure the convergence of the algorithm and a good compromise between convergence speed and steady-state misadjustment. Nevertheless, when dealing with nonstationary signals, it may be more appropriate to use time-dependent step-sizes, which lead to the following update relations:

$$\widehat{\mathbf{h}}_1(t) = \widehat{\mathbf{h}}_1(t-1) + \mu_{\widehat{\mathbf{h}}_1}(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t), \tag{3.83}$$

$$\widehat{\mathbf{h}}_2(t) = \widehat{\mathbf{h}}_2(t-1) + \mu_{\widehat{\mathbf{h}}_2}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t), \tag{3.84}$$

$$\widehat{\mathbf{h}}_3(t) = \widehat{\mathbf{h}}_3(t-1) + \mu_{\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t). \tag{3.85}$$

For deriving the expressions of the step-size parameters, we take into account the stability conditions and we target to cancel the a posteriori error signals [45]:

$$\varepsilon_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) = d(t) - \widehat{\mathbf{h}}_1^T(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t), \tag{3.86}$$

$$\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) = d(t) - \widehat{\mathbf{h}}_2^T(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t), \tag{3.87}$$

$$\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) = d(t) - \widehat{\mathbf{h}}_3^T(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t). \tag{3.88}$$

By replacing (3.74) in (3.86), (3.75) in (3.87) and (3.76) in (3.88), respectively, and by imposing the conditions $\varepsilon_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) = 0$, $\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) = 0$, and $\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) = 0$, we obtain that:

$$\varepsilon_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) = e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)\left[1 - \mu_{\widehat{\mathbf{h}}_1}(t)\mathbf{x}^T_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)\right] = 0, \tag{3.89}$$

$$\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) = e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)\left[1 - \mu_{\widehat{\mathbf{h}}_2}(t)\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)\right] = 0, \tag{3.90}$$

$$\varepsilon_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) = e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)\left[1 - \mu_{\widehat{\mathbf{h}}_3}(t)\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)\right] = 0. \tag{3.91}$$

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

$$\mu_{\widehat{\mathbf{h}}_1}(t) = \frac{1}{\mathbf{x}^T_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)}, \tag{3.92}$$

$$\mu_{\widehat{\mathbf{h}}_2}(t) = \frac{1}{\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)}, \tag{3.93}$$

$$\mu_{\widehat{\mathbf{h}}_3}(t) = \frac{1}{\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)}. \tag{3.94}$$

In order to achieve a good compromise between convergence rate and misadjustment, three positive constants $0 < \alpha_{\widehat{\mathbf{h}}_1} < 1$, $0 < \alpha_{\widehat{\mathbf{h}}_2} < 1$, and $0 < \alpha_{\widehat{\mathbf{h}}_3} < 1$ are employed [1]. In addition to that, three regularization constants $\delta_{\widehat{\mathbf{h}}_1} > 0$, $\delta_{\widehat{\mathbf{h}}_2} > 0$, and $\delta_{\widehat{\mathbf{h}}_3} > 0$, usually chosen to be proportional to the input signal variance [46], are added to the denominators of the step-size parameters. Finally, the updates of the NLMS algorithm for TF (NLMS-TF) become:

$$\widehat{\mathbf{h}}_1(t) = \widehat{\mathbf{h}}_1(t-1) + \frac{\alpha_{\widehat{\mathbf{h}}_1}(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)e_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)}{\mathbf{x}^T_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_2\widehat{\mathbf{h}}_3}(t) + \delta_{\widehat{\mathbf{h}}_1}}, \tag{3.95}$$

$$\widehat{\mathbf{h}}_2(t) = \widehat{\mathbf{h}}_2(t-1) + \frac{\alpha_{\widehat{\mathbf{h}}_2}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)}{\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_3}(t) + \delta_{\widehat{\mathbf{h}}_2}}, \tag{3.96}$$

$$\widehat{\mathbf{h}}_3(t) = \widehat{\mathbf{h}}_3(t-1) + \frac{\alpha_{\widehat{\mathbf{h}}_3}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)e_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)}{\mathbf{x}^T_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t)\mathbf{x}_{\widehat{\mathbf{h}}_1\widehat{\mathbf{h}}_2}(t) + \delta_{\widehat{\mathbf{h}}_3}}. \tag{3.97}$$

The initializations of the estimated filters may be the same as (3.77), (3.78), and (3.79). In a similar way as for the LMS algorithm, the global impulse response can be identified using the regular NLMS:

$$\widehat{\mathbf{h}}(t) = \widehat{\mathbf{h}}(t-1) + \frac{\alpha_{\widehat{\mathbf{h}}}\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{x}(t) + \delta_{\widehat{\mathbf{h}}}}, \tag{3.98}$$

where $e(t)$ is given in (3.82). The parameters $\alpha_{\widehat{\mathbf{h}}}$ and $\delta_{\widehat{\mathbf{h}}}$ represent the normalized step-size parameter and the regularization constant for the global filter, respectively. As it was shown before in [35] for BF, the global misalignment can be controlled by using a constraint on the sum of the normalized step-sizes, and this sum should be smaller than 1. In this way, for different values of $\alpha_{\widehat{\mathbf{h}}_1}$, $\alpha_{\widehat{\mathbf{h}}_2}$, $\alpha_{\widehat{\mathbf{h}}_3}$ fulfilling this condition, the misalignment of the global filter is the same. On the other hand, in the case when $\alpha_{\widehat{\mathbf{h}}_1} = \alpha_{\widehat{\mathbf{h}}_2} = \alpha_{\widehat{\mathbf{h}}_3}$, the three filters reach the same misalignment level.

Again, we can notice that the global impulse response identification involves the use of a filter of length $L_1L_2L_3$. Because the trilinear approach uses three much shorter impulse responses of lengths L_1, L_2, and L_3, respectively, it is expected that this new solution will yield a faster convergence. This will be shown through simulations.

The NLMS-TF algorithm proposed here is similar to the one presented in [80]. However, our choice of the system impulse responses used in simulations is different from the one in [80], where one of the three paths contains only one nonzero element. On the contrary, we aim to show the performance of the algorithm in a scenario which includes a real echo path. In addition, we also study the tracking capability of the algorithm.

3.4.3 Experimental Results

For the simulations in this section, the setup is the same as for the previous experiments, in subchapter 3.3.3. First, we aim to show the influence of the constant step-size values on the performance of the LMS-TF algorithm. The performance measure is the NM:

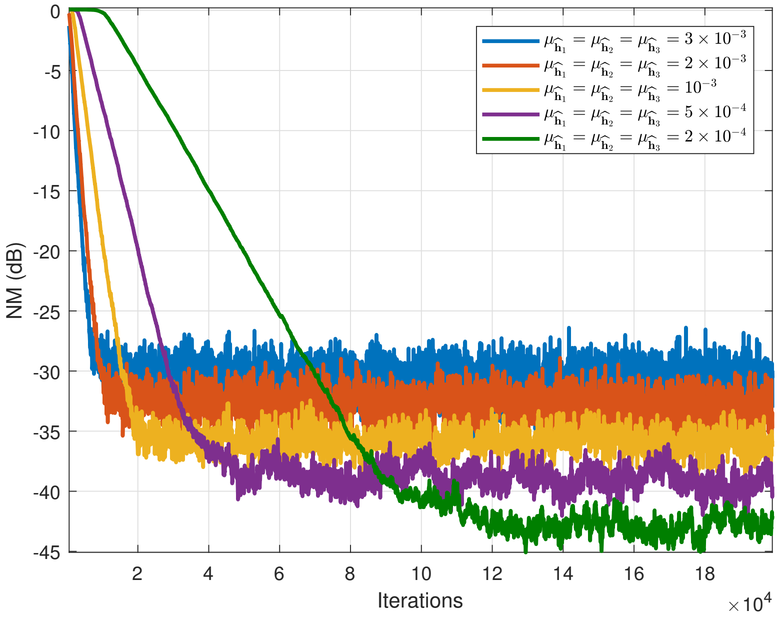


Figure 3.5: NM of the LMS-TF algorithm using different values of the step-size parameters.

$$\mathrm{NM}\left[\mathbf{h}, \widehat{\mathbf{h}}(t)\right] = \frac{\|\mathbf{h} - \widehat{\mathbf{h}}(t)\|^2}{\|\mathbf{h}\|^2}. \tag{3.99}$$

In Fig. 3.5 it can be seen that if the step-sizes take large values, the LMS-TF algorithm reaches convergence after less than 10^4 iterations. Then, as these values decrease, the convergence becomes slower but the steady-state value of the NM also decreases, highlighting the compromise between convergence rate and NM value.

Next, we illustrate the improvement brought by the proposed solution, by comparing the LMS-TF algorithm to its regular counterpart, applied for the identification of the global filter. Fig. 3.6 shows the values of the NM for the regular LMS filter and the LMS-TF. The first observation is that, in order to reach the same steady-state value of the NM, the regular LMS algorithm needs many more iterations. On the other hand, for a similar convergence speed, the final NM provided by the LMS-TF is much lower than that offered by its regular counterpart. This proves that the proposed solution offers a significant improvement with respect to the classical approach.

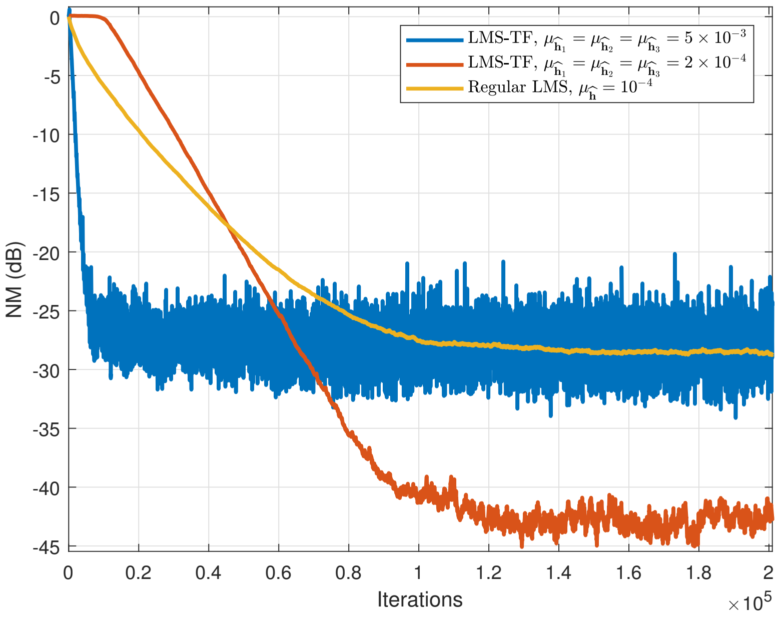


Figure 3.6: NM of the LMS-TF and regular LMS algorithms.

The next step is to study the behavior of the NLMS-TF filter. First, the performance of the NLMS-TF algorithm is depicted in Fig. 3.7, for different values of the normalized step-sizes. The regularization constants are $\delta_{\widehat{\mathbf{h}}_1} = \delta_{\widehat{\mathbf{h}}_2} = \delta_{\widehat{\mathbf{h}}_3} = 0.001$. The same conclusion as for the LMS-TF case is valid, namely that the decrease in the normalized step-sizes leads to a smaller value of the final NM but at the cost of a slower convergence rate. When the step-size values decrease 10 times, the number of iterations needed to reach convergence increases almost 10 times, while the steady-state NM value decreases by a bit more than 10 dB.

We then compare the NLMS-TF algorithm with its regular counterpart (applied on the global filter) in Fig. 3.8. Again, we observe that the NLMS-TF behaves better than the regular NLMS algorithm, from the perspective of both convergence rate and final NM value.

Finally, the tracking capability of the NLMS-TF algorithm is of interest, that is, the capability of the algorithm to react to abrupt changes to the impulse responses. In order to study this characteristic, we simulated a sudden change of the random

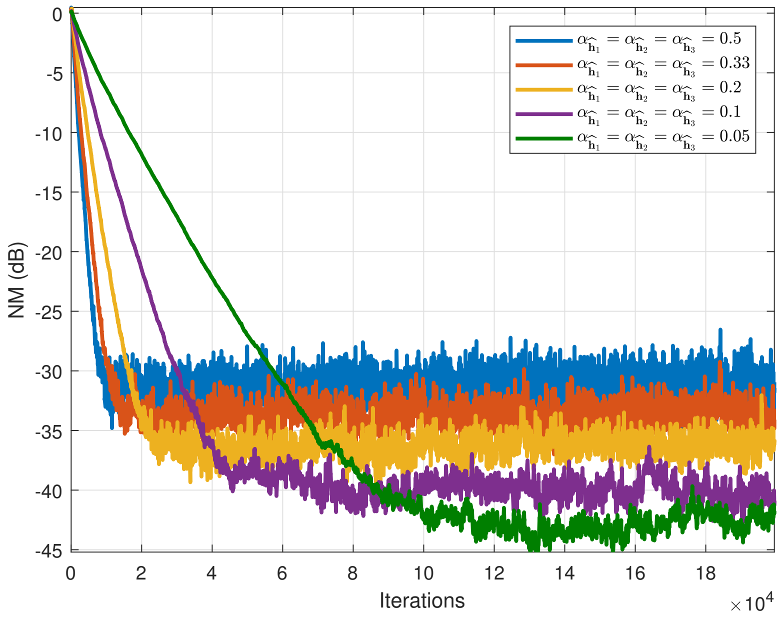


Figure 3.7: NM of the NLMS-TF algorithm using different values of the step-size parameters.

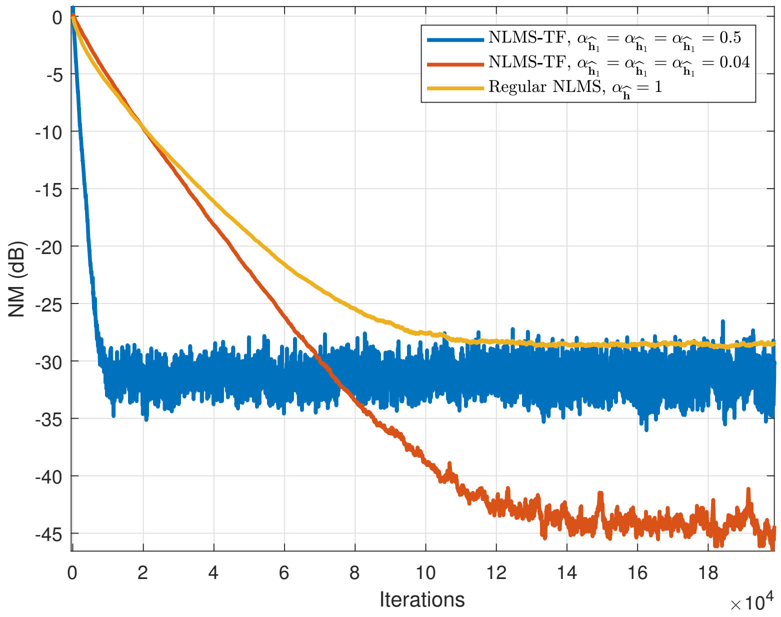


Figure 3.8: NM of the NLMS-TF and regular NLMS algorithms.

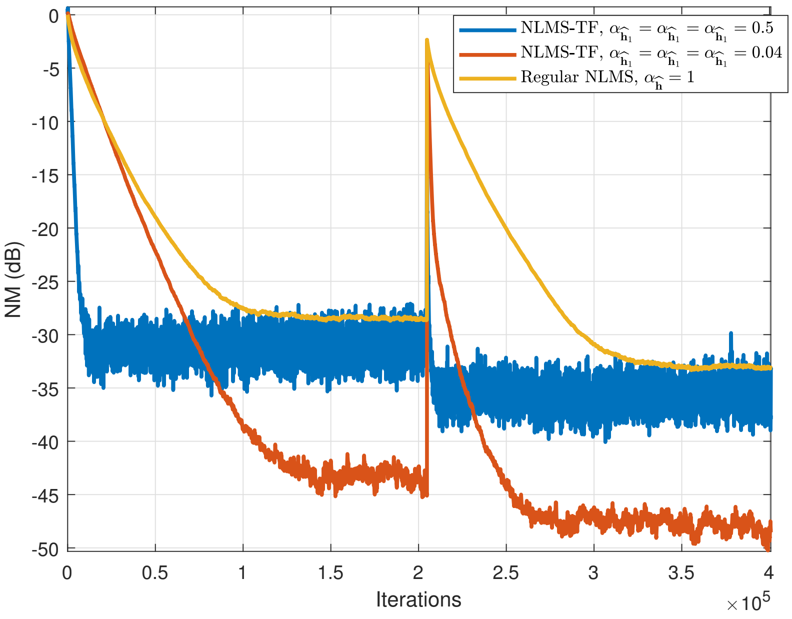


Figure 3.9: NM of the NLMS-TF and regular NLMS algorithms. The impulse response $\mathbf{h}_2$ changes in the middle of the experiment.

impulse response $\mathbf{h}_2$ in the middle of the experiment. The results are presented in Fig. 3.9. The improvement brought by the NLMS-TF algorithm is clear. The algorithm tracks faster after the change of the system, while the value of the NM is smaller, as compared to the regular NLMS filter. This proves that even if the environment changes, the proposed approach exhibits good behavior.

3.5 Summary and Conclusions

In this chapter, we addressed the problem of multilinear system identification, focusing in particular on TF in the framework of MISO systems. TF are defined with respect to the impulse responses of the system and are treated using third-order tensors. In this context, we derived the corresponding Wiener filter, as well as the LMS and NLMS adaptive algorithms, tailored for such TF (LMS-TF and NLMS-TF). Following this development, the solution can then be extended to higher-order multilinear systems. The proposed approach presents an advantage from the perspective of exploiting the decomposition of the global impulse response. One potential limitation of such a method is related to the particular form of the global impulse response to be identified, which is a result of separable systems. In perspective, it would be useful to extend this approach to identify more general forms of impulse responses. We have made a first step in direction and we developed a Kalman filter applicable to impulse responses that are not perfectly separable, as it will be seen in Chapter 4.

We have shown through simulations that the proposed algorithms lead to better solutions as compared to their regular counterparts, due to the reformulation of the system identification problem of high dimension in lower dimension problems. Experimental results support the theoretical analysis and highlight the good performance of the proposed solutions for the problem of system identification. Future work can focus on extending the approach to higher-order multilinear systems.

Chapter 4

Nearest Kronecker Product (NKP) Decomposition and Low-Rank Approximation

The work in this chapter was motivated by a well-known limitation of the previous system identification solutions. System identification problems are very difficult to treat in case of long length impulse responses [81, 47]. In these cases, the length of the filter may reach hundreds or even thousands of coefficients, raising challenges in terms of convergence, complexity, and accuracy of the solution. Moreover, another possible issue is the large parameter space [80, 82]. Nowadays, such scenarios are related to very important topics, for example, big data [83], machine learning [84], and source separation [85]. However, we can take advantage of the characteristics of the impulse response, in order to improve the overall performance. In this context, a recently introduced approach exploits a Kronecker product decomposition of the impulse response in tandem with low-rank approximations [43]. Also, a recursive least-squares (RLS) algorithm was developed based on this idea, showing appealing results for the identification of low-rank systems, like typical echo paths [86]. In this context, we developed a KF tailored for the identification of such low-rank systems. Simulations performed in the context of echo cancellation indicate that the proposed algorithm outperforms the regular KF, but also its RLS-based counterpart.

The results from this chapter were submitted for publication in [87].

4.1 An Efficient Approach for Low-Rank System Identification

Let us consider a linear system defined by the real-valued impulse response $\mathbf{h}$ of length $L = L_1 L_2$, where $L_1 \geq L_2$. This impulse response can be decomposed in L_2 short impulse responses of length L_1 each, such that $\mathbf{h} = \begin{bmatrix} \mathbf{s}_1^T & \mathbf{s}_2^T & \dots & \mathbf{s}_{L_2}^T \end{bmatrix}^T$, where $\mathbf{s}_l,\ l = 1, 2, \dots, L_2$ are the short impulse responses and the superscript ^{T} denotes the transpose operator. Alternatively, we can reshape $\mathbf{h}$ into a matrix of size $L_1 \times L_2$, which is $\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_{L_2} \end{bmatrix}$. At this point, the question is how well $\mathbf{h}$ can be approximated by $\mathbf{h}_2 \otimes \mathbf{h}_1$, where $\mathbf{h}_1$ and $\mathbf{h}_2$ are two impulse responses of lengths L_1 and L_2, respectively, and $\otimes$ is the Kronecker product. Hence, we can define a measure based on the NM [88], which can be expressed into two equivalent forms:

$$\mathcal{M}\left(\mathbf{h}_1, \mathbf{h}_2\right) = \frac{\left\|\mathbf{h} - \mathbf{h}_2 \otimes \mathbf{h}_1\right\|}{\left\|\mathbf{h}\right\|} = \frac{\left\|\mathbf{H} - \mathbf{h}_1 \mathbf{h}_2^T\right\|_{\mathrm{F}}}{\left\|\mathbf{H}\right\|_{\mathrm{F}}}, \tag{4.1}$$

where $\|\cdot\|$ and $\|\cdot\|_{\mathrm{F}}$ denote the Euclidean and Frobenius norms, respectively. To find the optimal values of $\mathbf{h}_1$ and $\mathbf{h}_2$, we need to minimize $\mathcal{M}\left(\mathbf{h}_1, \mathbf{h}_2\right)$. Alternatively, minimizing this quantity is equivalent to finding the nearest rank-1 matrix to $\mathbf{H}$ [88], whose solution is obtained from the well-known singular value decomposition (SVD) [62].

Next, using the SVD, we can express $\mathbf{H} = \mathbf{U}_1 \mathbf{\Sigma} \mathbf{U}_2^T = \sum_{l=1}^{L_2} \sigma_l \mathbf{u}_{1,l} \mathbf{u}_{2,l}^T$, where $\mathbf{U}_1$ and $\mathbf{U}_2$ are two orthogonal matrices of sizes $L_1 \times L_1$ and $L_2 \times L_2$, respectively, and $\mathbf{\Sigma}$

is an $L_1 \times L_2$ rectangular diagonal matrix with nonnegative real numbers on its main diagonal. The vectors $\mathbf{u}_{1,l}$ (resp. $\mathbf{u}_{2,l}$), with $l = 1, 2, \ldots, L_2$, denote the columns of $\mathbf{U}_1$ (resp. $\mathbf{U}_2$); in fact, they are the left-singular (resp. right-singular) vectors of $\mathbf{H}$, while the diagonal entries $\sigma_l,\ l = 1, 2, \ldots, L_2$ of $\boldsymbol{\Sigma}$ represent the singular values of $\mathbf{H}$, with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{L_2} \geq 0$. Hence, the optimal impulse responses that minimize $\mathcal{M}(\mathbf{h}_1, \mathbf{h}_2)$ are $\overline{\mathbf{h}}_1 = \sqrt{\sigma_1}\mathbf{u}_{1,1}$ and $\overline{\mathbf{h}}_2 = \sqrt{\sigma_1}\mathbf{u}_{2,1}$, where $\mathbf{u}_{1,1}$ (resp. $\mathbf{u}_{2,1}$) is the first column of $\mathbf{U}_1$ (resp. $\mathbf{U}_2$). Therefore, the optimal approximation of $\mathbf{h}$ is $\overline{\mathbf{h}} = \overline{\mathbf{h}}_2 \otimes \overline{\mathbf{h}}_1$. Clearly, there is no unique set of solutions, since $\overline{\mathbf{h}}_2 \otimes \overline{\mathbf{h}}_1 = \eta\overline{\mathbf{h}}_2 \otimes (1/\eta)\overline{\mathbf{h}}_1$, where $\eta \neq 0$ is a real-valued number. However, the global impulse response $\overline{\mathbf{h}}$ results with no scaling ambiguity.

In the more general case, the short impulse responses that compose $\mathbf{h}$ (i.e., $\mathbf{s}_l,\ l = 1, 2, \ldots, L_2$) may not be that linearly dependent. Consequently, we can use the approximation $\mathbf{h} \approx \sum_{p=1}^{P} \mathbf{h}_{2,p} \otimes \mathbf{h}_{1,p} = \text{vec}\left(\mathbf{H}_1\mathbf{H}_2^T\right)$, where $P \leq L_2$, $\mathbf{h}_{1,p}$ and $\mathbf{h}_{2,p}$ are impulse responses of lengths L_1 and L_2, respectively, vec$(\cdot)$ denotes the vectorization operation (i.e., conversion of a matrix into a vector [42]), and $\mathbf{H}_1 = \begin{bmatrix} \mathbf{h}_{1,1} & \mathbf{h}_{1,2} & \ldots & \mathbf{h}_{1,P} \end{bmatrix}$ and $\mathbf{H}_2 = \begin{bmatrix} \mathbf{h}_{2,1} & \mathbf{h}_{2,2} & \ldots & \mathbf{h}_{2,P} \end{bmatrix}$ are matrices of sizes $L_1 \times P$ and $L_2 \times P$, respectively. In this context, the problem is to minimize

$$\mathcal{M}(\mathbf{H}_1, \mathbf{H}_2) = \frac{\left\|\mathbf{H} - \mathbf{H}_1\mathbf{H}_2^T\right\|_{\mathrm{F}}}{\|\mathbf{H}\|_{\mathrm{F}}}, \tag{4.2}$$

which leads to the optimal solutions

$$\overline{\mathbf{H}}_1 = \begin{bmatrix} \overline{\mathbf{h}}_{1,1} & \overline{\mathbf{h}}_{1,2} & \ldots & \overline{\mathbf{h}}_{1,P} \end{bmatrix} = \begin{bmatrix} \sqrt{\sigma_1}\mathbf{u}_{1,1} & \sqrt{\sigma_2}\mathbf{u}_{1,2} & \ldots & \sqrt{\sigma_P}\mathbf{u}_{1,P} \end{bmatrix} \tag{4.3}$$

$$\overline{\mathbf{H}}_2 = \begin{bmatrix} \overline{\mathbf{h}}_{2,1} & \overline{\mathbf{h}}_{2,2} & \ldots & \overline{\mathbf{h}}_{2,P} \end{bmatrix} = \begin{bmatrix} \sqrt{\sigma_1}\mathbf{u}_{2,1} & \sqrt{\sigma_2}\mathbf{u}_{2,2} & \ldots & \sqrt{\sigma_P}\mathbf{u}_{2,P} \end{bmatrix}, \tag{4.4}$$

where $\mathbf{u}_{1,p},\ p = 1, 2, \ldots, P$ (resp. $\mathbf{u}_{2,p},\ p = 1, 2, \ldots, P$) are the first P columns of $\mathbf{U}_1$ (resp. $\mathbf{U}_2$). Thus, the optimal approximation of $\mathbf{h}$ is

$$\overline{\mathbf{h}}(P) = \sum_{p=1}^{P} \overline{\mathbf{h}}_{2,p} \otimes \overline{\mathbf{h}}_{1,p} = \sum_{p=1}^{P} \sigma_p \mathbf{u}_{2,p} \otimes \mathbf{u}_{1,p}, \tag{4.5}$$

while the approximation error is $\mathbf{e}(P) = \mathbf{h} - \overline{\mathbf{h}}(P) = \sum_{i=P+1}^{L_2} \sigma_i \mathbf{u}_{2,i} \otimes \mathbf{u}_{1,i}$. To evaluate this approximation error we can use a performance measure based on the NM, i.e., $\|\mathbf{e}(P)\| / \|\mathbf{h}\|$.

As we can notice, the exact decomposition is obtained for $P = L_2$. However, if the rank of the matrix $\mathbf{H}$ is equal to $P < L_2$ (i.e., $\sigma_i = 0$, for $P < i \leq L_2$), then we can estimate $\mathbf{h}$ at least as well as in the conventional approach. Moreover, if the approximation of the impulse response is a good one (for a reasonably low value of P, as compared to L_2), important advantages can be obtained by using this decomposition approach for low-rank system identification problems [43]. The expected gain could be twofold, in terms of both performance and complexity.

A Wiener solution based on the nearest Kronecker product (NKP) was developed in previous works, along with an RLS adaptive filter [86]. However, we do not present these results here, since they are not necessary for the ease of understanding of the following developments. Instead, we continue by deriving a KF suitable for the identification of long length impulse responses, based on the approach discussed earlier in this chapter.

4.2 Kalman Filter Based on the Nearest Kronecker Product Decomposition (KF-NKP)

The KF [11] is a very popular signal processing tool, which is commonly used in the framework of many important applications, e.g., see [89] and references therein. Basically, it recursively estimates a set of unknown variables, based on a set of observations (corrupted by noise). Due to its practical features, the KF fits very well in the context of different real-world system identification problems.

One of the most challenging system identification applications is echo cancellation [47]. Its main difficulty is related to the long length of the impulse response to be identified (i.e., the echo path), which can be of the order of hundreds or even thousands of coefficients. Nevertheless, the KF also shows appealing results in the context of echo cancellation [16, 90, 91, 92].

It is known that some specific characteristics of the impulse response could be exploited during the identification process, in order to improve the overall performance. For example, the echo paths are sparse in nature (i.e., most of the coefficients are zero or small), which inspired the development of the proportionate-type algorithms, e.g., see [15, 49, 51] and references therein. Another approach exploits the decomposition of the impulse response, reformulating the problem in terms of the identification of bilinear/trilinear forms [19, 35, 79, 80]. However, most of these algorithms are designed in the framework of perfectly separable systems, which is not always the case in practical applications.

Recently, an improved solution was presented in [43], by exploiting the NKP decomposition of the impulse response, together with low-rank approximation, thus being applicable for the identification of more general forms of low-rank systems (like echo paths). These techniques were previously addressed in the context of tensor decompositions and modeling, in the framework of different applications [85, 84, 69, 93, 94, 95]. In [86], an RLS algorithm was developed based on the approach introduced in [43]. In this work, we further extend this idea using the KF, showing improved results as compared to the RLS-based solution.

In the general framework of a system identification problem, the reference signal is usually defined as the output of an unknown system, corrupted by an additive noise. Hence, the signal model is

$$d(t) = \mathbf{h}^T(t)\mathbf{x}(t) + v(t) = y(t) + v(t), \tag{4.6}$$

where $d(t)$ is the zero-mean desired (or reference) signal at the discrete-time index t, $\mathbf{h}(t)$ is the impulse response of the unknown system (of length L), $\mathbf{x}(t) = \begin{bmatrix} x(t) & x(t-1) & \cdots & x(t-L+1) \end{bmatrix}^T$ is a vector containing the most recent L time samples of the zero-mean input signal $x(t)$, and $v(t)$ is the zero-mean additive noise with the variance $\sigma_v^2 = E\left[v^2(t)\right]$, where $E[\cdot]$ denotes mathematical expectation. It is assumed that all the data is real valued and $\mathbf{x}(t)$ and $v(t)$ are uncorrelated.

The main goal is to estimate $\mathbf{h}(t)$ with an adaptive filter, $\widehat{\mathbf{h}}(t)$, of length L. In this context, the estimation error is defined as the difference between the desired signal, $d(t)$, and the output of the adaptive filter (i.e., the estimated signal), $\widehat{y}(t)$, which results in

$$e(t) = d(t) - \widehat{y}(t) = d(t) - \widehat{\mathbf{h}}^T(t-1)\mathbf{x}(t). \tag{4.7}$$

Based on the approach presented in Section 4.1, let us assume that $L = L_1 L_2$ (with $L_1 \geq L_2$) and $\text{rank}\,(\mathbf{H}) = P < L_2$, so that the impulse response can be decomposed as in (4.5), i.e., $\mathbf{h}(t) = \sum_{p=1}^{P} \mathbf{h}_{2,p}(t) \otimes \mathbf{h}_{1,p}(t)$, where $\mathbf{h}_{1,p}(t)$ and $\mathbf{h}_{2,p}(t)$ are impulse responses of lengths L_1 and L_2, respectively. These short impulse responses can be grouped into two vectors of lengths PL_1 and PL_2, respectively, so that $\underline{\mathbf{h}}_1(t) = \begin{bmatrix} \mathbf{h}_{1,1}^T(t) & \mathbf{h}_{1,2}^T(t) & \cdots & \mathbf{h}_{1,P}^T(t) \end{bmatrix}^T$ and $\underline{\mathbf{h}}_2(t) = \begin{bmatrix} \mathbf{h}_{2,1}^T(t) & \mathbf{h}_{2,2}^T(t) & \cdots & \mathbf{h}_{2,P}^T(t) \end{bmatrix}^T$. A similar decomposition can be used for the adaptive filter, i.e.,

$$\widehat{\mathbf{h}}(t) = \sum_{p=1}^{P} \widehat{\mathbf{h}}_{2,p}(t) \otimes \widehat{\mathbf{h}}_{1,p}(t), \tag{4.8}$$

where $\widehat{\mathbf{h}}_{1,p}(t)$ and $\widehat{\mathbf{h}}_{2,p}(t)$ are filters of lengths L_1 and L_2, respectively. Using the relationships [42]:

$$\widehat{\mathbf{h}}_{2,p}(t) \otimes \widehat{\mathbf{h}}_{1,p}(t) = \left[\widehat{\mathbf{h}}_{2,p}(t) \otimes \mathbf{I}_{L_1}\right] \widehat{\mathbf{h}}_{1,p}(t) = \left[\mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{1,p}(t)\right] \widehat{\mathbf{h}}_{2,p}(t), \tag{4.9}$$

where $\mathbf{I}_{L_1}$ and $\mathbf{I}_{L_2}$ are the identity matrices of sizes $L_1 \times L_1$ and $L_2 \times L_2$, respectively, into (4.8), together with the notation $\mathbf{x}_{2,p}(t) = \left[\widehat{\mathbf{h}}_{2,p}(t-1) \otimes \mathbf{I}_{L_1}\right]^T \mathbf{x}(t)$ and $\mathbf{x}_{1,p}(t) = \left[\mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{1,p}(t-1)\right]^T \mathbf{x}(t)$, the error signal in (4.7) can be expressed into two equivalent forms:

$$e_1(t) = d(t) - \sum_{p=1}^{P} \widehat{\mathbf{h}}_{1,p}^T(t-1)\mathbf{x}_{2,p}(t) = d(t) - \widehat{\underline{\mathbf{h}}}_1^T(t-1)\underline{\mathbf{x}}_2(t), \tag{4.10}$$

$$e_2(t) = d(t) - \sum_{p=1}^{P} \widehat{\mathbf{h}}_{2,p}^T(t-1)\mathbf{x}_{1,p}(t) = d(t) - \widehat{\underline{\mathbf{h}}}_2^T(t-1)\underline{\mathbf{x}}_1(t), \tag{4.11}$$

where
$\widehat{\underline{\mathbf{h}}}_1(t) = \begin{bmatrix} \widehat{\mathbf{h}}_{1,1}^T(t) & \widehat{\mathbf{h}}_{1,2}^T(t) & \cdots & \widehat{\mathbf{h}}_{1,P}^T(t) \end{bmatrix}^T$, $\underline{\mathbf{x}}_2(t) = \begin{bmatrix} \mathbf{x}_{2,1}^T(t) & \mathbf{x}_{2,2}^T(t) & \cdots & \mathbf{x}_{2,P}^T(t) \end{bmatrix}^T$,
$\widehat{\underline{\mathbf{h}}}_2(t) = \begin{bmatrix} \widehat{\mathbf{h}}_{2,1}^T(t) & \widehat{\mathbf{h}}_{2,2}^T(t) & \cdots & \widehat{\mathbf{h}}_{2,P}^T(t) \end{bmatrix}^T$, $\underline{\mathbf{x}}_1(t) = \begin{bmatrix} \mathbf{x}_{1,1}^T(t) & \mathbf{x}_{1,2}^T(t) & \cdots & \mathbf{x}_{1,P}^T(t) \end{bmatrix}^T$.

Following this decomposition approach, the original system identification problem (of length L_1L_2) can be reformulated in terms of two shorter filters, of lengths PL_1 and PL_2, respectively. Furthermore, the problem can be developed similarly to a bilinear optimization strategy [19]. As it was seen in the previous chapters, in the framework of the KF, the impulse responses can be modeled as state equations, while the expression of the desired signal represents the observation equation. Thus, we consider that $\underline{\mathbf{h}}_1(t)$ and $\underline{\mathbf{h}}_2(t)$ are zero-mean random vectors, which follow the simplified first-order Markov models:

$$\underline{\mathbf{h}}_1(t) = \underline{\mathbf{h}}_1(t-1) + \mathbf{w}_1(t), \tag{4.12}$$

$$\underline{\mathbf{h}}_2(t) = \underline{\mathbf{h}}_2(t-1) + \mathbf{w}_2(t), \tag{4.13}$$

where $\mathbf{w}_1(t)$ and $\mathbf{w}_2(t)$ are zero-mean WGN vectors, with correlation matrices $\mathbf{R}_{\mathbf{w}_1} = \sigma_{w_1}^2 \mathbf{I}_{PL_1}$ and $\mathbf{R}_{\mathbf{w}_2} = \sigma_{w_2}^2 \mathbf{I}_{PL_2}$, respectively; here, $\mathbf{I}_{PL_1}$ and $\mathbf{I}_{PL_2}$ denote the identity matrices of size $PL_1 \times PL_1$ and $PL_2 \times PL_2$, respectively. It is considered that $\mathbf{w}_1(t)$ is uncorrelated with $\underline{\mathbf{h}}_1(t-1)$ and $v(t)$, while $\mathbf{w}_2(t)$ is uncorrelated with $\underline{\mathbf{h}}_2(t-1)$

and $v(t)$. The variances $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ capture the uncertainties in $\underline{\mathbf{h}}_1(t)$ and $\underline{\mathbf{h}}_2(t)$, respectively.

In the context of the linear sequential Bayesian approach, the optimal estimates of the state vectors result as [63]

$$\widehat{\underline{\mathbf{h}}}_1(t) = \widehat{\underline{\mathbf{h}}}_1(t-1) + \mathbf{k}_1(t)e_1(t), \tag{4.14}$$

$$\widehat{\underline{\mathbf{h}}}_2(t) = \widehat{\underline{\mathbf{h}}}_2(t-1) + \mathbf{k}_2(t)e_2(t), \tag{4.15}$$

where $\mathbf{k}_1(t)$ and $\mathbf{k}_2(t)$ are the Kalman gain vectors. Next, let us define the a posteriori misalignments (which represent the state estimation errors) related to the two impulse responses as $\boldsymbol{\mu}_1(t) = \underline{\mathbf{h}}_1(t) - \widehat{\underline{\mathbf{h}}}_1(t)$ and $\boldsymbol{\mu}_2(t) = \underline{\mathbf{h}}_2(t) - \widehat{\underline{\mathbf{h}}}_2(t)$, for which their correlation matrices are $\mathbf{R}_{\boldsymbol{\mu}_1}(t) = E\left[\boldsymbol{\mu}_1(t)\boldsymbol{\mu}_1^T(t)\right]$ and $\mathbf{R}_{\boldsymbol{\mu}_2}(t) = E\left[\boldsymbol{\mu}_2(t)\boldsymbol{\mu}_2^T(t)\right]$, respectively. As mentioned in Section 4.1, we can only identify the impulse responses up to an arbitrary scaling factor η; however, the pair $\underline{\mathbf{h}}_1(t)/\eta$ and $\eta\underline{\mathbf{h}}_2(t)$ is equivalent to the pair $\underline{\mathbf{h}}_1(t)$ and $\underline{\mathbf{h}}_2(t)$. Therefore, in order to simplify the notation, the scaling factor η does not appear explicitly in the following.

Furthermore, we can define the a priori misalignments related to these two impulse responses:

$$\mathbf{m}_1(t) = \underline{\mathbf{h}}_1(t) - \widehat{\underline{\mathbf{h}}}_1(t-1) = \boldsymbol{\mu}_1(t-1) + \mathbf{w}_1(t), \tag{4.16}$$

$$\mathbf{m}_2(t) = \underline{\mathbf{h}}_2(t) - \widehat{\underline{\mathbf{h}}}_2(t-1) = \boldsymbol{\mu}_2(t-1) + \mathbf{w}_2(t), \tag{4.17}$$

whose correlation matrices are $\mathbf{R}_{\mathbf{m}_1}(t) = E\left[\mathbf{m}_1(t)\mathbf{m}_1^T(t)\right]$ and $\mathbf{R}_{\mathbf{m}_2}(t) = E\left[\mathbf{m}_2(t)\mathbf{m}_2^T(t)\right]$, respectively. Also, developing in (4.16) and (4.17), we get

$$\mathbf{R}_{\mathbf{m}_1}(t) = \mathbf{R}_{\boldsymbol{\mu}_1}(t-1) + \mathbf{R}_{\mathbf{w}_1}, \tag{4.18}$$

$$\mathbf{R}_{\mathbf{m}_2}(t) = \mathbf{R}_{\boldsymbol{\mu}_2}(t-1) + \mathbf{R}_{\mathbf{w}_2}. \tag{4.19}$$

The Kalman gain vectors are obtained by minimizing the criterions $J_1(t) = 1/(PL_1)\mathrm{tr}\left[\mathbf{R}_{\boldsymbol{\mu}_1}(t)\right]$ and $J_2(t) = 1/(PL_2)\mathrm{tr}\left[\mathbf{R}_{\boldsymbol{\mu}_2}(t)\right]$, with respect to $\mathbf{k}_1(t)$ and $\mathbf{k}_2(t)$, respectively, where $\mathrm{tr}\left[\cdot\right]$ denotes the trace of a square matrix. From these minimizations, we find that

$$\mathbf{k}_1(t) = \mathbf{R}_{\mathbf{m}_1}(t)\underline{\mathbf{x}}_2(t)\left[\underline{\mathbf{x}}_2^T(t)\mathbf{R}_{\mathbf{m}_1}(t)\underline{\mathbf{x}}_2(t) + \sigma_v^2\right]^{-1}, \tag{4.20}$$

$$\mathbf{k}_2(t) = \mathbf{R}_{\mathbf{m}_2}(t)\underline{\mathbf{x}}_1(t)\left[\underline{\mathbf{x}}_1^T(t)\mathbf{R}_{\mathbf{m}_2}(t)\underline{\mathbf{x}}_1(t) + \sigma_v^2\right]^{-1}, \tag{4.21}$$

together with

$$\mathbf{R}_{\boldsymbol{\mu}_1}(t) = \left[\mathbf{I}_{PL_1} - \mathbf{k}_1(t)\underline{\mathbf{x}}_2^T(t)\right]\mathbf{R}_{\mathbf{m}_1}(t), \tag{4.22}$$

$$\mathbf{R}_{\boldsymbol{\mu}_2}(t) = \left[\mathbf{I}_{PL_2} - \mathbf{k}_2(t)\underline{\mathbf{x}}_1^T(t)\right]\mathbf{R}_{\mathbf{m}_2}(t). \tag{4.23}$$

The resulting KF based on the NKP decomposition (namely KF-NKP) is summarized in Table 4.1.

As we can notice, the KF-NKP involves matrices of size $PL_1 \times PL_1$ and $PL_2 \times PL_2$, thus resulting a complexity order of $\mathcal{O}\left[(PL_1)^2 + (PL_2)^2\right]$. On the other hand, the regular KF would involve matrices of size $L_1L_2 \times L_1L_2$, so that the complexity order would be $\mathcal{O}\left[(L_1L_2)^2\right]$. Consequently, when $P \ll L_2 \leq L_1$, it is much more convenient to use the proposed KF-NKP instead of the regular KF. In this context, in order to maximize the gain in terms of complexity, it is reasonable to select the value of L_1 close to the value of L_2, such that the sum $L_1 + L_2$ should be as small as possible as compared to the product L_1L_2 [43, 86].

The computational complexity is also illustrated in Fig. 4.1, in terms of the number of multiplications (per iteration) required by the proposed KF-NKP and the regular KF, considering two impulse responses of different lengths. When $L = 500$ [Fig. 4.1(a)], the decomposition is performed using $L_1 = 25$ and $L_2 = 20$, while for $L = 1024$ [Fig. 4.1(b)], we can use $L_1 = L_2 = 32$. As we can notice in both cases, the

Table 4.1: Kalman filter based on the nearest Kronecker product decomposition (KF-NKP).

Initialization:

$\widehat{\mathbf{h}}_{1,p}(0) = \left[\begin{array}{cccc} \epsilon & 0 & \cdots & 0 \end{array}\right]^T, \quad \widehat{\mathbf{h}}_{2,p}(0) = \left[\begin{array}{cccc} \epsilon & 0 & \cdots & 0 \end{array}\right]^T, \ p = 1, 2, \ldots, P \ (0 < \epsilon \leq 1)$

$\widehat{\underline{\mathbf{h}}}_1(0) = \left[\begin{array}{cccc} \widehat{\mathbf{h}}_{1,1}^T(0) & \widehat{\mathbf{h}}_{1,2}^T(0) & \cdots & \widehat{\mathbf{h}}_{1,P}^T(0) \end{array}\right]^T, \quad \widehat{\underline{\mathbf{h}}}_2(0) = \left[\begin{array}{cccc} \widehat{\mathbf{h}}_{2,1}^T(0) & \widehat{\mathbf{h}}_{2,2}^T(0) & \cdots & \widehat{\mathbf{h}}_{2,P}^T(0) \end{array}\right]^T$

$\mathbf{R}_{\boldsymbol{\mu}_1}(0) = \epsilon \mathbf{I}_{PL_1}, \quad \mathbf{R}_{\boldsymbol{\mu}_2}(0) = \epsilon \mathbf{I}_{PL_2}, \ \epsilon =$ small positive constant

Parameters: $\sigma_{w_1}^2, \ \sigma_{w_2}^2, \ \sigma_v^2$ known or estimated; $\quad \mathbf{R}_{\mathbf{w}_1} = \sigma_{w_1}^2 \mathbf{I}_{PL_1}, \ \mathbf{R}_{\mathbf{w}_2} = \sigma_{w_2}^2 \mathbf{I}_{PL_2}$

Algorithm:

For $t = 1, 2, \ldots$

$\mathbf{x}_{2,p}(t) = \left[\widehat{\mathbf{h}}_{2,p}(t-1) \otimes \mathbf{I}_{L_1}\right]^T \mathbf{x}(t), \quad \mathbf{x}_{1,p}(t) = \left[\mathbf{I}_{L_2} \otimes \widehat{\mathbf{h}}_{1,p}(t-1)\right]^T \mathbf{x}(t), \ p = 1, 2, \ldots, P$

$\underline{\mathbf{x}}_2(t) = \left[\begin{array}{cccc} \mathbf{x}_{2,1}^T(t) & \mathbf{x}_{2,2}^T(t) & \cdots & \mathbf{x}_{2,P}^T(t) \end{array}\right]^T, \quad \underline{\mathbf{x}}_1(t) = \left[\begin{array}{cccc} \mathbf{x}_{1,1}^T(t) & \mathbf{x}_{1,2}^T(t) & \cdots & \mathbf{x}_{1,P}^T(t) \end{array}\right]^T$

$\mathbf{R}_{\mathbf{m}_1}(t) = \mathbf{R}_{\boldsymbol{\mu}_1}(t-1) + \mathbf{R}_{\mathbf{w}_1}, \quad \mathbf{R}_{\mathbf{m}_2}(t) = \mathbf{R}_{\boldsymbol{\mu}_2}(t-1) + \mathbf{R}_{\mathbf{w}_2}$

$\mathbf{k}_1(t) = \mathbf{R}_{\mathbf{m}_1}(t)\underline{\mathbf{x}}_2(t)\left[\underline{\mathbf{x}}_2^T(t)\mathbf{R}_{\mathbf{m}_1}(t)\underline{\mathbf{x}}_2(t) + \sigma_v^2\right]^{-1}, \quad \mathbf{k}_2(t) = \mathbf{R}_{\mathbf{m}_2}(t)\underline{\mathbf{x}}_1(t)\left[\underline{\mathbf{x}}_1^T(t)\mathbf{R}_{\mathbf{m}_2}(t)\underline{\mathbf{x}}_1(t) + \sigma_v^2\right]^{-1}$

$\mathbf{R}_{\boldsymbol{\mu}_1}(t) = \left[\mathbf{I}_{PL_1} - \mathbf{k}_1(t)\underline{\mathbf{x}}_2^T(t)\right]\mathbf{R}_{\mathbf{m}_1}(t), \quad \mathbf{R}_{\boldsymbol{\mu}_2}(t) = \left[\mathbf{I}_{PL_2} - \mathbf{k}_2(t)\underline{\mathbf{x}}_1^T(t)\right]\mathbf{R}_{\mathbf{m}_2}(t)$

$e(t) = d(t) - \widehat{\underline{\mathbf{h}}}_1^T(t-1)\underline{\mathbf{x}}_2(t) = d(t) - \widehat{\underline{\mathbf{h}}}_2^T(t-1)\underline{\mathbf{x}}_1(t)$

$\widehat{\underline{\mathbf{h}}}_1(t) = \widehat{\underline{\mathbf{h}}}_1(t-1) + \mathbf{k}_1(t)e(t) = \left[\begin{array}{ccc} \widehat{\mathbf{h}}_{1,1}^T(t) & \cdots & \widehat{\mathbf{h}}_{1,P}^T(t) \end{array}\right]^T$

$\widehat{\underline{\mathbf{h}}}_2(t) = \widehat{\underline{\mathbf{h}}}_2(t-1) + \mathbf{k}_2(t)e(t) = \left[\begin{array}{ccc} \widehat{\mathbf{h}}_{2,1}^T(t) & \cdots & \widehat{\mathbf{h}}_{2,P}^T(t) \end{array}\right]^T$

$\widehat{\mathbf{h}}(t) = \sum_{p=1}^{P} \widehat{\mathbf{h}}_{2,p}(t) \otimes \widehat{\mathbf{h}}_{1,p}(t)$

computational complexity of the KF-NKP exceeds its regular counterpart for large values of P (i.e., closer to L_2). However, the proposed approach is not designed for such cases, but for the identification of low-rank impulse responses, where the rank of the matrix $\mathbf{H}$ is usually much lower as compared to L_2. As will be illustrated in the experiments reported in Section 4.2.2, the KF-NKP performs very well for the identification of such systems (using a reasonably low value of P, as compared to L_2).

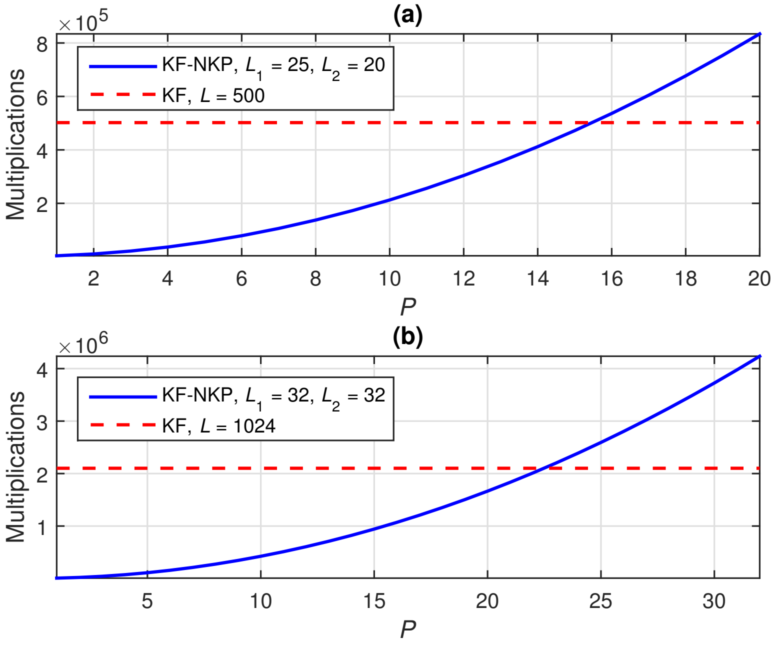


Figure 4.1: Number of multiplications (per iteration) required by the KF-NKP and KF, as a function of P. The KF-NKP uses two shorter filters of lengths PL_1 and PL_2 (with $P \leq L_2$), while the length of the KF is $L = L_1 L_2$: (a) $L_1 = 25$, $L_2 = 20$, and (b) $L_1 = L_2 = 32$.

4.2.1 Study of the KF-NKP Parameters

There are three main parameters that have to be set or estimated within the proposed KF-NKP. The first one is the system noise power, σ_v^2, which can be estimated in different ways, depending on the application. For example, in echo cancellation scenarios, this parameter can be estimated during the silence periods of the near-end talker [96]. Also, other practical methods can be found in [12]. Their analysis is beyond the scope of this work.

The other two specific parameters of the KF-NKP are $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$, which are related to the model uncertainties and reflect the capability of the algorithm to track the changes of the system. Nevertheless, there is always a compromise between good tracking and low misalignment. When the values of $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ are close to zero, the KF-NKP would lead to an accurate solution in terms of system identification (i.e., low misalignment), but its tracking capability would be reduced. On the other hand, increasing the values of $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ improves the tracking, but by paying the price of a higher misalignment. Based on these considerations, different strategies can be used to set these parameters in practice. For example, in case of the regular KF, several methods can be found in [16, 91]. Due to the nature of our system identification problem (i.e., low-rank systems, where sparsity is a helping factor), the solution proposed in [91] is more reliable to be adapted in case of the KF-NKP, as explained in the following.

In (4.12)–(4.13), $\mathbf{w}_1(t)$ and $\mathbf{w}_2(t)$ are considered zero-mean WGN signals; thus, their correlation matrices are diagonal and all the coefficients have the same level of uncertainty. However, in the context of low-rank systems, we could consider a more realistic case, with independent fluctuations of each coefficient, so that

$$\underline{\mathbf{h}}_1(t) = \underline{\mathbf{h}}_1(t-1) + \widetilde{\mathbf{w}}_1(t), \tag{4.24}$$

$$\underline{\mathbf{h}}_2(t) = \underline{\mathbf{h}}_2(t-1) + \widetilde{\mathbf{w}}_2(t), \tag{4.25}$$

where

$$\widetilde{\mathbf{w}}_1(t) = \left[\begin{array}{cccc} \widetilde{w}_{1,0}(t) & \widetilde{w}_{1,1}(t) & \cdots & \widetilde{w}_{1,PL_1-1}(t) \end{array} \right]^T, \tag{4.26}$$

$$\widetilde{\mathbf{w}}_2(t) = \left[\begin{array}{cccc} \widetilde{w}_{2,0}(t) & \widetilde{w}_{2,1}(t) & \cdots & \widetilde{w}_{2,PL_2-1}(t) \end{array} \right]^T, \tag{4.27}$$

with

$$E\left[\widetilde{w}_{1,k}(t)\widetilde{w}_{1,l}(t)\right] = \begin{cases} \sigma_{\widetilde{w}_{1,l}}^2, & k = l \\ 0, & k \neq l \end{cases}, \ k, l = 0, 1, \ldots, PL_1 - 1, \tag{4.28}$$

$$E\left[\widetilde{w}_{2,i}(t)\widetilde{w}_{2,j}(t)\right] = \begin{cases} \sigma^2_{\widetilde{w}_{2,j}}, & i = j \\ 0, & i \neq j \end{cases}, \ i,j = 0,1,\ldots,PL_2-1. \tag{4.29}$$

Thus, the correlation matrices that should appear in (4.18)–(4.19) become $\mathbf{R}_{\widetilde{\mathbf{w}}_1} = \text{diag}\left[\sigma^2_{\widetilde{w}_{1,0}}, \sigma^2_{\widetilde{w}_{1,1}}, \ldots, \sigma^2_{\widetilde{w}_{1,PL_1-1}}\right]$ and $\mathbf{R}_{\widetilde{\mathbf{w}}_2} = \text{diag}\left[\sigma^2_{\widetilde{w}_{2,0}}, \sigma^2_{\widetilde{w}_{2,1}}, \ldots, \sigma^2_{\widetilde{w}_{2,PL_2-1}}\right]$, where diag[$\cdot$] denotes a diagonal matrix.

Next, based on the state equations (4.24)–(4.25), we can express $\widetilde{w}_{1,l}(t) = \underline{h}_{1,l}(t) - \underline{h}_{1,l}(t-1),\ l = 0,1,\ldots,PL_1-1$ and $\widetilde{w}_{2,j}(t) = \underline{h}_{2,j}(t) - \underline{h}_{2,j}(t-1),\ j = 0,1,\ldots,PL_2-1$, so that

$$\sigma^2_{\widetilde{w}_{1,l}} = E\left[\widetilde{w}^2_{1,l}(t)\right] = E\left\{\left[\underline{h}_{1,l}(t) - \underline{h}_{1,l}(t-1)\right]^2\right\}, \tag{4.30}$$

$$\sigma^2_{\widetilde{w}_{2,j}} = E\left[\widetilde{w}^2_{2,j}(t)\right] = E\left\{\left[\underline{h}_{2,j}(t) - \underline{h}_{2,j}(t-1)\right]^2\right\}. \tag{4.31}$$

These parameters reflect the individual levels of uncertainty for the coefficients of $\underline{\mathbf{h}}_1(t)$ and $\underline{\mathbf{h}}_2(t)$, respectively. Therefore, we obtain a version of the KF-NKP with individual control factors, taking into consideration the variation of each coefficient from one iteration to the next. Based on (4.30)–(4.31), we can recursively estimate

$$\widehat{\sigma}^2_{\widetilde{w}_{1,l}}(t) = \alpha_1\widehat{\sigma}^2_{\widetilde{w}_{1,l}}(t-1) + (1-\alpha_1)\left[\widehat{\underline{h}}_{1,l}(t-1) - \widehat{\underline{h}}_{1,l}(t-2)\right]^2, \quad l = 0,1,\ldots,PL_1-1, \tag{4.32}$$

$$\widehat{\sigma}^2_{\widetilde{w}_{2,j}}(t) = \alpha_2\widehat{\sigma}^2_{\widetilde{w}_{2,j}}(t-1) + (1-\alpha_2)\left[\widehat{\underline{h}}_{2,j}(t-1) - \widehat{\underline{h}}_{2,j}(t-2)\right]^2, \quad j = 0,1,\ldots,PL_2-1, \tag{4.33}$$

with $\alpha_1 = 1 - 1/(\kappa_1 PL_1),\ \kappa_1 \geq 1$, and $\alpha_2 = 1 - 1/(\kappa_2 PL_2),\ \kappa_2 \geq 1$. As we can notice, when using $\alpha_1 = \alpha_2 = 0$ (i.e., without temporal averaging), the mean values of (4.32) and (4.33) become

$$\widehat{\sigma}^2_{\widetilde{w}_1}(t) = \frac{1}{PL_1}\sum_{l=0}^{PL_1-1}\left[\widehat{\underline{h}}_{l,1}(t-1) - \widehat{\underline{h}}_{l,1}(t-2)\right]^2 = \frac{1}{PL_1}\left\|\widehat{\underline{\mathbf{h}}}_1(t-1) - \widehat{\underline{\mathbf{h}}}_1(t-2)\right\|^2, \tag{4.34}$$

$$\widehat{\sigma}^2_{\widetilde{w}_2}(t) = \frac{1}{PL_2} \sum_{l=0}^{PL_2-1} \left[\underline{\widehat{h}}_{l,2}(t-1) - \underline{\widehat{h}}_{l,2}(t-2) \right]^2 = \frac{1}{PL_2} \left\| \underline{\widehat{\mathbf{h}}}_2(t-1) - \underline{\widehat{\mathbf{h}}}_2(t-2) \right\|^2 . \tag{4.35}$$

In order to prevent large variations of the estimators (e.g., when the system suddenly changes), we may additionally control the parameters from (4.32)–(4.33), by limiting the levels of uncertainty to certain thresholds. These can be chosen as constant values (depending on the application) or even the mean values of the individual control parameters. To this purpose, we can use (4.32)–(4.33) in conjunction with (4.34)–(4.35), such that the final individual control factors become

$$\underline{\widehat{\sigma}}^2_{\widetilde{w}_{1,l}}(t) = \min\left\{ \widehat{\sigma}^2_{\widetilde{w}_{1,l}}(t), \widehat{\sigma}^2_{\widetilde{w}_1}(t) \right\},\ l = 0, 1, \ldots, PL_1 - 1, \tag{4.36}$$

$$\underline{\widehat{\sigma}}^2_{\widetilde{w}_{2,j}}(t) = \min\left\{ \widehat{\sigma}^2_{\widetilde{w}_{2,j}}(t), \widehat{\sigma}^2_{\widetilde{w}_2}(t) \right\},\ j = 0, 1, \ldots, PL_2 - 1. \tag{4.37}$$

Also, in some scenarios, when the impulse response to identify could be more dispersive (like the acoustic echo paths), it may be preferable to limit the uncertainty levels to a certain value that is associated to the global filter, e.g., $(1/L) \left\| \widehat{\mathbf{h}}(t-1) - \widehat{\mathbf{h}}(t-2) \right\|^2$, similar to the solution used in the context of the regular KF [16, 91].

4.2.2 Experimental Results

Simulations are performed in the context of echo cancellation, where the input signal $\mathbf{x}(t)$ is either an AR(1) process [generated by filtering a WGN through a first-order system $1/\left(1 - 0.9z^{-1}\right)$] or a speech signal, and the additive noise $v(t)$ is white and Gaussian. The SNR is defined related to (4.6) as σ_y^2/σ_v^2, where $\sigma_y^2 = E\left[y^2(t)\right]$ is the variance of $y(t)$. We consider that σ_v^2 is available in simulations.

In the first set of experiments, we provide a toy example in order to outline the relation between the "keywords" of the proposed approach. i.e., decomposition, low-rank, and sparsity. To this purpose, let us consider the impulse responses ($\mathbf{h}$) from

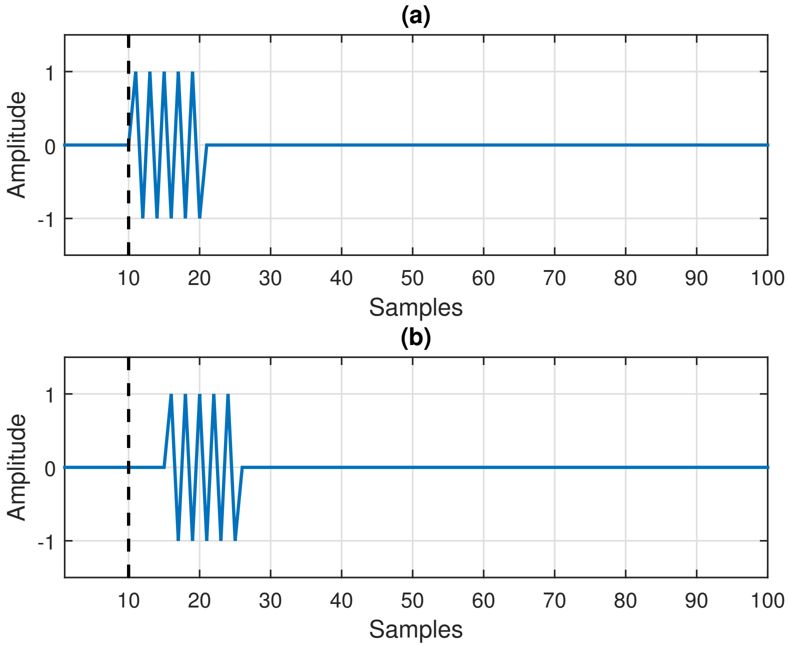


Figure 4.2: Impulse responses of length $L = 100$, which are decomposed using $L_1 = L_2 = 10$: (a) a cluster of 10 samples (alternating the amplitudes 1 and -1) padded with zero, with $\text{rank}\,(\mathbf{H}) = 1$; and (b) the same cluster shifted to the right by 5 samples, so that $\text{rank}\,(\mathbf{H}) = 2$.

Fig. 4.2, of length $L = 100$. The first impulse response [Fig. 4.2(a)] is a cluster of 10 samples (alternating the amplitudes 1 and -1) padded with zero; in Fig. 4.2(b), this cluster is shifted to the right by 5 samples. Therefore, the sparsity of these impulse responses is the same. However, their decomposition could be different, in terms of the rank of the corresponding matrix $\mathbf{H}$. For example, using $L_1 = L_2 = 10$, we obtain $\text{rank}\,(\mathbf{H}) = 1$ for the impulse response from Fig. 4.2(a), so that we can use $P = 1$ in (4.5). On the other hand, $\text{rank}\,(\mathbf{H}) = 2$ for the impulse response from Fig. 4.2(b), which implies that $P = 2$ is suitable in (4.5); in this case, for $P = 1$, the approximation error related to (4.5) (evaluated in terms of the NM, in dB) results as $20\log_{10}\left[\|\mathbf{e}(1)\| / \|\mathbf{h}\|\right] \approx -3$ dB.

In order to support the previous discussion, in Fig. 4.3, the performance of the KF-NKP is illustrated for the identification of the impulse responses from Fig. 4.2, using $P = 1$ and 2. The performance measure is the NM (in dB), which is defined as

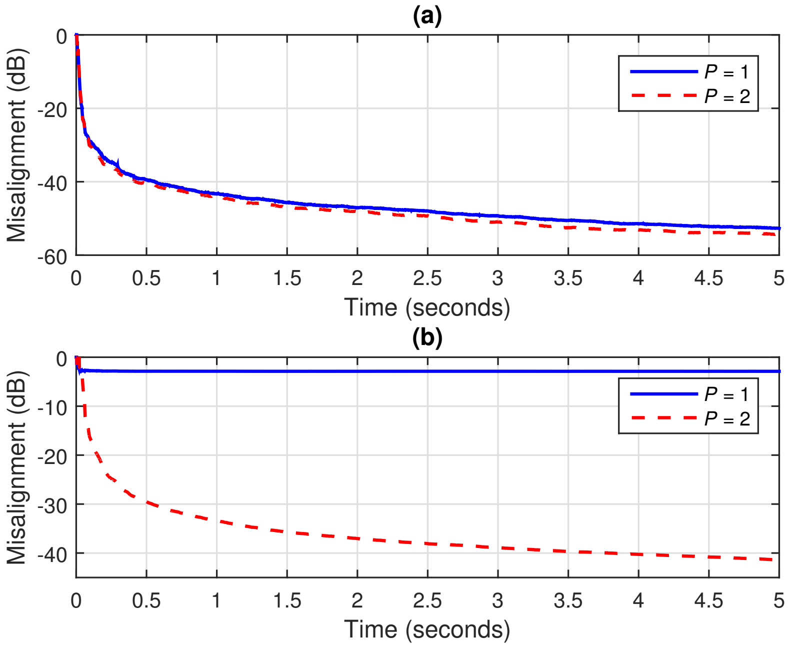


Figure 4.3: Normalized misalignment of the KF-NKP using $\sigma^2_{w_1} = \sigma^2_{w_2} = 0$, $L_1 = L_2 = 10$, and $P = 1$ or 2, corresponding to the impulse responses from Figs. 4.2(a) and (b). The input signal is an AR(1) process and SNR = 30 dB.

$20\log_{10}\left[\left\|\mathbf{h} - \widehat{\mathbf{h}}(t)\right\| / \|\mathbf{h}\|\right]$. The input signal is an AR(1) process and SNR = 30 dB. As we can notice in Fig. 4.3(a), the impulse response from Fig. 4.2(a) can be well identified using $P = 1$, while $P = 2$ leads to similar results. On the other hand, as shown in Fig. 4.3(b), $P = 2$ is suitable for identification of the impulse response from Fig. 4.2(a), while the misalignment level achieved for $P = 1$ is close to the approximation error computed before. Concluding, the sparsity of the system helps (by reducing the rank of the matrix $\mathbf{H}$), but it is not the only factor; the proposed approach is also related to the decomposition of the impulse response.

In the second set of experiments, we assess the performance of the proposed KF-NKP in the context of echo cancellation, which is one of the most challenging system identification scenarios. Three impulse responses are used, as illustrated in Fig. 4.4. The first two impulse responses provided in Figs. 4.4(a) and (b) are network echo paths from G168 Recommendation [55], with $L = 500$, while the impulse response

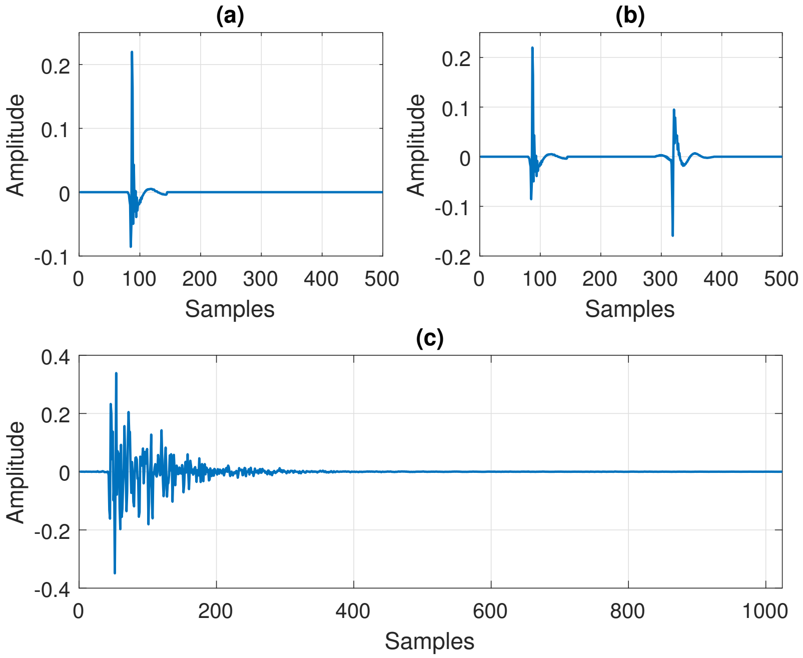


Figure 4.4: Impulse responses used in simulations: (a) the first impulse response from G168 Recommendation [55], with $L = 500$; (b) the first and the fifth impulse responses (concatenated) from G168 Recommendation [55], with $L = 500$; and (c) acoustic impulse response, with $L = 1024$.

from Fig. 4.4(c) represents an acoustic echo path of length $L = 1024$. In order to evaluate the decomposition features related to these impulse responses, we can also use the approximation error related to (4.5), in terms of the NM (in dB), i.e., $20\log_{10} [\|\mathbf{e}(P)\| / \|\mathbf{h}\|]$. The results are presented in Fig. 4.5, where the representation is limited to -100 dB (for better visualization). As we can notice in Figs. 4.5(a) and (b), where $L_1 = 25$ and $L_2 = 20$, the network impulse responses from Figs. 4.4(a) and (b) can be very well approximated for $P \ll L_2$. On the other hand, as supported in Fig. 4.5(c) (where $L_1 = L_2 = 32$), a larger value of P could be required for the identification of the acoustic impulse response from Fig. 4.4(c), since its corresponding matrix $\mathbf{H}$ is close to full rank. These aspects will be illustrated in the following simulations.

First, in Fig. 4.6, the performance of the KF-NKP (using $L_1 = 25$, $L_2 = 20$, and different values of P) is compared to the regular KF, when identifying the impulse

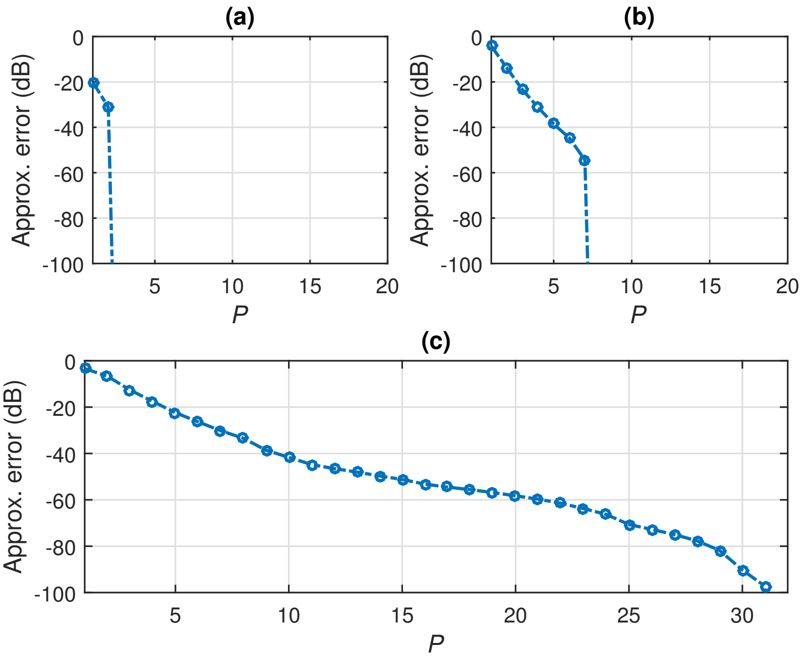


Figure 4.5: Approximation error (in terms of the NM) related to (4.5), for the identification of the impulse responses from Fig. 4.4: (a) impulse response from Fig. 4.4(a), of length $L = 500$, with $L_1 = 25$ and $L_2 = 20$; (b) impulse response from Fig. 4.4(b), of length $L = 500$, with $L_1 = 25$ and $L_2 = 20$; and (c) impulse response from Fig. 4.4(c), of length $L = 1024$, with $L_1 = L_2 = 32$.

responses from Figs. 4.4(a) and (b); the performance measure is the NM (in dB). The impulse response changes in the middle of the simulation [from Fig. 4.4(a) to Fig. 4.4(b)], the input signal is an AR(1) process, and SNR = 20 dB. The specific parameters of the KF-NKP are set to $\sigma^2_{w_1} = \sigma^2_{w_2} = 10^{-8}$; the same value is used for the uncertainty parameter of the KF. As we can notice in the first part of Fig. 4.6, the KF-NKP is able to identify the impulse response from Fig. 4.4(a) using $P \ll L_2$; for example, even for $P = 2$, the KF-NKP outperforms the regular KF. When $P \geq 3$, the KF-NKP leads to similar performances [see also Fig. 4.5(a)], since $\text{rank}(\mathbf{H}) = 3$ for the impulse response from Fig. 4.4(a). On the other hand, as we can notice in the second part of Fig. 4.6, a larger value of P is required for the identification of the impulse response from Fig. 4.4(b). As expected [based on Fig. 4.5(b)], the KF-NKP using $P = 2$ cannot provide an accurate result in this case; however, when $P = 3$,

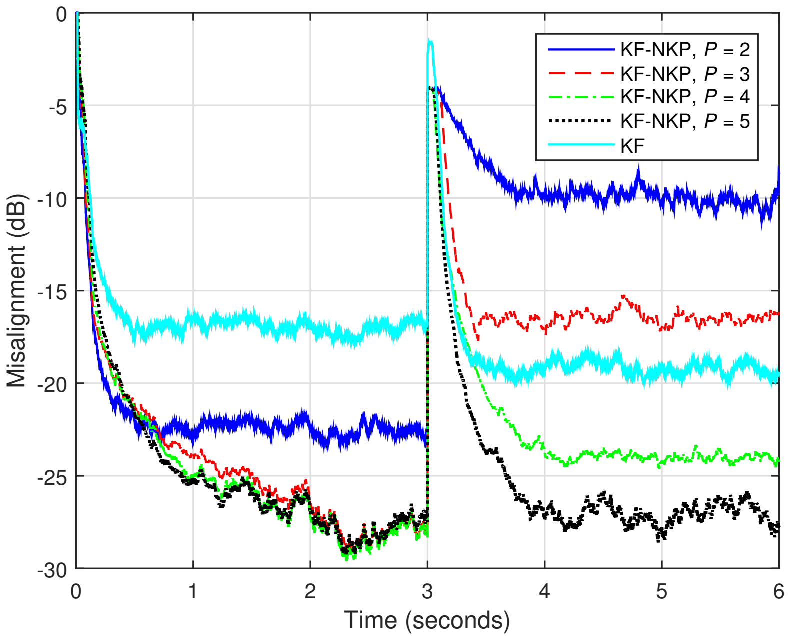


Figure 4.6: Normalized misalignment of the KF-NKP (using different values of P) and KF, for the identification of the impulse responses from Figs. 4.4(a) and (b). The impulse response changes after 3 seconds. The input signal is an AR(1) process, $L = 500$, and SNR = 20 dB. The KF-NKP uses $L_1 = 25$, $L_2 = 20$, and $\sigma_{w_1}^2 = \sigma_{w_2}^2 = 10^{-8}$; the KF uses the same value of its uncertainty parameter.

its performance is similar to the KF, while for $P \geq 4$, the KF-NKP outperforms its regular counterpart.

A similar experiment is reported in Fig. 4.7, for the identification of the acoustic impulse response from Fig. 4.4(c); an abrupt change is introduced in the middle of simulation, by shifting the impulse response to the right by 12 samples. The decomposition is performed using $L_1 = L_2 = 32$. As we can notice, the KF-NKP requires a larger value of P in order to reach the performance of the regular KF; however, this value is reasonably low as compared to L_2.

The influence of the uncertainty parameters ($\sigma_{w_1}^2$ and $\sigma_{w_2}^2$) on the performance of the KF-NKP algorithm is illustrated in Fig. 4.8, when identifying the impulse responses from Figs. 4.4(a) and (b). The echo path changes in the middle of the simulation, the input signal is an AR(1) process, and SNR = 20 dB. As outlined in

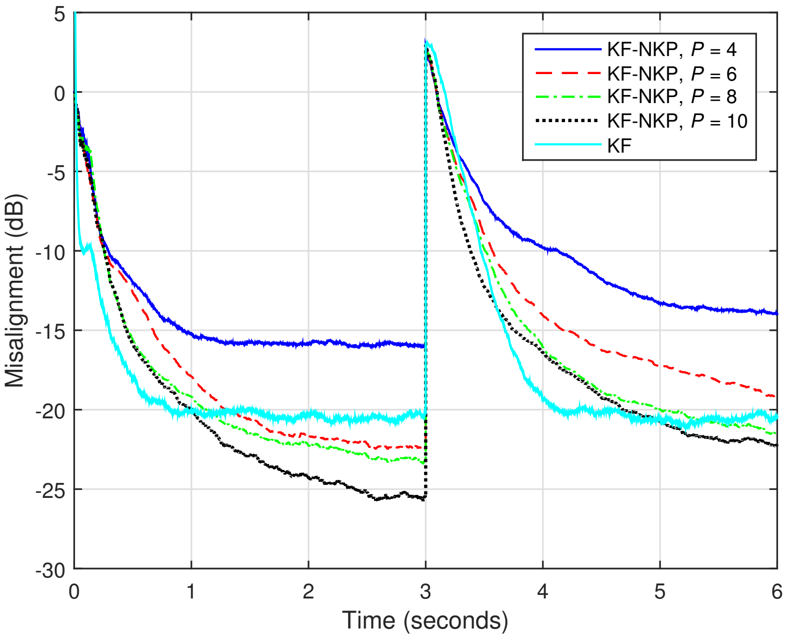


Figure 4.7: Normalized misalignment of the KF-NKP (using different values of P) and KF, for the identification of the impulse response from Fig. 4.4(c). The impulse response changes after 3 seconds. The input signal is an AR(1) process, $L = 1024$, and SNR = 20 dB. The KF-NKP uses $L_1 = L_2 = 32$ and $\sigma^2_{w_1} = \sigma^2_{w_2} = 10^{-8}$; the KF uses the same value of its uncertainty parameter.

Section 4.2.1, the values of $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$ lead to a compromise between fast convegence/tracking and low misalignment, so that a proper estimation of these parameters is required. These aspects are supported in Fig. 4.8, where the KF-NKP uses $P = 5$ and different values of the uncertainty parameters, including the estimated ones [based on (4.36)–(4.37)]. Clearly, lower values of $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$ improve the performance in terms of misalignment, but at the price of a slower tracking reaction. Most important, the estimators from (4.36)–(4.37) lead to a proper compromise between the performance criteria.

Next, in Fig. 4.9, the performances of the KF-NKP and KF are compared, when identifying the impulse response from Fig. 4.4(c) and using the practical estimators of the uncertainty parameters. To this purpose, the KF-NKP uses (4.36)–(4.37), while the regular KF involves the solution proposed in [91], which is similar to the individual

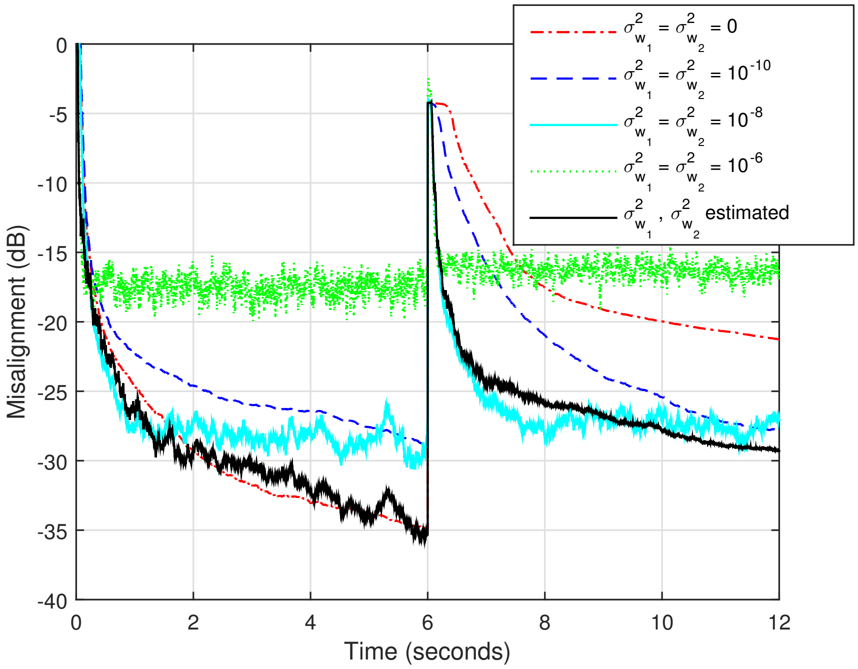


Figure 4.8: Normalized misalignment of the KF-NKP, for the identification of the impulse responses from Figs. 4.4(a) and (b). The impulse response changes after 6 seconds. The input signal is an AR(1) process, $L = 500$, and SNR = 20 dB. The KF-NKP uses $L_1 = 25$, $L_2 = 20$, $P = 5$, and different values of $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$, including the estimation from (4.36)–(4.37).

control approach from Section 4.2.1. Both algorithms use the same threshold for the uncertainty parameters, which is evaluated based on the global filter (as discussed in the end of Section 4.2.1). The KF-NKP algorithm uses $L_1 = L_2 = 32$ and different values of P. In this experiment, the input signal is an AR(1) process and SNR = 20 dB. As we can notice in Fig. 4.9, the KF-NKP achieves a similar performance as compared to the regular KF, even for a reasonably low value of P.

Finally, the performance of the proposed KF-NKP is compared to the recently developed RLS-NKP algorithm [86]. It is known that the KF and the RLS algorithms have striking resemblances [97]. However, the additional parameters specific to the KF allow for an additional control of the algorithm, which could further improve the performance [16]. Consequently, a similar behavior is expected in case of the KF-NKP and the RLS-NKP algorithm. In Figs. 4.10 and 4.11, the performances of

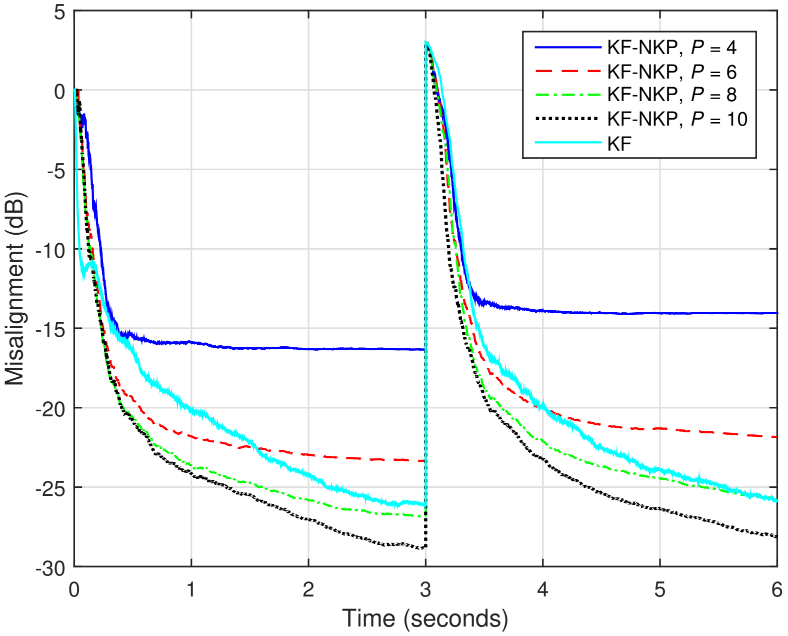


Figure 4.9: Normalized misalignment of the KF-NKP (using different values of P) and KF, for the identification of the impulse response from Fig. 4.4(c). The impulse response changes after 3 seconds. The input signal is an AR(1) process, $L = 1024$, and SNR = 20 dB. The KF-NKP uses $L_1 = L_2 = 32$, while the specific parameters $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$ are estimated based on (4.36)–(4.37); the KF uses the uncertainty parameter estimated as in [91].

these algorithms are compared for the identification of the impulse responses from Figs. 4.4(a) and (c), respectively. In both scenarios, the impulse response changes in the middle of the simulation (by shifting to the right by 12 samples). The input signal is a speech sequence and SNR = 20 dB. In Fig. 4.10, the algorithms use $L_1 = 25$, $L_2 = 20$, and $P = 5$, while in Fig. 4.11 the setup is $L_1 = L_2 = 32$ and $P = 10$. In both cases, the KF-NKP uses the estimated uncertainty parameters (presented in Section 4.2.1). The specific parameters of the RLS-NKP algorithm are the forgetting factors [86], which are set to $\lambda_1 = \lambda_2 = 0.999$ in Fig. 4.10, while in Fig. 4.11 their values are $\lambda_1 = \lambda_2 = 0.9999$. As we can notice in Fig. 4.10, the RLS-NKP algorithm and the KF-NKP achieve a similar convergence rate (and tracking reaction), but the KF-NKP reaches a lower misalignment level. The gain is even more apparent in

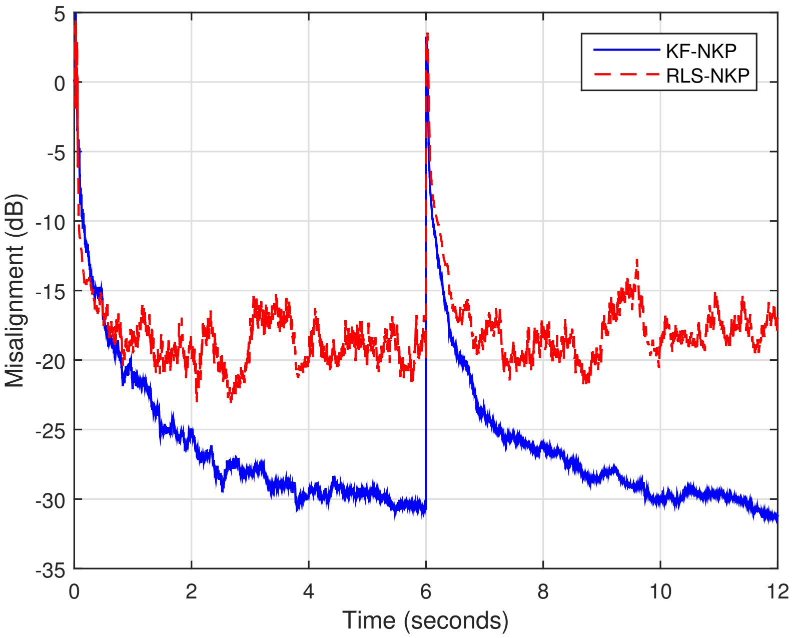


Figure 4.10: Normalized misalignment of the KF-NKP and RLS-NKP algorithm (using $L_1 = 25$, $L_2 = 20$, and $P = 5$), for the identification of the impulse response from Fig. 4.4(a). The impulse response changes after 6 seconds. The input signal is a speech sequence, $L = 500$, and SNR = 20 dB. The KF-NKP uses $\sigma^2_{w_1}$ and $\sigma^2_{w_2}$ estimated based on (4.36)–(4.37); the RLS-NKP algorithm uses the forgetting factors $\lambda_1 = \lambda_2 = 0.999$.

Fig. 4.11, where the KF-NKP outperforms the RLS-NKP algorithm in terms of both performance criteria.

4.3 Summary and Conclusions

In this chapter, we have started by introducing a newly developed approach for the identification of low-rank systems, based on the NKP decomposition and low-rank approximation. We continued by deriving a KF tailored for the identification of low-rank systems. The proposed KF-NKP is based on the NKP decomposition of the impulse response, together with low-rank approximation. Consequently, a system identification problem of size $L = L_1L_2$ could be reformulated using two smaller filters of lengths PL_1 and PL_2, with $P \ll L_2$. Hence, it is efficient for the identification

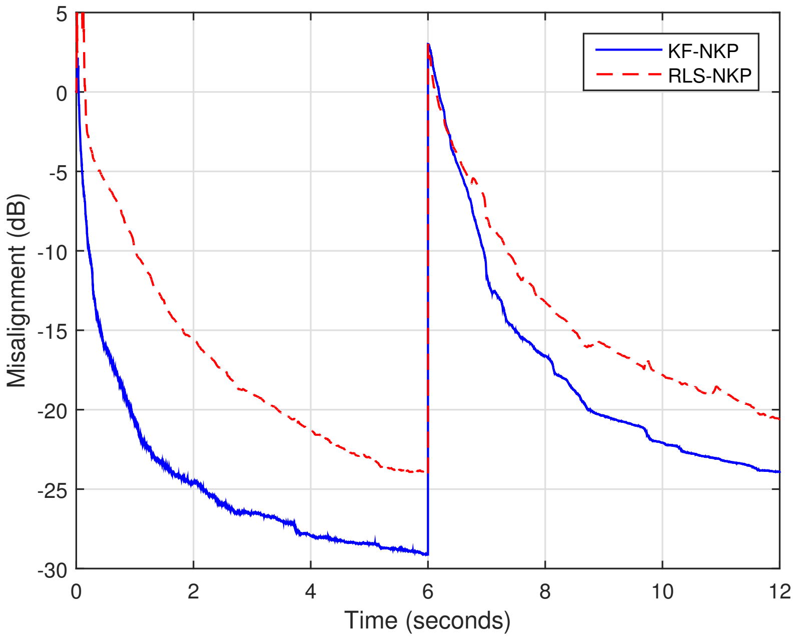


Figure 4.11: Normalized misalignment of the KF-NKP and RLS-NKP algorithm (using $L_1 = L_2 = 32$ and $P = 10$), for the identification of the impulse response from Fig. 4.4(c). The impulse response changes after 6 seconds. The input signal is a speech sequence, $L = 1024$, and SNR = 20 dB. The KF-NKP uses $\sigma_{w_1}^2$ and $\sigma_{w_2}^2$ estimated based on (4.36)–(4.37); the RLS-NKP algorithm uses the forgetting factors $\lambda_1 = \lambda_2 = 0.9999$.

of long length (low-rank) impulse responses, like the echo paths. In this case, the proposed KF-NKP achieves a better performance, but also a lower computational complexity as compared to the regular KF. Moreover, due to an additional control of its specific parameters, the KF-NKP is also able to outperform its RLS-based counterpart that was recently developed in [86]. Simulations performed in the framework of echo cancellation support these advantages.

Chapter 5

An Adaptive Solution for Nonlinear System Identification

5.1 Motivation

This short chapter presents a new and very simple method to perform identification of nonlinearities in systems that are supposed to be linear. The aim of this study is somewhat different from the rest of the work presented before in this thesis. The reason is that the focus is not on the study of multilinear systems, but rather on solving the problem of small nonlinearities in supposedly linear systems. In this context, the proposed solution offers an improvement with respect to the previous approaches, which lies in the low computational complexity of the solution and the convenient possibility of online evaluation. Simulation results show that the proposed method provides good performance in terms of coefficients identification, resulting in both low steady-state error and high convergence rate of the algorithm, which justifies its applicability in practice. The results shown in this chapter were published in [98].

There are two different ways of identifying nonlinearities in a system: the Volterra-based approach and the method based on neural networks. The first method is a well-known solution that has been proposed in the 1960s and has been continuously developed and improved since then [99, 100, 101]. The approximation of the nonlinear systems is performed using a finite sum of the Volterra series expansion that relates the system's inputs and outputs. This method has been used in different applications, for example, [102, 103, 104], among others. Although the Volterra filter [105] is a popular choice, one of its main drawbacks is its high computational complexity. On the other hand, neural networks represent a newer topic in the research community, which has been analyzed during the past decades and has proven to give excellent results in different applications, such as classification problems or nonlinear adaptive filters design [33, 106]. Despite its higher computational efficiency, there is still a considerable amount of complexity in performing nonlinearities identification using neural networks.

The method that we propose distinguishes itself by its simplicity. Although not very general, it is applicable in situations where supposedly linear systems present small nonlinearities. It allows online evaluation of the nonlinearities of sensors, amplifiers, transducers, without having to use a special measurement session. It is based on the idea that in case of small nonlinearities, the Taylor series expansion of a function around zero has a finite number of significant coefficients. The problem of identifying the system translates into finding the unknown coefficients by means of an adaptive algorithm. In this context, the purpose becomes to determine the coefficients of the Taylor series expansion and, consequently, to determine the nonlinearities of the system. A similar idea was presented in [39] and [17], but in a different context (echo cancellation).

In the following subchapter, the nonlinearities identification system model is introduced and the solution obtained according to the optimal filtering theory is presented. Then, the adaptive approach in solving the problem is described in 5.3. Subchapter 5.4 illustrates the experimental results, showing the effectiveness of the proposed solution, while the last section contains a brief summary and discussion to conclude the subject.

5.2 The Nonlinearities Identification Problem

Let x be a zero-mean real-valued random input signal and let $g(x)$ be a nonlinear, bijective, odd-type function, which is applied to the input signal. As examples, we can consider the exponential limiter function [i.e., $g(x) = (1 - \exp(-\alpha|x|))\text{sign}(x), 0 < \alpha < 1$], the arctangent function [i.e., $g(x) = \arctan(\alpha x), 0 < \alpha < 1$], or any other function that satisfies the conditions. We define the vector containing the first M powers of the input as

$$\mathbf{x}(n) = [x(n), x^2(n), ..., x^M(n)]^T, \tag{5.1}$$

where T denotes transposition.

Assuming the function can be expanded in a Taylor series around the origin, it means that by truncating the series we can write it in the following form:

$$g(x) \cong \sum_{k=1}^{M}(g_k x^k). \tag{5.2}$$

The function is characterized by the vector of coefficients:

$$\mathbf{g} = [g_1, g_2, ..., g_M]^T. \tag{5.3}$$

Our goal is to obtain an estimation of the coefficient vector:

$$\hat{\mathbf{g}}(n) = [\hat{g}_1(n), \hat{g}_2(n), ..., \hat{g}_M(n)]^T, \tag{5.4}$$

based on the knowledge of the input signal $\mathbf{x}(n)$ and by measuring the unknown

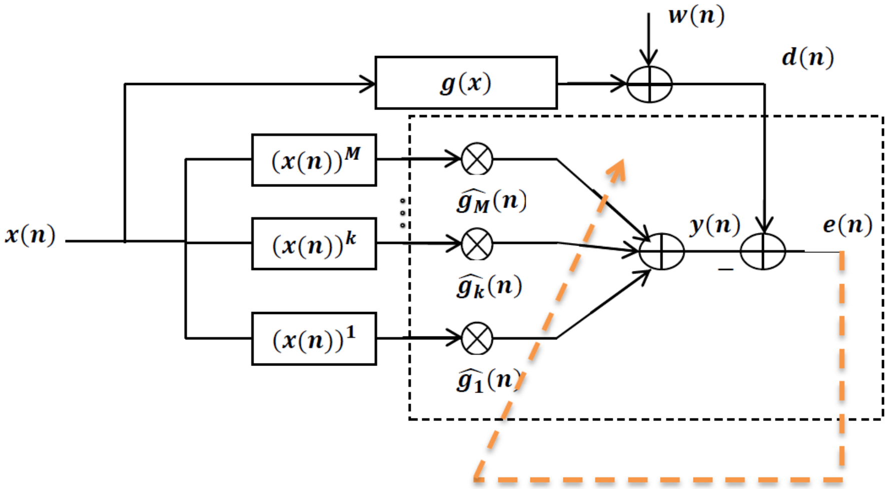


Figure 5.1: System model.

system output corrupted by the measurement noise:

$$d(n) = \mathbf{g}^T \mathbf{x}(n) + w(n). \tag{5.5}$$

The number M of coefficients that need to be found depends on the particular function; the greater M, the more precise is the Taylor expansion. A representation of this situation can be found in Fig. 5.1, where $x(n)$ is the input signal, the sum of products $\sum_{k=1}^{M}(g_k x^k)$ is denoted by $y(n)$, and the error $e(n) = d(n) - y(n)$ is used in order to perform the adaptation.

By denoting the output error:

$$e(n) = d(n) - \hat{\mathbf{g}}^T \mathbf{x}(n) = (\mathbf{g}^T - \hat{\mathbf{g}}^T)\mathbf{x}(n) + w(n) \tag{5.6}$$

and assuming the measurement noise has zero mean and is uncorrelated to the input signal, our initial problem translates into the minimization of the MSE, $J(n) = E[e^2(n)]$, with respect to the vector $\hat{\mathbf{g}}$. It can be noticed that the MSE can be expressed as

$$J(n) = \sigma_d^2 - 2\hat{\mathbf{g}}^T \mathbf{p} + \hat{\mathbf{g}}^T \mathbf{R} \hat{\mathbf{g}}, \tag{5.7}$$

where $\sigma_d^2 = E[d^2(n)]$ is the desired signal variance, $\mathbf{p} = E[\mathbf{x}(n)d(n)]$ is the cross-covariance between the input signal $x(n)$ and the desired signal $d(n)$, and $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ is the covariance matrix of the vector $\mathbf{x}(n)$. The optimal solution is given by the Wiener-Hopf equations [107]:

$$\mathbf{g}_o = \mathbf{R}^{-1}\mathbf{p}. \tag{5.8}$$

This solution can only be used when the unknown coefficients are fixed (usually they may change in time) and the involved statistical expectations are known. In order to overcome these limitations, we propose the use of the adaptive approach instead.

5.3 The Adaptive Approach

Firstly, one must note that, although the Wiener-Hopf equations are related to finding the optimal coefficients of a linear filter, the theory does not imply that the vector $\mathbf{x}(n)$ needs to contain successive delayed versions of $x(n)$. In our case, this vector contains M successive powers of $x(n)$. Consequently, a more attractive solution would be to use an adaptive algorithm, derived from the Wiener-Hopf theory, to estimate the optimal solution. One of the simplest algorithms that may be used is the LMS algorithm, with the update:

$$\hat{\mathbf{g}}(n) = \hat{\mathbf{g}}(n-1) + \mu\mathbf{x}(n)e_a(n), \tag{5.9}$$

where $e_a(n) = d(n) - \hat{\mathbf{g}}^T(n-1)\mathbf{x}(n)$ is the a priori error, which is computed using the coefficients at time index $n-1$ (i.e., from the previous step) and the new updated data. The step-size μ has to be chosen according to $0 < \mu < \frac{2}{M\sigma_x^2}$, in order to ensure the algorithm convergence [1, 3, 108], where σ_x^2 is the mean power of the input signal. Alternatively, the NLMS algorithm can be used, which is equivalent to an LMS algorithm with a variable step-size given by

$$\mu(n) = \frac{\alpha}{||\mathbf{x}(n)||^2 + \delta}, \tag{5.10}$$

where α is the normalized step-size ($0 < \alpha < 2$), which should be chosen in such a way to ensure a good compromise between the convergence rate and the asymptotic MSE $J(\infty)$, and δ is the regularization parameter [46].

The particular structure of the input vector $\mathbf{x}(n)$ leads to the necessity of an analysis of its covariance matrix:

$$\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\} = [r_{i,j}], i, j = 1, ..., M, \tag{5.11}$$

$$r_{i,j} = E\{x^{i+j}(n)\}. \tag{5.12}$$

Some remarks result from here regarding $\mathbf{R}$, which are stated in the following.

- $\mathbf{R}$ is symmetric and positive semidefinite, but it is not Toeplitz (the elements of the main diagonal are not equal).

- It can be noticed that the value of each element of $\mathbf{R}$ is a function of the sum of the indices i and j. This means that $\mathbf{R}$ is a Hankel matrix.

- For an input signal with zero mean and an even probability density function, $r_{i,j} = 0$ for $i + j$ odd. This means that a number of matrix diagonals, parallel to the main diagonal, have all zero elements.

Two issues have to be considered. First of all, the matrix must be non-singular [see (5.8)]. Second, some information about the condition number is of interest, because it is well-known that in the case of an ill-conditioned covariance matrix (very large condition number), the convergence rate of the adaptive algorithm is very slow [1, 3, 108].

Some examples of covariance matrix elements computation for well-known signals are as follows.

- White and Gaussian, with zero mean and variance σ_x^2. In this case it follows that

$$E\{x^{i+j}(n)\} = \begin{cases} 0, & \text{for} \quad i+j \quad \text{odd} \\ 3 \cdot 5 \cdot \ldots (i+j-1)\sigma_x^{i+j}, & \text{for} \quad i+j \quad \text{even} \end{cases} \tag{5.13}$$

- Uniformly distributed between $-L$ and L. Then,

$$E\{x^{i+j}(n)\} = \begin{cases} 0, & \text{for} \quad i+j \quad \text{odd} \\ \frac{(\sqrt{3}\sigma_x)^{i+j}}{i+j+1}, & \text{for} \quad i+j \quad \text{even} \end{cases} \tag{5.14}$$

- White and Gaussian, with zero mean and variance σ_x^2, symmetrically clipped at level L. The condition number in this case was computed using Matlab and plotted in Fig. 5.2 as a function of the variance, along with the condition number for the previous two cases.

From Fig. 5.2 it follows that if we want the covariance matrix not to be ill-conditioned, then ideally, regardless of the type of input signal that is chosen, its variance needs to be somewhere in the interval [0.3; 0.8]; moreover, if the variance increases above approximately 1 or decreases below approximately 0.3, the condition number of the matrix **R** also increases a lot, the matrix becoming ill-conditioned. This leads to the idea that the variance will be chosen between reasonable limits. In practice, this is feasible, considering that real systems do not use inputs of very large (or too small) power. Hence, we can conclude that any of the signals presented can be used, provided that its power is between acceptable limit values.

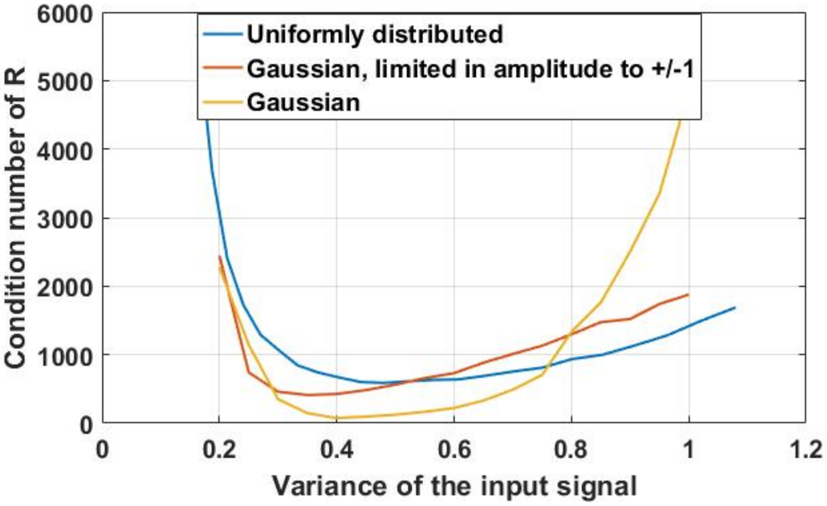


Figure 5.2: Condition number of $\mathbf{R}$ as a function of the signal's variance.

5.4 Experimental Results

Simulations have been performed using as input of the filter the first M powers of a zero mean Gaussian signal limited in amplitude to ±1, where M is the length of the NLMS filter (in our case, 6) and as function whose coefficients need to be identified, a polynomial function with nonzero odd power coefficients, respectively null valued coefficients corresponding to the even powers. The value of the normalized step-size α in (5.10) is 0.5. Fig. 5.3 illustrates the evolution of the coefficients for a polynomial of the form: $g(x) = x + 0.3x^3 + 0.2x^5$.

The black dotted lines represent the actual coefficients; we can see that all the computed coefficients converge to the optimal values after less than 1000 iterations. In Fig. 5.4, the values of the average squared error for 1000 simulations are plotted in dB. Fig. 5.5 shows the graphical representation of the polynomial function in two cases: when using the actual coefficients (which represent the optimal coefficients $\mathbf{g}_o$), on one hand, and when using the coefficients computed using our method, on the other hand.

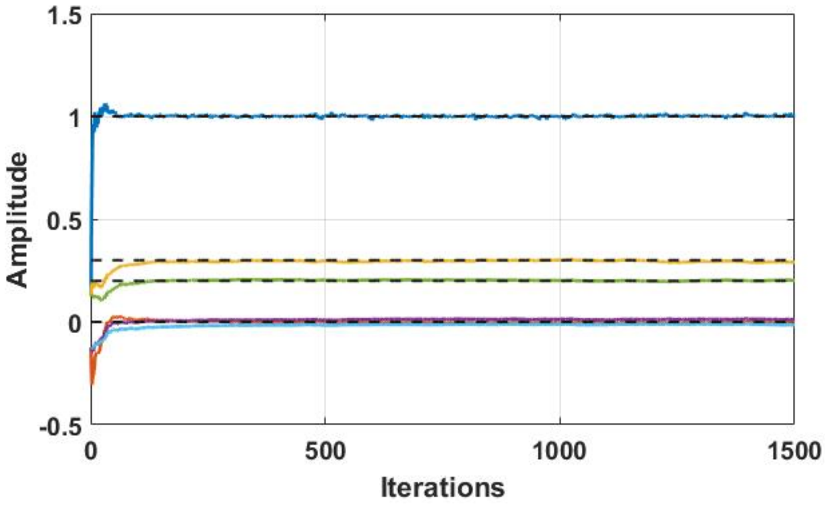


Figure 5.3: Evolution of the coefficients g_k computed using the NLMS algorithm for a polynomial function.

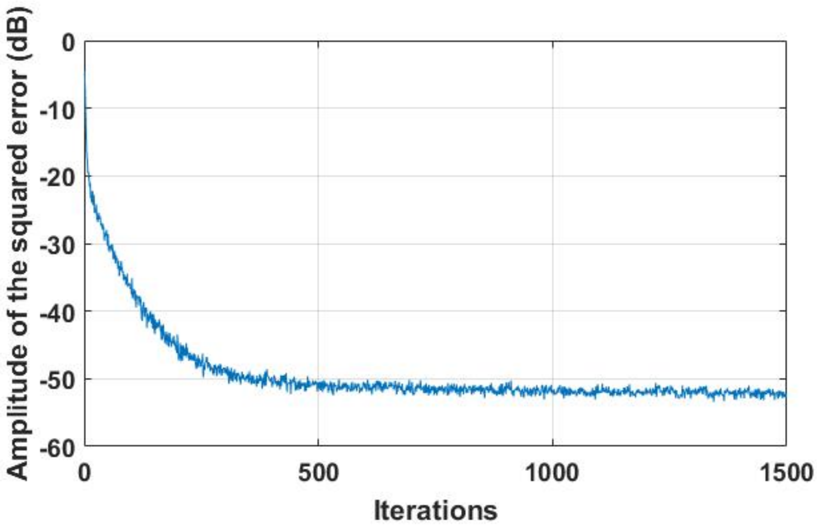


Figure 5.4: Squared error values (in dB).

We can see from Fig. 5.3 that the convergence of the algorithm is fast, but at the same time, the calculated coefficients are very close to their actual values which were known apriori. This can be observed from Fig. 5.4, which shows that the value of the squared error is smaller than -20 dB even from the first iterations of the algorithm, as well as from Fig. 5.5, which illustrates an almost perfect overlap between the graph of the function g when using the two sets of coefficients, i.e., the ones that were initially generated and the ones computed using the proposed approach. The results obtained

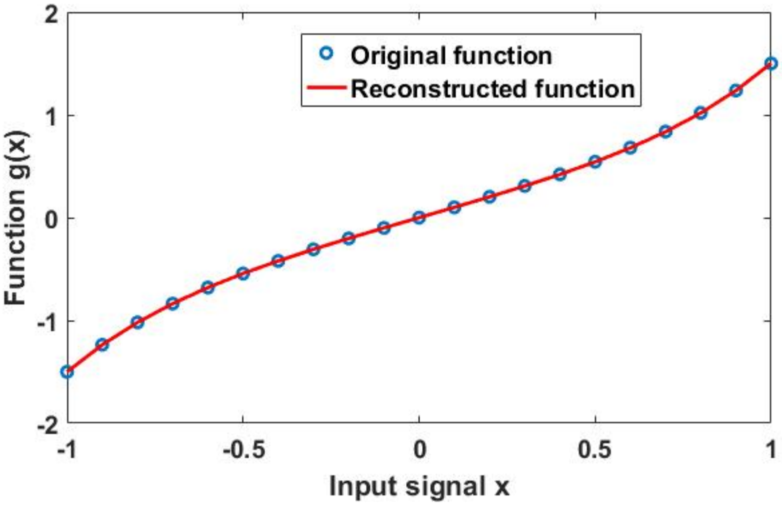


Figure 5.5: Representation of the polynomial function and the reconstructed function when the input $x \in [-1; 1]$.

through simulations are very promising, proving that the method is suitable to be used in this kind of problems, where the coefficients of a nonlinear system need to be identified.

The behavior of the algorithm when applied to a system that changes in time was also studied. Simulations were conducted using the same polynomial function described before for the first half of the experiment, then the coefficients were changed: the term corresponding to the third power, whose initial value was 0.3, became 0.4, whereas the fifth power coefficient switched its value from 0.2 to 0.1. Fig. 5.6 shows the evolution of the coefficients. The black and red dotted lines represent the actual values of the coefficients (black - for the first half of the time, red - the values after the change in the system occured). It can be seen that the algorithm was able to track the changes in the system and the coefficients converged to their new values in less than 3000 iterations after the change.

Simulations were also repeated for other nonlinear functions, in order to prove that the results above are still valid. In Fig. 5.7, the original arctangent function is represented along with the function obtained using relation 5.2 from the first six

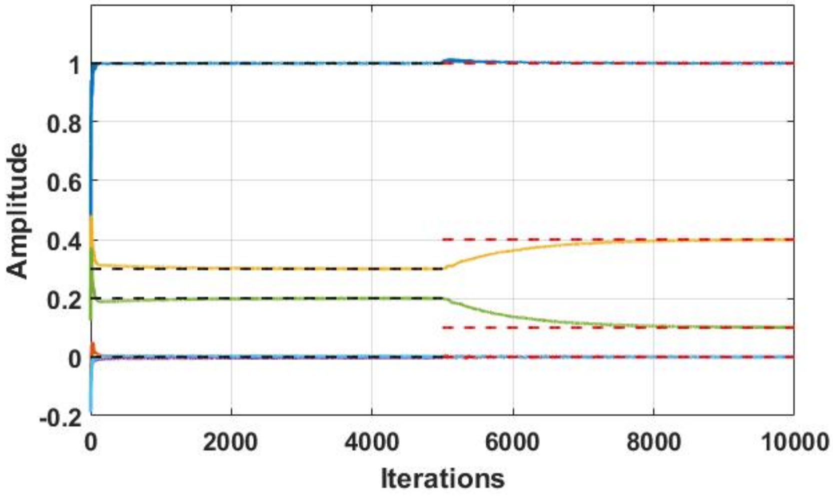


Figure 5.6: Evolution of the coefficients g_k when a change in their values occurs.

significant coefficients in the Taylor series expansion, computed using the same NLMS algorithm. Because in reality, the number of coefficients in the Taylor series expansion is infinite, we would expect to have a noticeable difference between the plot of the actual function $g(x) = \arctan(x)$ for $x \in [-1, 1]$ and the plot of the function $g(x) = \sum_{k=1}^{M}(g_k x^k)$, because of the approximation in the equality between the two. However, Fig. 5.7 shows that the plots of the two functions overlap almost perfectly, which proves that the arctangent function can be very accurately expanded by only taking into consideration the first six terms of the Taylor series, on one hand, and that our algorithm is very efficient in computing these coefficients, on the other hand.

5.5 Summary and Conclusions

In this chapter, we proposed a new method for performing a simple and fast identification of the nonlinearities in a system, by using the well-known NLMS algorithm and by taking advantage of the fact that the Taylor series expansion of a function has only a finite number of significant coefficients.

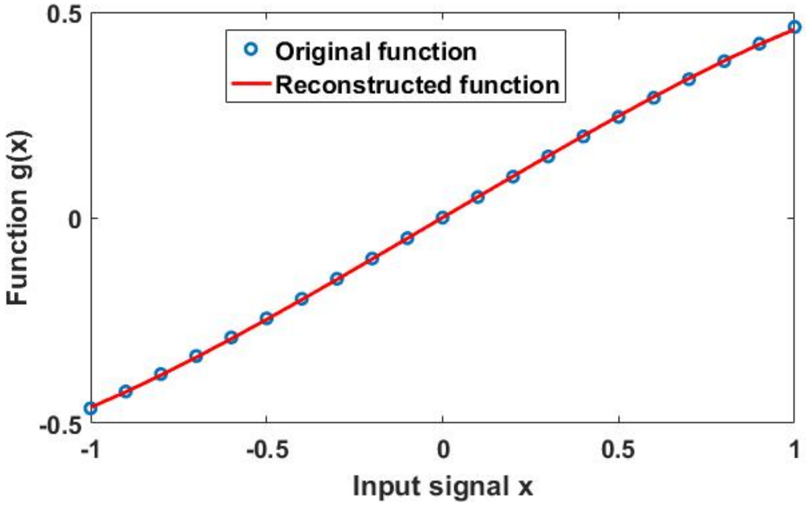


Figure 5.7: Representation of the arctangent function and the reconstructed function when the input $x \in [-1; 1]$.

The great advantages of the proposed solution are its simplicity, as well as the fact that it can be performed online, without the need to allocate time and a special setup for the measurements. Simulations have been conducted on different types of functions and the results have shown that the method gives very good results, with small error values and high convergence speed. Despite the method not being generally applicable, it may help decrease both the complexity and the computation time of the nonlinear system identification process, in situations where its usage is suitable.

A further simplification of the presented method would rely on the fact that the functions which were considered are odd functions, which means that the coefficients of their Taylor series expansion corresponding to the even powers will be null. If we take that into account when developing the LMS-based algorithm, we can compute only the odd terms, in this way significantly decreasing the length of the adaptive filter.

Future work might also include research on the impact that a very small coefficient has on the efficiency of the identification: in the case when the system has coefficients with very different values among them (even a few orders of magnitude), the computation of the smallest coefficients will probably not have a high enough sensitivity.

Chapter 6

Summary of Contributions and Future Work

This thesis aimed to present our contributions during the past three years to the field of nonlinear system identification, with a special focus on multilinear systems.

Because the conclusions and remarks were inserted at the end of each chapter, there is no need to repeat them here. However, in the following we list the major original contributions of this work, divided by chapters. Since all contributionsare either published or accepted for publication, we provide for each one the corresponding reference.

In Chapter 2:

– Development of an optimized LMS algorithm for the identification of bilinear forms (OLMS-BF) [20];

– A mathematical analysis of the properties of a system mismatch covariance matrix (SMCM) in the LMS algorithm [22];

– Derivation of a Kalman filter tailored for bilinear forms (KF-BF), together with a computationally simplified version (SKF-BF) [19];

– A comparative study of the SKF-BF and OLMS-BF algorithm, highlighting the similarities between the two [21];

– Development of an improved proportionate affine projection algorithm for the special case of BF (IPAPA-BF) [23];

– Illustration of the experimental results obtained for each proposed solution, which show the algorithms' performance [20, 22, 19, 21, 23].

In Chapter 3:

– A Wiener filter conceived for the identification of trilinear forms, comprising a direct and an iterative version [67];

– An LMS algorithm suitable for TF (LMS-TF), along with a normalized version (NLMS-TF) [66];

– Simulations to prove the good behavior of the proposed approaches [67, 66].

In Chapter 4:

– A Kalman filter based on the nearest Kronecker product decomposition combined with the low-rank approximation (KF-NKP) [87];

– A study of how to estimate the KF-NKP parameters, such as to ensure good performance [87];

– Experimental study of the system identification performance [87].

In Chapter 5:

– A simple technique for the identification of nonlinear systems possessing small nonlinearities [98];

– Particular case of a function on which the method is applied and illustration of the results [98].

Future work will focus on extending the techniques displayed in this thesis to higher-order multilinear structures. Given the numerous applications of nonlinear and, in particular, multilinear system identification in the development of technologies nowadays, it would be of interest to obtain a general method for dealing with such problems. In this context, it would be highly interesting to assess the NKP decomposition approach for the identification of higher-order complex impulse responses, such as real-world network and acoustic echo paths which are not perfectly decomposable in a tensorial form. This requires the use of higher-order tensors for the mathematical formalization and analysis of the problem, which is not straight-forward in the context of real-world systems.

Publications

International journal papers:

[21] L.-M. Dogariu, S. Ciochină, C. Paleologu, and J. Benesty, “A connection between the Kalman filter and an optimized LMS algorithm for bilinear forms,” *Algorithms*, vol. 11(12), p. 211, Dec. 2018 **(ISI Indexed, ISSN: 1999-4893)**

[66] L.-M. Dogariu, S. Ciochină, J. Benesty, and C. Paleologu, “System identification based on tensor decompositions: a trilinear approach,” *Symmetry*, vol. 11, p. 556, 2019 **(Q2, IF: 2.143, ISSN: 2073-8994)**

[87] L.-M. Dogariu, C. Paleologu, J. Benesty, and S. Ciochină, “An efficient Kalman filter for the identification of low-rank systems,” *Signal Processing*, vol. 166, p. 107239, January 2020 **(Q1, IF: 4.086, ISSN: 0165-1684)**

International conference papers [ISI Proceedings]:

[19] L. Dogariu, C. Paleologu, S. Ciochină, J. Benesty, and P. Piantanida, “Identification of bilinear forms with the Kalman filter,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Calgary, AB, Canada), pp. 4134–4138, 15–20 April 2018

[20] L.-M. Dogariu, S. Ciochină, C. Paleologu, J. Benesty, and P. Piantanida, “An Optimized LMS algorithm for bilinear forms,” in *Proc. IEEE International Symposium on Electronics and Telecommunications (ISETC)*, (Timisoara, Romania),

pp. 1–4, Nov. 2018

[22] L.-M. Dogariu, S. Ciochină, and C. Paleologu, “On the Properties of the System Mismatch Covariance Matrix in the LMS Adaptive Algorithm,” in *Proc. IEEE International Conference on Communications (COMM)*, (Bucharest, Romania), pp. 39–44, 2018

[23] L.-M. Dogariu, C. Elisei-Iliescu, C. Paleologu, J. Benesty, and S. Ciochină, “A Proportionate Affine Projection Algorithm for the Identification of Sparse Bilinear Forms,” in *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, (Iasi, Romania), 11–12 July 2019

[67] L.-M. Dogariu, S. Ciochină, J. Benesty, and C. Paleologu, “An Iterative Wiener Filter for the Identification of Trilinear Forms,” in *Proc. IEEE International Conference on Telecommunications and Signal Processing (TSP)*, (Budapest, Hungary), 1–3 July 2019 **(Best Paper Award)**

[98] L.-M. Dogariu, S. Ciochină, C. Paleologu, J. Benesty, and P. Piantanida, “An adaptive solution for nonlinear system identification,” in *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, (Iasi, Romania), pp. 1–4, July 2017

Bibliography

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

[2] J. Benesty and Y. H. (editors), *Adaptive Signal Processing – Applications to Real-World Problems.* Springer-Verlag, Berlin, Germany, 2003.

[3] A. H. Sayed, *Adaptive Filters.* New York, NY: Wiley, 2008.

[4] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications.* MIT Press, 1949.

[5] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *Proc. IRE WESCON Conv. Rec.*, pp. 96–104, 1960.

[6] J. S. Koford and G. F. Groner, "The use of an adaptive threshold element to design a linear optimal pattern classifier," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 42–50, Jan. 1966.

[7] B. Widrow, "Adaptive filters," *Aspects of Network and System Theory, New York:Holt, Rinehart, and Winston*, pp. 563–587, 1971.

[8] M. H. Hayes, *Statistical Digital Signal Processing and Modeling.* Wiley, 1996.

[9] F. Albu and K. Nishikawa, "Nonlinear Adaptive Filtering with a Family of Kernel Affine Projection Algorithms," *Handbook of Research on Advanced Intelligent Control Engineering and Automation*, pp. 66–89, 2015.

[10] K. Ozeki, *Theory of Affine Projection Algorithms for Adaptive Filtering.* Springer, 2016.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.

[12] C. Paleologu, J. Benesty, S. L. Grant, and C. Osterwise, "Variable step-size NLMS algorithms designed for echo cancellation," in *Proc. IEEE Asilomar*, pp. 633–637, 2009.

[13] C. Paleologu, J. Benesty, and S. Ciochină, "A variable step-size affine projection algorithm designed for acoustic echo cancellation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16 (8), pp. 1466–1478, 2008.

[14] C. Paleologu, S. Ciochină, and J. Benesty, "An efficient proportionate affine projection algorithm for echo cancellation," *IEEE Signal Processing Letters*, vol. 17 (2), pp. 165–168, 2009.

[15] C. Paleologu, J. Benesty, and S. Ciochină, *Sparse Adaptive Filters for Echo Cancellation.* Morgan & Claypool Publishers, 2010.

[16] C. Paleologu, J. Benesty, and S. Ciochină, "Study of the general Kalman filter for echo cancellation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, pp. 1539–1549, 2013.

[17] Y. Huang, J. Skoglund, and A. Luebs, "Practically efficient nonlinear acoustic echo cancellers using cascaded block RLS and FLMS adaptive filters," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (New Orleans, LA, USA), pp. 596–600, 5–9 March 2017.

[18] E.-W. Bai and D. Li, "Convergence of the iterative Hammerstein system identification algorithm," *IEEE Trans. Automatic Control*, vol. 49, pp. 1929–1940, Nov. 2004.

[19] L. Dogariu, C. Paleologu, S. Ciochină, J. Benesty, and P. Piantanida, "Identification of bilinear forms with the Kalman filter," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Calgary, AB, Canada), pp. 4134–4138, 15–20 April 2018.

[20] L.-M. Dogariu, S. Ciochină, C. Paleologu, J. Benesty, and P. Piantanida, "An Optimized LMS algorithm for bilinear forms," in *Proc. IEEE International Symposium on Electronics and Telecommunications (ISETC)*, (Timisoara, Romania), pp. 1–4, Nov. 2018.

[21] L.-M. Dogariu, S. Ciochină, C. Paleologu, and J. Benesty, "A connection between the Kalman filter and an optimized LMS algorithm for bilinear forms," *Algorithms*, vol. 11(12), p. 211, Dec. 2018.

[22] L.-M. Dogariu, S. Ciochină, and C. Paleologu, "On the Properties of the System Mismatch Covariance Matrix in the LMS Adaptive Algorithm," in *Proc. IEEE International Conference on Communications (COMM)*, (Bucharest, Romania), pp. 39–44, 2018.

[23] L.-M. Dogariu, C. Elisei-Iliescu, C. Paleologu, J. Benesty, and S. Ciochină, "A Proportionate Affine Projection Algorithm for the Identification of Sparse Bilinear Forms," in *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, (Iasi, Romania), 11–12 July 2019.

[24] R. R. Mohler and W. J. Kolodziej, "An overview of bilinear system theory and applications," *IEEE Trans. Systems, Man, Cybernetics*, vol. SMC-10, pp. 683–688, Oct. 1980.

[25] H. K. Baik and V. J. Mathews, "Adaptive lattice bilinear filters," *IEEE Trans. Signal Process.*, vol. 41, pp. 2033–2046, June 1993.

[26] S. Han, J. Kim, and K. Sung, "Extended generalized total least squares method for the identification of bilinear systems," *IEEE Trans. Signal Process.*, vol. 44, pp. 1015–1018, Apr. 1996.

[27] V. Tsoulkas, P. Koukoulas, and N. Kalouptsidis, "Identification of input-output bilinear systems using cumulants," *IEEE Trans. Signal Process.*, vol. 51, pp. 2753–2761, Nov. 2001.

[28] P. Lopes dos Santos, J. A. Ramos, and J. L. Martins de Carvalho, "Identification of bilinear systems with white noise inputs: an iterative deterministic-stochastic subspace approach," *IEEE Trans. Control Systems Technology*, vol. 17, pp. 1145–1153, Sept. 2009.

[29] U. Forssén, "Adaptive bilinear digital filters," *IEEE Trans. Circuits Systems-II: Analog and Digital Signal Process.*, vol. 40, pp. 729–735, Nov. 1993.

[30] R. Hu and H. M. Hassan, "Echo cancellation in high speed data transmission systems using adaptive layered bilinear filters," *IEEE Trans. Communications*, vol. 42, pp. 655–663, Feb./Mar./Apr. 1994.

[31] Z. Zhu and H. Leung, "Adaptive identification of nonlinear systems with application to chaotic communications," *IEEE Trans. Circuits Systems-I: Fundamental Theory and Applications*, vol. 47, pp. 1072–1080, July 2000.

[32] S. M. Kuo and H.-T. Wu, "Nonlinear adaptive bilinear filters for active noise control systems," *IEEE Trans. Circuits Systems-I: Regular Papers*, vol. 52, pp. 617–624, Mar. 2005.

[33] H. Zhao, X. Zeng, and Z. He, "Low-complexity nonlinear adaptive filter based on a pipelined bilinear recurrent neural network," *IEEE Trans. Neural Networks*, vol. 22, pp. 1494–1507, Sept. 2011.

[34] J. Benesty, C. Paleologu, and S. Ciochină, "On the identification of bilinear forms with the Wiener filter," *IEEE Signal Process. Lett.*, vol. 24, pp. 653–657, 2017.

[35] C. Paleologu, J. Benesty, and S. Ciochină, "Adaptive filtering for the identification of bilinear forms," *Digital Signal Process.*, vol. 75, pp. 153–167, Apr. 2018.

[36] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, and S. Ciochină, "Efficient recursive least-squares algorithms for the identification of bilinear forms," *Digital Signal Process.*, vol. 83, pp. 280–296, Dec. 2018.

[37] D. Gesbert and P. Duhamel, "Robust blind joint data/channel estimation based on bilinear optimization," in *Proceedings of the 8th Workshop on Statistical Signal and Array Processing*, (Corfu, Greece), pp. 168–171, 24–26 June 1996.

[38] A. Stenger, W. Kellermann, and R. Rabenstein, "Adaptation of acoustic echo cancellers incorporating a memoryless nonlinearity," in *Proc. IEEE IWAENC*, 1999.

[39] A. Stenger and W. Kellermann, "Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling," *Signal Processing*, vol. 80, pp. 1747–1760, Sept. 2000.

[40] D.R. Morgan, J. Benesty, and M.M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Process. Lett.*, vol. 5, pp. 174–176, 1998.

[41] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, pp. 3–34, Jun. 2015.

[42] D. A. Harville, *Matrix Algebra From a Statistician's Perspective.* New York, NY, USA: Springer-Verlag, 1997.

[43] C. Paleologu, J. Benesty, and S. Ciochină, "Linear system identification based on a Kronecker product decomposition," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 26, pp. 1793–1808, Oct. 2018.

[44] A. Yener, R. D. Yates, and S. Ulukus, "Interference management for CDMA systems through power control, multiuser detection, and beamforming," *IEEE Trans. Commun.*, vol. 49(7), pp. 1227–1239, Jul. 2001.

[45] D. R. Morgan and S. G. Kratzer, "On a class of computationally efficient, rapidly converging, generalized NLMS algorithms," *IEEE Signal Process. Lett.*, vol. 3, pp. 245–247, Aug. 1996.

[46] J. Benesty, C. Paleologu, and S. Ciochină, "On regularization in adaptive filtering," *IEEE Trans. Audio, Speech, Language Processing*, vol. 19, pp. 1734–1742, Aug. 2011.

[47] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation.* Berlin, Germany: Springer-Verlag, 2001.

[48] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE ICASSP*, pp. II–1881–II–1884, 2002.

[49] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech, Audio Processing*, vol. 8, pp. 508–518, Sept. 2000.

[50] H. Deng and M. Doroslovački, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Trans. Signal Processing*, vol. 54, pp. 1794–1803, May 2006.

[51] J. Liu and S. L. Grant, "Proportionate adaptive filtering for block-sparse system identification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, pp. 623–630, Apr. 2016.

[52] C. Paleologu, J. Benesty, C. Elisei-Iliescu, C. Stanciu, and S. Ciochină, "A proportionate NLMS algorithm for the identification of sparse bilinear forms," in *Proc. IEEE TSP*, pp. 698–701, 2018.

[53] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Jpn.*, vol. 67-A, pp. 19–27, May 1984.

[54] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, "A generalized proportionate variable step-size algorithm for fast changing acoustic environments," in *Proc. IEEE ICASSP*, pp. IV–161–IV–164, 2004.

[55] "Digital Network Echo Cancellers." `https://www.itu.int/rec/T-REC-G.168/en`, 2002. Online; accessed on 16 April 2019.

[56] S. Haykin and B. Widrow, *(Eds.) Least-Mean-Square Adaptive Filters.* Wiley: Hoboken, NJ, USA, 2003.

[57] A.I. Sulyman and A. Zerguine, "Convergence and steady-state analysis of a variable step-size NLMS algorithm," *Signal Processing*, vol. 83, pp. 1255–1273, 2003.

[58] S. Ciochină, C. Paleologu, and J. Benesty, "An optimized NLMS algorithm for system identification," *Signal Process.*, vol. 118, pp. 115–121, Jan. 2016.

[59] E. Eweda, "Convergence analysis of adaptive filtering algorithms with singular data covariance matrix," *IEEE Trans. Signal Processing*, vol. 49, pp. 334–343, Feb. 2001.

[60] S.-E Kim, J.-W. Lee, and W.-J. Song, "A theory on the convergence behavior of the affine projection algorithm," *IEEE Trans. Signal Processing*, vol. 59, pp. 6233–6239, Dec. 2011.

[61] P. G. Park, C. H. Lee, and J. W. Ko, "Mean-square deviation analysis of affine projection algorithm," *IEEE Trans. Signal Processing*, vol. 59, pp. 5789–5799, Dec. 2011.

[62] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: The Johns Hopkins University Press, 1996.

[63] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[64] M.A. Iqbal and S.L. Grant, "Novel variable step size NLMS algorithm for echo cancellation," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Las Vegas, NV, USA), pp. 241–244, 31 March–4 April 2008.

[65] C. Paleologu, S. Ciochină, and J. Benesty, "Double-talk robust VSS-NLMS algorithm for under-modeling acoustic echo cancellation," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Las Vegas, NV, USA), pp. 245–248, 31 March–4 April 2008.

[66] L.-M. Dogariu, S. Ciochină, J. Benesty, and C. Paleologu, "System identification based on tensor decompositions: a trilinear approach," *Symmetry*, vol. 11, p. 556, 2019.

[67] L.-M. Dogariu, S. Ciochină, J. Benesty, and C. Paleologu, "An Iterative Wiener Filter for the Identification of Trilinear Forms," in *Proc. IEEE International Conference on Telecommunications and Signal Processing (TSP)*, (Budapest, Hungary), 1–3 July 2019.

[68] L. N. Ribeiro, S. Schwarz, M. Rupp, A. L. F. de Almeida, J. C. M. Mota, "A low-complexity equalizer for massive MIMO systems based on array separability," in *Proc. EUSIPCO*, (Kos, Greece), pp. 2522–2526, 28 August–2 September 2017.

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

[70] L. Ljung, *System Identification: Theory for the User (2nd ed.).* Prentice-Hall: Upper Saddle River, NJ, USA, 1999.

[71] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, pp. 105–122, 2000.

[72] P. Kroonenberg, *Applied Multiway Data Analysis.* Hoboken, NJ: Wiley, 2008.

[73] L. D. Lathauwer, *Signal Processing Based on Multilinear Algebra.* PhD thesis, Katholieke Universiteit Leuven, 1997.

[74] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, pp. 455–500, Sept. 2009.

[75] P. Comon, "Tensors: a brief introduction," *IEEE Signal Processing Mag.*, vol. 31, pp. 44–53, May 2014.

[76] A. Cichocki, D. P. Mandic, A. Huy Phan, C. F. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications," *IEEE Signal Processing Mag.*, vol. 32, pp. 145–163, Mar. 2015.

[77] J. Chen and Y. Saad, “On the tensor SVD and the optimal low rank orthogonal approximation of tensors,” *SIAM J. Matrix Anal. & Appl.*, vol. 30, pp. 1709–1734, Apr. 2009.

[78] D. Bertsekas, *Nonlinear Programming (2nd edition).* Athena Scientific, Belmont, Massachusetts, 1999.

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

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

[81] S. L. Gay and J. B. (Editors), *Acoustic Signal Processing for Telecommunication.* Kluwer Academic Publisher: Boston, MA, USA, 2000.

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

[83] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, “Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis,” *IEEE Signal Processing Mag.*, vol. 31, pp. 71–79, Sept. 2014.

[84] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Trans. Signal Processing*, vol. 65(13), pp. 3551–3582, July 2017.

[85] M. Boussé, O. Debals, and L. De Lathauwer, “A tensor-based method for large-scale blind source separation using segmentation,” *IEEE Trans. Signal Processing*, vol. 65(2), pp. 346–358, Jan. 2017.

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

[87] L.-M. Dogariu, C. Paleologu, J. Benesty, and S. Ciochină, “An efficient Kalman filter for the identification of low-rank systems,” *Signal Processing*, vol. 166, p. 107239, January 2020.

[88] C. F. Van Loan, “The ubiquitous Kronecker product,” *J. Computational Applied Mathematics*, vol. 123, pp. 85–100, 2000.

[89] G. Wang, S. S. Ge, R. Xue, J. Zhao, and C. Li, “Complex-valued Kalman filters based on Gaussian entropy,” *Signal Processing*, vol. 160, pp. 178–189, July 2019.

[90] G. Enzner and P. Vary, “Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones,” *Signal Processing*, vol. 86, pp. 1140–1156, 2006.

[91] C. Paleologu, J. Benesty, S. Ciochină, and S. L. Grant, “A Kalman filter with individual control factors for echo cancellation,” in *Proc. IEEE ICASSP*, pp. 6015–6019, 2014.

[92] F. Yang, G. Enzner, and J. Yang, “Frequency-domain adaptive Kalman filter with fast recovery of abrupt echo-path changes,” *IEEE Signal Processing Lett.*, vol. 24, pp. 1778–1782, Dec. 2017.

[93] F. Van Eeghem, O. Debals, N. Vervliet, and L. De Lathauwer, “Coupled and incomplete tensors in blind system identification,” *IEEE Trans. Signal Processing*, vol. 66, pp. 6137–6147, Dec. 2018.

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

[95] N. Iqbal, E. Liu, J. H. McClellan, A. Al-Shuhail, S. I. Kaka, and A. Zerguine, “Detection and denoising of microseismic events using time-frequency representation and tensor decomposition,” *IEEE Access*, vol. 6, pp. 22993–23006, May 2018.

[96] J. Benesty, H. Rey, L. Rey Vega, and S. Tressens, “A non-parametric VSS NLMS algorithm,” *IEEE Signal Processing Lett.*, vol. 13, pp. 581–584, Oct. 2006.

[97] A. H. Sayed and T. Kailath, “A state-space approach to adaptive RLS filtering,” *IEEE Signal Processing Mag.*, vol. 11, pp. 18–60, July 1994.

[98] L.-M. Dogariu, S. Ciochină, C. Paleologu, J. Benesty, and P. Piantanida, “An adaptive solution for nonlinear system identification,” in *Proc. IEEE International Symposium on Signals, Circuits and Systems (ISSCS)*, (Iasi, Romania), pp. 1–4, July 2017.

[99] R. Roy and J. Sherman, “A learning technique for Volterra series representation,” *IEEE Trans. Automatic Control*, vol. 12, pp. 761–764, Dec. 1967.

[100] H. A. Barker, S. N. Obidegwu, and T. Pradisthayon, “Performance of antisymmetric pseudorandom signals in the measurement of 2nd-order Volterra kernels by crosscorrelation,” *Proc. IEEE*, vol. 119, pp. 353–362, Mar. 1972.

[101] M. Annabestani, N. Naghavi, and M. M. Nejad, “Nonautoregressive nonlinear identification of IPMC in large deformation situations using generalized Volterra-based approach,” *IEEE Trans. Instrumentation and Measurement*, vol. 65, pp. 2866–2872, Dec. 2016.

[102] Z. Zhang and Y. Ma, “Modeling of rate-dependent hysteresis using a GPO-based adaptive filter,” *Sensors*, vol. 16, p. 205, 2016.

[103] W. J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach.* Johns Hopkins University Press: Baltimore, MD, USA, 1981.

[104] L. Carassale and A. Kareem, “Modeling nonlinear systems by Volterra series,” *J. Eng. Mech.*, vol. 136, pp. 801–818, 2010.

[105] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing.* NY: Wiley, 2001.

[106] S. Haykin, *Neural Networks: A Comprehensive Foundation.* Englewood Cliffs, NJ: Prentice-Hall, 1994.

[107] B. Noble, *Methods Based on the Wiener-Hopf Technique for the Solution of Partial Differential Equations.* NY: Pergamon Press, 1958.

[108] S. Ciochină and C. Negrescu, *Sisteme adaptive (in Romanian).* Bucharest: Ed. Tehnică, 1999.