
Federal University of Ceará
Department of Teleinformatics Engineering
Postgraduate Program in Teleinformatics Engineering

# On Supervised Multilinear Filtering: Applications to System Identification and Antenna Beamforming

***Master of Science Thesis***

**Author**
Lucas Nogueira Ribeiro

**Advisor**
Prof. Dr. João César Moura Mota

**Co-Advisor**
Prof. Dr. André Lima Férrer de Almeida

Fortaleza – Brazil
February 2016

Universidade Federal do Ceará
Departamento de Engenharia de Teleinformática
Programa de Pós-graduação em Engenharia de Teleinformática

# Sobre a Filtragem Multilinear Supervisionada: Aplicações em Identificação de Sistemas e Formatação de Feixes de Antenas

**Autor**
Lucas Nogueira Ribeiro

**Orientador**
Prof. Dr. João César Moura Mota

**Co-Orientador**
Prof. Dr. André Lima Férrer de Almeida

*Dissertação apresentada à Coordenação do Programa de Pós-graduação em Engenharia de Teleinformática da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de* ***Mestre em Engenharia de Teleinformática****. Área de concentração: Sinais e sistemas.*

Fortaleza – Brasil
Fevereiro 2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Pós-Graduação em Engenharia - BPGE

R369o Ribeiro, Lucas Nogueira.
On supervised multilinear filtering: applications to system identification and antenna beamforming / Lucas Nogueira Ribeiro. – 2016.
85 f. : il. color. , enc. ; 30 cm.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia de Teleinformática, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2016.
Área de concentração: Sinais e Sistemas.
Orientação: Prof. Dr. João César Moura Mota.
Coorientação: Prof. Dr. André Lima Férrer de Almeida.

1. Teleinformática. 2. Tensores. 3. Identificação de sistemas. 4. Antenas. I. Título.

CDD 621.38

UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

CAMPUS DO PICI, CAIXA POSTAL 6007 CEP 60.738-640
FORTALEZA – CEARÁ - BRASIL

**LUCAS NOGUEIRA RIBEIRO**

**ON SUPERVISED MULTILINEAR FILTERING: APPLICATIONS TO SYSTEM IDENTIFICATION AND ANTENNA BEAMFORMING**

**Dissertação submetida à Coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática, da Universidade Federal do Ceará, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Teleinformática.**
**Área de concentração: Sinais e Sistemas**

**Aprovada em: 24/02/2016.**

**BANCA EXAMINADORA**

**Prof. Dr. JOÃO CESAR MOURA MOTA (Orientador)**
**Universidade Federal do Ceará**

**Profa. Dra. ANDRÉ LIMA FÉRRER DE ALMEIDA**
**Universidade Federal do Ceará**

**Prof. Dr. CHARLES CASIMIRO CAVALCANTE**
**Universidade Federal do Ceará**

**Prof. Dr. RODRIGO CAIADO DE LAMARE**
**Pontifícia Universidade Católica do Rio de Janeiro**

# Abstract

Linear filtering methods are well known and have been successfully applied to many engineering problems. However, they become unpractical when the parameter space is very large. The recently proposed assumption of system separability allows the development of computationally efficient alternatives to classical filtering methods in this scenario. In this work, we show that system separability calls for multilinear system representation and filtering. Based on this parallel, the proposed filtering framework consists of a multilinear extension of the classical Wiener-Hopf (WH) filter that exploits the separability property to solve the supervised multilinear filtering problem. System identification and antenna beamforming computer simulations were conducted to assess the performance of the proposed method. Our numerical results show our approach has smaller computational complexity and that it provides better estimation accuracy than the classical WH filter, which ignores the multilinear system structure.

**Keywords**: Multilinear filtering, tensors, system identification, antenna array.

# Resumo

Métodos de filtragem linear estão bem estabelecidos e têm sido aplicados em diversos problemas de engenharia. Entretanto, eles tornam-se impraticáveis quando o espaço de parâmetros é grande. A recente hipótese de separabilidade de sistema permite o desenvolvimento de métodos computacionalmente eficientes neste cenário. Neste trabalho, nós mostramos que a separabilidade de um sistema leva à sua representação multilinear. Em vista disso, o método de filtragem proposto consiste em uma extensão multilinear do filtro de Wiener-Hopf (WH) clássico, que explora a separabilidade para resolver o problema de filtragem multilinear supervisionada. Simulações computacionais de identificação de sistemas e formatação de feixes de antenas foram realizadas para a avaliação do desempenho do método proposto. Nosso resultados numéricos mostram que nossa abordagem possui menor complexidade computacional e que ela fornece melhor acurácia de estimação que o filtro de WH clássico, que ignora a estrutura multilinear do sistema.

**Palavras-chave**: Filtragem multilinear, tensores, identificação de sistemas, arranjo de antenas.

# Acknowledgments

I would like to express my deepest gratitude towards:

– God, for taking care of me;

– My family, for always supporting me;

– Priscilla, for the love;

– CAPES, for the financial support;

– Prof. João César and Prof. André, for the invested time in the supervision of this thesis and for all fruitful scientific discussions;

# List of Acronyms

| | |
|---|---|
| **ALS** | Alternating Least Squares |
| **CDMA** | Code Division Multiple Access |
| **FIR** | Finite Impulse Response |
| **FLOPS** | Floating-point Operations Per Second |
| **LMS** | Least Mean Square |
| **MMSE** | Minimum Mean Square Error |
| **MSE** | Mean Square Error |
| **NLMS** | Normalized Least Mean Square |
| **PARAFAC** | Parallel Factors |
| **PNLMS** | Proportionate Normalized Least Mean Square |
| **QPSK** | Quadrature Phase-Shift Keying |
| **SepWH** | Separable Wiener-Hopf |
| **SNR** | Signal-to-Noise Ratio |
| **SOI** | Signal of Interest |
| **STAP** | Space-Time Adaptive Processing |
| **TLMS** | Tensor Least Mean Square |
| **TWH** | Tensor Wiener-Hopf |
| **WH** | Wiener-Hopf |
| **WSS** | Wide Sense Stationary |

# List of Symbols

| | |
|---|---|
| $\jmath$ | Square root of $-1$ |
| $\mathbb{Z}$ | Field of integer numbers |
| $\mathbb{R}$ | Field of real numbers |
| $\mathbb{C}$ | Field of complex numbers |
| $\mathbb{E}[\cdot]$ | Statistical expectation |
| $(\cdot)^{\mathsf{H}}$ | Hermitian operator |
| $(\cdot)^{\mathsf{T}}$ | Transpose operator |
| $\cdot \coloneqq \cdot$ | Definition |
| $\Theta(\cdot)$ | Tensorization operator |
| $\mathrm{vec}(\cdot)$ | Vectorization operator |
| $\cdot \circ \cdot$ | Outer product |
| $\cdot \otimes \cdot$ | Kronecker product |
| $\cdot \times_n \cdot$ | $n$-mode product |
| $\langle \cdot, \cdot \rangle$ | Inner product |
| $\left\|\cdot\right\|_{\mathrm{F}}$ | Frobenius norm |
| $\left\| \cdot \right\|_2$ | Euclidean norm |
| $\cdot \sqcup_N \cdot$ | Tensor concatenation along the $N$th dimension |
| $\cdot * \cdot$ | Discrete-time convolution |
| $\mathcal{O}(\cdot)$ | Big-O notation for computational complexity |

# List of Figures

# Contents

# Chapter 1

# Introduction

The filtering problem consists of designing a set of parameters which extract information from a set of noisy data [1]. A filter can be mathematically represented by a function that transforms an input signal into an output signal. A filter is classified with respect to the mathematical properties of its model: linear or non-linear, instantaneous (memoryless) or dynamic (with memory), time variant or invariant, among others [2]. The statistical approach to the filtering problem consists of designing a filter that minimizes the error between its output signal and a given desired signal. In system identification problems, for instance, a filter is employed to model an unknown system. In this context, both devices are fed by the same input signal, and the output signal produced by the unknown system is regarded as the desired signal. The unknown system model parameters can be identified by minimizing the mean square error between the filter output and the desired signal [1]. This process is said to be supervised because the error minimization depends on the availability of a desired signal, also called reference signal. There are basically two types of supervised filtering techniques: optimal methods, such as the Wiener-Hopf (WH) solution, which computes the filter coefficients after observing all available data and using known statistics, and adaptive filtering algorithms, such as the Least Mean Square (LMS) algorithm, which update the filter coefficients as new data samples are available. The WH filter and the LMS algorithm are widely employed in practical problems and hereafter they will be referred to as classical filtering methods.

Although these methods are straightforward to implement and provide sufficiently accurate solutions, they present serious drawbacks in some scenarios. For instance, Big Data problems demands low-complexity filtering techniques capable of dealing with very large datasets [3]. In this context, optimum methods

need to cope with large covariance matrices, whereas adaptive filters face slow convergence rate [4]. Furthermore, these methods are not adapted to process multidimensional signals, whose information is defined in multiple signal domains. Instead of employing different filters to process different information domains, classical filtering methods usually make use of a unique filter to jointly process all information domains, resulting in a large filtering problem. By contrast, taking the multidimensionality into account allows the use of filters with smaller dimensions, reducing the overall computational complexity of the filtering. Another important drawback of classical filtering methods is that their performance strongly depends on the power spectrum of the input signal. It is known that the condition number of the input signal covariance matrix determines the numerical stability of the WH filter and the convergence rate of LMS-based algorithms [1]. It is defined as the ratio between the largest and smallest eigenvalues (eigenvalue spread) of the input signal covariance matrix. Larger eigenvalue spread characterizes singular covariance matrices, resulting in unstable WH solutions and slow LMS convergence. Nonlinear system models, such as Volterra systems, are known to present large eigenvalue spreading, which decreases the convergence rate of adaptive filtering methods [5].

Many ideas have been proposed for increasing the numerical stability and the convergence rate of classical filtering methods. Variable step size factors [4], transform domain implementations [6], multistage nested Wiener filtering [7, 8], affine projection algorithms [9], and Newton-based solutions [10] are among the most popular approaches to increasing the convergence speed of adaptive filtering algorithms. Another idea consists of exploiting *a priori* information on the system impulse response and signals. For instance, the proportionate normalized least mean square (PNLMS) algorithm was proposed in [11] to increase the convergence speed by exploiting the impulse response sparsity. However, its performance severely deteriorates in large-scale scenarios and it is not adapted to multidimensional problems. System separability is another *a priori* information that has been used for many years to reduce the implementation costs of large filters. A system is said to be separable when it can be expressed as a "product" of subsystems. This "product" can either be a convolution or a Kronecker product depending on the application. In image processing, separable filters are usually expressed in terms of convolutions, whereas Kronecker products are used in other contexts such as telecommunications. Efficient design methods for separable bidimensional filters were proposed several years ago [12, 13]. More recently, large

multidimensional filters were approximated as convolution separable systems in [14, 15]. A tensor LMS (TLMS) algorithm was proposed in [16] to identify a separable finite impulse response (FIR) filter with large parameter space. This method provides an important increase of convergence rate, which is due to the division of the large identification task into smaller ones. In some applications, each separable factor is associated with a signal domain, and, in this case, filtering methods that exploit separability can be also regarded as multidimensional filtering methods. System separability can be found in many practical problems such as nonlinear systems modeling [17], compressed sensing [18], and array signal processing [19].

Tensor filtering, also called multiway filtering [20], has been widely used to process multidimensional signals. It consists of using a multidimensional (tensor) filter to process the multiple signal domains. In the last few years, it has been shown that tensor methods provides some advantages over vector and matrix methods: parameter reduction [21], lower approximation error [16], and faster convergence [16]. In the following, we briefly review some of these works:

- Tensor filtering methods based on a generalized signal subspace approach were reviewed in [20] and applied to color image denoising problem. Computational results showed that the tensor generalization provided less noisy estimates than the matrix method.

- In [22], a generalization of the WH filter based on a generalized signal subspace approach was proposed and applied to denoising problem as well.

- A low-rank space-time adaptive processing (STAP) tensor filter was proposed in [23] to jointly process a space-time-polarization tensor. Such processing presented superior performance compared to vector methods.

- Tensor filtering was also used in the multilinear regression method proposed in [24] to predict clinical risk of Alzheimer's disease and heart failure. The multilinear regression problem was formulated as a multilinear optimization problem that was solved using a Block Coordinate Descent method, providing the state-of-art prediction performance.

- We have proposed a tridimensional batch filtering method in [25] inspired by [26], where a rank-constrained spatiotemporal matrix filter for Code Division Multiple Access (CDMA) systems was computed using an alternat-

ing optimization approach. The performance of the proposed batch filtering method was assessed in signal denoising problems.

– In [27, 28], a multilinear least-squares method based on alternating minimization was proposed to decompose multidimensional filters into a cascade of subfilters. Such decomposition would decrease the computational cost of multidimensional filtering problems. The proposed decomposition technique provided more robustness to initialization than other alternating minimization based methods. The considered multidimensional filters were separable in the sense of the convolution product.

– Learning methods for separable multidimensional image filters based on the parallel factors (PARAFAC) tensor decomposition were proposed in [15]. These methods were employed to approximate a bank of 2D image filters as a sum of rank-1 tensors. This approximation provided parameter reduction and lower approximation errors compared to competing methods.

As most adaptive methods, the performance of the TLMS algorithm depends on a step size factor. This algorithm demands a small step size factor to attain lower error levels, which drastically reduces its convergence rate. This problem is more pronounced when the filter parameter space is large. In view of this problem, we questioned ourselves whether an alternating optimization method using batch filtering (WH filtering) instead of stochastic gradient descent (LMS) would decrease the overall computational costs. Although this proposed approach, named Tensor Wiener-Hopf (TWH) algorithm, would still be subject to convergence issues, the WH filters would provide fast iterations towards the optimal solution. Since this approach divides a large-scale optimization problem into simpler subproblems, it should demand less computational cost than classical filtering methods as well. It is known that system separability provides a reduction of computational costs and approximation errors, but to what extent does it provide these gains? Higher separability order leads to smaller separable factors, which can be accurately estimated using less computational resources. Therefore we expect the parameter estimation accuracy to increase and the computational complexity to decrease with the system separability order up to a certain degree.

To answer these questions, we have conducted system identification and beamforming computer simulations. In the system identification problem, we have considered the parameter identification of a separable FIR filter, whereas, in the beamforming problem, we have proposed a multilinear model for trans-

lation invariant arrays that calls for tensor beamforming. We have observed in our simulations that TWH provides inferior computational complexity than its classical counterpart and that system separability provides advantages up to a certain separability order. We have also managed to show that every separable FIR system can be factorized in terms of sparse subsystems. In conclusion, the main contributions of this thesis are:

- The TWH algorithm, which provides state-of-art parameter estimation accuracy with small computational costs in large-scale separable filtering problems;
- The Extended Subsystem Theorem, which decomposes any separable FIR system into a cascade of sparse subsystems;
- Insights on the gains provided by separable filtering methods;
- The proposed separable multilinear sensor array model;

## 1.1 Scientific Output

Two papers were produced from the results obtained in this thesis:

- **L. N. Ribeiro**, A. L. F. de Almeida and J. C. M. Mota, *"Identification of Separable Systems Using Trilinear Filtering"* published in the 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (IEEE CAMSAP 2015) proceedings;
- **L. N. Ribeiro**, A. L. F. de Almeida and J. C. M. Mota, *"Tensor Beamforming for Multilinear Translation Invariant Arrays"* to appear in the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016);

## 1.2 Thesis organization

This thesis is structured as follows:

– The multilinear filtering methods used in this thesis are introduced in Chapter 2;

– Results from the supervised system identification experiments are presented in Chapter 3;

– A tensor beamformer based on the proposed multilinear filtering method and results from computational simulations are presented in Chapter 4;

– Chapter 5 summarizes our conclusions and lists some research perspectives on tensor filtering;

– The classical linear filtering methods are reviewed in Appendix A;

– The separability of exponential transversal filters is discussed in Appendix B;

– The papers mentioned in Section 1.1 are attached in Appendices C and D;

# Chapter 2

# Supervised Filtering Methods

This chapter is devoted to the presentation of the multilinear supervised filtering methods proposed in this thesis. First, the supervised linear filtering problem is stated and the classical filtering methods are presented. Afterwards, the multilinear filtering problem is introduced in Section 2.2. We propose the SepWH method, which is based on the factorization of rank-1 multilinear filters. Next we propose the TWH algorithm, which solves the multilinear filtering problem using an alternating optimization approach. Finally, we present the TLMS algorithm, which can be regarded as the stochastic gradient descent counterpart of the TWH algorithm.

## 2.1 Supervised Linear Filtering

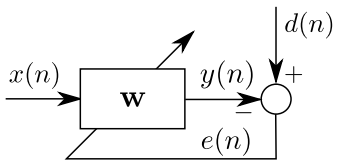



Figure 2.1: Supervised linear filtering model.

The supervised linear filtering problem consists of processing an input signal $x(n)$ using a linear filter to produce an estimate $y(n)$ of a desired signal $d(n)$ [1]. The linear filter is designed to minimize the estimation error $e(n) \coloneqq d(n) - y(n)$ according to some statistic criterion. In this work, we assume that $x(n)$ is a wide sense stationary (WSS) discrete-time signal and that the linear filter is a $N$th order tapped delay line. This filter produces an output signal $y(n)$, which is given

by the convolution sum

$$y(n) = \sum_{k=0}^{N-1} w_k^* x(n-k) = \mathbf{w}^{\mathsf{H}} \mathbf{x}(n), \tag{2.1}$$

where

$$\mathbf{w} \coloneqq [w_0, w_1, \ldots, w_{N-1}]^{\mathsf{T}} \in \mathbb{C}^N$$

and

$$\mathbf{x}(n) \coloneqq [x(n), x(n-1), \ldots, x(n-N+1)]^{\mathsf{T}} \in \mathbb{C}^N$$

denote the filter coefficients vector and the input signal regression vector, respectively. The supervised linear filtering problem is illustrated in Figure 2.1.

The mean square error (MSE) is an attractive optimization criterion due to its mathematical tractability. Since the input signal $x(n)$ is assumed to be WSS, it can be shown that the MSE curve is convex and has a unique global minimum [1]. In this context, we seek to minimize the MSE between $y(n)$ and $d(n)$, i.e. we aim at solving the following optimization problem:

$$\min_{\mathbf{w}} \mathbb{E}\left[\left|d(n) - \mathbf{w}^{\mathsf{H}} \mathbf{x}(n)\right|^2\right], \tag{2.2}$$

where $\mathbb{E}[\cdot]$ denotes the statistical expectation operator.

Let the objective function be defined as $J_{\mathbf{w}} \coloneqq \mathbb{E}\left[\left|d(n) - \mathbf{w}^{\mathsf{H}} \mathbf{x}(n)\right|^2\right]$. The WH filter (Section A.1) provides the minimum mean square error (MMSE) solution to (2.2) by solving the $\nabla J_{\mathbf{w}} = \frac{\partial J_{\mathbf{w}}}{\partial \mathbf{w}^{\mathsf{H}}} = \mathbf{0}$ for $\mathbf{w}$. On the other hand, the LMS algorithm (Section A.2) solves (2.2) by using the stochastic gradient descent method. The Normalized Least Mean Square (NLMS) algorithm, reviewed in Section A.3, is a modified LMS which employs a variable step size factor. The reader is referred to Appendix A for more information on these classical filtering methods.

## 2.2 Supervised Multilinear Filtering

In the last Section, we have considered the linear filtering problem, in which a FIR filter was regarded as a linear operator. We will show in this Section that it can be also regarded as a multilinear operator under the hypothesis that the filter coefficients are separable. This hypothesis leads to the multilinear filtering problem, which will be introduced in the following. Before introducing this

problem, multilinear (tensor) algebra prerequisites will be presented for convenience. Subsequently multilinear extensions of the linear filtering methods will be introduced.

### 2.2.1 Tensor Prerequisites

An $N$th order tensor is essentially an element of a tensor product between $N$ vector spaces [29]. Any $N$th order tensor can be represented by a multidimensional coordinates array whenever the basis of the vector spaces $\mathbb{V}_n$ are set for $n = 1, \ldots, N$. Hereafter the term "tensor" will refer to its multidimensional array representation, denoted by uppercase calligraphic letters, e.g. $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$, where $I_n \coloneqq \dim(\mathbb{V}_n)$ for $n = 1, \ldots, N$. The scalar tensor element indexed by $(i_1, i_2, \ldots, i_N)$ is denoted by $[\mathcal{X}]_{i_1 i_2 \ldots i_N} \coloneqq x_{i_1 i_2 \ldots i_N}$, where $i_n \in \{1, \ldots, I_n\}$ for $n = 1, \ldots, N$. Note that scalars can be regarded as 0th order tensors, vectors as 1st order tensors, and matrices as 2nd order tensors. Other tensor definitions and operators will be defined next.

**Definition 1** (Tensorization)**.** *Let $\Theta : \mathbb{C}^{\prod_{n=1}^{N} I_n} \to \mathbb{C}^{\times_{n=1}^{N} I_n}$ denote the tensorization operator, which transforms a vector $\mathbf{v} \in \mathbb{C}^{\prod_{n=1}^{N} I_n}$ into a tensor $\Theta(\mathbf{v}) \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ whose components are defined as*

$$[\Theta(\mathbf{v})]_{i_1 i_2 \ldots i_N} \coloneqq [\mathbf{v}]_j, \quad j \coloneqq i_1 + \sum_{n=2}^{N} (i_n - 1) \prod_{v=1}^{n-1} I_v. \tag{2.3}$$

For example, let $N = 3$, $I_1 = 2$, $I_2 = 3$, $I_3 = 4$, and $\mathbf{v} \in \mathbb{C}^{I_1 I_2 I_3}$. The tensorization of $\mathbf{v}$ is $\Theta(\mathbf{v}) \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ and its elements are given by

$$[\Theta(\mathbf{v})]_{i_1 i_2 i_3} \coloneqq [\mathbf{v}]_j, \quad j \coloneqq i_1 + (i_2 - 1) I_1 + (i_3 - 1) I_1 I_2$$

for $i_1 \in \{1, 2\}$, $i_2 \in \{1, 2, 3\}$, and $i_3 \in \{1, 2, 3, 4\}$.

**Definition 2** (Vectorization)**.** *Let $vec : \mathbb{C}^{\times_{n=1}^{N} I_n} \to \mathbb{C}^{\prod_{n=1}^{N} I_n}$ denote the vectorization operator, which transforms a tensor $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ into a vector $vec(\mathcal{X}) \in \mathbb{C}^{\prod_{n=1}^{N} I_n}$ whose components are defined as*

$$[vec(\mathcal{X})]_j \coloneqq [\mathcal{X}]_{i_1 i_2 \ldots i_N}, \quad j \coloneqq i_1 + \sum_{n=2}^{N} (i_n - 1) \prod_{v=1}^{n-1} I_v. \tag{2.4}$$

**Definition 3** (Tensor fiber [30])**.** *The $n$-mode tensor fiber of an $N$th order tensor $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ is defined as the vector formed by fixing every index but the $i_n$th.*

For instance, consider a third-order tensor $\mathcal{U} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$. Its 1-, 2-, and 3-mode fibers are given by

$$\begin{aligned}\mathbf{u}_{\cdot i_2 i_3} &\in \mathbb{C}^{I_1},\\ \mathbf{u}_{i_1 \cdot i_3} &\in \mathbb{C}^{I_2},\\ \mathbf{u}_{i_1 i_2 \cdot} &\in \mathbb{C}^{I_3},\end{aligned}$$

where "$\cdot$" denotes the varying index. Note that the vec$(\cdot)$ operator can be seen as the concatenation of the 1-mode fibers along the first dimension of a vector.

**Definition 4** (Tensor unfolding [30])**.** *The $n$-mode unfolding of an $N$th order tensor $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ is a matrix $\mathbf{X}_{(n)} \in \mathbb{C}^{I_n \times I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N}$ whose entries are defined as*

$$[\mathbf{X}_{(n)}]_{i_n j} \coloneqq [\mathcal{X}]_{i_1 \ldots i_N}, \quad j \coloneqq 1 + \sum_{\substack{u=1 \\ u \neq n}}^{N} (i_u - 1) \prod_{\substack{v=1 \\ v \neq n}}^{u-1} I_v.$$

For instance, the unfoldings of $\mathcal{U} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ are

$$\begin{aligned}\mathbf{U}_{(1)} &\in \mathbb{C}^{I_1 \times I_2 I_3},\\ \mathbf{U}_{(2)} &\in \mathbb{C}^{I_2 \times I_1 I_3},\\ \mathbf{U}_{(3)} &\in \mathbb{C}^{I_3 \times I_1 I_2}.\end{aligned}$$

Note that the $n$-mode matrix unfolding can be seen as the concatenation of the $n$-mode fibers along the matrix columns.

**Definition 5** (Tensor concatenation [31])**.** *The concatenation of $T$ tensors $\mathcal{U}_t \in \mathbb{C}^{\times_{n=1}^{N} I_n}, t = 1, \ldots, T$, along the $(M+1)$th dimension is denoted by*

$$\mathcal{U} = \mathcal{U}_1 \sqcup_{M+1} \mathcal{U}_2 \sqcup_{M+1} \ldots \sqcup_{M+1} \mathcal{U}_T \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_M \times T},$$

*where $[\mathcal{U}]_{i_1 i_2 \ldots i_M t} = [\mathcal{U}_t]_{i_1 i_2 \ldots i_M}$.*

**Definition 6** (Outer product)**.** *Consider an $N$th order tensor $\mathcal{U} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ and an $M$th order tensor $\mathcal{V} \in \mathbb{C}^{\times_{m=1}^{M} J_m}$. The elements of the outer product $\mathcal{T} = \mathcal{U} \circ \mathcal{V} \in \mathbb{C}^{I_1 \times \ldots \times I_N \times J_1 \times \ldots \times J_M}$ are defined as*

$$[\mathcal{T}]_{i_1 \ldots i_N j_1 \ldots j_M} \coloneqq u_{i_1 \ldots i_N} v_{j_1 \ldots j_M}, \tag{2.5}$$

*where "$\circ$" denotes the outer product, and $\mathcal{T}$ is a $(N+M)$th order tensor.*

**Definition 7** (Rank-1 tensor [30]). *Any Nth order rank-1 tensor* $\mathcal{T} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *can be written as*

$$\mathcal{T} \coloneqq \mathbf{t}_1 \circ \mathbf{t}_2 \circ \ldots \circ \mathbf{t}_N, \tag{2.6}$$

*where* $\mathbf{t}_n \in \mathbb{C}^{I_n}$ *for* $n = 1, \ldots, N$.

**Definition 8** (Tensor rank). *Tensor rank is defined as the minimum number of rank-1 components that exactly decomposes additively a tensor.*

**Definition 9** (Inner product [32]). *Consider the tensors* $\mathcal{U} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *and* $\mathcal{V} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *with same dimensions. The inner product* $\langle \mathcal{U}, \mathcal{V} \rangle$ *is defined as*

$$\langle \mathcal{U}, \mathcal{V} \rangle \coloneqq \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_N=1}^{I_N} u_{i_1 i_2 \ldots i_N} v^*_{i_1 i_2 \ldots i_N}, \tag{2.7}$$

*where* $(\cdot)^*$ *denotes complex conjugation.*

**Definition 10.** *The Frobenius norm of an Nth order tensor* $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *is defined as*

$$\|\mathcal{X}\|_F \coloneqq \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_N=1}^{I_N} |x_{i_1 i_2 \ldots i_N}|^2}. \tag{2.8}$$

The Frobenius norm (2.8) can be expressed in terms of an inner product $\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$.

**Definition 11** ($n$-mode product [30]). *The elements of the n-mode product between an Nth order tensor* $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *and a matrix* $\mathbf{U} \in \mathbb{C}^{J \times I_n}$ *is defined as*

$$[\mathcal{X} \times_n \mathbf{U}]_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} \coloneqq \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}, \quad j \in 1, \ldots, J, \tag{2.9}$$

*where "*$\times_n$*" denotes the n-mode product operator.*

**Definition 12.** *The elements of the* $\{1, \ldots, N\}$*-mode product between an Nth order tensor* $\mathcal{X} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ *and a sequence of* $(J_n \times I_n)$*-dimensional matrices* $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$*, denoted by* $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)} \ldots \times_N \mathbf{U}^{(N)} \in \mathbb{C}^{\times_{n=1}^{N} J_n}$*, are defined as*

$$[\mathcal{Y}]_{j_1 \ldots j_N} \coloneqq \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} x_{i_1 \ldots i_N} u_{j_1 i_1} \ldots u_{j_N i_N}. \tag{2.10}$$

It can be shown that $\mathbf{Y}_{(n)}$, the $n$-mode unfolding of $\mathcal{Y}$, is given by [30]

$$\mathbf{Y}_{(n)} = \mathbf{U}^{(n)} \mathbf{X}_{(n)} \mathbf{U}^{\otimes n^{\mathsf{T}}}, \tag{2.11}$$

where $\mathbf{X}_{(n)}$ denotes the $n$-mode unfolding of $\mathcal{X}$, "$\otimes$" the Kronecker product, and

$$\mathbf{U}^{\otimes n} \coloneqq \mathbf{U}^{(N)} \otimes \ldots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \ldots \otimes \mathbf{U}^{(1)} \tag{2.12}$$

the Kronecker product of the matrices $\{\mathbf{U}^{(j)}\}_{\substack{j=1 \\ j\neq n}}^{N}$ in the decreasing order.

**Definition 13** (Multilinear operator [29])**.** *An operator $f$ that maps $\mathbb{C}_1 \times \ldots \times \mathbb{C}_N$ onto $\mathbb{C}$ is said to be multilinear if $f(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ is linear with respect to every input $\mathbf{v}_n$ for $n = 1, \ldots, N$.*

**Definition 14** (Kronecker product)**.** *The elements of the Kronecker product $\mathbf{v} = \bigotimes_{n=N}^{1} \mathbf{v}_n \in \mathbb{C}^{\prod_{n=1}^{N} I_n}$ are given by*

$$[\mathbf{v}]_j = \prod_{n=1}^{N} [\mathbf{v}_n]_{i_n}, \tag{2.13}$$

*where $\mathbf{v}_n \in \mathbb{C}^{I_n}$ for $n = 1, \ldots, N$, and $j \coloneqq i_1 + \sum_{n=2}^{N} (i_n - 1) \prod_{v=1}^{n-1} I_v$.*

**Proposition 1.** *It follows that $vec(\mathcal{T}) = \bigotimes_{n=N}^{1} \mathbf{t}_n \in \mathbb{C}^{\prod_{n=1}^{N} I_n}$, where $\mathcal{T} = \mathbf{t}_1 \circ \mathbf{t}_2 \circ \ldots \circ \mathbf{t}_N \in \mathbb{C}^{\times_{n=1}^{N}}$ is a rank-1 tensor.*

*Proof.* Let us substitute $\mathcal{T} = \mathbf{t}_1 \circ \mathbf{t}_2 \circ \ldots \circ \mathbf{t}_N$ into (2.4):

$$\begin{aligned} [\mathrm{vec}(\mathcal{T})]_j &= [\mathbf{t}_1 \circ \mathbf{t}_2 \circ \ldots \circ \mathbf{t}_N]_{i_1 i_2 \ldots i_N} \\ &= \prod_{n=1}^{N} [\mathbf{t}_n]_{i_n}. \end{aligned}$$

From Definition 2, we have that $j = i_1 + \sum_{n=2}^{N} (i_n - 1) \prod_{v=1}^{n-1} I_v$. Therefore, $\mathrm{vec}(\mathcal{T}) = \mathbf{t}_N \otimes \mathbf{t}_{N-1} \otimes \ldots \otimes \mathbf{t}_1$ holds according to Definition 14. □

For example, let $\mathbf{a} = [a_1, a_2]^{\mathsf{T}} \in \mathbb{R}^2$ and $\mathbf{b} = [b_1, b_2, b_3]^{\mathsf{T}} \in \mathbb{R}^3$. The Kronecker product between these vectors is given by

$$\mathbf{a} \otimes \mathbf{b} = [a_1 b_1, a_1 b_2, a_1 b_3, a_2 b_1, a_2 b_2, a_2 b_3]^{\mathsf{T}}.$$

The outer product between $\mathbf{b}$ and $\mathbf{a}$, and its vectorization can be written as

$$\mathbf{b} \circ \mathbf{a} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \begin{bmatrix} a_1, a_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_2 b_1 \\ a_1 b_2 & a_2 b_2 \\ a_1 b_3 & a_2 b_3 \end{bmatrix}_{3 \times 2}$$

and $\text{vec}(\mathbf{b} \circ \mathbf{a}) = [a_1b_1, a_1b_2, a_1b_3, a_2b_1, a_2b_2, a_2b_3]^{\mathsf{T}}$, respectively.

**Proposition 2.** *The inner product between an $N$th order tensor $\mathcal{U} \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ and a rank-1 $N$th order tensor $\mathcal{V} = \mathbf{v}_1 \circ \mathbf{v}_2 \circ \ldots \circ \mathbf{v}_N \in \mathbb{C}^{\times_{n=1}^{N} I_n}$ is given by*

$$y(\mathbf{v}_1, \ldots, \mathbf{v}_N) \coloneqq \langle \mathcal{U}, \mathcal{V} \rangle = \mathcal{U} \times_1 \mathbf{v}_1^{\mathsf{H}} \times_2 \mathbf{v}_2^{\mathsf{H}} \ldots \times_N \mathbf{v}_N^{\mathsf{H}} \in \mathbb{C}, \tag{2.14}$$

*where $(\cdot)^{\mathsf{H}}$ denotes the Hermitian operator.*

*Proof.* From the definitions of the rank-1 tensor (2.6) and inner product (2.7), it follows that

$$\begin{aligned} y(\mathbf{v}_1, \ldots, \mathbf{v}_N) &= \langle \mathcal{U}, \mathcal{V} \rangle \\ &= \langle \mathcal{U}, [\mathbf{v}_1 \circ \mathbf{v}_2 \ldots \circ \mathbf{v}_N] \rangle \\ &= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} u_{i_1 i_2 \ldots i_N} \, [\mathbf{v}_1 \circ \mathbf{v}_2 \ldots \circ \mathbf{v}_N]^*_{i_1 i_2 \ldots i_N} \\ &= \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} u_{i_1 i_2 \ldots i_N} [\mathbf{v}_1]^*_{i_1} [\mathbf{v}_2]^*_{i_2} \ldots [\mathbf{v}_N]^*_{i_N}. \end{aligned} \tag{2.15}$$

Equation (2.15) is recognized as a $\{1, \ldots, N\}$-mode product (2.10), which is equal to (2.14). □

**Proposition 3.** *The inner product $y(\mathbf{v}_1, \ldots, \mathbf{v}_N) = \mathcal{U} \times_1 \mathbf{v}_1^{\mathsf{H}} \times_2 \mathbf{v}_2^{\mathsf{H}} \ldots \times_N \mathbf{v}_N^{\mathsf{H}}$ is a multilinear operator.*

*Proof.* Substituting equation (2.11) into $y(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ gives:

$$y(\mathbf{v}_1, \ldots, \mathbf{v}_N) = \mathbf{v}_n^{\mathsf{H}} \mathbf{x}_n \quad n = 1, \ldots, N, \tag{2.16}$$

where $\mathbf{x}_n = \mathbf{U}_{(n)} [\mathbf{v}^{\otimes n}]^*$. Note that $y(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ is linear with respect to the components of $\mathbf{v}_n$ for $n = 1, \ldots, N$, completing the proof. □

**Proposition 4.** *Let*

$$\mathcal{X} \coloneqq [\mathcal{X}^{(1)} \sqcup_{M+1} \ldots \sqcup_{M+1} \mathcal{X}^{(K)}] \in \mathbb{C}^{N_1 \times \ldots \times N_M \times K}$$

*denote the concatenation of $K$ tensors with same dimensions along the $(M+1)$th dimension. The $\{1, \ldots, m-1, m+1, \ldots, N\}$-mode product between $\mathcal{X}$ and $\{\mathbf{w}_j^{\mathsf{H}}\}_{j=1, j \neq m}^{M}$*

*for* $m = 1, \ldots, M$ *is given by*

$$\begin{aligned} \mathbf{U}^{(m)} &= \mathcal{X} \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_{m-1} \mathbf{w}_{m-1}^{\mathsf{H}} \times_{m+1} \mathbf{w}_{m+1}^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}} \\ &= \left[\mathbf{u}_m^{(1)}, \ldots, \mathbf{u}_m^{(K)}\right] \in \mathbb{C}^{N_m \times K}, \end{aligned} \tag{2.17}$$

*where* $\mathbf{u}_m^{(k)} \coloneqq \mathbf{X}_{(m)}^{(k)} \mathbf{w}^{\otimes m} \in \mathbb{C}^{N_m}$, $m \in \{1, \ldots, M\}$.

*Proof.* The elements of (2.17) are given by:

$$\left[\mathbf{U}^{(m)}\right]_{n_m,k} = \sum_{n_1=1}^{N_1} \cdots \sum_{n_{m-1}=1}^{N_{m-1}} \sum_{n_{m+1}=1}^{N_{m+1}} \cdots \sum_{n_M=1}^{N_M} [\mathcal{X}]_{n_1 \ldots n_M k} [\mathbf{w}_1]_{n_1}^* \ldots [\mathbf{w}_{m-1}]_{n_{m-1}}^* [\mathbf{w}_{m+1}]_{n_{m+1}}^* \ldots [\mathbf{w}_M]_{n_M}^* \tag{2.18}$$

for $n_m = 1, \ldots, N_m$ and $k = 1, \ldots, K$. From Definition 5, it follows that $[\mathcal{X}]_{n_1 n_2 \ldots n_M k} = [\mathcal{X}^{(k)}]_{n_1 n_2 \ldots n_M}$ and that the elements of (2.18) become:

$$\left[\mathbf{U}^{(m)}\right]_{n_m,k} = \sum_{n_1=1}^{N_1} \cdots \sum_{n_{m-1}=1}^{N_{m-1}} \sum_{n_{m+1}=1}^{N_{m+1}} \cdots \sum_{n_M=1}^{N_M} [\mathcal{X}^{(k)}]_{n_1 \ldots n_M} [\mathbf{w}_1]_{n_1}^* \ldots [\mathbf{w}_{m-1}]_{n_{m-1}}^* [\mathbf{w}_{m+1}]_{n_{m+1}}^* \ldots [\mathbf{w}_M]_{n_M}^*,$$

From the equation above, we notice that the $k$th column of $\mathbf{U}^{(m)}$ is given by the $\{1, \ldots, m-1, m+1, \ldots, M\}$-mode product $\mathbf{u}_m^{(k)} \coloneqq \mathbf{X}_{(m)}^{(k)} \mathbf{w}^{\otimes m}$.

□

### 2.2.2 Problem Formulation

Now let us introduce the multilinear generalization of the supervised linear filtering problem presented in the last Section. For simplicity's sake, we consider rank-1 separable systems. Consider an $M$th order separable FIR system whose coefficients vector is given by

$$\mathbf{h} = \bigotimes_{m=M}^{1} \mathbf{h}_m \in \mathbb{C}^N, \tag{2.19}$$

where $\mathbf{h}_m \in \mathbb{C}^{N_m}$ denotes its $m$th order subsystem for $m = 1, \ldots, M$. An input signal $x(n)$ feeds the unknown system, producing

$$\begin{aligned}\tilde{d}(n) &= \mathbf{h}^{\mathsf{H}}\mathbf{x}(n) \\ &= \sum_{k=1}^{N} [\mathbf{h}]_k^* [\mathbf{x}(n)]_k \\ &= \sum_{k_1=1}^{N_1} \sum_{k_2=1}^{N_2} \cdots \sum_{k_M=1}^{N_M} [\mathbf{h}_1]_{k_1}^* [\mathbf{h}_2]_{k_2}^* \ldots [\mathbf{h}_M]_{k_M}^* [\mathbf{x}(n)]_{k'}, \end{aligned} \tag{2.20}$$

where $k' \coloneqq k_1 + \sum_{m=2}^{M} (k_m - 1) \prod_{v=1}^{m-1} N_v$. A separable FIR filter

$$\mathbf{w} = \bigotimes_{m=M}^{1} \mathbf{w}_m \in \mathbb{C}^N, \tag{2.21}$$

where $\mathbf{w}_m \in \mathbb{C}^{N_m}$ is employed to identify $\mathbf{h}$. This filter is fed by $x(n)$ as well, yielding an output signal

$$y(n) = \sum_{k_1=1}^{N_1} \sum_{k_2=1}^{N_2} \cdots \sum_{k_M=1}^{N_M} [\mathbf{w}_1]_{k_1}^* [\mathbf{w}_2]_{k_2}^* \ldots [\mathbf{w}_M]_{k_M}^* [\mathbf{x}(n)]_{k'}. \tag{2.22}$$

From Proposition 1, $\mathbf{h}$ and $\mathbf{w}$ are vectorizations of the $M$th order rank-1 tensors $\mathcal{H} = \mathbf{h}_1 \circ \ldots \circ \mathbf{h}_M$ and $\mathcal{W} = \mathbf{w}_1 \circ \ldots \circ \mathbf{w}_M$, respectively. The input regression vector can be regarded as the vectorization of $\mathcal{X}(n) \coloneqq \Theta[\mathbf{x}(n)] \in \mathbb{C}^{\times_{m=1}^{M} N_m}$. According to Proposition 2, equations (2.20) and (2.22) become

$$d(n) = \mathcal{X}(n) \times_1 \mathbf{h}_1^{\mathsf{H}} \ldots \times_M \mathbf{h}_M^{\mathsf{H}}$$

and

$$y(n) = \mathcal{X}(n) \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}},$$

respectively. These filtering operations are multilinear with respect to each subsystem (filter) as shown in Proposition 3. In view of this, equations (2.20) and (2.22) are hereafter referred to as *multilinear filtering*.

**Definition 15** (Multilinear filtering)**.** *Consider an $M$th order input tensor $\mathcal{X}(n) \in \mathbb{C}^{\times_{m=1}^{M} N_m}$ and a rank-1 tensor filter $\mathcal{W} = \mathbf{w}_1 \circ \mathbf{w}_2 \circ \ldots \circ \mathbf{w}_M$ with same dimensions.*

*This filter produces a complex valued output signal*

$$\begin{aligned} y(n) &= \langle \mathcal{X}(n), \mathcal{W} \rangle \\ &= \mathcal{X}(n) \times_1 \mathbf{w}_1^{\mathsf{H}} \times_2 \mathbf{w}_2^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}. \end{aligned} \tag{2.23}$$

The supervised multilinear filtering approach to system identification consists of approximating a distorted desired signal $d(n) = \tilde{d}(n) + b(n)$ using a rank-1 tensor filter $\mathcal{W}$, where $b(n)$ is a zero mean and unit variance additive white Gaussian measurement noise component uncorrelated to $\tilde{d}(n)$ and $x(n)$. The supervised multilinear filtering can be expressed as the following optimization problem:

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_M} \mathbb{E}\left[\left|d(n) - \mathcal{X}(n) \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}\right|^2\right]. \tag{2.24}$$

The multilinear system identification model is illustrated in Figure 2.2. Note that when $M = 1$ (not separable case), the supervised multilinear filtering problem reduces to its linear counterpart.

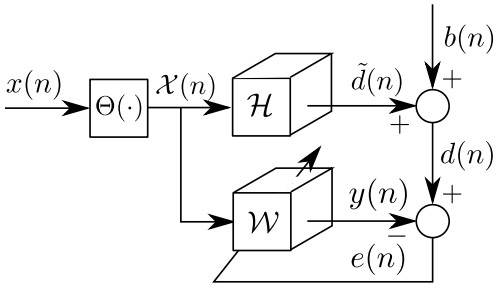



Figure 2.2: Supervised multilinear system identification model.

We have remarked a link between Kronecker products and cascade filters, and we will show that Kronecker factorization leads to a system factorization in terms of sparse subsystems, which we refer to as *extended* subsystems. We propose in the following the Extended Subsystem Factorization Theorem:

**Theorem 1** (Extended Subsystem Factorization)**.** *Any $M$th order separable FIR system whose coefficients vector decomposes into $\mathbf{h} = \bigotimes_{m=M}^{1} \mathbf{h}_m$ can be expressed as*

$$\mathbf{h} = \tilde{\mathbf{h}}_1 * \tilde{\mathbf{h}}_2 * \ldots * \tilde{\mathbf{h}}_M, \tag{2.25}$$

*where "$*$" denotes discrete-time convolution, $\tilde{\mathbf{h}}_m \in \mathbb{C}^{(N_m-1)\Psi_m+1}$ denotes the $m$th order extended subsystem, and $\Psi_m = \prod_{v=1}^{m-1} N_v$. The coefficients of the extended*

*subsystems are given by*

$$[\tilde{\mathbf{h}}_m]_{n_m} = \begin{cases} [\mathbf{h}_m]_{(n_m-1)/\Psi_m} & , \textit{ if } (n_m-1) \textit{ mod } \Psi_m = 0 \\ 0 & , \textit{ otherwise} \end{cases}, \tag{2.26}$$

*for* $n_m = 1, \ldots, N_m$.

*Proof.* From Definition 14, it follows that the $(n+1)$th element of $\mathbf{h}$ can be written as:

$$[\mathbf{h}]_{n+1} = [\mathbf{h}_1]_{n_1+1}[\mathbf{h}_2]_{n_2+1} \ldots [\mathbf{h}_M]_{n_M+1}, \tag{2.27}$$

where $n = n_1 + \sum_{m=2}^{M} n_m \prod_{v=1}^{m-1} N_v$ for $n_m \in \{0, \ldots, N_m - 1\}$ and $m = 1, \ldots, M$. Let us study the separability effect on the Z-transform of $\mathbf{h}$, which is denoted by $H(z)$:

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} [\mathbf{h}]_{n+1} z^{-n} \\ &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \cdots \sum_{n_m=0}^{N_M-1} [\mathbf{h}_1]_{n_1+1}[\mathbf{h}_2]_{n_2+1} \ldots [\mathbf{h}_M]_{n_M+1} z^{-\left(n_1+\sum_{m=2}^{M} n_m \prod_{v=1}^{m-1} N_v\right)} \\ &= \left(\sum_{n_1=0}^{N_1-1} [\mathbf{h}_1]_{n_1+1} z^{-n_1}\right)\left(\sum_{n_2=0}^{N_2-1} [\mathbf{h}_2]_{n_2+1} z^{-n_2 N_1}\right) \ldots \left(\sum_{n_M=0}^{N_M-1} [\mathbf{h}_M]_{n_M+1} z^{-n_M \prod_{v=1}^{M-1} N_v}\right) \\ &= \tilde{H}_1(z)\tilde{H}_2(z) \ldots \tilde{H}_M(z). \end{aligned} \tag{2.28}$$

The term $\tilde{H}_m(z)$ denotes the Z-transform of the $m$th extended subfilter. It can be expressed as:

$$\tilde{H}_m(z) = [\mathbf{h}_m]_1 + [\mathbf{h}_m]_2 z^{-\Psi_m} + [\mathbf{h}_m]_3 z^{-2\Psi_m} + \ldots + [\mathbf{h}_m]_{N_m} z^{-(N_m-1)\Psi_m}. \tag{2.29}$$

Note that the elements of $\tilde{H}_m(z)$ are nonzero when the exponent of $z$ is a multiple of $\Psi_m$. In view of this, the inverse Z-transform of (2.29) is given by:

$$[\tilde{\mathbf{h}}_m]_{n_m} = \begin{cases} [\mathbf{h}_m]_{(n_m-1)/\Psi_m} & , \text{ if } (n_m-1) \text{ mod } \Psi_m = 0 \\ 0 & , \text{ otherwise} \end{cases}, \tag{2.30}$$

for $n_m = 1, \ldots, N_m$. Considering (2.30), the inverse Z-transform of (2.28) is given by $\mathbf{h} = \tilde{\mathbf{h}}_1 * \tilde{\mathbf{h}}_2 * \ldots * \tilde{\mathbf{h}}_M$. □

To illustrate this result, let us consider the following example. Let $\mathbf{h} = \mathbf{h}_1 \otimes \mathbf{h}_2 \otimes \mathbf{h}_3 \in \mathbb{C}^{I_1 I_2 I_3}$, $\mathbf{h}_1 \in \mathbb{C}^{I_1}$, $\mathbf{h}_2 \in \mathbb{C}^{I_2}$, and $\mathbf{h}_3 \in \mathbb{C}^{I_3}$, where $I_1 = 2$, $I_2 = 3$,

and $I_3 = 4$. According to Theorem 1, $\mathbf{h}$ can be convolutionaly decomposed into $\mathbf{h} = \tilde{\mathbf{h}}_1 * \tilde{\mathbf{h}}_2 * \tilde{\mathbf{h}}_3$, where

$$\tilde{\mathbf{h}}_1 = \left[ h_1^{(1)} \; h_2^{(1)} \right]_{1\times 2}^{\mathsf{T}},$$
$$\tilde{\mathbf{h}}_2 = \left[ \begin{array}{ccccc} h_1^{(2)} & 0 & h_2^{(2)} & 0 & h_3^{(2)} \end{array} \right]_{1\times 5}^{\mathsf{T}},$$
$$\tilde{\mathbf{h}}_3 = \left[ \begin{array}{cccccccccccccccccc} h_1^{(3)} & 0 & 0 & 0 & 0 & 0 & h_2^{(3)} & 0 & 0 & 0 & 0 & 0 & h_3^{(3)} & 0 & 0 & 0 & 0 & 0 & h_4^{(3)} \end{array} \right]_{1\times 19}^{\mathsf{T}}.$$

and $h_n^{(m)} \coloneqq [\mathbf{h}_m]_n$. This factorization is depicted in Figure 2.3. We notice that the extended subsystems $\tilde{\mathbf{h}}_m$ are oversampled/sparse versions of their correspondent subfilter $\mathbf{h}_m$.

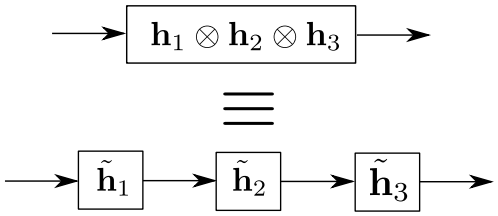



Figure 2.3: Extended subsystem factorization example.

In the next Section, we will introduce a method for estimating the extended subfilters using WH filters.

### 2.2.3 Separable Wiener-Hopf Algorithm

The Separable Wiener-Hopf (SepWH) algorithm consists of dividing the multilinear filtering problem (2.24) into $M$ *independent* linear subproblems. This division is carried out by factorizing the unknown FIR system $\mathbf{h}$ in terms of its extended subsystems $\tilde{\mathbf{h}}_m$ and individually identifying them using $\tilde{\mathbf{w}}_m$, the WH filter presented in Appendix A. Afterwards, the system identification is completed by computing $\mathbf{w} = \tilde{\mathbf{w}}_1 * \ldots * \tilde{\mathbf{w}}_M$. Since the solution of a linear subproblem does not depend on the other solutions, they are said to be independent.

The linear subproblem corresponding to the $m$th extended subfilter is given by

$$\min_{\tilde{\mathbf{w}}_m} \mathbb{E}\left[ \left| d(n) - \tilde{\mathbf{w}}_m^{\mathsf{H}} \tilde{\mathbf{x}}_m(n) \right|^2 \right], \quad m \in \{1, \ldots, M\}, \tag{2.31}$$

where it is supposed that $\tilde{\mathbf{x}}_m(n) \coloneqq \mathbf{x}(n) * \tilde{\mathbf{h}}_{\neq m}$, the prefiltered regression vector, is observable, and $\tilde{\mathbf{h}}_{\neq m} \coloneqq *_{\substack{k=1 \\ k\neq m}}^{M} \tilde{\mathbf{h}}_k$. Notice that $\tilde{\mathbf{h}}_{\neq m} * \tilde{\mathbf{h}}_m = \mathbf{h} \; \forall m \in \{1, \ldots, M\}$ due to the convolution commutative property. The filtering operation (2.31) is

illustrated in Figure 2.4. The SepWH has a computational complexity given by $\mathcal{O}\left[((N_M-1)\Psi_M+1)^3\right]$ since it inverts a $[(N_M-1)\Psi_M+1\times(N_M-1)\Psi_M+1]$-dimensional autocorrelation matrix.

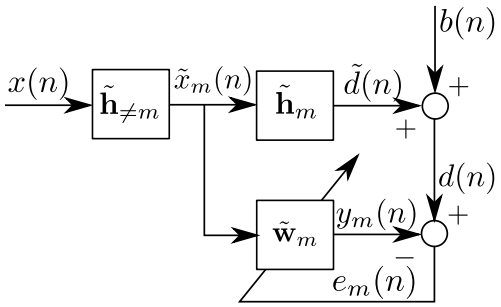



Figure 2.4: Separable Wiener-Hopf algorithm structure.

Let us illustrate the SepWH algorithm structure by using an example. Consider a third-order separable FIR system whose coefficients vector is given by $\mathbf{h}=\mathbf{h}_3\otimes\mathbf{h}_2\otimes\mathbf{h}_1\in\mathbb{C}^{I_1I_2I_3}$, where $\mathbf{h}_1\in\mathbb{C}^{I_1}$, $\mathbf{h}_2\in\mathbb{C}^{I_2}$, $\mathbf{h}_3\in\mathbb{C}^{I_3}$, $I_1=2$, $I_2=3$, and $I_3=4$. The estimation of $\tilde{\mathbf{w}}_1$, $\tilde{\mathbf{w}}_2$, and $\tilde{\mathbf{w}}_3$ is illustrated in Figures 2.5. The SepWH algorithm is summarized in Algorithm 1.

**Algorithm 1** Separable Wiener-Hopf algorithm

**procedure** SEPWH($\mathbf{x}(n)$, $\mathbf{h}_1,\ldots,\mathbf{h}_M$)
  $\mathbf{h}\leftarrow\mathbf{h}_1*\mathbf{h}_2*\ldots*\mathbf{h}_M$
  Calculate $\tilde{\mathbf{h}}_m$ using Eq. (2.26) for $m=1,\ldots,M$
  **for** $m=1,\ldots,M$ **do**
    **for** $t=0,\ldots,T-1$ **do**
      $\tilde{\mathbf{h}}_{\neq m}\leftarrow *_{\substack{k=1\\k\neq m}}^{M}\tilde{\mathbf{h}}_k$
      $\tilde{\mathbf{x}}_m(n-t)\leftarrow\mathbf{x}(n-t)*\mathbf{h}_{\neq m}$
      $d(n-t)\leftarrow\mathbf{h}^{\mathsf{H}}\tilde{\mathbf{x}}(n)+b(n-t)$
      $\mathbf{R}_{\tilde{\mathbf{x}}}\leftarrow(1/T)\mathbf{x}(n-t)\mathbf{x}(n-t)^{\mathsf{H}}$
      $\mathbf{p}_{d\tilde{\mathbf{x}}}\leftarrow(1/T)d^*(n-t)\mathbf{x}(n-t)$
      $\tilde{\mathbf{w}}_m\leftarrow\mathbf{R}_{\tilde{\mathbf{x}}}^{-1}\mathbf{p}_{d\tilde{\mathbf{x}}}$
    **end for**
  **end for**
  $\mathbf{w}\leftarrow\tilde{\mathbf{w}}_1*\tilde{\mathbf{w}}_2*\ldots\tilde{\mathbf{w}}_M$
**end procedure**

### 2.2.4 Tensor Wiener-Hopf Algorithm

Standard optimization methods do not guarantee global convergence when minimizing (2.24) due to its nonconvexity with respect to all the variables. The

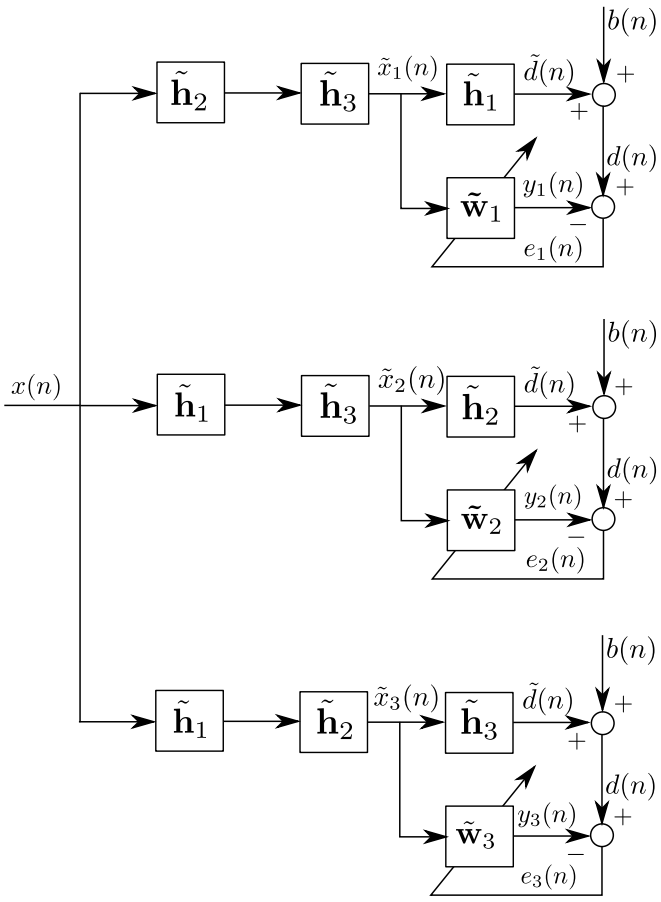



Figure 2.5: Identification of each extended subsystem of $\mathbf{h}$ using the SepWH algorithm.

alternating optimization approach [26, 33] has demonstrated to solve the global nonlinear problem in terms of $M$ smaller linear problems. It consists of updating the $m$th mode subfilter each time by solving (2.24) for $\mathbf{w}_m$, while $\{\mathbf{w}_j\}_{j=1,j\neq m}^{M}$ remain fixed, $m = 1, \ldots, M$, conditioned on the previous updates of the other subfilters. In this sense, (2.24) can be divided in $M$ *interdependent* linear subproblems using the $m$-mode unfolding equation (2.11). This allows us to represent the multilinear filtering (2.23) in terms of $M$ different but equivalent linear filtering:

$$\begin{aligned} y(n) &= \mathbf{w}_1^{\mathsf{H}} \mathbf{X}_{(1)}(n) \left(\mathbf{w}_M \otimes \mathbf{w}_{M-1} \otimes \ldots \otimes \mathbf{w}_2\right)^* && = \mathbf{w}_1^{\mathsf{H}} \mathbf{u}_1(n), \\ &= \mathbf{w}_2^{\mathsf{H}} \mathbf{X}_{(2)}(n) \left(\mathbf{w}_M \otimes \mathbf{w}_{M-1} \otimes \ldots \otimes \mathbf{w}_1\right)^* && = \mathbf{w}_2^{\mathsf{H}} \mathbf{u}_2(n), \\ &\qquad\qquad \vdots && = \quad \vdots \\ &= \mathbf{w}_M^{\mathsf{H}} \mathbf{X}_{(M)}(n) \left(\mathbf{w}_{M-1} \otimes \mathbf{w}_{M-2} \otimes \ldots \otimes \mathbf{w}_1\right)^* && = \mathbf{w}_M^{\mathsf{H}} \mathbf{u}_M(n), \end{aligned}$$

where

$$\mathbf{u}_m(n) = \mathbf{X}_{(m)}(n) \mathbf{w}^{\otimes m} \in \mathbb{C}^{N_m},\ m \in \{1, \ldots, M\}. \tag{2.32}$$

Substituting the equations above in (2.24) gives

$$\min_{\mathbf{w}_m} \mathbb{E}\left[\left|d(n) - \mathbf{w}_m^{\mathsf{H}} \mathbf{u}_m(n)\right|^2\right], \quad m \in \{1, \ldots, M\}. \tag{2.33}$$

The solution of (2.33) is given by the Wiener-Hopf filter (A.3):

$$\mathbf{w}_{\text{opt}}^m = \mathbf{R}_m^{-1} \mathbf{p}_m, \quad m \in \{1, \ldots, M\}, \tag{2.34}$$

where

$$\begin{aligned} \mathbf{R}_m &\coloneqq \mathbb{E}\left[\mathbf{u}_m(n) \mathbf{u}_m(n)^{\mathsf{H}}\right] \\ &= \mathbb{E}\left[\mathbf{X}_{(m)}(n) \mathbf{w}^{\otimes m}\ \mathbf{w}^{\otimes m\,\mathsf{H}}\ \mathbf{X}_{(m)}(n)^{\mathsf{H}}\right] \end{aligned} \tag{2.35}$$

and

$$\begin{aligned} \mathbf{p}_m &\coloneqq \mathbb{E}\left[\mathbf{u}_m(n) d^*(n)\right] \\ &= \mathbb{E}\left[\mathbf{X}_{(m)}(n) \mathbf{w}^{\otimes m} d^*(n)\right] \end{aligned} \tag{2.36}$$

for $m = 1, \ldots, M$. The term $\mathbf{w}^{\otimes m}$ in $\mathbf{R}_m$ and $\mathbf{p}_m$ indicates that eqs. (2.34) are *interdependent* in the sense that the computation of $\mathbf{w}_{\text{opt}}^m$ depends on $\{\mathbf{w}_j\}_{j=1,j\neq m}^{M}$, motivating the alternating optimization approach.

We propose the TWH algorithm, which is based on the alternating least squares (ALS) method to estimate $\mathbf{w}_{\text{opt}}^m$ by data averaging in the time interval $T$. Let

$$\mathcal{X} = [\mathcal{X}(n) \sqcup_{M+1} \ldots \sqcup_{M+1} \mathcal{X}(n-T+1)] \in \mathbb{C}^{N_1 \times \ldots \times N_M \times T}$$

denote the concatenation of $T$ time snapshots of $\mathcal{X}(n)$ along the $(M+1)$th dimension. Let $\mathbf{U}^{(m)} \in \mathbb{C}^{N_m \times T}$ denote the $\{1, \ldots, m-1, m+1, \ldots, N\}$-mode product between $\mathcal{X}$ and $\{\mathbf{w}_j^{\mathsf{H}}\}_{j=1, j\neq m}^{M}$ for $m = 1, \ldots, M$:

$$\mathbf{U}^{(m)} = \mathcal{X} \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_{m-1} \mathbf{w}_{m-1}^{\mathsf{H}} \times_{m+1} \mathbf{w}_{m+1}^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}. \tag{2.37}$$

From Proposition 4, it follows that $\mathbf{U}^{(m)} = [\mathbf{u}_m(n), \ldots, \mathbf{u}_m(n-T+1)]$. Therefore the sample estimates of $\mathbf{R}_m$ and $\mathbf{p}_m$ are given by:

$$\hat{\mathbf{R}}_m = \frac{1}{T}\mathbf{U}^{(m)}\mathbf{U}^{(m)^{\mathsf{H}}}, \tag{2.38}$$

$$\hat{\mathbf{p}}_m = \frac{1}{T}\mathbf{U}^{(m)}\mathbf{d}^*, \tag{2.39}$$

where $\mathbf{d} = [d(n), d(n-1), \ldots, d(n-T+1)]^{\mathsf{T}} \in \mathbb{C}^T$. Now the $m$th order subfilter updating rule (2.34) becomes $\hat{\mathbf{w}}_{\text{opt}}^m = \hat{\mathbf{R}}_m^{-1}\hat{\mathbf{p}}_m$. At the end of the subfilter estimation stage, the global filter is computed as $\hat{\mathbf{w}} = \bigotimes_{m=M}^{1} \hat{\mathbf{w}}_m$. The subfilters are estimated in an alternate fashion until convergence, which is attained when the relative system mismatch, defined as $\|\mathbf{h} - \hat{\mathbf{w}}\|_2^2 / \|\mathbf{h}\|_2^2$, between two consecutive iterations is smaller than a threshold $\varepsilon$. This procedure is summarized in Algorithm 2.

The TWH algorithm has a computational complexity of $\mathcal{O}\left(\sum_{m=1}^{M} N_m^3\right)$. Such an alternating minimization procedure has a monotonic convergence [34]. It is worth mentioning that an analytical convergence analysis of this algorithm is a challenging research topic which is under investigation.

### 2.2.5 Tensor LMS Algorithm [16]

As in the linear filtering scenario, the optimization problems (2.24) can be solved using a stochastic gradient approach. The gradient of the instantaneous objective function $J_{\mathbf{w}_m}(n) = \mathbb{E}\left[\left|d(n) - \mathbf{w}_m^{\mathsf{H}}\mathbf{u}_m(n)\right|^2\right]$ corresponding to the $m$th filtering mode is given by

$$\hat{\nabla} J_{\mathbf{w}_m}(n) = -\mathbf{u}_m(n)\mathbf{u}_m(n)^{\mathsf{H}}\mathbf{w}_m + \mathbf{u}_m(n)d^*(n). \tag{2.40}$$

**Algorithm 2** Tensor Wiener-Hopf algorithm

**procedure** TWH($\mathcal{X}$, $\mathbf{s}$, $\varepsilon$)
  $k \leftarrow 1$
  Initialize $e(k)$, $\mathbf{w}_m(k)$, $m = 1, \ldots, M$.
  **repeat**
    **for** $m = 1, \ldots, M$ **do**
      Calculate $\mathbf{U}^{(m)}(k)$ using Equation (2.37)
      $\hat{\mathbf{R}}_m \leftarrow (1/T)\mathbf{U}^{(m)}\mathbf{U}^{(m)^{\mathsf{H}}}$
      $\hat{\mathbf{p}}_m \leftarrow (1/T)\mathbf{U}^{(m)}\mathbf{s}^*$
      $\hat{\mathbf{w}}_m(k+1) \leftarrow \hat{\mathbf{R}}_m^{-1}\hat{\mathbf{p}}_m$
    **end for**
    $k \leftarrow k + 1$
    $\mathbf{y}(k) \leftarrow \mathcal{X} \times_1 \hat{\mathbf{w}}_1(k)^{\mathsf{H}} \ldots \times_M \hat{\mathbf{w}}_M(k)^{\mathsf{H}}$
    $e(k) = \|\mathbf{s} - \mathbf{y}(k)\|_2^2/T$
  **until** $|e(k) - e(k-1)| < \varepsilon$
**end procedure**

In [16], M. Rupp and S. Schwarz proposed the Tensor Least Mean Square (TLMS) update rule for the $m$th order filter using a variable step size $\mu(n)$:

$$\begin{aligned}
\mathbf{w}_m(n+1) &= \mathbf{w}_m(n) - \mu(n)\hat{\nabla} J_{\mathbf{w}_m}(n) \\
&= \mathbf{w}_m(n) - \mu(n)\left[-\mathbf{u}_m(n)\mathbf{u}_m(n)^{\mathsf{H}}\mathbf{w}_m + \mathbf{u}_m(n)d^*(n)\right] \\
&= \mathbf{w}_m(n) + \mu(n)\mathbf{u}_m(n)\left[\mathbf{w}_m^{\mathsf{H}}\mathbf{u}_m(n) - d(n)\right]^* \\
&= \mathbf{w}_m(n) + \mu(n)\mathbf{u}_m(n)e_m^*(n). \qquad (2.41)
\end{aligned}$$

The authors indicated that TLMS converges for $M = 2$ if

$$0 < \mu(n) < \frac{2}{\|\mathbf{u}_1(n)\|_2^2 + \|\mathbf{u}_2(n)\|_2^2}. \qquad (2.42)$$

In this work, we have used $\mu(n) = \frac{\tilde{\mu}}{\sum_{m=1}^{M}\|\mathbf{u}_m(n)\|_2^2}$ for some values of $\tilde{\mu}$ such that TLMS attained convergence. This algorithm could be named "Tensor Normalized Least Square" due to the step size normalization w.r.t. $\mathbf{u}_m(n)$, $m = 1, \ldots, M$, but we preferred to keep the original acronym. Each TLMS iteration, which consists of updating all $M$ subfilters, has computational complexity $\mathcal{O}\left(\sum_{m=1}^{M} N_m\right)$. Analytical study of convergence of the TLMS algorithm remains an open research problem. TLMS is summarized in Algorithm 3. In line 11, $\gamma \in \mathbb{R}^+$ is a constant chosen to avoid divergence when $\|\mathbf{u}_m(n-k)\|_2^2$ becomes negligible.

**Algorithm 3** Tensor Least Mean Square algorithm [16]

**procedure** TLMS($\mathbf{x}(n) \in \mathbb{C}^N$, $d(n) \in \mathbb{C}^T$, $\tilde{\mu} \in \mathbb{R}^+$)
  $k \leftarrow 1$
  Initialize $\mathbf{w}_m(k)$ for $m = 1, \ldots, M$.
  **for** $k = 0, \ldots, T-1$ **do**
    $\mathcal{X}(n-k) \leftarrow \Theta(\mathbf{x}(n-k))$
    $e(k) \leftarrow d(n-k) - \mathcal{X}(n-k) \times_1 \mathbf{w}_1(k)^{\mathsf{H}} \times \mathbf{w}_2(k)^{\mathsf{H}} \ldots \times_M \mathbf{w}_M(k)^{\mathsf{H}}$
    **for** $m = 1, \ldots, M$ **do**
      Calculate $\mathbf{u}_m(n-k)$ using Eq. (2.32)
    **end for**
    **for** $m = 1, \ldots, M$ **do**
      $\mathbf{w}(k+1) \leftarrow \mathbf{w}(k) + \frac{\tilde{\mu}}{\gamma + \|\mathbf{u}_m(n-k)\|_2^2} \mathbf{u}_m(n-k) e^*(k)$
    **end for**
  **end for**
**end procedure**

| Method | Computational complexity |
|---|---|
| SepWH | $\mathcal{O}\left[((N_M - 1)\Psi_M + 1)^3\right]$ |
| TWH | $\mathcal{O}(\sum_{m=1}^{M} N_m^3)$ |
| TLMS [16] | $\mathcal{O}(\sum_{m=1}^{M} N_m)$ |

Table 2.1: Computational complexity of the SepWH, TWH, and TLMS algorithms.

It is not straightforward to compute the exact amount of FLOPS demanded by the methods discussed in this chapter due to the matrix inversions, however their computational complexity is listed in Table 2.1. In the next chapters, we will show the results and analysis of the numerical experiments on the applications to system identification and antenna beamforming using the methods presented in this chapter.

# Chapter 3

# Supervised Identification of Multilinear Systems

In this chapter, computer simulations were conducted to assess the performance of the proposed filtering methods in the system identification context. The system model is described in Section 3.1, the obtained results are shown in Section 3.2, next they are discussed in Section 3.3.

## 3.1 System Model

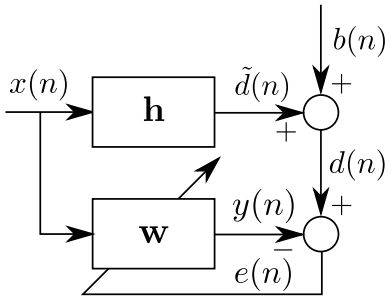



Figure 3.1: Supervised system identification model.

System identification is the process of estimating the model of a system by analyzing the relationship between the input and output signals. System identification techniques can be used to model, design, and predict the behavior of dynamical systems [35]. When designing complex dynamical systems, it is important to predict and simulate their behavior under diverse scenarios to predict failures. Modeling and estimating the dynamic characteristics of system components is a necessary stage prior to the system simulation and can be achieved

using system identification techniques.

Consider an unknown system modeled by an $N$th order real FIR system whose coefficients vector $\mathbf{h} \in \mathbb{R}^N$ is $M$th order separable, i.e.

$$\mathbf{h} = \mathbf{h}_M \otimes \mathbf{h}_{M-1} \otimes \ldots \otimes \mathbf{h}_1, \tag{3.1}$$

where $\mathbf{h}_m \in \mathbb{R}^{N_m}$ is the coefficients vector corresponding to the $m$th order subsystem for $m = 1, \ldots, M$, and $\prod_{m=1}^{M} N_m = N$. An $N$th order real FIR filter whose coefficients vector $\mathbf{w} = \mathbf{w}_M \otimes \mathbf{w}_{M-1} \otimes \ldots \otimes \mathbf{w}_1$ is $M$th order separable was placed parallel to the unknown system to identify its model, as illustrated in Figure 3.1. Both systems were excited by a zero mean and unit variance signal $x(n)$. The unknown system produced a distorted desired signal $d(n) = \tilde{d}(n) + b(n)$, where $\tilde{d}(n) \coloneqq \mathbf{h}^{\mathsf{T}}\mathbf{x}(n)$, $\mathbf{x}(n) \coloneqq [x(n), x(n-1), \ldots, x(n-N+1)]^{\mathsf{T}}$, and $b(n)$ denotes a zero mean and unit variance additive white Gaussian measurement noise component uncorrelated with $\tilde{d}(n)$ and $x(n)$.

## 3.2 Numerical Results

Monte Carlo (MC) simulations of $R = 100$ independent realizations were conducted to calculate the figures of merit of the NLMS, WH, SepWH, TLMS, and TWH methods. Tensors operations were computed using the Tensorlab toolbox [36]. Each figure of merit was expressed as a function of the input sample size $K$, the filter length $N$, and the SNR, which is defined as

$$\text{SNR} \coloneqq \frac{\mathbb{E}\left[|\tilde{d}(n)|^2\right]}{\mathbb{E}\left[|b(n)|^2\right]}.$$

In this chapter, the mean relative system mismatch

$$\rho \coloneqq \frac{1}{R}\sum_{r=1}^{R} \frac{\|\mathbf{h}^{(r)} - \mathbf{w}^{(r)}\|_2^2}{\|\mathbf{h}^{(r)}\|_2^2},$$

and the number of floating-points operations per second (FLOPS) demanded to attain convergence were chosen as figures of merit. The former allows to assess the system identification accuracy, whereas the latter provides the algorithm overall computational cost. The superscript $(\cdot)^{(r)}$ denotes the $r$th MC run. The number of FLOPS demanded by each algorithm was calculated using the Lightspeed toolbox [37].

The coefficients of all (sub)filters were randomly initialized in all iterative algorithms. The step size of the LMS-based algorithms was set to $\mu = 10^{-2}$. The autocorrelation matrix and the cross-correlation vector of the WH filter were estimated directly from data. Convergence in TWH was attained when the relative system mismatch difference between two consecutive ALS steps was smaller than $\varepsilon = 10^{-3}$. To check convergence, TLMS compares the relative system mismatch at the current iteration with that of 2000 past iterations. This is because the TLMS mean system mismatch difference in a few consecutive iterations may be too small due to slow convergence with a small step size factor.

In the first simulation scenario, the coefficients vector of the unknown system (3.1) was 3rd order separable ($M = 3$). Its subsystems $\mathbf{h}_1 \in \mathbb{R}^{N_1}$, $\mathbf{h}_2 \in \mathbb{R}^{N_2}$, and $\mathbf{h}_3 \in \mathbb{R}^{N_3}$ were randomly generated accordingly to a zero mean Gaussian distribution with identity covariance matrix and null mean vector. The influence of the input sample size $K$, the filter length $N$, and the SNR on the methods performance was assessed in three simulations where one parameter varied while the others remained fixed. In Figures 3.2, 3.3, and 3.4, the figures of merit are depicted as functions of $K$, $N$, and SNR, respectively.

Another simulation was conducted to analyze the effect of the step size factor on the TLMS performance and to compare it to the TWH algorithm. The obtained learning curves for a step size $\mu \in \{10^{-1}, 10^{-2}, 10^{-3}\}$ are depicted in Figure 3.6 and the corresponding results are shown at Table 3.3.

In the second simulation scenario, the influence of the separability order $M$ on the TWH method performance was studied. In order to consistently compare this influence, it was necessary to consider a system whose coefficients vector $\mathbf{h}$ was equivalently separable across different separability orders, i.e.

$$\mathbf{h} = \bigotimes_{m=M}^{1} \mathbf{h}_m = \bigotimes_{j=J}^{1} \mathbf{h}_j, \quad \forall M, J \in \mathbb{Z}^+. \tag{3.2}$$

Exponential systems fulfills the condition (3.2) and were considered in this scenario. More specifically, the coefficients of $\mathbf{h}$ were defined as

$$[\mathbf{h}]_n \coloneqq \exp\left[\frac{-0.1(n-1)}{N}\right] \quad \text{for } n = 1, \ldots, N \tag{3.3}$$

The reader is referred to Appendix B for more information on the exponential system separability.

The figures of merit were computed as functions of the sample size $K$, and

SNR for $M = 2, 3, \ldots, 6$ and plotted in Figures 3.7, and 3.10, respectively. When analyzing the effect of the system length, it was necessary to assure that the system length remained constant across different values of $M$, i.e. $\prod_{m=1}^{M} N_m = \prod_{j=1}^{J} N_j$, $\forall M, J \in \mathbb{Z}^+$. In view of this, two sets of subsystem dimensions were considered. In the first set (Table 3.1), the subsystem dimensions were chosen to be similar, whereas, in the second set (Table 3.2), there was a subfilter with a larger dimension. TWH performance is plotted as a function of the system length $N$ in Figures 3.9 and 3.8 for dimensions sets at Table 3.1 and Table 3.2, respectively.

| $N$ \ $M$ | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 128 | $8 \times 16$ | $8 \times 4 \times 4$ | $4 \times 4 \times 4 \times 2$ | $2 \times 2 \times 2 \times 4 \times 4$ | $2 \times 2 \times 2 \times 2 \times 2 \times 4$ |
| 256 | $16 \times 16$ | $8 \times 8 \times 4$ | $4 \times 4 \times 4 \times 4$ | $2 \times 2 \times 4 \times 4 \times 4$ | $2 \times 2 \times 2 \times 2 \times 4 \times 4$ |
| 512 | $16 \times 32$ | $8 \times 8 \times 8$ | $8 \times 4 \times 4 \times 4$ | $2 \times 4 \times 4 \times 4 \times 4$ | $2 \times 2 \times 2 \times 4 \times 4 \times 4$ |
| 1024 | $32 \times 32$ | $8 \times 8 \times 16$ | $8 \times 8 \times 4 \times 4$ | $2 \times 4 \times 4 \times 4 \times 8$ | $2 \times 2 \times 2 \times 4 \times 4 \times 8$ |

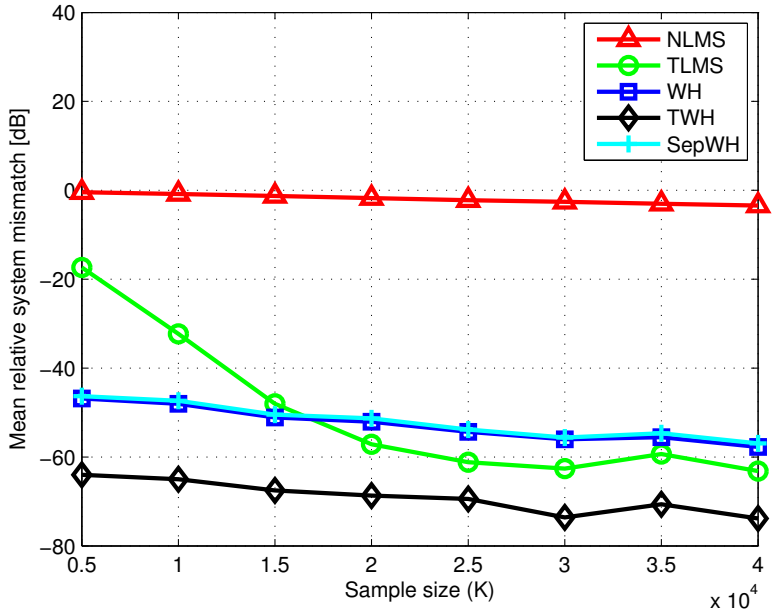
Table 3.1: Similar subsystem dimensions.

| $N$ \ $M$ | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 200 | $20 \times 20$ | $2 \times 10 \times 20$ | $2 \times 2 \times 10 \times 10$ | $2 \times 2 \times 2 \times 5 \times 10$ | $2 \times 2 \times 2 \times 2 \times 5 \times 5$ |
| 600 | $20 \times 30$ | $2 \times 10 \times 30$ | $2 \times 2 \times 10 \times 15$ | $2 \times 2 \times 2 \times 5 \times 15$ | $2 \times 2 \times 2 \times 3 \times 5 \times 5$ |
| 800 | $20 \times 40$ | $2 \times 10 \times 40$ | $2 \times 2 \times 10 \times 20$ | $2 \times 2 \times 2 \times 5 \times 20$ | $2 \times 2 \times 2 \times 4 \times 5 \times 5$ |
| 1000 | $20 \times 50$ | $2 \times 10 \times 50$ | $2 \times 2 \times 10 \times 25$ | $2 \times 2 \times 2 \times 5 \times 25$ | $2 \times 2 \times 2 \times 5 \times 5 \times 5$ |

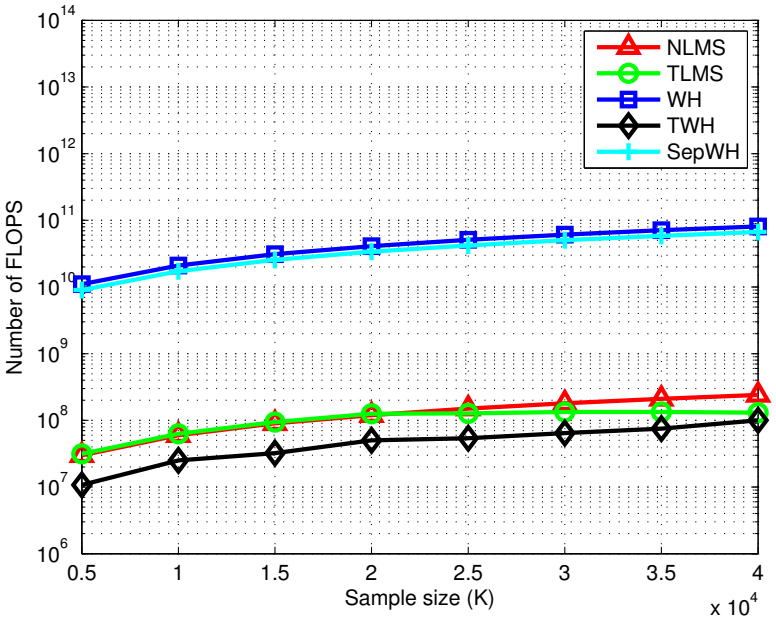
Table 3.2: Different subsystem dimensions.

| Method | Number of FLOPS | $\rho$ [dB] |
|---|---|---|
| TLMS ($\mu = 10^{-1}$) | $4.82 \times 10^7$ | $-49$ |
| TLMS ($\mu = 10^{-2}$) | $1.53 \times 10^8$ | $-59$ |
| TLMS ($\mu = 10^{-3}$) | $1.88 \times 10^9$ | $-70$ |
| WH | $4.01 \times 10^{11}$ | $-60$ |
| TWH | $4.30 \times 10^8$ | $-75$ |

Table 3.3: Number of FLOPS and mean relative system mismatch ($\rho$) for the simulation depicted in Figure 3.6.

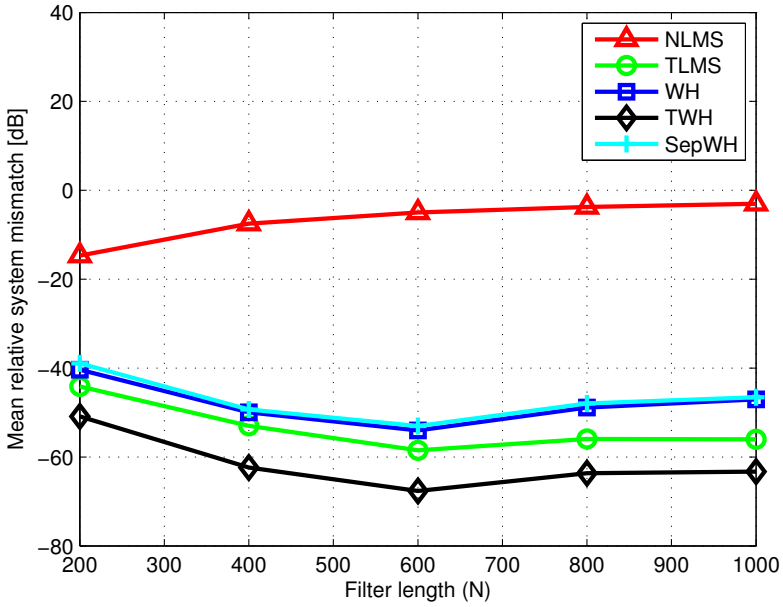



(a) Mean relative system mismatch vs. Sample size.

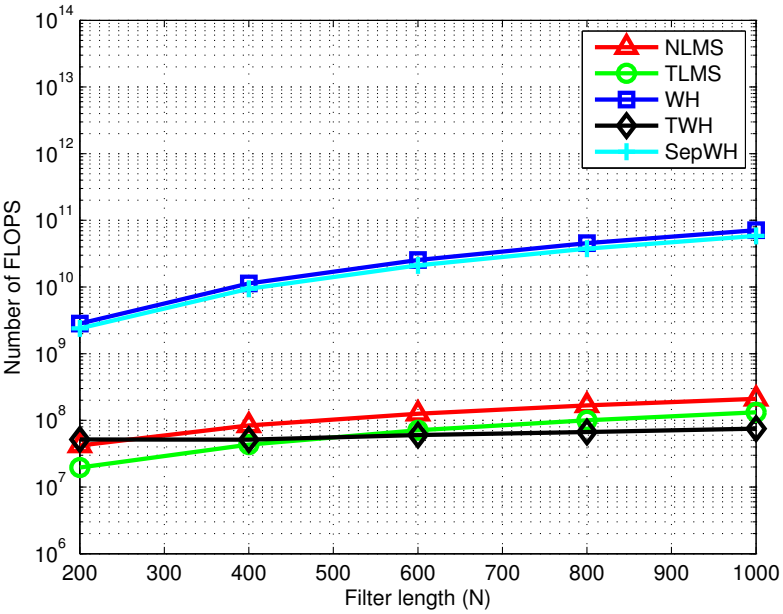



(b) Number of FLOPS vs. Sample size.

Figure 3.2: Filtering methods performance as a function of the sample size for $N = 1000$ and SNR $= 10$ dB. The step size of the adaptive filtering methods was set to $\mu = 10^{-2}$.

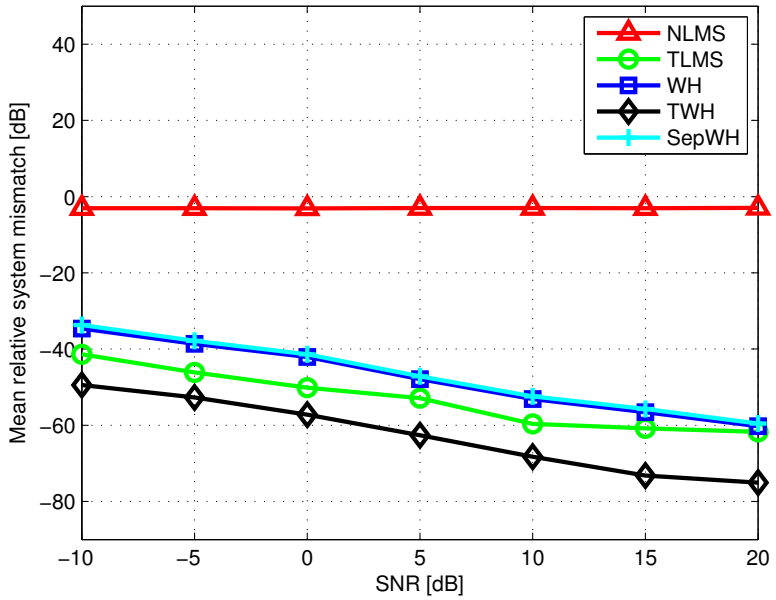



(a) Mean relative system mismatch vs. Filter length.

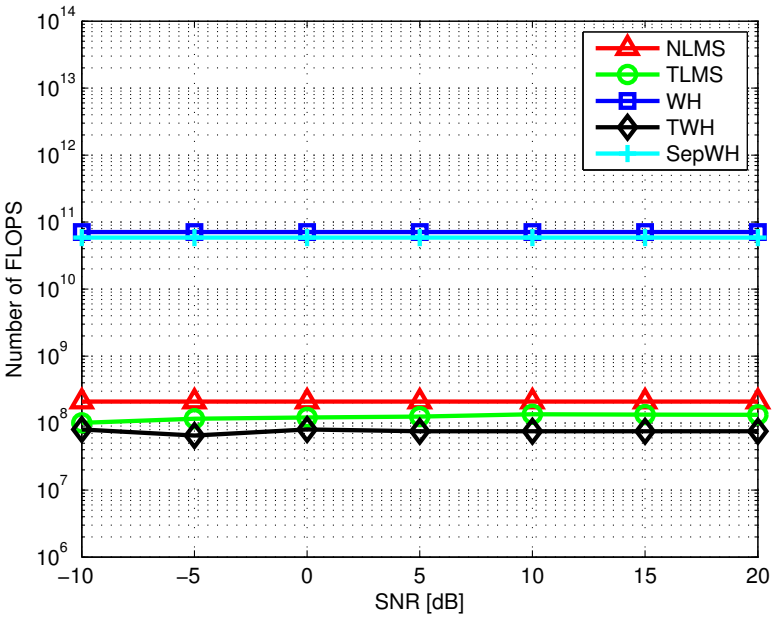



(b) Number of FLOPS vs. Filter length.

Figure 3.3: Filtering methods performance as a function of the filter length for $K = 35.000$ samples and SNR $= 10\,\text{dB}$. The step size of the adaptive filtering methods was set to $\mu = 10^{-2}$.




(a) Mean relative system mismatch vs. SNR.




(b) Number of FLOPS vs. SNR.

Figure 3.4: Filtering methods performance as a function of the SNR for $K = 35.000$ samples and $N = 1000$. The step size of the adaptive filtering methods was set to $\mu = 10^{-2}$.

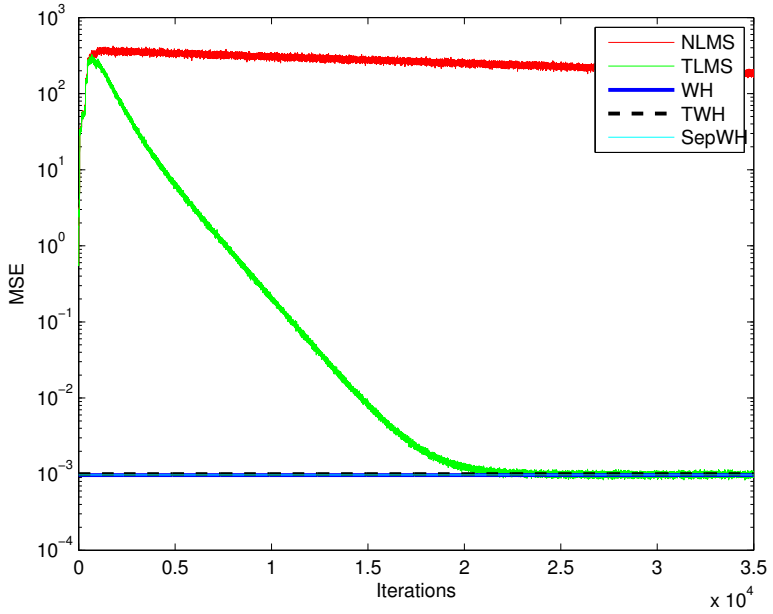



Figure 3.5: MSE performance for $\mu = 10^{-2}$, $K = 35.000$ samples, $N = 1000$, and SNR $= 30\,\mathrm{dB}$.

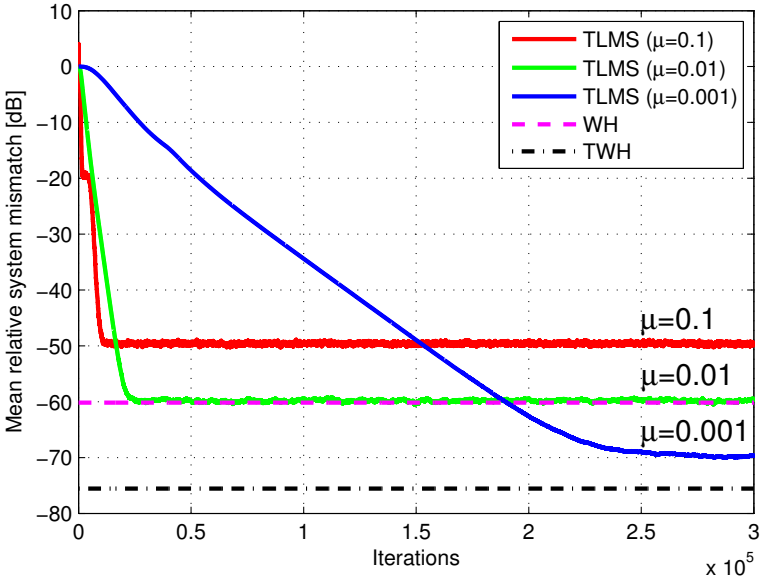
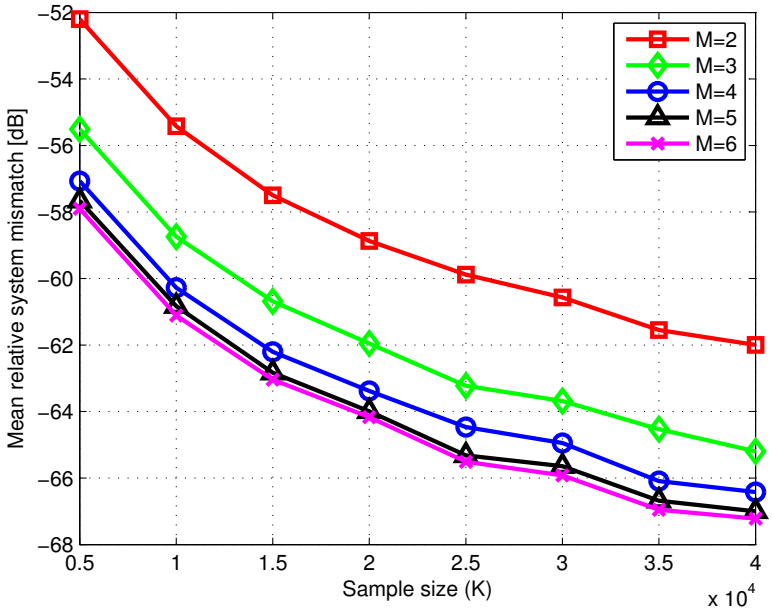
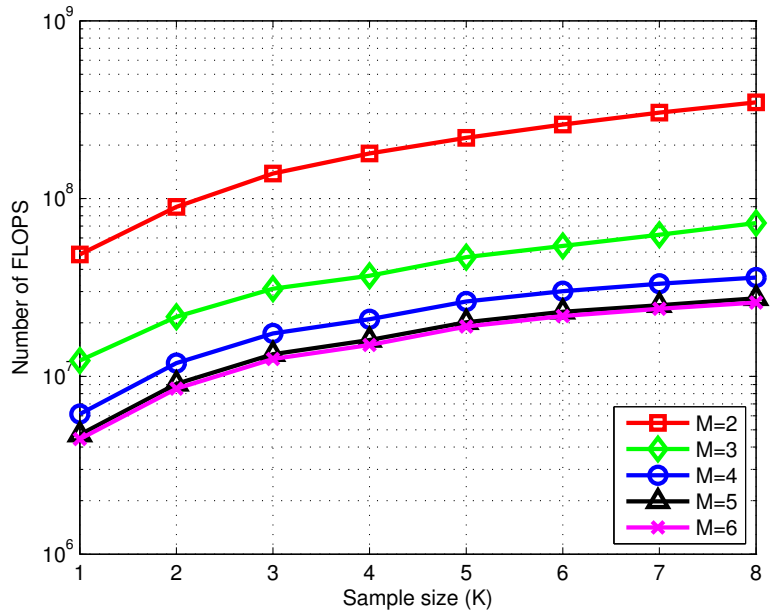



Figure 3.6: TLMS learning curves compared to the performance of the WH filter and TWH algorithm for $K = 3 \times 10^5$, $N = 1000$, and SNR$=10\,\mathrm{dB}$.

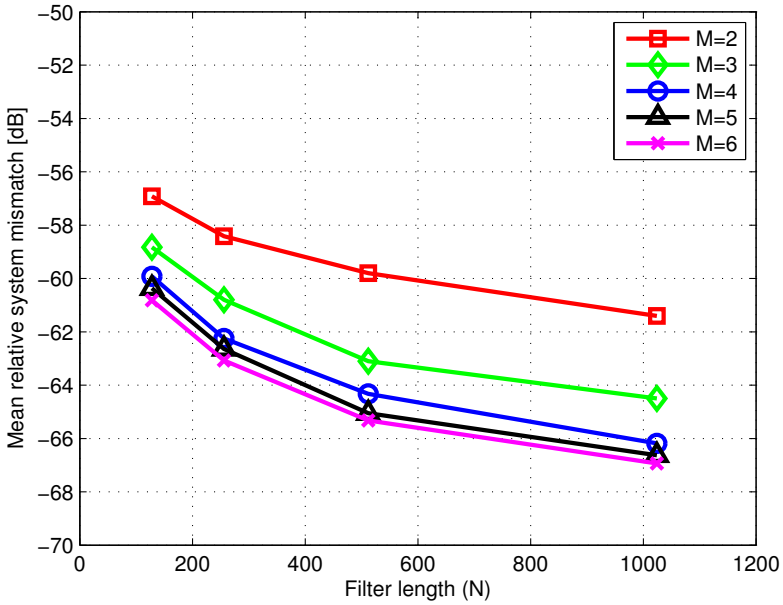



(a) Mean relative system mismatch vs. Sample size.

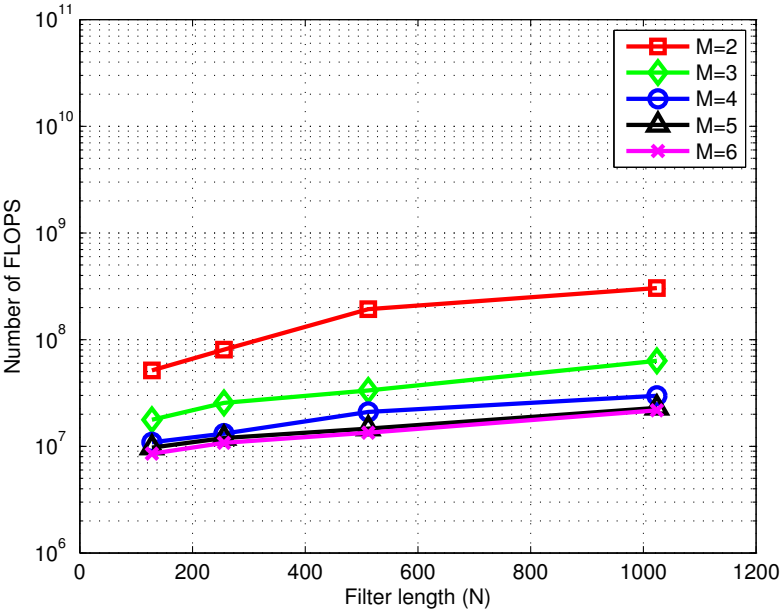



(b) Number of FLOPS vs. Sample size.

Figure 3.7: TWH performance for varying separability orders and sample size, $N = 1024$ and SNR $= 10\,\mathrm{dB}$.

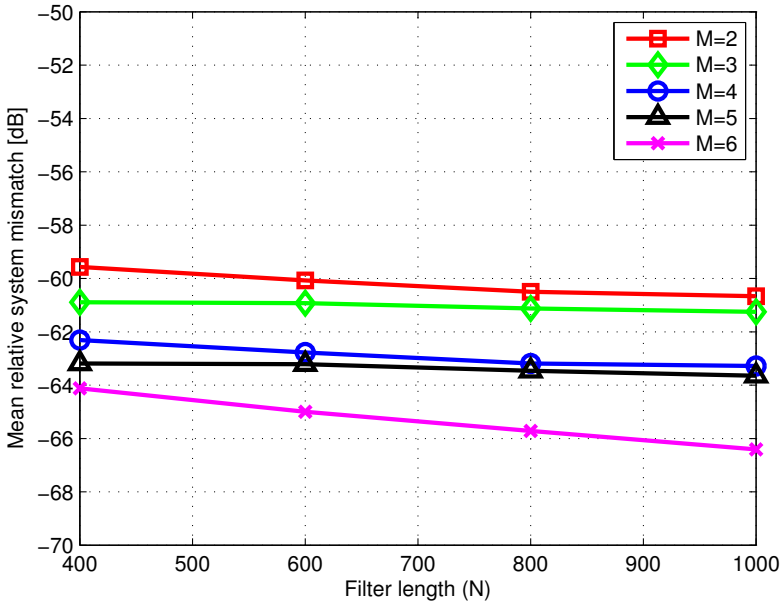



(a) Mean relative system mismatch vs. Filter length.

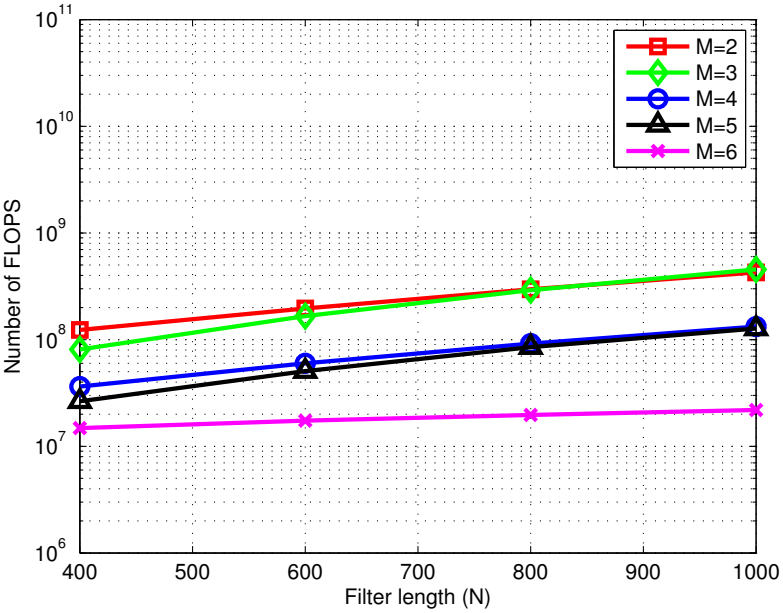



(b) Number of FLOPS vs. Filter length.

Figure 3.8: TWH performance for varying separability orders and filter lengths according to the dimensions at Table 3.1, $K = 35.000$ samples and SNR $= 10\,\mathrm{dB}$.

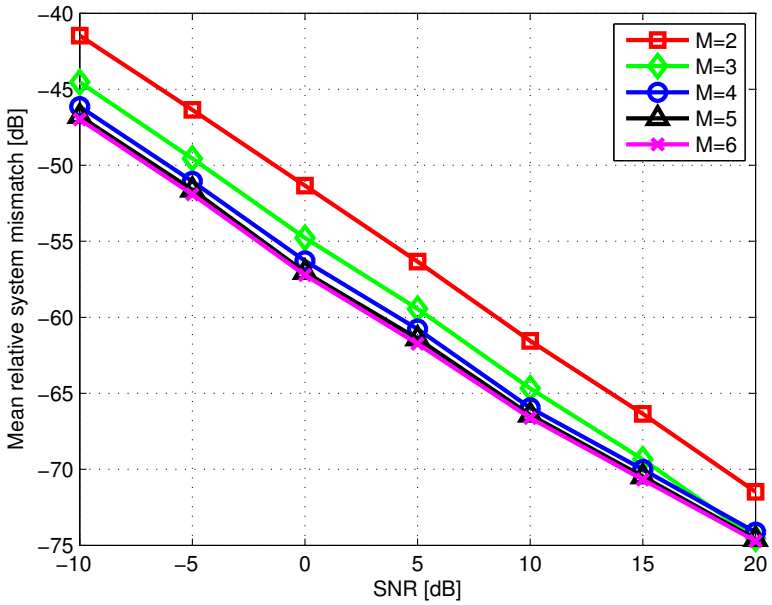



(a) Mean relative system mismatch vs. Filter length.

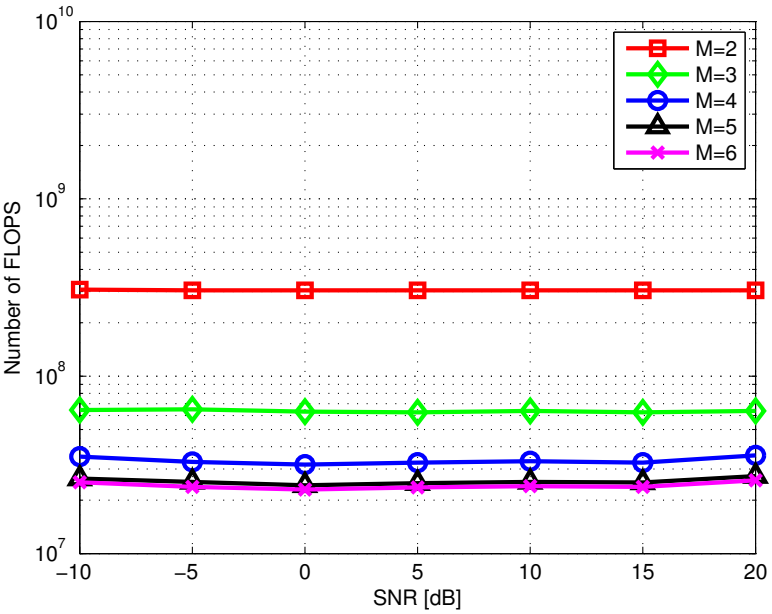



(b) Number of FLOPS vs. Filter length.

Figure 3.9: TWH performance for varying separability orders and filter lengths according to the dimensions at Table 3.2, $K = 35.000$ samples and SNR $= 10\,\mathrm{dB}$.




(a) Mean relative system mismatch vs. SNR.




(b) Number of FLOPS vs. SNR.

Figure 3.10: TWH performance for varying separability orders and SNR, $K = 35.000$ samples and $N = 1024$.

## 3.3 Discussion

The NLMS algorithm performed poorly in estimating the parameters of the unknown systems, as depicted in Figures 3.2a, 3.4a, and 3.3a. Since the parameter space was very large, much more iterations would be necessary to attain convergence. The effect of the filter length $N$ on NLMS performance is clear on Figure 3.3a. Smaller systems demanded less iterations, allowing the NLMS algorithm to achieve mismatch levels as low as $-17\,\mathrm{dB}$. Although this algorithm provides the worst parameter estimation accuracy, it has a small overall computational complexity.

Although the SepWH method exploited the system separability, it did not present any advantages with respect to the WH filter. As can be seen in Figures 3.2, 3.3, and 3.4, this method offered the same parameter estimation accuracy and computational complexity as its classic counterpart. In fact, the optimization stages in this method consists of *independent* least-squares subproblems, which do not incorporate the multilinear interdependence on their solution as TLMS and TWH do. Furthermore, the oversampling of the extended subsystems augmented their dimension, increasing the overall computational complexity and counterpoising the separability exploitation.

The TLMS algorithm provided better estimation accuracy than the methods discussed above in specific situations. As can be seen in Figure 3.2a, it started to be more accurate than the WH filter after the sample size was greater than 20.000. Since the alternating optimization approach employed by TLMS properly exploited the subfilter interdependency, it provided better accuracy than the WH filter. Figures 3.2b, 3.4b, and 3.3b showed that its computational complexity was similar to that of NLMS. The results shown in Figure 3.6 and Table 3.3 show that, to attain a system mismatch as low as $-70\,\mathrm{dB}$, it is necessary to use small step size factors. This restriction consequently demands a massive number of FLOPS from TLMS, whereas TWH requires less FLOPS to provide an even more accurate parameter estimation.

Our proposed solution, the TWH algorithm, provided the most accurate estimates with the lowest overall computational costs. Alike TLMS, the alternating optimization approach properly exploited system separability, providing very accurate estimates. However, since each ALS step computes multiple MMSE filter updates, our method moved faster towards the optimum solution. TWH usually converged within 5 ALS steps.Figure 3.2a shows that TWH had already attained $-60\,\mathrm{dB}$ for $K = 5000$ samples, while TLMS was still at $-20\,\mathrm{dB}$, since

it demanded more samples to achieve convergence. In Figure 3.3a, the TWH algorithm provided the best accuracy for all filter lengths.

In a first moment, it is surprising to see that any method performed better than the WH filter, which provides the MMSE estimate. However, it is important to stress that the results presented in Figures 3.2, 3.3, and 3.4 did not compare the MSE, but mean relative system mismatch. In fact, both linear and multilinear filtering methods are bounded by the MMSE, as observed in Figure 3.5. While 2nd order separable systems have a lower-bound on relative system mismatch estimation, as shown in [16], today there are similar no theoretical bounds for higher-order systems for the best of our knowledge.

It is interesting to remark that although TWH is a batch-based filtering method, it has an overall computational cost similar to that of adaptive filtering methods, as shown in Figures 3.2b, 3.3b, and 3.4b. The computational complexity of a TWH iteration is indeed superior than that of TLMS, as it can be seen in Table 2.1. However, TWH converges fastly, yielding a reduced overall computational complexity, whereas TLMS demands a large of amount of FLOPS to attain convergence in such large-scale scenarios.

The simulation in the second scenario showed that the separability order has an important influence on the multilinear filtering methods performance. Higher separability order leads to better accuracy and lower computational costs, as can be seen in Figures 3.7, 3.8, 3.9, and 3.10. Since the subfilters length decreases with $M$, the computational cost associated with the estimation of each subfilter is lower. Consequently, the overall computational cost reduces with $M$. Additionally, these figures show that there is a limit on the accuracy gain. Therefore, the system separability order offers an estimation accuracy / computational cost.

The results concerning the subfilter dimensions indicates that it determines the accuracy gain and the computational costs of multilinear filtering methods. When the system was partitioned into subsystems with similar dimensions, the performance depicted in Figure 3.8 was observed. By contrast, systems decomposed into subsystems whose dimensions differed from those of the other subsystems presented the performance described in Figure 3.9. We have observed that similar partitioning offered a $-2\,\mathrm{dB}$ gain for $M = 3, 4, 5$ with respect to the other partitioning, as can be seen in Figure 3.9. Furthermore, similar partitioning presented lower computational costs in the same $M$ interval.

We assumed in the conducted experiments that the unknown system impulse response was separable. In practice, however, there are systems that cannot be

represented as such. In this case, the proposed tensor filtering methods would fail to identify the unknown parameters. To address this problem, low-rank tensor models such as PARAFAC [14] could be used to approximate the non-separable system impulse as a finite sum of separable terms, allowing us to employ the proposed methods.

# Chapter 4

# Beamforming for Multilinear Antenna Arrays

In this chapter, we present the results obtained from computer simulations conducted to assess the performance of the TWH algorithm. The proposed system model was first introduced in [38], and it is described in Section 4.1. The obtained results from the simulations are shown in Section 4.2, next they are discussed in Section 4.3.

## 4.1 System Model

An antenna array consists of multiple sensors placed in different locations but physically connected that are used to process the impinging signals using a spatial filter. This filter is a *beamformer* when it is employed to enhance a signal of interest (SOI) arriving from a certain direction while attenuating any possible interfering signal [39].

Let us consider a translation invariant array of $N$ isotropic antennas located at $\tilde{\mathbf{p}}_n \in \mathbb{R}^{3\times 1}$ for $n = 1, \ldots, N$. It is formed by $N_1$ reference sensors located at $\mathbf{p}_{n_1}^{(1)}$ for $n_1 = 1, \ldots, N_1$. The $n_1$th reference sensor is translated $M-1$ times by means of the translation vectors $\mathbf{p}_{n_2}^{(2)}, \ldots, \mathbf{p}_{n_M}^{(M)}$, yielding the following decomposition for the $n$th global sensor location vector [40]

$$\tilde{\mathbf{p}}_n = \mathbf{p}_{n_1}^{(1)} + \mathbf{p}_{n_2}^{(2)} + \ldots + \mathbf{p}_{n_M}^{(M)}, \tag{4.1}$$

where $n = n_1 + \sum_{\mu=2}^{M}(n_\mu - 1)\prod_{v=1}^{\mu-1} N_v$, $n_\mu \in \{1, \ldots, N_\mu\}$. Figure 4.1 illustrates an example of a translation invariant array.

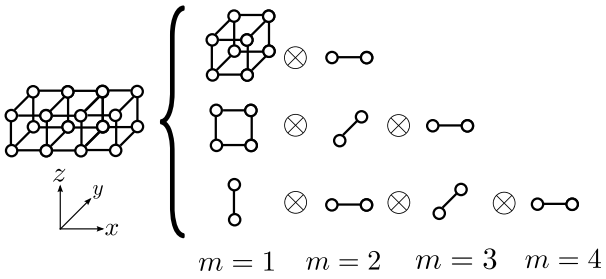



Figure 4.1: A $4 \times 2 \times 2$ volumetric array decomposed into three equivalent forms. Reference subarrays are indexed by $m = 1$, whereas $m > 1$ refers to translation.

Now consider that $R$ narrowband source signals with complex amplitudes $s_r(k)$ impinge on the global array from directions

$$\mathbf{d}_r = [\sin\theta_r \cos\phi_r, \sin\theta_r \sin\phi_r, \cos\theta_r]^{\mathsf{T}},$$

where $\theta_r$ and $\phi_r$ denote the elevation and azimuth angles, respectively, for $r = 1, \ldots, R$. The sources are assumed to be in the far-field and uncorrelated. It is also assumed that there are no reflection components. The received signals at instant $k$ can be written as:

$$\mathbf{x}(k) = \sum_{r=1}^{R} \mathbf{a}(\mathbf{d}_r) s_r(k) + \mathbf{b}(k) \in \mathbb{C}^{N \times 1}, \tag{4.2}$$

where

$$\mathbf{a}(\mathbf{d}_r) = \begin{bmatrix} e^{\jmath \frac{\omega}{c} \tilde{\mathbf{p}}_1^{\mathsf{T}} \mathbf{d}_r} \\ \vdots \\ e^{\jmath \frac{\omega}{c} \tilde{\mathbf{p}}_N^{\mathsf{T}} \mathbf{d}_r} \end{bmatrix} = \begin{bmatrix} e^{\jmath \frac{\omega}{c} {\mathbf{p}_1^{(1)}}^{\mathsf{T}} \mathbf{d}_r} \ldots e^{\jmath \frac{\omega}{c} {\mathbf{p}_1^{(M)}}^{\mathsf{T}} \mathbf{d}_r} \\ \vdots \\ e^{\jmath \frac{\omega}{c} {\mathbf{p}_{N_1}^{(1)}}^{\mathsf{T}} \mathbf{d}_r} \ldots e^{\jmath \frac{\omega}{c} {\mathbf{p}_{N_M}^{(M)}}^{\mathsf{T}} \mathbf{d}_r} \end{bmatrix} \tag{4.3}$$

is the array steering separable vector, $\mathbf{b}(k) \in \mathbb{C}^{N \times 1}$ is the additive zero-mean complex white Gaussian noise vector with covariance matrix equal to $\sigma^2 \mathbf{I}$, $\omega$ is the wave frequency, and $c$ the velocity of propagation in the medium. Here we assume that the array interelement spacing $d$ is smaller than $\lambda/2$, where $\lambda \coloneqq 2\pi c/\omega$ denotes the wavelength. Equation (4.3) can be expressed as

$$\mathbf{a}(\mathbf{d}_r) = \mathbf{a}^{(M)}(\mathbf{d}_r) \otimes \ldots \otimes \mathbf{a}^{(1)}(\mathbf{d}_r) \in \mathbb{C}^{\prod_{m=1}^{M} N_m}, \tag{4.4}$$

where $\mathbf{a}^{(m)}(\mathbf{d}_r) = \left[ e^{\jmath \frac{\omega}{c} {\mathbf{p}_1^{(m)}}^{\mathsf{T}} \mathbf{d}_r}, \ldots, e^{\jmath \frac{\omega}{c} {\mathbf{p}_{N_m}^{(m)}}^{\mathsf{T}} \mathbf{d}_r} \right]^{\mathsf{T}} \in \mathbb{C}^{N_m \times 1}$ denotes the subarray

vector associated with the $m$th order translation vector $\mathbf{p}_{n_m}^{(m)}$, $n_m \in \{1, \ldots, N_m\}$. Indeed, (4.4) is a vectorization of a rank-1 array steering tensor defined as

$$\mathcal{A}(\mathbf{d}_r) = \mathbf{a}^{(1)}(\mathbf{d}_r) \circ \ldots \circ \mathbf{a}^{(M)}(\mathbf{d}_r) \in \mathbb{C}^{\times_{m=1}^{M} N_m}. \tag{4.5}$$

In view of this, the received signals (4.2) can be expressed as a linear combination of $R$ rank-1 tensors:

$$\mathcal{X}(k) = \sum_{r=1}^{R} \mathcal{A}(\mathbf{d}_r) s_r(k) + \mathcal{B}(k), \tag{4.6}$$

where $\mathcal{B}(k) = \Theta(\mathbf{b}(k)) \in \mathbb{C}^{\times_{m=1}^{M} N_m}$ is the tensorized form of the noise vector $\mathbf{b}(k)$. A rank-1 tensor filter $\mathcal{W} \in \mathbb{C}^{\times_{m=1}^{M} N_m}$ is proposed to perform beamforming exploiting the multilinear separable array structure [38].

## 4.2 Numerical Results

Computer experiments were conducted in order to assess the SOI estimation performance and the computational complexity of the tensor beamformer. In this context, $R = 3$ QPSK signals with unitary variance, carrying $K$ symbols and arriving from the directions displayed at Table 4.1 were considered. The signal corresponding to $r = 1$ was set as SOI.

| Source | Azimuth [rad] | Elevation [rad] |
|---|---|---|
| $r = 1$ (SOI) | $\pi/3$ | $-\pi/4$ |
| $r = 2$ | $\pi/6$ | $\pi/3$ |
| $r = 3$ | $\pi/4$ | $-\pi/6$ |

Table 4.1: Direction of arrival of the QPSK sources.

The classical WH filter (A.3) was used as benchmark method. The autocorrelation matrix and the cross-correlation vector of the WH filter were estimated directly from data. The convergence threshold of TWH was set to $\varepsilon = 10^{-3}$. The mean performance indices were calculated by averaging the results obtained in $J = 100$ MC realizations. The SOI estimation performance was evaluated in terms of the MSE measure, defined as

$$\text{MSE} = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{K} \|\mathbf{s}^{(j)} - \hat{\mathbf{s}}^{(j)}\|_2^2,$$

where

$$\mathbf{s} = [s_{\mathrm{SOI}}(k), s_{\mathrm{SOI}}(k-1), \ldots, s_{\mathrm{SOI}}(k-K+1)]^{\mathsf{T}} \in \mathbb{C}^K$$

and $\hat{\mathbf{s}}^{(j)} \in \mathbb{C}^K$ is the decided symbols vector whose $k$th entry is the symbol decided from the statistic $\langle \mathcal{X}(k), \mathcal{W} \rangle$. As in the previous chapter, the number of FLOPS demanded by each method was computed using the Lightspeed MATLAB toolbox [37], and the Tensorlab toolbox [36] was used to implement the tensor operations of the multilinear methods.

Three simulation scenarios were considered. In the first scenario, the performance indices were calculated by varying the sample size $K$, as illustrated in Figure 4.3. In this case, the global array consisted of $N = 128$ sensors. In the second scenario, the performance indices were calculated by varying the number $N$ of sensors of the global array for $K = 5000$ samples, as depicted in Figure 4.4. In the third scenario, the performance was assessed by varying the SNR, while keeping the sample size and the number of sensors fixed, as illustrated in Figure 4.5. In all scenarios, the global array was formed by translating a reference uniform rectangular array along the $x$-axis, as illustrated in Figure 4.2.

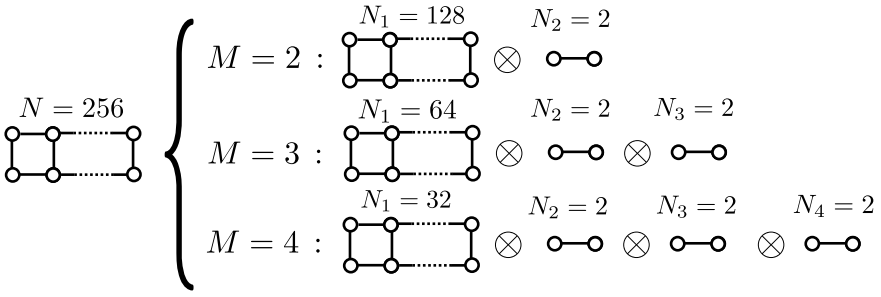
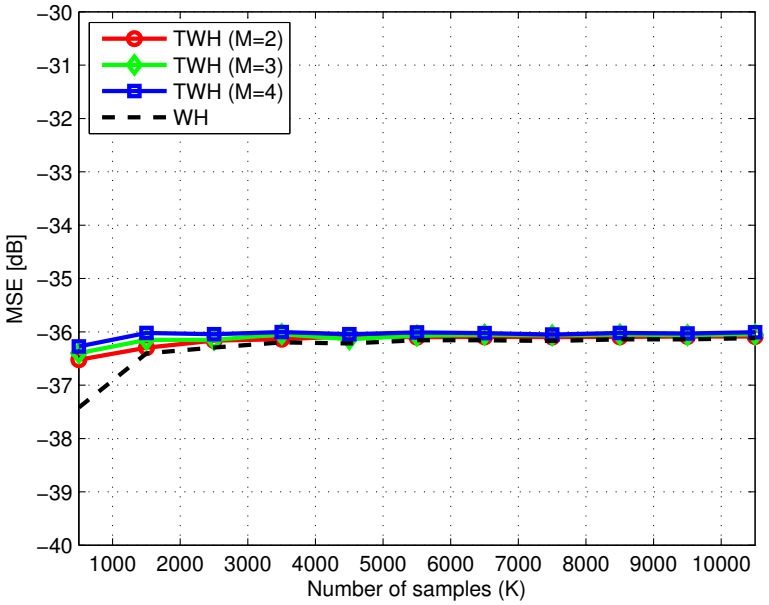


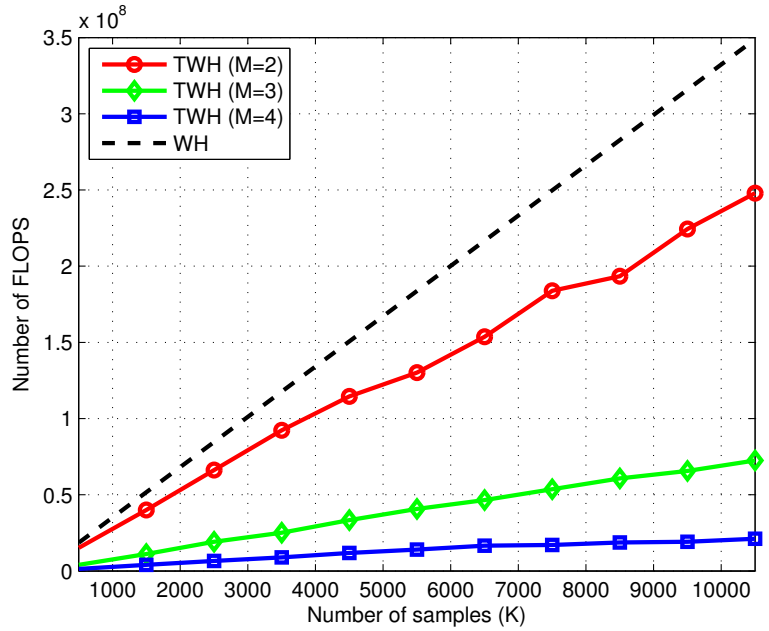

Figure 4.2: Illustration of the antenna array factorization considered in the computational simulations for $N = 256$ sensors.

## 4.3 Discussion

The proposed multilinear model for translation invariant arrays allowed us to employ the TWH algorithm presented in Chapter 2 to perform the array beamforming. In the conducted simulations, a translation invariant array was decomposed into multiple setups expressed in terms of virtual subarrays. This representation allowed us to perform the beamforming on each subarray instead of processing the global array. Any sensor array whose received signals and spatial signature are well modeled by equations (4.2) and (4.3), respectively, can be regarded as translation invariant. This property was exploited by TWH, providing an important reduction on computational costs as observed in Figures 4.3b, 4.4b, and 4.5b. The computational cost decreased with the separability order, as expected. It is important to stress that this reduction was obtained without significant loss on MSE performance, as observed in Figures 4.3a, 4.4a, and 4.5a. Indeed, the MSE performance of the TWH algorithm could not be better than the WH filter, since both are bounded by the MMSE (A.4), unlike the relative system mismatch in Chapter 3.

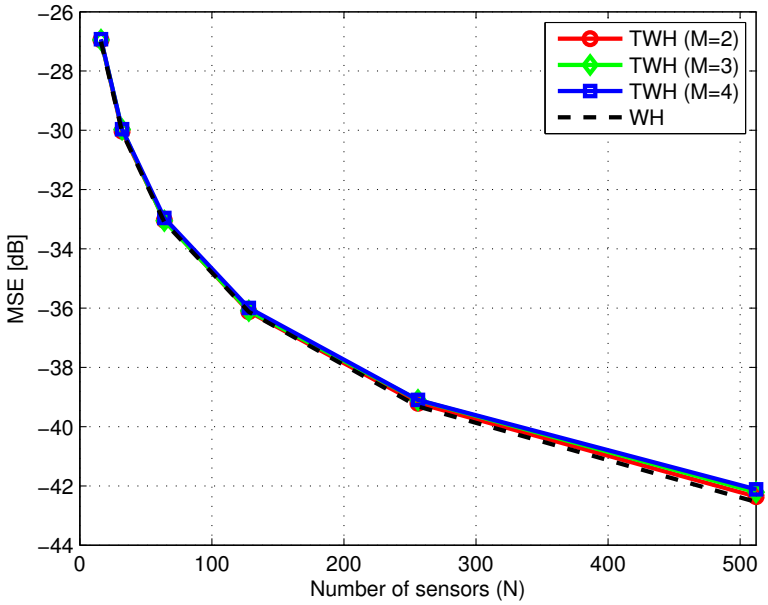



(a) MSE vs. Sample size.

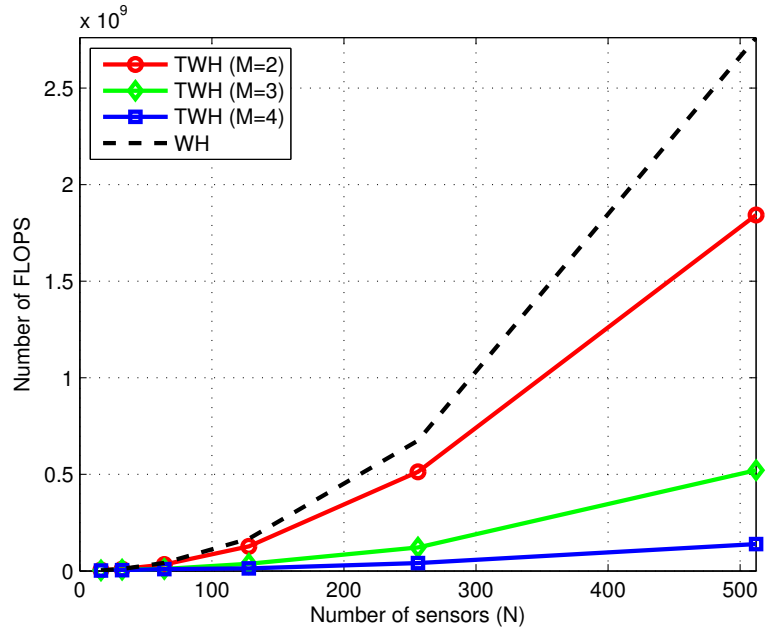



(b) Number of FLOPS vs. Sample size.

Figure 4.3: Performance for a varying number of samples, $N = 128$ sensors and SNR $= 15\,\mathrm{dB}$.

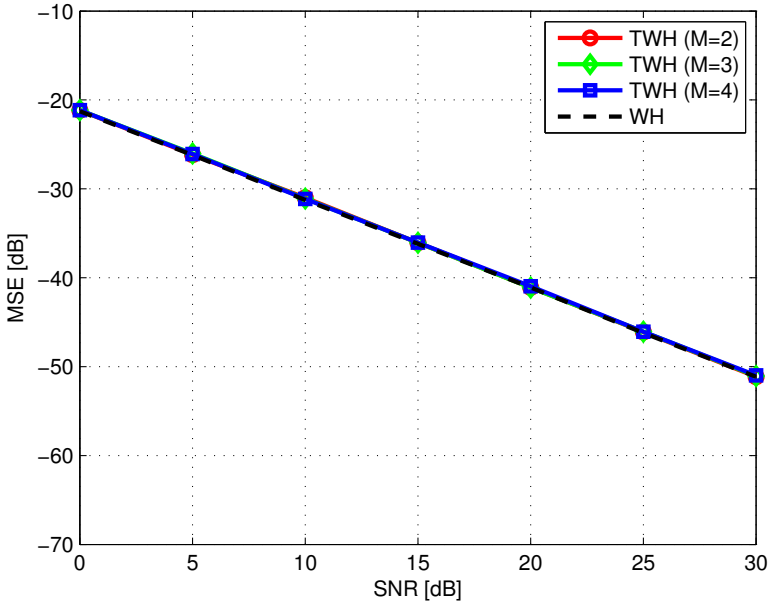



(a) MSE vs. Number of sensors.

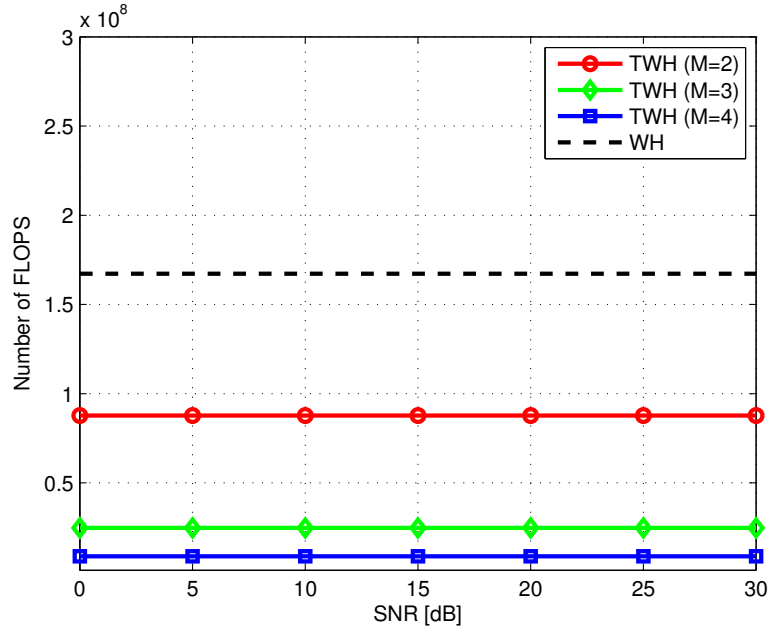



(b) Number of FLOPS vs. Number of sensors.

Figure 4.4: Performance for a varying number of sensors, $K = 5000$ samples and SNR $= 15\,\mathrm{dB}$.




(a) MSE vs. SNR.




(b) Number of FLOPS vs. SNR.

Figure 4.5: Performance for a varying SNR, $K = 5000$ samples and $N = 128$.

# Chapter 5

# Conclusion

In this thesis, we have proposed a computationally efficient tensor method for large-scale multilinear filtering problems. In the context of supervised system identification, we have shown that separable FIR filters admit a factorization in terms of its subfilters, which leads to a multilinear filtering structure. This multilinearity calls for a tensor formulation that leads to the separate estimation of each subfilter, allowing the computational complexity reduction and accurate system identification. In the context of array processing, we have proposed a multilinear model for translation invariant arrays that allowed us to reduce the computational costs of the MMSE beamforming without any significant trade-offs.

From the numerical results of Chapter 3, we have observed that tensor methods provides more accurate system parameter estimates in a computationally efficient manner. This is because of the subfilter interdependency present in each subproblem (2.33). More specifically, the subfilter interdependency takes into account updated versions of the other subfilters, increasing the estimation accuracy. We have observed that our proposed algorithm, TWH, is more efficient than the TLMS algorithm. This is due to its optimal ALS updates and fast convergence, whereas TLMS is subject to slow convergence when a small step size is used. Experiments also showed the strong dependence of the tensor method performance on the system separability order and partitioning. It is important to mention that since the relative system mismatch is not directly linked with the error surface, it is not bounded by the MMSE. The WH filter and the NLMS algorithm do not provide estimates as accurate as tensor methods in the considered multilinear large-scale problem, since they ignore the system separability. Although the SepWH method considers this *a priori* in some sense, the estimation step of each subfilter does not regard the other subfilters, resulting in the same accuracy of the WH filter.

The numerical results obtained in Chapter 4 confirm those from Chapter 3. TWH provides an important computational cost reduction compared to that of WH beamformer. This reduction becomes more important as the separability order grows. We have also remarked that these gains comes without compromising the MSE performance, which is bounded by the MMSE. We have shown that translation invariant arrays admit a representation in terms of virtual subarrays and translations. Since any array modeled by (4.2) and (4.3) is translation invariant, many practical sensor array systems can benefit from this multilinear representation.

In conclusion, the proposed TWH algorithm provides a significant computational complexity reduction compared to the classical filtering methods. It is also more efficient than the TLMS algorithm, which faces slow convergence when a very accurate system parameter estimate is demanded. System separability offers more accuracy and computational cost reduction up to a certain separability order.

Tensor filtering is an emergent field on signal processing that has been attracting more and more attention due to their advantages and open problems. In the following, we list some research perspectives and open problems concerning this topic.

- In this thesis we have considered rank-1 tensor filters. The low-rank tensor filtering extension is natural, allowing us, for example, to identify non-separable systems. Preliminary simulations indicated that the low-rank tensor filtering problem is subject to local minima. Since ALS is very sensitive local stationary points, more robust optimization methods such as Levenberg-Marquadt could be used.

- Tensor formulation of other filtering methods such as the Kalman filter and the Recursive Least Squares algorithm could provide, at least, computational complexity reduction. It is known that Kalman filtering can be interpreted in terms of Bayesian networks, which become unpractical to handle when a large number of variables is considered. Tensor networks models [21], such as Hierarchical Tucker Decomposition and Tensor Train, are able to efficiently represent large networks. Therefore, we may formulate Kalman filters as tensor networks to reduce their computational complexity in massive scenarios.

- Although the SepWH algorithm did not provide any improvements com-

pared to the WH filter, its idea can be further improved. We have observed that the extended subsystems are sparse, but we have not exploited this property in SepWH. Indeed, a L1 regularization term can be added to Equation (2.31) for exploiting the system sparsity. Furthermore, an heuristic method for sparsity exploitation could be employed the LMS algorithm to increase its convergence rate. Such method would set to zero elements previously known to be null, producing an instantaneous decrease in the MSE.

- *A priori* information usually leads to unsupervised filtering methods. The cross relation approach [41], which blindly identifies co-prime channels, could be generalized to the tensor scenario. Likewise, sparsity constraints could blindly identify separable systems.

- The multilinear filtering problem can be interpreted as a relaxed projection mapping (RPM) problem [42]. The solutions of the linear subproblems (2.33) can be interpreted as alternating projections. The RPM framework could provide a geometrical interpretation to this multilinear filtering operation.

- Although we have indicated that TWH is monotonically convergent, it is still necessary to study its convergence properties. However, the analytical convergence study of alternating methods is a challenging problem still under investigation.

We list some application problems that could benefit from tensor filtering.

- Tensor filtering can be employed to jointly process multiple signal domains. In biomedical signal processing problems, a space-time-frequency can be constructed and tensor filtering methods can be used to filter all domains incorporating *a priori* information. For instance, independent component analysis could be used in the time domain, beamforming techniques in the space domain, and adaptive filtering in the frequency domain. Tensor filtering could also be employed in the design of transceiver systems. Higher-order tensor signals can be obtained by exploiting the multiple diversities often present in communication systems.

- In [19, 38], tensor-based beamforming methods for narrowband signals were proposed. A natural extension is the design of wideband tensor beamformers, which would lead to a low-rank tensor filtering problem.

# Appendix A

# Classical Supervised Linear Filtering Methods

In this Appendix, we will present the supervised linear filtering methods used in this thesis: the Wiener-Hopf filter, the Least Mean Square algorithm, and its normalized version.

## A.1 Wiener-Hopf Filter

Let $J_{\mathbf{w}} \coloneqq \mathbb{E}\left[\left|d(n) - \mathbf{w}^{\mathsf{H}}\mathbf{x}(n)\right|^2\right]$ denote the objective function. Since $J_{\mathbf{w}}$ is a convex surface, it has an unique critical point which coincides with the global minimum. Let us expand the objective function:

$$\begin{aligned} J_{\mathbf{w}} &= \mathbb{E}\left[\left|d(n) - \mathbf{w}^{\mathsf{H}}\mathbf{x}(n)\right|^2\right] \\ &= \mathbb{E}\left\{\left[d(n) - \mathbf{w}^{\mathsf{H}}\mathbf{x}(n)\right]\left[d(n) - \mathbf{w}^{\mathsf{H}}\mathbf{x}(n)\right]^*\right\} \\ &= \sigma_d^2 - \mathbf{p}_{d\mathbf{x}}^{\mathsf{H}}\mathbf{w} - \mathbf{w}^{\mathsf{H}}\mathbf{p}_{d\mathbf{x}} + \mathbf{w}^{\mathsf{H}}\mathbf{R}_{\mathbf{x}}\mathbf{w}, \end{aligned} \tag{A.1}$$

where

$$\begin{aligned} \sigma_d^2 &\coloneqq \mathbb{E}\left[|d(n)d^*(n)|^2\right] \\ \mathbf{p}_{d\mathbf{x}} &\coloneqq \mathbb{E}\left[d^*(n)\mathbf{x}(\mathbf{n})\right] \\ \mathbf{R}_{\mathbf{x}} &\coloneqq \mathbb{E}\left[\mathbf{x}(n)\mathbf{x}(n)^{\mathsf{H}}\right] \end{aligned}$$

denote the desired signal variance, the cross-correlation vector between $d(n)$ and $\mathbf{x}(n)$, and the autocorrelation matrix of $\mathbf{x}(n)$, respectively.

The minimizer of (2.2) is obtained by setting the gradient

$$\nabla J_{\mathbf{w}} = \frac{\partial J_{\mathbf{w}}}{\partial \mathbf{w}^{\mathsf{H}}} = -\mathbf{p}_{d\mathbf{x}} + \mathbf{R}_{\mathbf{x}}\mathbf{w} \tag{A.2}$$

equal to $\mathbf{0} \in \mathbb{C}^N$, and solving for $\mathbf{w}$:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{\mathbf{x}}^{-1}\mathbf{p}_{d\mathbf{x}}. \tag{A.3}$$

This solution is known as the WH filter. To compute (A.3), a $N \times N$ matrix has to be inverted. Therefore, the WH filter has computational complexity $\mathcal{O}(N^3)$. The minimum mean square error $J_{\min}$ is found by substituting (A.3) into (A.1):

$$\begin{aligned} J_{\min} &= J_{\mathbf{w}_{\text{opt}}} \\ &= \sigma_d^2 - \mathbf{p}_{d\mathbf{x}}^{\mathsf{H}}\mathbf{w}_{\text{opt}} - \mathbf{p}_{d\mathbf{x}}^{\mathsf{H}}\mathbf{R}_{\mathbf{x}}^{-\mathsf{H}}\mathbf{p}_{d\mathbf{x}} + \mathbf{p}_{d\mathbf{x}}\mathbf{R}_{\mathbf{x}}^{-\mathsf{H}}\mathbf{R}_{\mathbf{x}}\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{p}_{d\mathbf{x}} \\ &= \sigma_d^2 - \mathbf{p}_{d\mathbf{x}}^{\mathsf{H}}\mathbf{w}_{\text{opt}}. \end{aligned} \tag{A.4}$$

The WH filter provides the MMSE solution, however solution, in practice, might be too computationally expensive since it depends on matrix inversion and on the statistical expectation operator. In large-scale problems, these operations might be unfeasible. A method to obtain an approximate WH filter is summarized in Algorithm 4, where $\mathbf{X} \coloneqq [\mathbf{x}(n), \ldots, \mathbf{x}(n-T+1)] \in \mathbb{C}^{N\times T}$, $\mathbf{d} \coloneqq [d(n), \ldots, d(n-T+1)]^{\mathsf{T}} \in \mathbb{C}^T$, and $T$ denotes the sample size. In this algorithm, we make use of averaging operations to approximate the statistical expectation in equation (A.3).

**Algorithm 4** Wiener-Hopf filter

1: **procedure** WH($\mathbf{X} \in \mathbb{C}^{N\times T}$, $\mathbf{d} \in \mathbb{C}^T$)
2: $\quad \hat{\mathbf{R}}_{\mathbf{x}} \leftarrow (1/T)\mathbf{X}\mathbf{X}^{\mathsf{H}}$
3: $\quad \hat{\mathbf{p}}_{d\mathbf{x}} \leftarrow (1/T)\mathbf{X}\mathbf{d}^*$
4: $\quad \mathbf{w} \leftarrow \hat{\mathbf{R}}_{\mathbf{x}}^{-1}\hat{\mathbf{p}}_{d\mathbf{x}}$
5: **end procedure**

## A.2 LMS Algorithm

Adaptive filtering is a popular alternative to analytical solutions due to its implementation simplicity and reduced computational cost. It consists of updating the filter coefficients according to a numerical algorithm which solves the problem

(2.2). Let us consider the steepest descent method, which takes an initial guess $\mathbf{w}(0)$ and computes the next guess by making a change in a direction opposed to that of the gradient of $J_{\mathbf{w}}$ [1]. This process is repeated in current time until the filter coefficients converge ergodically to the WH filter. The steepest descent update rule is given by

$$\begin{aligned}\mathbf{w}(k+1) &\coloneqq \mathbf{w}(k) - \mu \nabla J_{\mathbf{w}}(k) \\ &= \mathbf{w}(k) - \mu \left[\mathbf{p}_{d\mathbf{x}} - \mathbf{R}_{\mathbf{x}}\mathbf{w}(k)\right], \end{aligned} \tag{A.5}$$

where $\mu \in \mathbb{R}^+$ is the step size factor. It can be shown that the steepest descent method will converge to the MMSE solution, i.e. $\lim_{k\to\infty} J_{\mathbf{w}}(k) = J_{\min}$ provided that $0 < \mu < \frac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}_{\mathbf{x}}$. To compute the update rule (A.5), it is necessary to estimate $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{p}_{d\mathbf{x}}$, which we are trying to avoid.

The LMS algorithm solves (2.2) using the stochastic gradient descent, which approximates the deterministic gradient (A.2) by considering the instantaneous estimates of $\mathbf{p}_{d\mathbf{x}}$ and $\mathbf{R}_{\mathbf{x}}$:

$$\hat{\nabla} J_{\mathbf{w}}(k) \coloneqq -d^*(k)\mathbf{x}(n) + \mathbf{x}(n)\mathbf{x}(n)^{\mathsf{H}}. \tag{A.6}$$

Substituting (A.6) into (A.5) gives the LMS update rule:

$$\begin{aligned}\mathbf{w}(k+1) &\coloneqq \mathbf{w}(k) - \mu \left[-d^*(k)\mathbf{x}(k) + \mathbf{x}(k)\mathbf{x}(k)^{\mathsf{H}}\right] \\ &= \mathbf{w}(k) - \mu\mathbf{x}(k)\left[d(n) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\right] \\ &= \mathbf{w}(k) + \mu\mathbf{x}(k)e^*(k). \end{aligned} \tag{A.7}$$

The LMS algorithm will converge provided that $0 < \mu < \frac{2}{\lambda_{\max}}$. Unlike the steepest descent algorithm, it relies on an inexact gradient vector estimate, which only approximates the optimal solution $\mathbf{w}_{\text{opt}}$. The computation of (A.7) has a computational complexity of $\mathcal{O}(N)$.

The convergence rate of the LMS algorithm strongly depends on $\xi(\mathbf{R}_{\mathbf{x}})$, the eigenvalue spreading of $\mathbf{R}_{\mathbf{x}}$, the step size factor, and the filter length [43, Chapter 5]. It will converge faster as $\xi(\mathbf{R}_{\mathbf{x}}) \to 1$, and slower as $\xi(\mathbf{R}_{\mathbf{x}}) \to \infty$. However even when the eigenvalue spreading of $\mathbf{R}_{\mathbf{x}}$ is around 1 and $\mu$ is sufficiently large, the LMS algorithm will slowly converge when the filter is large [4]. This algorithm is summarized in Algorithm 5.

**Algorithm 5** Least Mean Square algorithm

1: **procedure** LMS($\mathbf{x}(n) \in \mathbb{C}^N$, $d(n) \in \mathbb{C}^T$, $\mu \in \mathbb{R}^+$)
2: $k \leftarrow 1$
3: Initialize $\mathbf{w}(k)$.
4: **for** $k = 0, \ldots, T-1$ **do**
5: $e(k) \leftarrow d(n-k) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(n-k)$
6: $\mathbf{w}(k+1) \leftarrow \mathbf{w}(k) + \mu\mathbf{x}(n-k)e^*(k)$
7: **end for**
8: **end procedure**

## A.3 NLMS Algorithm

A natural approach to increase the LMS convergence rate consists of employing a variable step size factor. The NLMS algorithm was designed to reduce the instantaneous squared error $e^2(k)$ as much as possible and minimize the gradient noise amplification [10]. This gradient noise amplification occurs when $\|\mathbf{x}(k)\|^2$ becomes too large, resulting in a large correction term $\mu\mathbf{x}(k)e^*(k)$. We will show in the following that minimizing $e^2(k)$ using a variable step size increases the convergence rate and solves the gradient noise amplification problem.

Consider the LMS update rule with a variable step size $\mu(k)$:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\mathbf{x}(k)e^*(k). \tag{A.8}$$

The filter coefficients in $\mathbf{w}(k)$ are corrected by a factor $\Delta\mathbf{w}(k) = \mu(k)\mathbf{x}(k)e^*(k)$, resulting in a corrected filter coefficients $\tilde{\mathbf{w}}(k) = \mathbf{w}(k) + \Delta\mathbf{w}(k)$. The corrected instantaneous error $\tilde{e}(k)$ at instant $k$ is given by

$$\begin{aligned}\tilde{e}(k) &= d(k) - \mathbf{w}(k+1)^{\mathsf{H}}\mathbf{x}(k) \\ &= d(k) - [\mathbf{w}(k) + \Delta\mathbf{w}(k)]^{\mathsf{H}}\mathbf{x}(k) \\ &= d(k) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k).\end{aligned} \tag{A.9}$$

It is produced by using $\mathbf{w}(k+1)$ to obtain the output at instant $k$ using the same $\mathbf{x}(k)$. We seek to minimize the corrected instantaneous squared error $\tilde{e}^2(k)$ as much as possible. In this sense, consider the instantaneous squared error *before*

the correction:

$$\begin{aligned} e^2(k) &= [d(k) - y(k)]^2 \\ &= \left[d(k) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\right]^2 \\ &= d^2(k) - 2d(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k). \end{aligned} \tag{A.10}$$

Now let us expand (A.9):

$$\begin{aligned} \tilde{e}^2(k) &= \left[d(k) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\right]^2 \\ &= \overbrace{d^2(k) - 2d(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)}^{e^2(k)} \\ &+ 2\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) \\ &- 2d(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) \\ &= e^2(k) + 2\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) + \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) \\ &- 2d(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) \\ &= e^2(k) + 2\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)y(k) - 2\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)d(k) \\ &+ \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k) \\ &= e^2(k) + \underbrace{\left[-2\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)e(k) + \Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\Delta\mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(k)\right]}_{\Delta e^2(k)}. \end{aligned} \tag{A.11}$$

Substituting $\Delta\mathbf{w}(k) = \mu(k)\mathbf{x}(k)e^*(k)$ into (A.11) gives:

$$\Delta e^2(k) = \tilde{e}^2(k) - e^2(k) = -2\mu(k)e^2(k)\mathbf{x}(k)^{\mathsf{H}}\mathbf{x}(k) + \mu^2(k)e^2(k)\left[\mathbf{x}(k)^{\mathsf{H}}\mathbf{x}(k)\right]^2. \tag{A.12}$$

The minimum value of $\Delta e^2(k)$ is given by $\mu_{\text{opt}}(k)$ such that $\left.\frac{\partial \Delta e^2(k)}{\partial \mu(k)}\right|_{\mu=\mu_{\text{opt}}} = 0$. It then follows that

$$\mu_{\text{opt}}(k) = \frac{1}{\mathbf{x}(k)^{\mathsf{H}}\mathbf{x}(k)} = \frac{1}{\|\mathbf{x}(k)\|_2^2}. \tag{A.13}$$

A fixed step size $\tilde{\mu} \in \mathbb{R}^+$ is considered (A.13) to regularize the algorithm convergence rate, and a small positive constant $\gamma$ is included to avoid large step size when $\|\mathbf{x}(k)\|_2^2$ is small:

$$\mu(k) = \frac{\tilde{\mu}}{\gamma + \|\mathbf{x}(k)\|_2^2}. \tag{A.14}$$

The update rule of the NLMS algorithm is obtained by substituting (A.14) into

(A.8):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\tilde{\mu}}{\gamma + \|\mathbf{x}(k)\|_2^2}\mathbf{x}(k)e^*(k). \tag{A.15}$$

The NLMS algorithm will converge given that $0 < \tilde{\mu} < 2$ [10]. Although the variable step size increases its convergence rate, this algorithm still converges slowly when the filter parameter space is too large. The computational complexity of (A.15) is $\mathcal{O}(N)$. The NLMS algorithm is summarized in Algorithm 6:

**Algorithm 6** Normalized Least Mean Square algorithm

1: **procedure** NLMS($\mathbf{x}(n) \in \mathbb{C}^N$, $d(n) \in \mathbb{C}^T$, $\tilde{\mu} \in \mathbb{R}^+$)
2: $k \leftarrow 1$
3: Initialize $\mathbf{w}(k)$.
4: **for** $k = 0, \ldots, T-1$ **do**
5: $e(k) \leftarrow d(n-k) - \mathbf{w}(k)^{\mathsf{H}}\mathbf{x}(n-k)$
6: $\mathbf{w}(k+1) \leftarrow \mathbf{w}(k) + \frac{\tilde{\mu}}{\gamma+\|\mathbf{x}(n-k)\|_2^2}\mathbf{x}(n-k)e^*(k)$
7: **end for**
8: **end procedure**

The computational complexity associated with the classical filtering methods discussed in this Appendix are listed in Table A.1

| Method | Computational complexity |
|---|---|
| WH | $\mathcal{O}(N^3)$ |
| LMS | $\mathcal{O}(N)$ |
| NLMS | $\mathcal{O}(N)$ |

Table A.1: Computational complexity of the classical filtering methods.

# Appendix B

# Exponential Filter Separability

To test the influence of the system separability order on the multilinear filtering methods performance, it was necessary to devise a linear time-invariant system whose coefficients vector was equivalently separable across different separability levels. We show that an exponential filter can be expressed in terms of $M$ exponential subfilters for any $M \in \mathbb{N}$.

**Proposition 5.** *Any exponential FIR filter* $\mathbf{w} \in \mathbb{C}^N$ *whose elements are defined as*

$$[\mathbf{w}]_n = b^n, \quad b \in \mathbb{C},\, n \in \{1, \ldots, N\} \tag{B.1}$$

*is equivalently separable across different separability levels, i.e.*

$$\mathbf{w} = \bigotimes_{m=1}^{M} \mathbf{w}_m = \bigotimes_{j=1}^{J} \mathbf{w}_j, \quad \forall M, J \in \mathbb{Z}^+. \tag{B.2}$$

*Proof.* Let $n = 1 + \sum_{u=1}^{M} (i_u - 1) \prod_{v=1}^{u-1} I_v$ and $n' = 1 + \sum_{u'=1}^{J} (i_{u'} - 1) \prod_{v'=1}^{u'-1} I_{v'}$ correspond to the indices of the $M$th order and $J$th order separable filters where $n, n' \in \{1, \ldots, N\}$. Substituting these indices in (B.1) gives

$$[\mathbf{w}]_n = b^n = b^1 b^{(i_1-1)} b^{(i_2-1)I_1} \ldots b^{(i_M-1)\prod_{v=1}^{M-1} I_v} \tag{B.3}$$

and

$$[\mathbf{w}]_{n'} = b^{n'} = b^1 b^{(i_1-1)} b^{(i_2-1)I_1} \ldots b^{(i_J-1)\prod_{v=1}^{J-1} I_v} \tag{B.4}$$

Since the indices $n$ and $n'$ span the same interval, Equations (B.3) and (B.4) are equal, completing the proof. □

# Appendix C

# CAMSAP 2015 Paper

Paper presented in the 2015 IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2015).

**Title**: "Identification of Separable Systems Using Trilinear Filtering"

**Authors**: Lucas N. Ribeiro, André L. F. de Almeida, João César M. Mota

# Identification of Separable Systems Using Trilinear Filtering

Lucas N. Ribeiro, André L. F. de Almeida, and João C. M. Mota
Wireless Communications Research Group (GTEL)
Federal University of Ceará, Fortaleza, Brazil
Emails: {nogueira, andre, mota}@gtel.ufc.br

***Abstract*—Linear filtering methods are well known and have been successfully applied in system identification and equalization problems. However, they become unpractical when the number of parameters to estimate is very large. The recently proposed assumption of system separability allows the development of computationally efficient alternatives to classic adaptive methods in this scenario. In this work, we show that system separability calls for multilinear system representation and filtering. Based on this parallel, the proposed filtering framework consists of a trilinear extension of the classical Wiener-Hopf (WH) solution that exploits the separability property to solve the supervised identification problem. Our numerical results shows the proposed algorithm can provide a better accuracy than the classical WH solution which ignores the multilinear system representation.**

## I. Introduction

System identification is the problem of identifying parameters of an unknown system [1]. When the system input and output are available, supervised techniques such as the so-called Wiener-Hopf (WH) solution can be employed to identify the system response. Indeed, the WH solution is the minimum mean square error (MMSE) estimator for linear filters. However, it becomes inadequate for online system identification due to its relatively high computational cost. An alternative to this approach consists of using an adaptive filter. The coefficients of this device are updated according to an algorithm that minimizes the energy of the estimation error. The so-called least mean squares (LMS) algorithm is the canonical to adaptive filtering method due to its implementation simplicity and low complexity. Nevertheless, LMS-based algorithms suffer from slow convergence rate when the parameter space becomes too large [2].

Many ideas have been proposed to ameliorate the convergence rate of adaptive algorithms. For instance, step-size adaptation [3] and sparsity constraints [4] are known approaches for learning rate improvement. Recently, the authors in [2] proposed a low complexity LMS algorithm with fast learning rate, therein referred to as TensorLMS, which exploits the system separability assumption, meaning that the system impulse response vector can be decomposed as the Kronecker products of two vectors representing its components. Indeed, this assumption is plausible in Volterra systems [5] and array processing [6]. In [2], the existence and uniqueness of second-order separable systems was discussed.

This work was partially supported by CAPES, CNPq, and FUNCAP (Brazil).

Higher-order tensor filtering presents itself as the proper signal processing framework for exploiting multilinearly structured systems. Multilinear filtering was first introduced in the context of noise reduction in color images and multicomponent seismic data [7]. In this context, the higher-order tensor data is corrupted by a multidimensional noise and the original data is recovered by filtering the observed tensor data by matrix filters operating on each mode. Since reference signals were not available, the optima filters are obtained from subspace decomposition. In [8], a tensor (trilinear) filtering framework was proposed for equalization problems.

In this work, the multilinearity of separable systems is exploited for solving the identification problem using higher-order tensor modeling and filtering. More specifically, we assume that the overall system impulse response is third-order separable, i.e. it can be factored as the Kronecker product of three components. A tensor formalism is used to devise an algorithm for the identification of the system impulse response, leading to the trilinear extension of the WH solution. Furthermore, this tensor formalism provides proper notation and insight regarding the filtering operations. The proposed method employs an alternating minimization approach, whereas TensorLMS is based on the stochastic gradient descent method. According to our numerical simulations, the proposed algorithm overcomes the drawbacks of its classic counterpart and provides better system estimation quality.

*Notation*: Lowercase letters denote scalars, lowercase boldface letters denote vectors, uppercase boldface letters denote matrices and calligraphic letters denote higher-order tensors. The symbol $\otimes$ denotes the Kronecker product, $\diamond$ Khatri-Rao product, $\|\cdot\|_2^2$, $\times_n$ $n$-mode product, Euclidean norm, $\mathbb{E}[\cdot]$ statistical expectation, and $(\cdot)^{\mathsf{T}}$ transpose operator.

## II. The trilinear filtering framework

Before presenting the trilinear filtering framework and its connection with the system identification problem, some useful tensor formalism is given for later use.

### A. Tensor preliminaries

The $\{1,\ldots,N\}$-*mode products* of $\mathcal{T}$ with $N$ matrices $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$ yield the tensor $\tilde{\mathcal{T}} = \mathcal{T} \times_1 \mathbf{U}^{(1)} \ldots \times_N \mathbf{U}^{(N)} \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ defined as [9]

$$[\tilde{\mathcal{T}}]_{j_1,\ldots,j_N} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} x_{i_1,\ldots,i_N} u^{(1)}_{j_1,i_1} \ldots u^{(N)}_{j_N,i_N},$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{J_n \times I_n}$, $i_n \in \{1,\ldots,I_n\}$, and $j_n \in \{1,\ldots,J_n\}$, $n = 1,\ldots,N$. The $n$-mode unfolding of $\tilde{\mathcal{T}}$ is given by

$$\tilde{\mathbf{T}}_{(n)} = \mathbf{U}^{(n)}\mathbf{T}_{(n)}\mathbf{U}^{\otimes n^{\mathsf{T}}}, \tag{1}$$

where $\mathbf{T}_{(n)}$ denotes the $n$-mode unfolding of $\mathcal{T}$, and

$$\mathbf{U}^{\otimes n} = \mathbf{U}^{(N)} \otimes \ldots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \ldots \otimes \mathbf{U}^{(1)}$$

denotes the Kronecker product of the matrices $\{\mathbf{U}^{(j)}\}_{j=1,j\neq n}^{N}$ in the decreasing order. Note that the $\{1,\ldots,N\}$-mode products of $\mathcal{T}$ with the $N$ vectors $\{\mathbf{u}^{(n)}\}_{n=1}^{N}$ yields a scalar $t = \mathcal{T} \times_1 \mathbf{u}^{(1)^{\mathsf{T}}} \ldots \times_N \mathbf{u}^{(N)^{\mathsf{T}}}$ where $\mathbf{u}^{(n)} \in \mathbb{R}^{I_n \times 1}$, $n = 1,\ldots,N$.

**Vector-Tensorization:** Let us consider the linear map $\Theta : \mathbb{R}^{I_1 I_2 I_3} \rightarrow \mathbb{R}^{I_3 \times I_2 \times I_1}$, defined as

$$[\Theta(\mathbf{v})]_{i_3,i_2,i_1} = [\mathbf{v}]_{i_3+I_3(i_2-1)+I_2 I_3(i_1-1)}\,, \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^{I_1 I_2 I_3}$ is the input vector, which can be partitioned into $I_1$ partitions of length $I_2 I_3$. These partitions can be further divided into $I_2$ subpartitions of length $I_3$. The input vector can be transformed into a third-order tensor $\mathcal{V} = \Theta(\mathbf{v}) \in \mathbb{R}^{I_3 \times I_2 \times I_1}$. The operation of transforming a vector (or matrix) into a tensor is referred to as "tensorization" [10], [11]. There are different forms of tensorization operations, and the specific transformation depends on the considered application. In our case, tensorization is deterministic and achieved by vector tri-partitioning, according to (2). The vector-tensorization is an isometric isomorphism for the $l_2$-norm of $\mathbf{v}$ on $\mathbb{R}^{I_1 I_2 I_3}$ and the Frobenius norm of $\mathcal{V}$ on $\mathbb{R}^{I_3 \times I_2 \times I_1}$.

### *B. Trilinear filtering*

Consider an $M$th order finite impulse response (FIR) filter with impulse response $\mathbf{z} \in \mathbb{R}^M$. The input regression vector and the output signal at instant $k$ are represented by $\mathbf{x}(k) \in \mathbb{R}^M$ and $y(k) = \mathbf{z}^{\mathsf{T}}\mathbf{x}(k)$, respectively. Let us suppose this filter is third-order separable, i.e. it can be expressed as $\mathbf{z} = \mathbf{z}_a \otimes \mathbf{z}_b \otimes \mathbf{z}_c$, where $\mathbf{z}_a \in \mathbb{R}^{M_a}$, $\mathbf{z}_b \in \mathbb{R}^{M_b}$, and $\mathbf{z}_c \in \mathbb{R}^{M_c}$ are its component subfilters, with $M_a M_b M_c = M$. This corresponds to a rank-one third-order tensor decomposition problem [11]. In view of this, its existence and uniqueness in the least squares (LS) sense is guaranteed [12], [13]. Since there are no results ensuring the estimation error bound for higher-order tensor decompositions, devising the closed-form MMSE expression for a third-order tensor decomposition is a challenge.

In order to express the output signal $y(k) = \mathbf{z}^{\mathsf{T}}\mathbf{x}(k)$ as the outcome of a trilinear filtering, let us partition the input signal into $\mathbf{x}(k) = [\mathbf{x}_1(k), \mathbf{x}_2(k), \ldots, \mathbf{x}_{M_a}(k)]^{\mathsf{T}}$ where $\mathbf{x}_{m_a}(k) \in \mathbb{R}^{M_b M_c}$ for $m_a = 1,\ldots,M_a$. Now, let us further divide each partition $\mathbf{x}_{m_a}(k)$ into $\mathbf{x}_{m_a}(k) = [\mathbf{x}_{1,m_a}(k), \mathbf{x}_{2,m_a}(k), \ldots, \mathbf{x}_{M_b,m_a}(k)]^{\mathsf{T}}$, where $\mathbf{x}_{m_b,m_a}(k) = [x_{1,m_b,m_a}(k), x_{2,m_b,m_a}(k), \ldots, x_{M_c,m_b,m_a}(k)]^{\mathsf{T}} \in \mathbb{R}^{M_c}$ for $m_b = 1,\ldots,M_b$ and $x_{m_c,m_b,m_a}(k) \in \mathbb{R}$ is an element of the third-order tensor $\mathcal{X}(k) = \Theta[\mathbf{x}(k)] \in \mathbb{R}^{M_c \times M_b \times M_a}$. The output signal can be written as:

$$\begin{aligned} y(k) &= (\mathbf{z}_a \otimes \mathbf{z}_b \otimes \mathbf{z}_c)^{\mathsf{T}}\mathbf{x}(k) \\ &= \sum_{m_a=1}^{M_a} [\mathbf{z}_a]_{m_a} (\mathbf{z}_b \otimes \mathbf{z}_c)^{\mathsf{T}} \mathbf{x}_{m_a}(k) \\ &= \sum_{m_a=1}^{M_a}\sum_{m_b=1}^{M_b} [\mathbf{z}_a]_{m_a} [\mathbf{z}_b]_{m_b} \mathbf{z}_c^{\mathsf{T}} \mathbf{x}_{m_b,m_a}(k) \\ &= \sum_{m_a=1}^{M_a}\sum_{m_b=1}^{M_b}\sum_{m_c=1}^{M_c} [\mathbf{z}_a]_{m_a}[\mathbf{z}_b]_{m_b}[\mathbf{z}_c]_{m_c} x_{m_c,m_b,m_a}(k). \end{aligned} \tag{3}$$

Note that $y(k)$ is trilinear with respect to the elements of the subfilters as seen on (3). This operation is identified as the $\{1,2,3\}$-mode product (1):

$$y(k) = \mathcal{X}(k) \times_1 \mathbf{z}_c^{\mathsf{T}} \times_2 \mathbf{z}_b^{\mathsf{T}} \times_3 \mathbf{z}_a^{\mathsf{T}}, \tag{4}$$

$$= \mathbf{z}_c^{\mathsf{T}}\mathbf{u}_c(k) = \mathbf{z}_c^{T}\mathbf{X}_{(1)}(k)(\mathbf{z}_a \otimes \mathbf{z}_b) \tag{5}$$

$$= \mathbf{z}_b^{\mathsf{T}}\mathbf{u}_b(k) = \mathbf{z}_b^{T}\mathbf{X}_{(2)}(k)(\mathbf{z}_a \otimes \mathbf{z}_c) \tag{6}$$

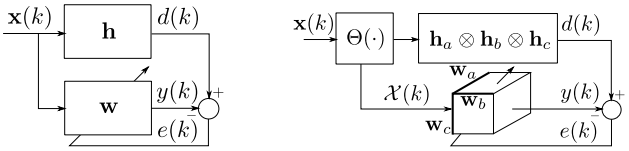
$$= \mathbf{z}_a^{\mathsf{T}}\mathbf{u}_a(k) = \mathbf{z}_a^{T}\mathbf{X}_{(3)}(k)(\mathbf{z}_b \otimes \mathbf{z}_c) \tag{7}$$

where $\mathbf{u}_c(k) = \mathbf{X}_{(1)}(k)(\mathbf{z}_a \otimes \mathbf{z}_b) \in \mathbb{R}^{M_c}$, $\mathbf{u}_b(k) = \mathbf{X}_{(2)}(k)(\mathbf{z}_a \otimes \mathbf{z}_c) \in \mathbb{R}^{M_b}$, and $\mathbf{u}_a(k) = \mathbf{X}_{(3)}(k)(\mathbf{z}_b \otimes \mathbf{z}_c) \in \mathbb{R}^{M_a}$ are the input of the subfilters $\mathbf{z}_c$, $\mathbf{z}_b$, and $\mathbf{z}_a$, respectively. The matrices $\mathbf{X}_{(1)}(k) \in \mathbb{R}^{M_c \times M_a M_b}$, $\mathbf{X}_{(2)}(k) \in \mathbb{R}^{M_b \times M_a M_c}$, and $\mathbf{X}_{(3)}(k) \in \mathbb{R}^{M_a \times M_b M_c}$ denote the $\{1,2,3\}$-mode unfoldings of $\mathcal{X}(k)$, respectively. Equations (5), (6), and (7) indicate that the product (4) can be equivalently represented by the output of the three linear subfilters

### *C. Optimum trilinear filtering*

Consider an $M$th order *unknown* trilinearly separable FIR system whose impulse response is $\mathbf{h} = \mathbf{h}_a \otimes \mathbf{h}_b \otimes \mathbf{h}_c$, where $\mathbf{h}_a \in \mathbb{R}^{M_a}$, $\mathbf{h}_b \in \mathbb{R}^{M_b}$, $\mathbf{h}_c \in \mathbb{R}^{M_c}$ and $M_a M_b M_c = M$. Now consider a trilinearly separable filter $\mathbf{w} = \mathbf{w}_a \otimes \mathbf{w}_b \otimes \mathbf{w}_c$, where $\mathbf{w}_a \in \mathbb{R}^{M_a}$, $\mathbf{w}_b \in \mathbb{R}^{M_b}$ and $\mathbf{w}_c \in \mathbb{R}^{M_c}$ are its subfilters. The input regression vector $\mathbf{x}(k) \in \mathbb{R}^M$ is tensorized, resulting in the input tensor signal $\mathcal{X}(k) = \Theta(\mathbf{x}(k)) \in \mathbb{R}^{M_c \times M_b \times M_a}$. Both the unknown system and the trilinear filter are driven by this tensor signal. The output of the unknown system is the desired signal $d(k) = \mathbf{h}^{\mathsf{T}}\mathbf{x}(k)$ of the trilinear filter, as depicted in Fig. 1(a). At the filter output, the estimation error $e(k) = d(k) - y(k)$ is calculated, where $y(k) = \mathbf{w}^{\mathsf{T}}\mathbf{x}(k) = \mathcal{X}(k) \times_1 \mathbf{w}_c^{\mathsf{T}} \times_2 \mathbf{w}_b^{\mathsf{T}} \times_3 \mathbf{w}_a^{\mathsf{T}}$ is the filter output. The mean square value of the estimation error is chosen to design the filter, leading to the following optimization problem:

$$\min_{\mathbf{w}_a,\mathbf{w}_b,\mathbf{w}_c} \mathbb{E}\left[|d(k) - \mathcal{X}(k) \times_1 \mathbf{w}_c^{\mathsf{T}} \times_2 \mathbf{w}_b^{\mathsf{T}} \times_3 \mathbf{w}_a^{\mathsf{T}}|^2\right]. \tag{8}$$




(a) System identification using classical adaptive filtering.

(b) System identification using trilinear filtering.

Fig. 1. Comparison between classical and trilinear WH approaches.

The objective function is clearly nonlinear with respect to the subfilters. Recalling that $y(k)$ can be represented in terms of its subfilters, the problem (8) can be divided in three subproblems:

$$\min_{\mathbf{w}_a} \mathbb{E}\left[\left|d(k) - \mathbf{w}_a^{\mathsf{T}}\mathbf{u}_a(k)\right|^2\right], \tag{9}$$

$$\min_{\mathbf{w}_b} \mathbb{E}\left[\left|d(k) - \mathbf{w}_b^{\mathsf{T}}\mathbf{u}_b(k)\right|^2\right], \tag{10}$$

$$\min_{\mathbf{w}_c} \mathbb{E}\left[\left|d(k) - \mathbf{w}_c^{\mathsf{T}}\mathbf{u}_c(k)\right|^2\right], \tag{11}$$

where $\mathbf{u}_a(k)$, $\mathbf{u}_b(k)$, and $\mathbf{u}_c(k)$ are defined as in Section II-B. Note that these vectors can be alternatively interpreted as weighted versions of the $\mathcal{X}(k)$ unfoldings. There is clearly an interdependency between the modes of $y(k)$ which hinders the optimization of (8). In view of this, the alternating least-squares (ALS) method can be used to solve (8). In this case, the subproblems (9), (10), and (11) are solved in a alternating manner. This method converges at least to a local minimum, and convergence to the global minimum cannot be guaranteed. Each subproblem corresponds to a classical LS estimation problem [1] (conditioned on the solutions provided by the other two subproblems). The solution of these subproblems is given by the WH equations:

$$\hat{\mathbf{w}}_a = \mathbf{R}_a^{-1}\mathbf{p}_a, \tag{12}$$

$$\hat{\mathbf{w}}_b = \mathbf{R}_b^{-1}\mathbf{p}_b, \tag{13}$$

$$\hat{\mathbf{w}}_c = \mathbf{R}_c^{-1}\mathbf{p}_c,, \tag{14}$$

where $\mathbf{R}_\lambda = \mathbb{E}\left[\mathbf{u}_\lambda(k)\mathbf{u}_\lambda(k)^{\mathsf{T}}\right] \in \mathbb{R}^{M_\lambda \times M_\lambda}$ is the autocorrelation matrix of $\mathbf{u}_\lambda(k)$, and $\mathbf{p}_\lambda = \mathbb{E}\left[d(k)\mathbf{u}_\lambda(k)\right] \in \mathbb{R}^{M_\lambda}$ is the crosscorrelation vector between $\mathbf{u}_\lambda(k)$ and $d(k)$ for $\lambda = a, b, c$. Note that these statistics can be expressed in terms of adaptive weighting matrices. For instance, consider the autocorrelation matrix of $\mathbf{u}_a(k)$:

$$\begin{aligned}\mathbf{R}_a &= \mathbb{E}\left[\mathbf{u}_a(k)\mathbf{u}_a(k)^{\mathsf{T}}\right] \\ &= \mathbb{E}\left[\mathbf{X}_{(3)}(k)(\mathbf{w}_b \otimes \mathbf{w}_c)(\mathbf{w}_b \otimes \mathbf{w}_c)^{\mathsf{T}}\mathbf{X}_{(3)}(k)^{\mathsf{T}}\right] \\ &= \mathbb{E}\left[\mathbf{X}_{(3)}(k)\mathbf{Q}_a\mathbf{X}_{(3)}(k)^{\mathsf{T}}\right].\end{aligned} \tag{15}$$

Now $\mathbf{R}_a$ is interpreted as the autocorrelation matrix of $\mathbf{X}_{(3)}(k)$ weighted by $\mathbf{Q}_a = (\mathbf{w}_b \otimes \mathbf{w}_c)(\mathbf{w}_b \otimes \mathbf{w}_c)^{\mathsf{T}}$. The crosscorrelation vector $\mathbf{p}_a$ can be interpreted as the weighted crosscorrelation between $\mathbf{X}_{(3)}(k)$ and $d(k)$:

$$\begin{aligned}\mathbf{p}_a &= \mathbb{E}\left[d(k)\mathbf{u}_a(k)\right] \\ &= \mathbb{E}\left[d(k)\mathbf{X}_{(3)}(k)(\mathbf{w}_b \otimes \mathbf{w}_c)\right] \\ &= \mathbb{E}\left[d(k)\mathbf{X}_{(3)}(k)\mathbf{q}_a\right],\end{aligned} \tag{16}$$

where $\mathbf{q}_a = (\mathbf{w}_b \otimes \mathbf{w}_c)$ and $\mathbf{Q}_a = \mathbf{q}_a\mathbf{q}_a^{\mathsf{T}}$.

### *D. Trilinear Wiener-Hopf algorithm*

To implement the ALS optimization, the batch Trilinear Wiener-Hopf (TriWH) algorithm is proposed. It consists of sequentially calculating (12), (13), and (14) using the sample estimate of the autocorrelation matrices and crosscorrelation vectors, as described in Algorithm 1. The convergence is attained when the normalized square error (NSE) between the true filter $\mathbf{h}$ and the estimated filter $\mathbf{w}$ is smaller than a threshold $\varepsilon$.

While the classical WH solution presents the standard complexity of $O(M^2)$ due to the inversion of a $(M \times M)$-dimensional autocorrelation matrix, TriWH presents a complexity of $O(Q(M_a^2 + M_b^2 + M_c^2))$, where $Q$ is the number of iterations necessary to attain the convergence. This computational complexity considerably smaller than its classic counterpart depending on how the input vector partitioning is done [2].

**Algorithm 1** TriWH

**procedure** TRIWH($\mathbf{x}(k)$, $d(k)$, $\mathbf{h}$, $\varepsilon$)
  $q \leftarrow 0$
  Initialize NSE($q$), $\mathbf{w}_a(q)$, $\mathbf{w}_b(q)$ and $\mathbf{w}_c(q)$.
  **repeat**
    Initialize $\mathbf{R}_\lambda(k)$ and $\mathbf{p}_\lambda(k)$ for $\lambda = a, b, c$
    $\mathbf{q}_c(q) \leftarrow (\mathbf{w}_a(q) \otimes \mathbf{w}_b(q))$
    $\mathbf{Q}_c(q) \leftarrow \mathbf{q}_c\mathbf{q}_c^{\mathsf{T}}$
    **for** $k = 0, \ldots, K-1$ **do**
      $\mathcal{X}(k) \leftarrow \mathbf{\Theta}(\mathbf{x}(k))$
      $\mathbf{R}_c(k+1) \leftarrow \mathbf{R}_c(k) + (1/K)\mathbf{X}_{(1)}(k)\mathbf{Q}_c(q)\mathbf{X}_{(1)}(k)^{\mathsf{T}}$
      $\mathbf{p}_c(k+1) \leftarrow \mathbf{p}_c(k) + (1/K)d(k)\mathbf{X}_{(1)}(k)\mathbf{q}_c(q)$
    **end for**
    $\mathbf{w}_c(q+1) \leftarrow \mathbf{R}_c(k+1)^{-1}\mathbf{p}_c(k+1)$

    $\mathbf{q}_b(q) \leftarrow (\mathbf{w}_a(q) \otimes \mathbf{w}_c(q+1))$
    $\mathbf{Q}_b(q) \leftarrow \mathbf{q}_b\mathbf{q}_b^{\mathsf{T}}$
    **for** $k = 0, \ldots, K-1$ **do**
      $\mathcal{X}(k) \leftarrow \mathbf{\Theta}(\mathbf{x}(k))$
      $\mathbf{R}_b(k+1) \leftarrow \mathbf{R}_b(k) + (1/K)\mathbf{X}_{(2)}(k)\mathbf{Q}_b(q)\mathbf{X}_{(1)}(k)^{\mathsf{T}}$
      $\mathbf{p}_b(k+1) \leftarrow \mathbf{p}_b(k) + (1/K)d(k)\mathbf{X}_{(2)}(k)\mathbf{q}_b(q)$
    **end for**
    $\mathbf{w}_b(q+1) \leftarrow \mathbf{R}_b(k+1)^{-1}\mathbf{p}_b(k+1)$

    $\mathbf{q}_a(q) \leftarrow (\mathbf{w}_b(q+1) \otimes \mathbf{w}_c(q+1))$
    $\mathbf{Q}_a(q) \leftarrow \mathbf{q}_a\mathbf{q}_a^{\mathsf{T}}$
    **for** $k = 0, \ldots, K-1$ **do**
      $\mathcal{X}(k) \leftarrow \mathbf{\Theta}(\mathbf{x}(k))$
      $\mathbf{R}_a(k+1) \leftarrow \mathbf{R}_a(k) + (1/K)\mathbf{X}_{(3)}(k)\mathbf{Q}_a(q)\mathbf{X}_{(3)}(k)^{\mathsf{T}}$
      $\mathbf{p}_a(k+1) \leftarrow \mathbf{p}_a(k) + (1/K)d(k)\mathbf{X}_{(3)}(k)\mathbf{q}_a(q)$
    **end for**
    $\mathbf{w}_a(q+1) \leftarrow \mathbf{R}_a(q+1)^{-1}\mathbf{p}_a(q+1)$
    $q \leftarrow q+1$
    $\mathbf{w}(q) \leftarrow \mathbf{w}_a(q) \otimes \mathbf{w}_b(q) \otimes \mathbf{w}_c(q)$
    NSE($q$) $= \|\mathbf{h} - \mathbf{w}(q)\|_2^2 / \|\mathbf{h}\|_2^2$
  **until** $|\text{NSE}(q) - \text{NSE}(q-1)| < \varepsilon$
**end procedure**

## III. NUMERICAL RESULTS

A separable FIR filter $\mathbf{h} = \mathbf{h}_a \otimes \mathbf{h}_b \otimes \mathbf{h}_c \in \mathbb{R}^M$ with $M = 1024$ parameters was estimated in the carried out experiments. The subfilters were defined similar to [2]: $\mathbf{h}_a \in \mathbb{R}^8$ is a vector whose $m_a$th element is given by $[\mathbf{h}_a]_{m_a} = 0.9^{m_a - 1}$ for $m_a = 1, 2, \ldots, 8$, $\mathbf{h}_b = [0, 0, \ldots, 0, 1, 0]^{\mathsf{T}} \in \mathbb{R}^{32}$, and $\mathbf{h}_c \in \mathbb{R}^4$ is a vector whose elements are Gaussian random variables with zero mean and unitary variance. This setup, depicted in Fig. 2, approximates a channel with echoes [2]. The input signal $x(k)$ was taken from a zero mean and unit variance Gaussian random process. An additive white Gaussian noise (AWGN) term with zero mean and variance $10^{-2}$ was added to the desired signal $d(k)$ for $k = 0, \ldots, K-1$. The sample size was set to $K = 15000$ in all experiments.

Monte Carlo simulations with $N$ independent realizations were performed to assess the performance of NLMS, TensorLMS, TriWH, and the WH solution. The step-size of the LMS-based algorithms was set to $0.5$. Independent impulse responses $\mathbf{h}^{(n)}$ were generated for each $n$th realization. The normalized mean square error (NMSE) between the

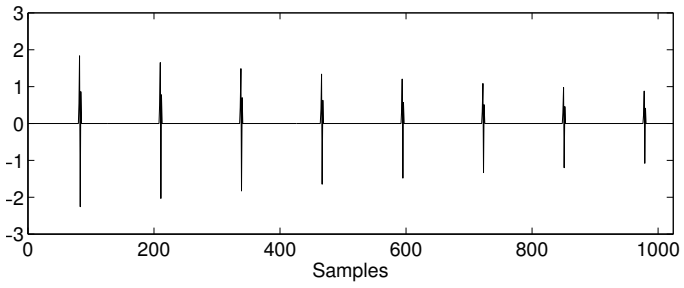



Fig. 2. Example of separable impulse response considered in this work.

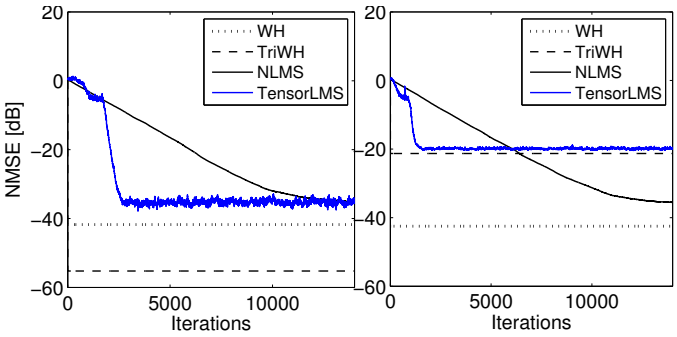



Fig. 3. Performance evaluation when the system is perfectly separable (left) and not perfectly separable (right).

actual and estimated filter at the $q$th iteration, defined as $\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{h}^{(n)} - \mathbf{w}^{(n)}(q)\|_2^2 / \|\mathbf{h}^{(n)}\|_2^2$, was used to measure the system mismatch. In the first experiment, the system to be identified was assumed to be perfectly separable. The performance of the studied methods is depicted in the left plot in Fig. 3. In the second experiment, their performances were assessed when the system was not perfectly separable. To reproduce this scenario, an AWGN component with zero mean and variance $10^{-4}$ was added to the true system impulse response. The right plot in Fig. 3 depicts their performance in this scenario.

When the system separability assumption holds, the TriWH solution presents the smallest system mismatch. It is an expected result since TriWH is a nonlinear method that was designed to properly explore the system separability, which is ignored by the classic methods. Furthermore, the adaptive weighting present in the autocorrelation and crosscorrelation matrices and vectors (c.f. (15) and (16)) contribute as well to the performance gain, playing the role of a weighted LS estimation with adjustable weights. It is important to recall that TriWH is less computationally expensive than its classical counterpart that do not exploit system separability. Regarding the iterative solutions, it can be seen that TriWH presented the best performance. In our simulations, the algorithm converged in about 3 iterations and presented a relative system mismatch much lower than the other algorithms. However, its computational complexity is greater than the iterative alternatives due to the calculation of matrix inverses. We also note that when the unknown system was not perfectly separable, TensorLMS and TriWH do not perform very well, since they could only identify the separable components. Since TensorLMS exploits the system separability, its performance could be similar to that of TriWH by setting a sufficiently small step-size, which would considerably decrease its convergence rate.

## IV. Conclusion and Perspectives

The tensor filtering framework was introduced in the supervised separable system estimation problem. Such type of system is useful to model multidimensional array of sensors. The tensor formalism present in this framework provided proper notation and interpretation for this trilinear problem. A nonlinear problem was considered to perform the identification. This problem was solved by exploiting its trilinear structure, leading to three linear subproblems. Based on this idea, the TriWH algorithm was proposed. According to our numerical experiments, it performed better than alternative solutions. A convergence analysis and the extension to the multichannel case will be provided in an extended version of this paper.

## Acknowledgment

The authors would like to thank Dr. Markus Rupp and Dr. Stefan Schwarz for publicly providing the MATLAB implementation of TensorLMS.

## References

[1] S. S. Haykin, *Adaptive filter theory*, 4th ed. Pearson Hall, 2002.

[2] M. Rupp and S. Schwarz, "A tensor LMS algorithm," in *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),*. IEEE, 2015.

[3] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1633–1642, 1992.

[4] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[5] R. D. Nowak and B. D. Van Veen, "Random and pseudorandom inputs for Volterra filter identification," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 2124–2135, 1994.

[6] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1260–1280, 2014.

[7] D. Muti and S. Bourennane, "Multidimensional filtering based on a tensor approach," *Signal Processing*, vol. 85, no. 12, pp. 2338–2353, 2005.

[8] L. N. Ribeiro, J. C. M. Mota, and A. L. F. de Almeida, "Trilinear Wiener filtering: Application to equalization problems," in *XXXI Simpósio Brasileiro de Telecomunicações, Fortaleza, Brazil*. SBrT, 2013.

[9] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[10] L. Grasedyck, "Polynomial approximation in hierarchical tucker format by vector-tensorization," Aachen, Germany, April 2010.

[11] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[13] T. Zhang and G. H. Golub, "Rank-one approximation to high order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 2, pp. 534–550, 2001.

# Appendix D

# ICASSP 2016 Paper

Paper accepted to the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016).

**Title**: "Tensor Beamforming for Multilinear Translation Invariant Arrays"

**Authors**: Lucas N. Ribeiro, André L. F. de Almeida, João César M. Mota

# TENSOR BEAMFORMING FOR MULTILINEAR TRANSLATION INVARIANT ARRAYS

*Lucas N. Ribeiro, André L. F. de Almeida, João C. M. Mota*

Wireless Communications Research Group (GTEL)
Federal University of Ceará, Fortaleza, Brazil
Emails: {nogueira, andre, mota}@gtel.ufc.br

**ABSTRACT**

In the past few years, multidimensional array processing emerged as the generalization of classic array signal processing. Tensor methods exploiting array multidimensionality provided more accurate parameter estimation and consistent modeling. In this paper, multilinear translation invariant arrays are studied. An $M$-dimensional translation invariant array admits a separable representation in terms of a reference subarray and a set of $M-1$ translations, which is equivalent to a rank-1 decomposition of an $M$th order array manifold tensor. We show that such a multilinear translation invariant property can be exploited to design tensor beamformers that operate multilinearly on the subarray level instead of the global array level, which is usually the case with a linear beamforming. An important reduction of the computational complexity is achieved with the proposed tensor beamformer with a negligible loss in performance compared to the classical minimum mean square error (MMSE) beamforming solution.

***Index Terms*—** Array processing, beamforming, tensor filtering.

## 1. INTRODUCTION

Array signal processing techniques have been used in the last decades in several area of applications such as: communications systems [1], audio processing [2], biomedical engineering [3], among others. An array consists of multiple sensors placed in different locations in space to process the impinging signals using a spatial filter. This filter is a *beamformer* when it is employed to enhance a signal of interest (SOI) arriving from a certain direction while attenuating any possible interfering signal [4].

In the past few years, generalized models for array processing have been proposed for taking advantage of the multidimensionality present in many types of arrays [1, 5, 6, 7]. For instance, in [8] a multidimensional harmonic retrieval method that improved the parameter estimation accuracy was proposed. Model selection methods for such multidimensional models were proposed in [9]. In [5], the authors introduced the concept of translation invariant arrays and proposed a joint channel and source estimation method based on the coherence properties of the sources. By contrast, very few works have concentrated on multidimensional beamforming. In [10], the authors proposed a MVDR-based beamformer that relies on the PARAFAC decomposition to estimate the DOA of the SOI. Recently, a multidimensional generalized sidelobe canceller (GSC) beamformer has been proposed in [11]. The separability of a uniform rectangular array was exploited by the proposed technique, resulting in better SOI estimation and reduced computational complexity compared to the classic GSC. Both tensor beamformers rely on a prior DOA estimation stage, which is then used to derive the filter coefficients. In [12], the authors exploited the separability of the impulse response of a linear time-invariant system and proposed a trilinear filtering system identification algorithm based on a tensor approach. Therein, it is shown that the tensor approach provides a more accurate system identification with a reduced computational complexity compared to its linear counterpart that ignores system separability property.

In this paper, we first extend the translation invariance property presented in [5] to multiple translation vectors. More specifically, we start from an $M$-dimensional translation invariant array that admits a separable representation in terms of a reference subarray and a set of $M-1$ translations, which is equivalent to a rank-1 decomposition of an $M$th order array manifold tensor. We show that such a multilinear translation invariant property can be exploited to design tensor beamformers that operate multilinearly on the subarray level instead of operating linearly on the global array level, which is the case with classical beamformers. Hence, an important reduction of the computational complexity can be achieved by the tensor beamformer, with a negligible loss in performance compared to the conventional linear minimum mean square error (MMSE) beamforming. According to our numerical results, the number of FLOPS demanded by the proposed method is remarkably lower than that of the linear (vector-based) MMSE filter for $M = 3, 4$ even though their SOI estimation quality are essentially the same. Moreover, since the separability degrees of freedom increase with the number of sensors in the multidimensional array, the tensor beamforming approach is particularly interesting for large-scale (massive) sensor arrays.

### 1.1. Notation

Scalars are denoted by lowercase letters, vectors by lowercase boldface letters, matrices by uppercase boldface letters, and higher-order tensors by calligraphic letters. The Kronecker, outer, and $n$-mode products are denoted by the symbols $\otimes$, $\circ$, and $\times_n$, respectively. The $\ell^2$ norm, statistical expectation, inner product, and $n$-mode tensor concatenation are denoted by $\|\cdot\|_2^2$, $\mathbb{E}[\cdot]$, $\langle\cdot,\cdot\rangle$, and $\sqcup_n$, respectively. The transpose and Hermitian operators are denoted by $(\cdot)^{\mathsf{T}}$ and $(\cdot)^{\mathsf{H}}$, respectively.

## 2. MULTILINEAR TRANSLATION INVARIANT ARRAYS

In this section, arrays enjoying the translation invariance property will be studied. Then, a tensor beamforming approach exploiting the multilinearity present in the translation invariant arrays will be formulated. First, let us review some tensor prerequisites for convenience.

### 2.1. Tensor prerequisites

In this work, an $N$th order tensor is defined as an $N$-dimensional array. For instance, $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ is an $N$th order tensor whose elements are denoted by $t_{i_1,i_2,\ldots,i_N} = [\mathcal{T}]_{i_1,i_2,\ldots,i_N}$ where $i_n \in \{1, \ldots, I_N\}$, $n = 1, 2, \ldots, N$.

This work was partially supported by CAPES, CNPq, and FUNCAP (Brazil).

The $\{1,\ldots,N\}$-*mode products* of $\mathcal{T}$ with $N$ matrices $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$ yield the tensor $\tilde{\mathcal{T}} = \mathcal{T} \times_1 \mathbf{U}^{(1)} \ldots \times_N \mathbf{U}^{(N)} \in \mathbb{C}^{J_1 \times \ldots \times J_N}$ defined as [13]

$$[\tilde{\mathcal{T}}]_{j_1,\ldots,j_N} = \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} t_{i_1,\ldots,i_N} u^{(1)}_{j_1,i_1} \ldots u^{(N)}_{j_N,i_N},$$

where $\mathbf{U}^{(n)} \in \mathbb{C}^{J_n \times I_n}$, $i_n \in \{1,\ldots,I_n\}$, and $j_n \in \{1,\ldots,J_n\}$, $n = 1,\ldots,N$. The $n$-mode unfolding of $\tilde{\mathcal{T}}$ is given by

$$\tilde{\mathbf{T}}_{(n)} = \mathbf{U}^{(n)} \mathbf{T}_{(n)} \mathbf{U}^{\otimes n^{\mathsf{T}}}, \tag{1}$$

where $\mathbf{T}_{(n)}$ denotes the $n$-mode unfolding of $\mathcal{T}$, and

$$\mathbf{U}^{\otimes n} = \mathbf{U}^{(N)} \otimes \ldots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \ldots \otimes \mathbf{U}^{(1)} \tag{2}$$

denotes the Kronecker product of the matrices $\{\mathbf{U}^{(j)}\}_{j=1,j\neq n}^{N}$ in the decreasing order. Note that the $\{1,\ldots,N\}$-mode products of $\mathcal{T}$ with the $N$ vectors $\{\mathbf{u}^{(n)}\}_{n=1}^{N}$ yields a scalar $t = \mathcal{T} \times_1 \mathbf{u}^{(1)^{\mathsf{T}}} \ldots \times_N \mathbf{u}^{(N)^{\mathsf{T}}}$ where $\mathbf{u}^{(n)} \in \mathbb{C}^{I_n \times 1}$, $n = 1,\ldots,N$.

The *inner product* between $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$ is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N},$$

where $a_{i_1,\ldots,i_N} = [\mathcal{A}]_{i_1,\ldots,i_N}$ and $b_{i_1,\ldots,i_N} = [\mathcal{B}]_{i_1,\ldots,i_N}$.

The *tensorization* operator $\Theta : \mathbb{C}^{\prod_{n=1}^{N} I_n} \to \mathbb{R}^{I_1 \times \ldots \times I_N}$ is defined as $[\Theta(\mathbf{v})]_{i_1,\ldots,i_N} = [\mathbf{v}]_j$, where $\mathbf{v} \in \mathbb{C}^{\prod_{n=1}^{N} I_n}$ is an input vector, and $j = i_1 + \sum_{\mu=2}^{M} (i_\mu - 1) \prod_{v=1}^{\mu-1} I_v$ for $i_\mu \in \{1,\ldots,I_\mu\}$.

### 2.2. Signal model

Consider a sensor array composed of $N$ isotropic sensors located at $\tilde{\mathbf{p}}_n \in \mathbb{R}^{3\times 1}$ for $n = 1,\ldots,N$. This array will be hereafter referred to as the *global array*. Consider that $R$ narrowband source signals with complex amplitudes $s_r(k)$ impinge on the global array from directions $\mathbf{d}_r = [\sin\theta_r \cos\phi_r, \sin\theta_r \sin\phi_r, \cos\theta_r]^{\mathsf{T}}$, where $\theta_r$ and $\phi_r$ denote the elevation and azimuth angles, respectively, $r = 1,\ldots,R$. The sources are assumed to be in the far-field and it is assumed that there are no reflection components. The steering vector associated with the $r$th source is given by

$$\mathbf{a}(\mathbf{d}_r) = \left[ e^{\jmath \frac{\omega}{c} \tilde{\mathbf{p}}_1^{\mathsf{T}} \mathbf{d}_r}, \ldots, e^{\jmath \frac{\omega}{c} \tilde{\mathbf{p}}_N^{\mathsf{T}} \mathbf{d}_r} \right]^{\mathsf{T}} \in \mathbb{C}^{N\times 1}, \tag{3}$$

where $\jmath = \sqrt{-1}$, $c$ denotes the velocity of propagation in the medium, and $\omega$ is the wave frequency. The signals collected by the global array at instant $k$ are modeled as

$$\mathbf{x}(k) = \sum_{r=1}^{R} \mathbf{a}(\mathbf{d}_r) s_r(k) + \mathbf{b}(k) \in \mathbb{C}^{N\times 1}, \tag{4}$$

where $\mathbf{b}(k) \in \mathbb{C}^{N\times 1}$ is the additive zero-mean complex white Gaussian noise vector with covariance matrix equal to $\sigma^2\mathbf{I}$.

Lim and Comon presented in [5, 14] the concept of translation invariant arrays formed by translating a *reference subarray*. However, the model discussed therein was limited to one translation vector only. This idea is now generalized to multiple translation vectors, leading to a multilinear array structure that will be useful in our context.

Let us assume that the global array enjoys the multilinear translation invariance property. Consider a reference subarray formed by $N_1$ reference sensors located at $\mathbf{p}_{n_1}^{(1)}$ for $n_1 = 1,\ldots,N_1$. The $n_1$th reference sensor is translated $M-1$ times by means of the translation vectors $\mathbf{p}_{n_2}^{(2)}, \ldots, \mathbf{p}_{n_M}^{(M)}$, yielding the following decomposition for the $n$th global sensor location vector

$$\tilde{\mathbf{p}}_n = \mathbf{p}_{n_1}^{(1)} + \mathbf{p}_{n_2}^{(2)} + \ldots + \mathbf{p}_{n_M}^{(M)}, \tag{5}$$

where $n = n_1 + \sum_{\mu=2}^{M} (n_\mu - 1) \prod_{v=1}^{\mu-1} N_v$, $n_\mu \in \{1,\ldots,N_\mu\}$. Note that $m = 1$ refers to the reference subarray, whereas $2 \leq m \leq M$ refers to the $m$th order translation vector. Substituting (5) into the global array vector (3) leads to the following separable form:

$$\begin{aligned} \mathbf{a}(\mathbf{d}_r) &= \begin{bmatrix} e^{\jmath \frac{\omega}{c} \mathbf{p}_1^{(1)^{\mathsf{T}}} \mathbf{d}_r} \ldots e^{\jmath \frac{\omega}{c} \mathbf{p}_1^{(M)^{\mathsf{T}}} \mathbf{d}_r} \\ \vdots \\ e^{\jmath \frac{\omega}{c} \mathbf{p}_{N_1}^{(1)^{\mathsf{T}}} \mathbf{d}_r} \ldots e^{\jmath \frac{\omega}{c} \mathbf{p}_{N_M}^{(M)^{\mathsf{T}}} \mathbf{d}_r} \end{bmatrix} \\ &= \mathbf{a}^{(1)}(\mathbf{d}_r) \otimes \ldots \otimes \mathbf{a}^{(M)}(\mathbf{d}_r) \in \mathbb{C}^{\prod_{m=1}^{M} N_m}, \end{aligned} \tag{6}$$

where $\mathbf{a}^{(m)}(\mathbf{d}_r) = [e^{\jmath \frac{\omega}{c} \mathbf{p}_1^{(m)^{\mathsf{T}}} \mathbf{d}_r}, \ldots, e^{\jmath \frac{\omega}{c} \mathbf{p}_{N_m}^{(m)^{\mathsf{T}}} \mathbf{d}_r}]^{\mathsf{T}} \in \mathbb{C}^{N_m \times 1}$ denotes the subarray vector associated with the $m$th order translation vector $\mathbf{p}_{n_m}^{(m)}$, $n_m \in \{1,\ldots,N_m\}$. A close idea was presented in [7] (therein referred to as multi-scale arrays), although the translation structure and its interpretation are different from the one we consider in this paper. Indeed, (6) is a vectorization of a rank-1 array steering tensor defined as

$$\mathcal{A}(\mathbf{d}_r) = \mathbf{a}^{(M)}(\mathbf{d}_r) \circ \ldots \circ \mathbf{a}^{(1)}(\mathbf{d}_r) \in {}^{N_1 \times \ldots \times N_M}. \tag{7}$$

In view of this, the received signals (4) can be expressed as a linear combination of $R$ rank-1 tensors:

$$\mathcal{X}(k) = \sum_{r=1}^{R} \mathcal{A}(\mathbf{d}_r) s_r(k) + \mathcal{B}(k), \tag{8}$$

where $\mathcal{B}(k) = \Theta(\mathbf{b}(k)) \in \mathbb{C}^{N_1 \times \ldots \times N_M}$ is the tensorized form of the noise vector $\mathbf{b}(k)$.

The multilinearity inherent to translation invariant arrays allows us to decompose the array response into multiple setups, as illustrated in Fig. 1. From this figure, it can be seen that the same 3-D global array can be decomposed as two separable (3-D and 1-D) subarrays ($M = 1$ translations), or three separable (2-D, 1-D, and 1-D) subarrays ($M = 2$ translations), or as four 1-D separable arrays ($M = 3$ translations). Other decompositions are possible, and the number of possibilities increases as a function of the number of sensors in the global array.

In the following, we exploit the multilinear translation invariant property to design tensor beamformers that operate on the subarray level instead of the global array level. By adopting a multilinear structure for the beamforming filters, we can obtain a considerable reduction on the computational complexity of the spatial filtering, with almost no loss in performance, as will be clear in later sections.

## 3. TENSOR BEAMFORMING

Classical linear beamforming methods [4] based on model (4) ignore the multilinearity that may be present in translation invariant arrays. For convenience, in this work we consider the MMSE beamforming solution, although the proposed approach is also applicable to other beamforming solutions. The well-known solution for the MMSE beamforming problem is given by

$$\mathbf{w}_{\text{MMSE}} = \mathbf{R}_x^{-1} \mathbf{p}_{d\mathbf{x}}, \tag{9}$$

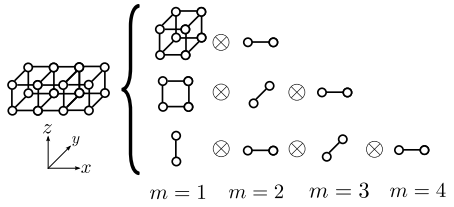



**Fig. 1**. An $4 \times 2 \times 2$ volumetric array decomposed in three different forms, in terms of subarrays translations ($M = 2, 3$, and 4). Reference subarrays are indexed by $m = 1$, whereas $m > 1$ refers to translation.

where $\mathbf{R}_x = \mathbb{E}\left[\mathbf{x}(k)\mathbf{x}(k)^{\mathsf{H}}\right] \in \mathbb{C}^{N\times N}$ is the autocorrelation matrix of the received vector, $\mathbf{p}_{d\mathbf{x}} = \mathbb{E}\left[s^*_{\text{SOI}}(k)\mathbf{x}(k)\right]$ is the cross-correlation vector between the received vector and the SOI. Since the MMSE filter depends on the inversion of an $N \times N$ matrix, the computational cost of (9) is $\mathcal{O}(N^2)$. Although the computational cost is not an issue for small sensor arrays, it may become prohibitively expensive for large-scale (massive) arrays of sensors. However, if the global array is translation invariant, then each $m$th subarray vector could be estimated/filtered in a lower-dimensional space, conditioned on the other subarray vectors thanks to the separability property of the array structure. Hereafter, instead of designing the beamforming coefficients from the spatial signatures or from prior DOA estimates (the common approach), we propose a direct beamforming method that exploits the array separability property in (8).

Let us consider an $M$th order tensor filter $\mathcal{W} \in \mathbb{C}^{N_1\times N_2\times\ldots\times N_M}$, each mode of which is associated with a different subarray. The output of the tensor beamformer is given by

$$y(k) = \langle \mathcal{X}(k), \mathcal{W}^* \rangle. \tag{10}$$

The tensor beamformer $\mathcal{W}$ can be designed to minimize the following cost function

$$J(\mathcal{W}) = \mathbb{E}\left[|s_{\text{SOI}}(k) - \langle \mathcal{X}(k), \mathcal{W}^* \rangle|^2\right], \tag{11}$$

defined as the MSE between $y(k)$ and the SOI $s_{\text{SOI}}(k)$. We assume that the tensor filter is rank-1, i.e. $\mathcal{W} = \mathbf{w}_1 \circ \ldots \circ \mathbf{w}_M$, where $\mathbf{w}_m \in \mathbb{C}^{N_m\times 1}$ for $m \in \{1, \ldots, M\}$. In this case, Eq. (10) becomes $\{1, \ldots, M\}$-mode products between $\mathcal{X}(k)$ and $\{\mathbf{w}^*_n\}_{m=1}^M$:

$$\begin{aligned} y(k) &= \sum_{n_1=1}^{N_1} \cdots \sum_{n_M=1}^{N_M} [\mathcal{X}(k)]_{n_1,\ldots,n_M} [\mathcal{W}]^*_{n_1,\ldots,n_M} \\ &= \sum_{n_1=1}^{N_1} \cdots \sum_{n_M=1}^{N_M} [\mathcal{X}(k)]_{n_1,\ldots,n_M} [\mathbf{w}_1]^*_{n_1} \ldots [\mathbf{w}_M]^*_{n_M} \\ &= \mathcal{X}(k) \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}. \end{aligned} \tag{12}$$

Substituting (8) into (12), ignoring the noise component for simplicity, and applying the $\{1, \ldots, M\}$-mode products yields the following output signal

$$y(k) = \sum_{r=1}^{R} \left[\mathbf{w}_1^{\mathsf{H}} \mathbf{a}^{(1)}(\mathbf{d}_r)\right] \circ \ldots \circ \left[\mathbf{w}_M^{\mathsf{H}} \mathbf{a}^{(M)}(\mathbf{d}_r)\right] s_r(k). \tag{13}$$

Equation (13) shows that the multilinearity imposed on the beamforming tensor $\mathcal{W}$ exploits the separability property of the translation invariant array, i.e., by processing each dimension of $\mathcal{X}(k)$ separately. Due to multilinearity of the tensor beamforming, the cost function $J(\mathcal{W})$ can be rewritten in $M$ equivalent forms, with respect to each subfilter:

$$J(\mathcal{W}) = \mathbb{E}\left[|s_{\text{SOI}}(k) - \mathcal{X}(k) \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}|^2\right] \tag{14}$$

$$= \mathbb{E}\left[|s_{\text{SOI}}(k) - \mathbf{w}_m^{\mathsf{H}} \mathbf{X}_{(m)}(k) \left[\mathbf{w}^{\otimes m}\right]^* |^2\right] \tag{15}$$

$$= \mathbb{E}\left[|s_{\text{SOI}}(k) - \mathbf{w}_m^{\mathsf{H}} \mathbf{u}_m(k)|^2\right], \tag{16}$$

where $\mathbf{u}_m(k) = \mathbf{X}_{(m)}(k)\left[\mathbf{w}^{\otimes m}\right]^* \in \mathbb{C}^{I_m\times 1}$ for $m = 1, \ldots, M$, and $\mathbf{w}^{\otimes m}$ is defined analogously to (2). Note that the $n$-mode unfolding (1) is used in (14) to obtain (15). Deriving (16) with respect to $\mathbf{w}^*_m$ and equating the result to $\mathbf{0} \in \mathbb{C}^{N_m\times 1}$ yields:

$$\frac{\partial J(\mathcal{W})}{\partial \mathbf{w}^*_m} = -\mathbf{p}_m + \mathbf{R}_m \mathbf{w}_m = \mathbf{0} \Rightarrow \hat{\mathbf{w}}_m = \mathbf{R}_m^{-1} \mathbf{p}_m, \tag{17}$$

where $\mathbf{p}_m = \mathbb{E}\left[\mathbf{u}_m(k) s^*_{\text{SOI}}(k)\right] \in \mathbb{C}^{N_m\times 1}$ is the cross-correlation vector between $\mathbf{u}_m(k)$ and $s_{\text{SOI}}(k)$, and $\mathbf{R}_m = \mathbb{E}\left[\mathbf{u}_m(k)\mathbf{u}_m(k)^{\mathsf{H}}\right] \in \mathbb{C}^{N_m\times N_m}$ is the autocorrelation matrix associated with the $m$th subarray.

**Multilinear MMSE beamforming**

Standard optimization methods do not guarantee global convergence when minimizing (14) due to its joint nonconvexity with respect to all the variables. The alternating minimization approach [12, 15] has demonstrated to be a solution to solve the global nonlinear problem in terms of $M$ smaller linear problems. It consists in updating the $m$th mode beamforming filter each time by solving (17) for $\mathbf{w}_m$, while $\{\mathbf{w}_j\}_{j=1,j\neq m}^M$ remain fixed, $m = 1, \ldots, M$, conditioned on the previous updates of the other filters.

Define $\mathcal{X} = [\mathcal{X}(k) \sqcup_{M+1} \ldots \sqcup_{M+1} \mathcal{X}(k-K+1)] \in \mathbb{C}^{N_1\times\ldots\times N_M\times K}$ as the concatenation of $K$ time snapshots of $\mathcal{X}(k)$ along the $(M+1)$th dimension. Let $\mathbf{U}^{(m)} \in \mathbb{C}^{N_m\times K}$ denote the $\{1, \ldots, m-1, m+1, \ldots, N\}$-mode products between $\mathcal{X}$ and $\{\mathbf{w}_j\}_{j=1,j\neq m}^M$:

$$\mathbf{U}^{(m)} = \mathcal{X} \times_1 \mathbf{w}_1^{\mathsf{H}} \ldots \times_{m-1} \mathbf{w}_{m-1}^{\mathsf{H}} \times_{m+1} \mathbf{w}_{m+1}^{\mathsf{H}} \ldots \times_M \mathbf{w}_M^{\mathsf{H}}. \tag{18}$$

It can be shown that $\mathbf{U}^{(m)} = [\mathbf{u}_m(k), \ldots, \mathbf{u}_m(k-K+1)]$. Therefore the sample estimate of $\mathbf{R}_n$ and $\mathbf{p}_n$ are given by:

$$\hat{\mathbf{R}}_n = \frac{1}{K} \mathbf{U}^{(m)} \mathbf{U}^{(m)^{\mathsf{H}}} \tag{19}$$

$$\hat{\mathbf{p}}_n = \frac{1}{K} \mathbf{U}^{(m)} \mathbf{s}^*, \tag{20}$$

where $\mathbf{s} = [s_{\text{SOI}}(k), s_{\text{SOI}}(k-1), \ldots, s_{\text{SOI}}(k-K+1)]^{\mathsf{T}} \in \mathbb{C}^{K\times 1}$. The $m$th order subfilter updating rule is given by $\hat{\mathbf{w}}_m = \hat{\mathbf{R}}_m^{-1} \hat{\mathbf{p}}_m$. The subfilters are estimated in an alternate fashion until convergence, which is attained when the error between two consecutive iterations is smaller than a threshold $\varepsilon$. This procedure is described in Algorithm 1.

The multilinear MMSE beamforming algorithm presents a computational complexity of $\mathcal{O}\left(Q \sum_{m=1}^M N_m^2\right)$, where $Q$ is the number of iterations necessary to attain the convergence. Such an alternating minimization procedure has a monotonic convergence. In this work, we do not assume any prior knowledge on the array response and a random initialization is used. In the chosen array configurations, convergence is usually achieved within 4 or 6 iterations. It is worth mentioning that an analytical convergence analysis of this algorithm is a challenging research topic which is under investigation.

An alternative approach to solve (16) would consist in using a gradient-based algorithm. The idea of this algorithm is similar to

that of [16], therein referred to as TensorLMS. However, such an approach would need small step sizes and convergence can be much slower in comparison with the multilinear MMSE algorithm.

**Algorithm 1** Multilinear MMSE

**procedure** MULTILINEARMMSE($\mathcal{X}$, $\mathbf{s}$, $\varepsilon$)
  $q \leftarrow 1$
  Initialize $e(q)$, $\mathbf{w}_m(q)$, $m = 1, \ldots, M$.
  **repeat**
    **for** $m = 1, \ldots, M$ **do**
      Calculate $\mathbf{U}^{(m)}(q)$ using Equation (18)
      $\hat{\mathbf{R}}_m \leftarrow (1/K)\mathbf{U}^{(m)}\mathbf{U}^{(m)\mathsf{H}}$
      $\hat{\mathbf{p}}_m \leftarrow (1/K)\mathbf{U}^{(m)}\mathbf{s}^*$
      $\mathbf{w}_m(q+1) \leftarrow \hat{\mathbf{R}}_m^{-1}\hat{\mathbf{p}}_m$
    **end for**
    $q \leftarrow q + 1$
    $\mathbf{y}(q) \leftarrow \mathcal{X} \times_1 \mathbf{w}_1(q)^{\mathsf{H}} \ldots \times_M \mathbf{w}_M(q)^{\mathsf{H}}$
    $e(q) = \|\mathbf{s} - \mathbf{y}(q)\|_2^2/K$
  **until** $|e(q) - e(q-1)| < \varepsilon$
**end procedure**

## 4. NUMERICAL RESULTS

Computer experiments were conducted in order to assess the SOI estimation performance and the computational complexity of the proposed tensor beamformer. In this context, $R = 3$ uncorrelated QPSK signals with unitary variance arriving from the directions $(\theta_r, \phi_r)$ rad $\in \left\{\left(\frac{\pi}{3}, -\frac{\pi}{4}\right), \left(\frac{\pi}{6}, \frac{\pi}{3}\right), \left(\frac{\pi}{4}, -\frac{\pi}{6}\right)\right\}$ were considered. The signal corresponding to $r = 1$ was set as SOI. The linear MMSE beamformer (9) was used as benchmark method. Recall that the linear beamformer ignores the multilinear translation invariant structure of the sensor array, by operating over the vectorized form of the received signal tensor. A noise component was added to the observed signals at the array and the signal-to-noise ratio was set to 15 dB. The convergence threshold of the multilinear MMSE algorithm was set to $\varepsilon = 10^{-6}$. The mean performance indices were calculated by averaging the results obtained in $J = 100$ Monte Carlo (MC) realizations. The SOI estimation performance was evaluated in terms of the MSE measure, defined as $\text{MSE} = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{K}\|\mathbf{s}^{(j)} - \langle\mathcal{X}^{(j)}, \mathcal{W}^{(j)^*}\rangle\|_2^2$, where the superscript $(\cdot)^{(j)}$ denotes the $j$th MC realization. The number of FLOPS demanded by each method was computed using the Lightspeed MATLAB toolbox [17]. The Tensorlab toolbox [18] was used to implement the tensor operations involved in the proposed algorithm.

Two simulation scenarios were considered. In the first one, the performance indices were calculated by varying the number $N$ of sensors of the global array for $K = 5000$ samples, as depicted in Fig. 2. In the second scenario, the performance indices were calculated by varying the sample size $K$, as illustrated in Fig. 3. In this case, the global array consisted of $N = 128$ sensors. In both scenarios, the global array was formed by translating a $2 \times 4$ uniform rectangular array, the reference array, along the $x$-axis.

The left plots of Figures 2 and 3 show that the multilinear MMSE algorithm offers a reduced computational complexity compared to the linear (vector) MMSE filter thanks to the exploitation of the array separability, as expected. The gains are particularly more pronounced for $M = 3$ and 4. On the other hand, the right plots of these figures indicate that the MSE of the proposed algorithm is 0.5 dB above that of the linear beamformer. Such a performance gap can be considered negligible in view of the computational gains, and it can be explained due to the loss of optimality of the rank-1 filter. Therefore, the multilinear algorithm offers a reduction on the computation cost with almost no trade-offs in terms of MSE performance for multilinear translation invariant sensor arrays.

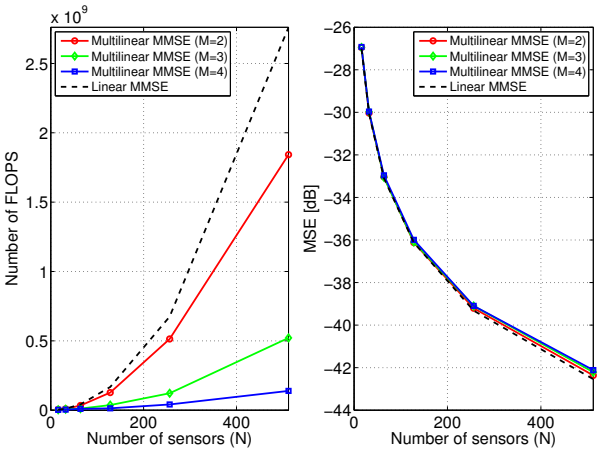



**Fig. 2**. Performance for a varying number of sensors for $K = 5000$ samples.

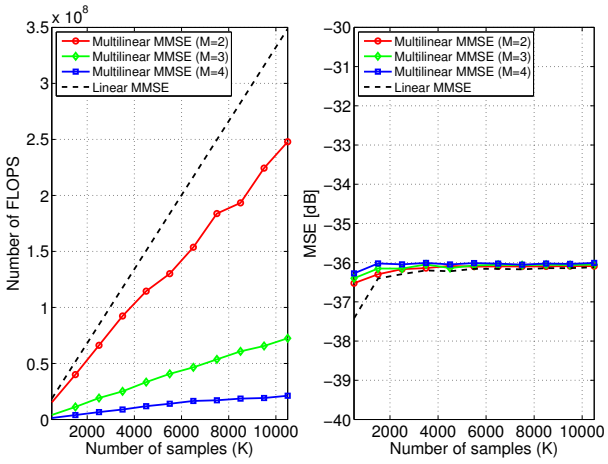



**Fig. 3**. Performance for a varying sample size for $N = 128$ sensors.

## 5. CONCLUSION AND PERSPECTIVES

There has been a growing interest on array processing systems capable of processing data received by a massive number of sensors. Multilinear array models are interesting in this context since they represent a sensor array in simpler terms, allowing the development of computationally efficient array processing methods. In this work, a tensor beamformer that exploits the separability present in multilinear translation invariant arrays model was presented. Numerical results showed that the presented method has a reduced processing time with almost no performance loss compared with the linear beamforming solution that operates on the global array by ignoring the array manifold separability. A future work includes the extension of the proposed tensor beamforming to the wideband filtering scenario, where separability can be further exploited in the joint space-time domain. In this work, we have adopted a rank-1 representation for the beamforming filter. The use of low-rank tensor beamformers will be addressed in the future.

## 6. REFERENCES


[1] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[2] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*, Springer Science & Business Media, 2001.

[3] J. C. Mosher and R. M. Leahy, "Recursive MUSIC: a framework for EEG and MEG source localization," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 11, pp. 1342–1354, 1998.

[4] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[5] L. H. Lim and P. Comon, "Multiarray signal processing: Tensor decomposition meets compressed sensing," *Comptes Rendus Mecanique*, vol. 338, no. 6, pp. 311–320, 2010.

[6] F. Roemer, *Advanced algebraic concepts for efficient multi-channel signal processing*, Ph.D. thesis, Universitätsbibliothek Ilmenau, 2013.

[7] S. Miron, S. Yang, D. Brie, and K. Wong, "A multilinear approach of direction finding using a sensor-array with multiple scales of spatial invariance," *IEEE Transactions on Aerospace and Electronic Systems*, p. 00, 2015.

[8] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3198–3213, 2008.

[9] J. P. C. L. Da Costa, F. Roemer, M. Haardt, and R. T. de Sousa Jr., "Multi-dimensional model order selection," *EURASIP Journal on Advances in Signal Processing*, vol. 26, pp. 1–13, 2011.

[10] X. F. Gong and Q. H. Lin, "Spatially constrained parallel factor analysis for semi-blind beamforming," in *2011 Seventh International Conference on Natural Computation (ICNC),*. IEEE, 2011, vol. 1, pp. 416–420.

[11] R. K. Miranda, J. P. C. L. da Costa, F. Roemer, A. L. F. de Almeida, and G. Del Galdo, "Generalized sidelobe cancellers for multidimensional separable arrays," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (IEEE CAMSAP 2015)*, Cancun, Mexico, Dec. 2015.

[12] L. N. Ribeiro, A. L. F. de Almeida, and J. C. M. Mota, "Identification of separable systems using trilinear adaptive filtering," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (IEEE CAMSAP 2015)*, Cancun, Mexico, Dec. 2015.

[13] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[14] L. H. Lim and P. Comon, "Blind multilinear identification," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1260–1280, 2014.

[15] Ö. Filiz and A. Yener, "Rank constrained temporal-spatial matrix filters for cdma systems," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 1974–1979, 2004.

[16] M. Rupp and S. Schwarz, "A tensor LMS algorithm," in *2015 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2015.

[17] T. Minka, "The Lightspeed MATLAB toolbox, efficient operations for MATLAB programming, version 2.2," *Microsoft Corp.*, 2007.

[18] L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v2. 0, january 2014," *URL: http://www. tensorlab. net*.

# Bibliography


[1] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[2] A. V. Oppenheim, R. W. Schafer, J. R. Buck, *et al.*, *Discrete-time signal processing*, vol. 2. Prentice hall Englewood Cliffs, NJ.

[3] G. B. Giannakis, F. Bach, R. Cendrillon, M. Mahoney, and J. Neville, "Signal processing for big data," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 15–16, 2014.

[4] V. H. Nascimento, "Improving the initial convergence of adaptive filters: variable-length LMS algorithms," in *2002 14th International Conference on Digital Signal Processing, 2002. DSP 2002.*, vol. 2, pp. 667–670, IEEE, 2002.

[5] T. Ogunfunmi and S.-L. Chang, "Second-order adaptive Volterra system identification based on discrete nonlinear Wiener model," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 148, no. 1, pp. 21–29, 2001.

[6] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 3, pp. 609–615, 1983.

[7] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2943–2959, 1998.

[8] R. C. De Lamare and R. Sampaio-Neto, "Reduced-rank adaptive filtering based on joint iterative optimization of adaptive filters," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 980–983, 2007.

[9] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan (Part I: Communications)*, vol. 67, no. 5, pp. 19–27, 1984.

[10] P. S. Diniz, *Adaptive filtering*. Springer, 1997.

[11] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[12] S. Treitel and J. L. Shanks, "The design of multistage separable planar filters," *IEEE Transactions on Geoscience Electronics*, vol. 9, no. 1, pp. 10–27, 1971.

[13] R. E. Twogood and S. K. Mitra, "Computer-aided design of separable two-dimensional digital filters," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 25, pp. 165–169, 1977.

[14] J.-P. Calliess, M. Mai, and S. Pfeiffer, "On the computational benefit of tensor separation for high-dimensional discrete convolutions," *Multidimensional Systems and Signal Processing*, vol. 23, no. 1-2, pp. 255–279, 2012.

[15] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 94–106, 2015.

[16] M. Rupp and S. Schwarz, "A tensor LMS algorithm," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3347–3351, IEEE, 2015.

[17] G. Favier, A. Y. Kibangou, and T. Bouilloc, "Nonlinear system modeling and identification using Volterra-PARAFAC models," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 1, pp. 30–53, 2012.

[18] M. F. Duarte and R. G. Baraniuk, "Kronecker compressive sensing," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 494–504, 2012.

[19] R. K. Miranda, J. P. C. L. da Costa, F. Roemer, A. L. F. de Almeida, and G. D. Galdo, "Generalized sidelobe cancellers for multidimensional separable arrays," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (IEEE CAMSAP 2015)*, (Cancun, Mexico), Dec. 2015.

[20] D. Muti and S. Bourennane, "Survey on tensor signal algebraic filtering," *Signal Processing*, vol. 87, no. 2, pp. 237–249, 2007.

[21] A. Cichocki, "Era of big data processing: a new approach via tensor networks and tensor decompositions," *arXiv preprint arXiv:1403.2048*, 2014.

[22] D. Muti and S. Bourennane, "Multidimensional filtering based on a tensor approach," *Signal Processing*, vol. 85, no. 12, pp. 2338–2353, 2005.

[23] M. Boizard, G. Ginolhac, F. Pascal, and P. Forster, "Low rank tensor STAP filter based on multilinear svd," in *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012*, pp. 305–308, IEEE, 2012.

[24] F. Wang, P. Zhang, B. Qian, X. Wang, and I. Davidson, "Clinical risk prediction with multilinear sparse logistic regression," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 145–154, ACM, 2014.

[25] L. N. Ribeiro, J. C. M. Mota, and A. L. F. de Almeida, "Trilinear Wiener filtering: Application to equalization problems," in *XXXI Simpósio Brasileiro de Telecomunicações, Fortaleza, Brazil*, SBrT, 2013.

[26] O. Filiz and A. Yener, "Rank constrained temporal-spatial matrix filters for cdma systems," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 1974–1979, Nov 2004.

[27] M. Andersson, O. Burdakov, H. Knutsson, and S. Zikrin, "Global search strategies for solving multilinear least-squares problems," *Research Report LiTH-MAT-R–2011/17–SE*, 2011.

[28] B. Norell, O. Burdakov, M. Andersson, and H. Knutsson, "Approximate spectral factorization for design of efficient sub-filter sequences," *Research Report LiTH-MAT-R-2011/14-SE*, 2011.

[29] P. Comon, "Tensors: a brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, 2014.

[30] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[31] I. V. Cavalcante, A. L. de Almeida, and M. Haardt, "Tensor-based approach to channel estimation in amplify-and-forward MIMO relaying systems," in *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014*, pp. 445–448, IEEE, 2014.

[32] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[33] L. Ribeiro, A. de Almeida, and J. C. M. Mota, "Identification of separable systems using trilinear filtering," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (IEEE CAMSAP 2015)*, (Cancun, Mexico), pp. 189–192, Dec. 2015.

[34] A. Yener, R. D. Yates, and S. Ulukus, "Combined multiuser detection and beamforming for CDMA systems: filter structures," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 5, pp. 1087–1095, 2002.

[35] L. Ljung, *System identification*. Springer, 1998.

[36] L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v2.0," *Available online, URL: www.tensorlab.net*, 2014.

[37] T. Minka, "The Lightspeed MATLAB toolbox, efficient operations for MATLAB programming, version 2.2," *Microsoft Corp.*, 2007.

[38] L. N. Ribeiro, A. L. F. de Almeida, and J. C. M. Mota, "Tensor beamforming for multilinear translation invariant arrays," in *accepted to the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*.

[39] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[40] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1260–1280, 2014.

[41] L. Tong and S. Perreau, "Multichannel blind identification: From subspace to maximum likelihood methods," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1951–1968, 1998.

[42] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, 2011.

[43] W. Jenkins, A. W. Hull, J. C. Strait, B. A. Schnaufer, and X. Li, *Advanced Concepts in Adaptive Signal Processing*, vol. 365. Springer Science & Business Media, 2012.