System identification under non-negativity constraints - Applications in adaptive filtering and hyperspectral image analysis

Jie Chen

System identification under non-negativity constraints

Applications in adaptive filtering and hyperspectral image analysis

A dissertation presented

by

Jie CHEN

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Optimization and security of systems

of

University of Technology of Troyes

January 28, 2013

Jury :

Reviewers :	Christian JUTTEN	-	Grenoble INP
	Jean-Yves TOURNERET	-	INP Toulouse
Examinators :	Jose-Carlos M. BERMUDEZ	-	Univ. Federal de Santa Catarina
	Jocelyn CHANUSSOT	-	Grenoble INP
	Pascal LARZABAL	-	Univ. Paris Sud 11
Advisors :	Cédric RICHARD	-	Univ. de Nice Sophia-Antipolis
	Paul HONEINE	-	Univ. de Technologie de Troyes

Acknowledgments

This thesis is the end of my journey in obtaining my PhD degree. I have not traveled in a vacuum in this journey. This thesis has been kept on track and been through to completion with the support and encouragement of numerous people including my well-wishers, friends, colleagues and various institutions. It is a pleasant aspect that I now have the opportunity to express my gratitude to all those people who made this thesis possible and an unforgettable experience for me.

It was eight years ago when I first arrived to France to pursue my study. Two years of engineering education and over three years of doctoral research in France are indelible memory that will influence me through my life. In all these years of my study and research, even in the foreseeable future research journey, the most influential person is definitely my advisor, Mr. Cédric Richard. This acknowledgment is addressed to him, not only for his guidance of my thesis, but also for directing my study and research career. We have recognized each other eight years ago, the first time I was in France. In those years, through his engineering courses, guided experimental work, guidance of internship, he introduced me step by step into the domain of signal processing. In the year of 2009, I came to France again to start my PhD thesis. His passion, guidance, and discipline have been indispensable to my development as a researcher and as a person over these years. He taught me to systematically formulate and analyze the problem rather than relying on intuition, and his focussed guidance helped me substantially during the research and writing of articles. This thesis would not have been possible without the help, support and patience of Cédric. Numerous discussions, writing and revising of articles with him made me grow, year by year. He also provided me with many opportunities to get in touch with other distinguished professors and to extend my perspective in the signal processing community. I would like to express my sincere gratitude to him.

My co-advisor, Mr. Paul Honeine, has always been there to listen and to give advice. I am deeply grateful to him for the valuable discussions, suggestions, comments and ideas for my work. Also, as I have conducted my research outside Troyes, he was always kindly enough to take time to help me deal with administrative documents and affairs. I thank him for all this support, and also for his kindness and friendship during these years.

I am especially indebted to Mr. Jose-Carlo M. Bermudez, professor of Federal University of Santa Catarina, Brazil. He gave me invaluable guidance in both methodology and technical details of designing and analyzing of adaptive filter algorithms. Discussing, writing and revising papers with him always help me to understand and learn more on the topic of adaptive filtering. I am grateful to him for all his contributions, comments to my thesis, and his kindness.

I gratefully acknowledge Langrange Laboratory of University of Nice Sophia-Antipolis, that accepted me to conduct my research within the group TDSI and provided me with all possible convenience for my work.

I would like to express my gratitude to other professors who helped me in my research work or other aspects : Régis Lengellé at UTT, Henri Lantéri, André Ferrari, Céline Theys, Jean-Pierre Folcher, David Mary and Claude Aime of Lagrange Laboratory at University of Nice, and Jean-Yves Tourneret at INP Toulouse. I would also thank Isabelle Leclercq, Pascale Denise at UTT and Caroline Daire, Catherine Blanc at University of Nice, for their help in administrative affairs.

I've also been fortunate to have a great group of friends. My time at UTT and University of Nice was made enjoyable in large part due to the many friends and groups that became a part of my life. This includes Jie Zhang, Yunfei Fang, Peng Li, Tian Wang, Silvia Paris, Ivan Milic, Nguyen Hoang Nguyen, Liyong Liu, Wei Gao. They had not only been my mates for discussions of my research, but also the confidants whose help I could count on.

Last but not the least, I would like to thank all my family members for their love and encouragement in my pursuit of this research, which would have been imponderable task for me without their support.

Table of contents

List of figures

List of tables

List of symbols and abbreviations

General notation

a, A	lowercase and uppercase lightface letters denote scalars
$\boldsymbol{a}$	lowercase boldface letters denote column vectors
$\boldsymbol{A}$	uppercase boldface letters denote matrices
$\approx$	approximated to
$a \sim F$	a follows distribution F
$\sum$, $\sum_{i=1}^{N}$	summation
$\log_{10}$	logarithm to the base 10
$\lvert a \rvert$	absolute value of a
$\lVert \boldsymbol{a} \rVert$	Euclidean norm of a vector $\boldsymbol{a}$
$\lVert \boldsymbol{a} \rVert_p$	ℓ_p-norm of a vector $\boldsymbol{a}$
$\boldsymbol{a} \succeq 0$	components of vector $\boldsymbol{a}$ are non-negative
$E\{\boldsymbol{a}\}$	expectation of $\boldsymbol{a}$
$\max(a, b)$	larger value in a and b

Sets and spaces

$\{\boldsymbol{x} : P\}$	set of $\boldsymbol{x}$ with property P
$\{i, \ldots, j\}$	set of integers between i and j
$\mathbb{R}^M$	M-dimensional Euclidean space
$\mathcal{H}$	functional Hilbert space

Matrix and vector

$[a_1, a_2, \ldots, a_N]$	$(1 \times N)$ vector with components a_i, $i = 1, \ldots, N$
$[\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N]$	matrix with column $\boldsymbol{a}_i$, $i = 1, \ldots, N$
$[\boldsymbol{A}]_{ij}$	(i, j)th element of a matrix $\boldsymbol{A}$

$\left(\begin{array}{c\|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right)$	partitioned matrix, consisting of submatrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$
$\boldsymbol{I}$	identity matrix
$\mathbf{1}$	vector of ones
$\mathbf{0}$	vector of zeros
$\boldsymbol{A}^\top$	transpose of a matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of a matrix $\boldsymbol{A}$
trace$\{\boldsymbol{A}\}$	trace of a matrix $\boldsymbol{A}$
Diag$\{\boldsymbol{A}\}$	vector composed by diagonal elements of a matrix $\boldsymbol{A}$
$\boldsymbol{D_a}$	a diagonal matrix with diagonal elements given by $\boldsymbol{a}$
$\|\boldsymbol{A}\|_{1,1}$	sum of the column ℓ_1-norms of a matrix $\boldsymbol{A}$

Abbreviations

EMSE	Excess mean squared error
FCLS	Fully constrained least-squares
i.i.d.	Independently and identically distributed
KKT	Karush-Kuhn-Tucker condition
K-Hype	Kernel hyperspectral unmixing algorithm
LASSO	Least absolute shrinkage and selection operator
LMS	Least-mean-square algorithm
MSE	Mean squared error
MVSA	Minimum volume simplex analysis
NLMS	Normalized least-mean-square algorithm
NNLMS	Non-negative least-mean-square algorithm
NMF	Non-negative matrix factorization
RKHS	Reproducing kernel Hilbert space
RMSE	Root mean square error
SK-Hype	Super kernel hyperspectral unmixing algorithm
SNR	Signal-to-noise radio
SVM	Support vector machine
SVR	Support vector regression
VCA	Vertex component analysis

Chapter 1

Introduction

Contents

1.1 Context

1.1.1 System identification and roles of constraints

The problem of inferring the structure of a system from observations of its behavior is an ancient one with many ramifications. The literature on the subject is vast, having its roots in the philosophical problems of the nature of reality and our inference of it from sensations. Plato's famous allegory of the prisoners in the cave who, "like ourselves ... see only their shadows, or the shadows of one another, which the fire throws on the opposite wall of the cave", epitomizes the inherent uncertainty and tenuous nature of inference processes that we all perform and take for granted in our everyday life [Gaines 1977]. In nowadays science community, constructing models from observed data is a fundamental element. Several methodologies and nomenclatures have been developed in different application areas. These techniques are often known under the term *System identification* in the signal processing and control community [Ljung 2010]. The term system identification was coined by Lotfi Zadeh in 1962. He defined system identification as [Zadeh 1962] :

Identification is the determination, on the basis on input and output, of a system within a specified class of systems, to which the system under test is equivalent.

This definition has become somewhat limited with the development of understanding and modeling of theoretical and practical problems, it remains nevertheless the core in most of tasks. As the art and science of building mathematical models of dynamic systems from observed input-output data, and being seen as the interface

between the real world of applications and the mathematical world of model abstractions, system identification plays a crucial role in the development of techniques for stationary and non-stationary signal processing.

System identification is a very large topic, with different techniques that depend on the character of the models to be estimated [Ljung 1999, Gevers 2006]. In many applications, system identification is concerned with the estimation of a set of coefficients or parameters characterizing an unknown system with an underlying mathematical structure. This parametric category of system identification problems is typically formulated as an optimization problem. A frequent and useful formulation is to minimize a predefined criterion, which generally involves available output/input observations, with respect to the parameters [Ljung 1999]

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \tag{1.1}$$

with $\boldsymbol{\theta}$ the parameters to estimate and Ψ the criterion. Approaches such as least-square regression, maximum likelihood estimation fall into this form.

Rather than leaving the parameters totally free and relying completely on the data, in practical it is often desirable to introduce constraints into the parameter space [Johansen 1998, Yeredor 2006]. With the introduced constraints on the parameter space, the system identification, or equivalent parameter estimation problem, is rewritten by restricting parameters into a feasible set Θ

$$\begin{aligned} \boldsymbol{\theta}^* = &\arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \\ &\text{subject to } \boldsymbol{\theta} \in \Theta. \end{aligned} \tag{1.2}$$

Introducing constraints is motivated by, but not limited to, the following reasons :

- Incorporating prior knowledge about the system so as to improve the estimation accuracy and the result interpretability by effectively reducing the size of feasible set, including assumed ranges of variables, zero or pole positions of system responses, the smoothness of signals, etc. This is usually a basic motivation for adding constraints.
- Avoiding physically absurd and uninterpretable results, such as non-negativity constraints imposed to preserve inherent characteristics of solutions corresponding to amounts and measurements associated to, for instance, frequency counts, pixel intensities and chemical concentrations. This is another frequent motivation for adding constraints.
- Avoiding a trivial minimizer of the criterion, where the associated optimization problem cannot yield a useful solution without excluding trivial solutions from the feasible set. For example, unit norm constraint is frequently imposed on homogeneous equations as we are obviously not interested in the null trivial solution.
- Imposing certain "natural" structure on some of the signal involved, such as imposing the Toeplitz structure on a matrix, or sparsity on a dictionary of coefficients.

– Other facts, such as guaranteeing the stability of the resulting estimated system, mitigating the bias induced by additive output noise or by the use of an inconsistent criterion, etc.

In the case where both the function Ψ and the set Θ are convex, the above optimization problem (1.2) is convex and has a unique solution. A frequently studied form is formulated using inequality and equality constraints

$$\begin{aligned} \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \\ \text{subject to } g_i(\boldsymbol{\theta}) \leq 0 \\ h_i(\boldsymbol{\theta}) = 0 \end{aligned} \tag{1.3}$$

with functions Ψ and g_i convex, and h_i affine. This standard form has been largely discussed in literature of convex optimization [Boyd 2004, Luenberger 2008]. Many practical problems, including those to be discussed in our work, can be formulated in this standard from.

In this thesis, our study mainly focuses on problems under the non-negativity and the ℓ_1-norm constraints.

1.1.2 Non-negativity constraints in system identification

In many real-life phenomena including biological and physiological ones, due to the inherent physical characteristics of systems under investigation, non-negativity is a desired constraint that can be imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results [Chen 2009a]. For instance, in the study of frequency counts, material fraction, pixel intensities and chemical concentration, it makes sense to respect this constraint. The feasible set with respect to the non-negativity constraint writes

$$\Theta_+ = \{\boldsymbol{\theta} : \theta_i \geq 0, \quad \forall i\}. \tag{1.4}$$

This constraint has received growing attention from the signal processing community during the last decade. It finds applications in several problems, to cite a few in what followings.

In support vector machine problems, we aim at maximizing predictive accuracy while automatically avoiding over-fitting to the data, by minimizing the norm of regressor subject to constraints with respect to classification or regression rules. Usually, this formulated problem is solved in its dual space, which leads to quadratic optimization problems subject to non-negativity constraints imposed on duality variables [Vapnik 2000]. This type of problem will be also encountered in the duality of our nonlinear hyperspectral data unmixing algorithms in Chapter 4 and 5.

Digital image data are represented using non-negative matrix arrays, due to the nature of the pixel intensities. Non-negativity is therefore a desired constraint during the image processing. In the image deblurring problem with the known point spread function or convolution operator, the estimated original image is often reconstructed by solving the non-negative least-square problem [Puetter 2005]. Literature

has shown that enforcing a non-negativity constraint could produce a much more accurate approximate solution [Bardsley 2008].

In the context of remote sensing, due to reflectance of photons and diversity of materials, the reflectance observed in each pixel, is a mixture of spectra of several material signatures. The spectral unmixing aims to decompose each mixed pixel into its pure spectra, and estimate the abundances associated to each material. To be physically interpretable, the estimated material signatures and fractional abundances should satisfy the non-negativity constraints, plus a frequently imposed sum-to-one constraint [Keshava 2002]. Considering the linear mixture model, a possibility to determine the material signatures and abundances at the same time is to solve the non-negative matrix factorization problem [Pauca 2006]. In the case where the material signatures have been determined in advance, the unmixing boils down to a non-negative least-square problem with additional constraints [Heinz 2001]. In the nonlinear unmixing cases, the non-negativity constraints are also necessary, as discussed in later chapters of this thesis.

In some applications of wireless sensor networks, such as concentration monitoring or thermal diffusion monitoring, the fields are described by non-negative values [Chen 2010a, Waterschoot 2011]. When inferring these fields from measures obtained at sensors, it is often assumed that the field can be expressed by a weighted sum of basis functions parametrized by coordinates of observed points. In order to ensure that the inference is non-negative at any position, the weights are constrained to be non-negative if non-negative basis functions are used.

There are many other applications involving the non-negativity constraints, such as speech recognition, text mining, deconvolution for room impulse response estimation, etc., see [Saul 2003, Chen 2009a] for other examples.

1.1.3 ℓ_1-norm constraints in system identification

Another constraint largely discussed in the signal processing community in recent years is the ℓ_1-norm constraint, due to its favorable properties in structuring the estimate. This thesis involves two types of ℓ_1-norm constraint.

The first one is the constant ℓ_1-norm constraint that restricts solutions on the ℓ_1-ball, i.e. :

$$\Theta_{\ell_1} = \{\boldsymbol{\theta} \,:\, \|\boldsymbol{\theta}\|_1 = \delta\} \tag{1.5}$$

with the vector ℓ_1-norm $\|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i|$. If this constraint is considered with the non-negativity constraint, the above set boils down to the ℓ_1-ball in the non-negative orthant, i.e. a sum constraint

$$\begin{aligned} \Theta_{\ell_1^+} &= \Theta_{\ell_1} \cap \Theta_+ \\ &= \{\boldsymbol{\theta} \,:\, \sum_i \theta_i = \delta \text{ and } \theta_i \geq 0 \quad \forall i\} \end{aligned} \tag{1.6}$$

This constraint plays an important role in several problems. Especially the case where δ takes 1, means that each value of θ_i represents a proportion of its associated

mode. For instance, in the multi-kernel learning problem, combination of candidate kernels is used instead of a single kernel to improve the classification/regression accuracy [Gönen 2011]. In order to guarantee the positive-definiteness of the constructed kernel and avoid trivial solutions, the weight coefficients to be estimated should be an element of $\Theta_{\ell_1^+}$. Another example is the hyperspectral data unmixing problem, where the fractional abundances associated to different materials have to be determined. To be physically interpretable, the driving abundances are often required to satisfy two constraints : all abundances must be nonnegative, and their sum must be equal to one [Keshava 2002], i.e., an element of the set $\Theta_{\ell_1^+}$.

The other important constraint is defined by limiting the ℓ_1-norm of a vector, via the inequality expression

$$\Theta_{\ell_1'} = \{\boldsymbol{\theta} \,:\, \|\boldsymbol{\theta}\|_1 \leq \delta\} \tag{1.7}$$

The importance of this constraint arises with the emergence of the compressed sensing (CS) theory. While the exact sparsity measurement with ℓ_0-norm has been shown a NP-hard problem, the constraint $\Theta_{\ell_1'}$ allows an exact reconstruction with a high probability and it is much more computationally tractable [Candès 2008]. Thus ℓ_1-norm based algorithms have a large range of applications due to its sparsity-inducing property. Many real-life systems admit sparse representations with few non-zero coefficients. Examples include multipath wireless communication channels where reflections reach the receiver with long delays, imaging, video, etc [Bruckstein 2009, Berger 2010, Elad 2010]. Many of the above applications require adaptive estimation techniques with minimum computational complexity due to time-varying dynamics and a large number of potential parameters. Wireless communication channels are a typical example of the above setup. The wireless channel is described by sparse fading rays and long zero samples and thus admits a sparse representation [Cotter 2002]. If power amplifiers at the transmitter and receiver ends operate in the linear regime, the channel is represented by a time-varying linear filter whose unit sample response is a sparse vector. In the spectral unmixing problem, the number of materials to be identified in the scene represents only a very small portion of a large candidate library. Sparse regression is a direction explored for the unmixing. The problem is formulated as that of fitting the observed hyperspectral vectors with sparse mixtures of spectral signatures, from a large spectrum library [Iordache 2011].

Study combining the non-negativity constraints Θ_+ and the ℓ_1-norm constraint $\Theta_{\ell_1'}$ has also been investigated within the context of machine learning [O'Grady 2008, Khajehnejad 2011].

Note that this constraint is also closely related to the non-negativity constraint as (1.7) can be reformulated with two non-negative vectors such that

$$\Theta_{\ell_1'} = \{\boldsymbol{\theta} = \boldsymbol{\theta}^+ - \boldsymbol{\theta}^- \,:\, \mathbf{1}^\top\boldsymbol{\theta}^+ + \mathbf{1}^\top\boldsymbol{\theta}^- \leq \delta \text{ and } \boldsymbol{\theta}^+ \in \Theta_+,\, \boldsymbol{\theta}^- \in \Theta_+\} \tag{1.8}$$

1.2 Motivation

The objective of this thesis is to investigate theories and algorithms for system identification under side constraints, in particular the non-negativity constraint and ℓ_1-type constraint over the vector of parameters to estimate. The motivation of this thesis is twofold regarding to the linear case and the nonlinear case respectively. In the linear case, the thesis focuses on designing online algorithms with sequential inputs under such constraints for adaptive filter design and analysis. In the nonlinear case, the thesis considers a particular system identification problem - nonlinear unmixing of hyperspectral images and studies how the non-negativity and constant-sum constraints (a trivial ℓ_1-norm constraint) are involved in this case.

The linear system identification problem under non-negativity constraint consisting of minimizing the mean-squared error can be formulated as a quadratic optimization problem. A variety of methods have been developed to tackle non-negative least-square problems (NNLS), including active set methods, projected gradient methods, interior point methods, etc. In recent years, multiplicative update algorithms have also become popular for dealing with the non-negativity constraint, e.g., with the non-negative matrix factorization (NMF) problem. Many approaches and their variants have also been proposed for the ℓ_1-norm constraint, including interior points methods, homotopy method and its variants, and greedy pursuits. These algorithms however require batch processing, which is not suitable for online system identification problems. Therefore, this thesis will firstly investigate online adaptive algorithms subject to the non-negativity constraint and processing inputs sequentially. After that, we intend to extend this online algorithm so as to enable it to solve ℓ_1-norm constraint problem in an online manner.

In the nonlinear case, instead of solving a specific quadratic problem, we focus on how to model a nonlinear system with such constraints. As briefly presented previously, due to the physical interpretation of fractional abundances in hyperspectral image processing, the non-negativity and sum-to-one constraints are frequently imposed. When conducting linear unmixing, these constraints usually lead to a constrained quadratic programming problem. However, it is noteworthy that in an observed scene, photons may interact among several materials due to reflectance and intimate interaction of substances. Deriving a more reasonable and general nonlinear model for describing these effects is important in hyperspectral data analysis. Incorporating the two mentioned constraints into the nonlinear model is also of interest for unmixing data. We propose a general nonlinear model, and the associated constrained unmixing problem, and to derive the corresponding optimization method.

1.3 Main contributions

Concentrating on the problems and the motivations presented above, the main contributions of this thesis are the following :

- Derivation of an online system identification algorithm under the non-negativity constraints, i.e. non-negative least-mean-square algorithm (NNLMS) and analysis of the stochastic behavior of this algorithm.
- Proposition of three useful variants of NNLMS (Normalized NNLMS, Exponential NNLMS, Sign-Sign NNLMS) to improve the NNLMS properties in different senses and extension of the stochastic behavior analysis of these variants for nonstationary enviroments.
- Proposition of an algorithm to solve the identification problem with the ℓ_1-norm constraint in an online manner based on NNLMS, and derivation of the corresponding stochastic behavior models.
- Proposition of a general linear mixture/nonlinear fluctuation model for hyperspectral mixture nonlinearity modeling.
- Development of techniques for material abundance estimation of hyperspectral images by solving the optimization problem in a reproducing kernel Hilbert space.
- Generalization of the above nonlinear unmixing method to develop an intelligent unmixing algorithm that can automatically balance the linear and nonlinear components in the model automatically.
- Integration of spatial-correlation of hyperspectral images into the nonlinear unmixing problem and derivation of the associated algorithm.

1.4 Thesis organization

The main body of this thesis is divided in two parts. The first part, consists of Chapter 2 and 3, and concerns online adaptive algorithms for constrained linear system identification problems. In Chapter 2, we present the central method of this part, the non-negative least-mean square algorithm. In Chapter 3, we exploit how this algorithm can be generalized to address the identification problem with the ℓ_1-norm constraint. The second part, consists of Chapter 4 and 5, and investigates nonlinear spectral unmixing of hyperspectral data under the non-negativity and the sum-to-one (constant ℓ_1-norm) constraints. In Chapter 4, we introduce a new nonlinear mixture model and associated kernel-based unmixing scheme under such constraints. In Chapter 5, we extend the proposed nonlinear unmixing method to take the spatial correlation of pixels into account, using ℓ_1-norm regularization.

Part I - Online system identification under non-negativity constraints

Chapter 2 : Considering the non-negativity constraint in online system identification problems, we propose a general method for system identification under non-negativity constraints. We derive the so-called non-negative least-mean-square (NNLMS) algorithm based on stochastic gradient descent, and we analyze its first-order and second-order convergence properties. In addition, in order to extend the adaptability of this algorithm in some practical applications, we derive useful va-

riants of the NNLMS algorithm. Each of these variants is derived to improve the NNLMS properties in some sense. In order to study these algorithms in a more practical context, we derive their stochastic behavior for non-stationary environments.

Chapter 3 : The ℓ_1-norm constraint has received considerable attention in many applications due to physical constraints and its sparsity-inducing property, as the sparsity has been one of the most popular topics in signal processing. Firstly we briefly present an online algorithm with constant ℓ_1-norm. After, examining the fact that ℓ_1-norm of a vector can be split into the sum of two non-negative vectors with some proper transformation, see (1.8), we extend the proposed NNLMS algorithm to address the sparse online system identification problems, and we derive the stochastic behavior model of convergence of this new algorithm.

Part II - Constrained nonlinear system identification : the problem of hyperspectral image unmixing

Chapter 4 : Spectral unmixing is an important issue to analyze remotely sensed hyperspectral data. Due to the physical properties, the non-negativity and the sum-to-one constraints are frequently imposed on the abundance vector to be estimated. Although the linear mixture model has obvious practical advantages, there are many situations in which it may not be appropriate and could be advantageously replaced by a nonlinear one. We hence formulate a new kernel-based paradigm that relies on the assumption that the mixing mechanism can be described by a linear mixture of endmember spectra, with additive nonlinear fluctuations defined in a reproducing kernel Hilbert space. The parameters in the linear part are considered as abundances and subject to such constraints. This family of models has a clear interpretation, and allows to take complex interactions of material spectral signatures into account.

Chapter 5 : Incorporating spatial information into hyperspectral unmixing procedures has been shown to have a positive effect on the estimation of fractional abundances, due to the inherent spatial-spectral duality in hyperspectral scenes. Benefiting from our proposed unmixing scheme, we investigate how to incorporate spatial correlation into a nonlinear abundance estimation process. A nonlinear unmixing algorithm operating in reproducing kernel Hilbert spaces, coupled with an ℓ_1-type spatial regularization, is thus derived.

At the end of this thesis, Chapter 6 summarizes our contributions and discusses possible extensions and other open problems for future works.

1.5 List of related publications

Journal articles

1. Jie Chen, Cédric Richard and Paul Honeine. "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model". *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pages 480-492, Jan. 2013.

2. Jie Chen, Cédric Richard, Jose-Carlos M. Bermudez and Paul Honeine. "Nonnegative least-mean-square algorithm". *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pages 5525-5535, Nov. 2011.

3. Jie Chen, Cédric Richard and Paul Honeine. "Nonlinear abundance estimation of hyperspectral images with ℓ_1-norm spatial regularization". *IEEE Transactions on Geoscience and Remote Sensing*, 2012 (second round review).

4. Jie Chen, Cédric Richard, Jose-Carlos M. Bermudez and Paul Honeine. "Variants of non-negative least mean square algorithm and converge analysis". *IEEE Transactions on Signal Processing*, 2012 (second round review).

5. Paul Honeine, Cédric Richard, Jose-Carlos M. Bermudez, Jie Chen and Hichem Snoussi. "A decentralized approach for nonlinear prediction of time series data in sensor networks". *EURASIP Journal on Wireless Communications and Networking, special issue on "Theoretical and algorithmic foundations of wireless adhoc and sensor networks"*, 12 pages, 2010.

International conference articles

6. Jie Chen, Cédric Richard, André Ferrari, and Paul Honeine. "Nonlinear unmixing of hyperspectral data with partially linear least-squares support vector machine". *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

7. Jie Chen, Cédric Richard, Paul Honeine and Jean-Yves Tourneret. "Prediction of rain attenuation series with discretized spectral model". *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Munich, Germany, Jul. 2012.

8. Jie Chen, Cédric Richard and Paul Honeine. "Nonlinear unmixing of hyperspectral images with multi-kernel learning". *IEEE Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS)*, Shanghai, China, Jun. 2012.

9. Jie Chen, Cédric Richard and Paul Honeine. "A novel kernel-based nonlinear unmixing scheme of hyperspectral images". *The 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove (CA), USA, Nov. 2011.

10. Jie Chen, Cédric Richard, Jose-Carlos M. Bermudez and Paul Honeine. "A Modified non-negative LMS algorithm and its stochastic behavior analysis".

The 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove (CA), USA, Nov. 2011.

11. Jie Chen, Cédric Richard, Henri Lantéri, Céline Theys and Paul Honeine. "Online system identification undernon-negativity and ℓ_1-norm Constraints - Algorithm and weight behavior analysis". *The 19th European Conference on Signal Processing (EUSIPCO)*, Barcelone, Spain, Aug. 2011.

12. Jie Chen, Cédric Richard, Paul Honeine, Céline Theys and Henri Lantéri. "A gradient based method for fully constrained least-squares unmixing of hyperspectral images". *IEEE Workshop on Statistical Signal Processing (SSP)*, Nice, France, Jun. 2011.

13. Jie Chen, Cédric Richard, Paul Honeine and Jose-Carlos M. Bermudez. "Non-negative distributed regression for data inference in wireless sensor networks". *The 44th Asilomar Conference on Signals, Systems, and Computers (ASILOMAR)*, Pacific Grove (CA), USA, Nov. 2010.

14. Jie Chen, Cédric Richard, Paul Honeine, Henri Lantéri and Céline Theys. "System identification under non-negativity constraints". *The 18th European Conference on Signal Processing (EUSIPCO)*, Aalborg, Denmark, Aug. 2010.

Francophone conference articles

15. Jie Chen, Cédric Richard and Paul Honeine. "Un nouveau paradigme pour le démélange nonlinéaire des images hyperspectrales". *23ème Colloque GRETSI sur le Traitement du Signal et des Images (GRETSI)*, Bordeaux, France, Sep. 2011. (session spéciale)

16. Cédric Richard, Jose-Carlos M. Bermudez and Paul Honeine. "Filtrage adatatif avec contrainte de non-négativité - principes de l'algorihtme NN-LMS et modèle de convergence". *23ème Colloque GRETSI sur le Traitement du Signal et des Images (GRETSI)*, Bordeaux, France, Sep. 2011.

17. Jie Chen, Cédric Richard, Paul Honeine, Hichem Snoussi, Henri Lantéri and Céline Theys. "Techniques d'apprentissage nonlineaires en ligne avec contraintes de positivité". *6ème Conférence Internationale Francophone d'Automatique (CIFA)*, Nancy, France, Jun. 2010. (article invité)

Part I

Online system identification subject to non-negativity constraints

Context

Suppose we have an unknown dynamic system that is linear and time varying. The system is characterized by a set of real-valued discrete-time measurements that describe the variation of the system output response to a known stationary input. The linear estimation problem is associated to a transversal filter model. The model consists of a finite number of unit-delay elements and a corresponding set of adjustable parameters. The online and non-stationary properties require to develop adaptive filtering algorithms for this system.

Let the available input signal at time n be denoted by $\boldsymbol{x}(n)$, composed by the set of samples $x(n)$, $x(n-1)$, $\ldots$, $x(n-N+1)$, with N the number of adjustable parameters in the model. This input signal is applied simultaneously to the system and the model. The system output $y(n)$ serves as the desired response for the adaptive filter to adjust the model parameters

$$y(n) = \boldsymbol{\alpha}^\top \boldsymbol{x}(n) + z(n) \tag{I.1}$$

with $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^\top$ the vector of model parameters. The input signal $x(n)$ and the desired output signal $y(n)$ are assumed zero-mean stationary. The sequence $z(n)$ accounts for measurement noise and modeling error. This scheme is illustrated in Figure I.1.

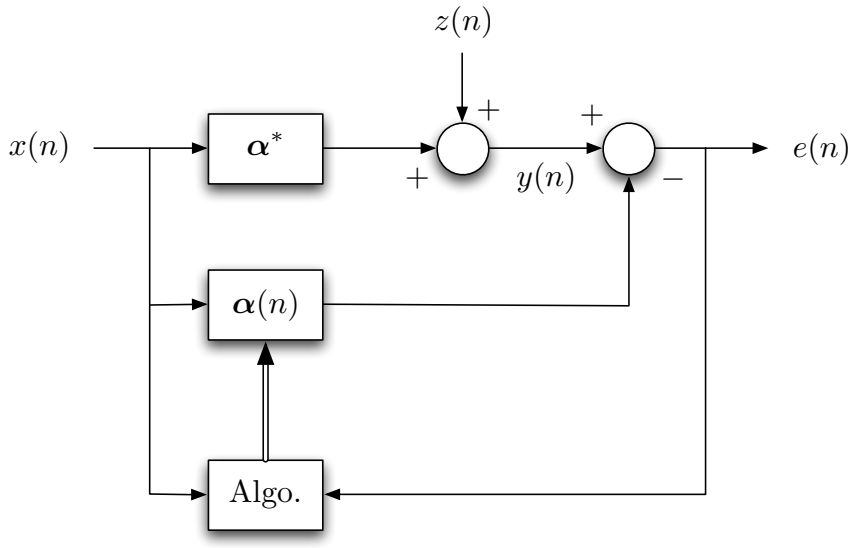


FIGURE I.1 – Adaptive system for identification problem.

As we introduced in Chapter 1, this problem is frequently addressed by minimizing a predefined cost function with respect to the unknown parameter vector $\boldsymbol{\alpha}$. Let us denote by $\boldsymbol{\alpha}^*$ the solution of the unconstrained least-mean-square problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} E\left\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\right\} \tag{I.2}$$

whose solution $\boldsymbol{\alpha}^*$ satisfies the Wiener-Hopf equations

$$\boldsymbol{R}_x \, \boldsymbol{\alpha}^* = \boldsymbol{r}_{xy} \tag{I.3}$$

where $\boldsymbol{R}_x$ is the autocorrelation matrix of $\boldsymbol{x}(n)$ and $\boldsymbol{r}_{xy}$ the correlation vector between $\boldsymbol{x}(n)$ and $y(n)$. Instead of solving these Wiener-Hopf equations, a steepest-descend method can be used to minimize the cost function J_{mse} defined in (I.2), and calculate the solution iteratively. The gradient is given by

$$\nabla J_{mse}(\boldsymbol{\alpha}) = \boldsymbol{R}_x \boldsymbol{\alpha} - \boldsymbol{r}_{xy} \tag{I.4}$$

In order to solve this problem in an online manner and to track possible non-stationarities of the system, recursive update of the weight vector using stochastic gradient at each instant was developed. This leads to the well-known least-mean-square (LMS) algorithm [Sayed 2008]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta \, \boldsymbol{x}(n) \, [y(n) - \boldsymbol{\alpha}(n)^\top \boldsymbol{x}(n)] \tag{I.5}$$

However, the LMS algorithm can not be simply applied when additional constraints, such as the non-negativity constraint, are imposed on the parameters to be estimated. Such problems present specific difficulties for online applications. Common real-time signal processing restrictions on computational complexity and memory requirements tend to rule out several good solutions to the constrained optimization problem. Algorithms for updating the adaptive filter under constraints will be explored in the Part I of this thesis. We shall study online algorithms for adaptive filtering when the non-negativity constraint is imposed (Chapter 2), and extend the proposed algorithm for solving ℓ_1-regularized problem in an online way (Chapter 3).

Chapter 2

Non-negative least-mean-square algorithm

Contents

2.1 Introduction

In many real-life phenomena including biological and physiological ones, due to the inherent physical characteristics of systems under investigation, non-negativity is a desired constraint that can be imposed on the parameters to estimate in order to avoid physically absurd and uninterpretable results. For instance, in the study of a concentration field or a thermal radiation field, any observation is described with non-negative values (ppm, joule). Non-negativity as a physical constraint has received growing attention from the signal processing community during the last decade. For instance, consider the following non-negative least-square inverse problem

$$\begin{aligned} &\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2 \\ &\text{subject to } [\boldsymbol{x}]_i \geq 0, \quad \forall i, \end{aligned} \tag{2.1}$$

with $\boldsymbol{A}$ a real $M \times N$ matrix of rank $k \leq \min(M,N)$, $\boldsymbol{b}$ an M-length real vector, and $\boldsymbol{x}$ an N-length real vector. $\|\cdot\|$ denotes the Euclidean 2-norm and $[\,\cdot\,]_i$ is the i-th entry of the vector. This problem has been addressed in various contexts, with applications ranging from image deblurring in astrophysics [Benvenuto 2010] to deconvolution of emission spectra in chemometrics [Van Benthem 2004]. Another similar problem is the non-negative matrix factorization (NMF), which is now a popular dimension reduction technique [Lee 1999, Lee 2001, Cichocki 2009]. Given a matrix $\boldsymbol{X}$ with non-negative entries, the squared error version of this problem can be stated as follows

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}\|_F^2 \\ &\text{subject to } [\boldsymbol{W}]_{ij} \geq 0, [\boldsymbol{H}]_{ij} \geq 0, \quad \forall i,j \end{aligned} \tag{2.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This problem is closely related to the blind deconvolution one, and has found direct application in hyperspectral imaging [Berry 2007]. Other problems such as Bayesian source separation, independent component analysis involving the non-negativity constraint have also been considered in [Plumbley 2003, Moussaoui 2006, Lin 2006].

Over the last fifteen years, a variety of methods have been developed to tackle non-negative least-square problems (NNLS). Active set techniques for NNLS use the fact that if the set of variables which activate constraints is known, then the solution of the constrained least-square problem can be obtained by solving an unconstrained one that only includes inactive variables. The active set algorithm of Lawson and Hanson [Lawson 1995] is a batch resolution technique for NNLS problems. It has become a standard among the most frequently used methods. In [Bro 1997], Bro and De Jong introduced a modification of the latter, called fast NNLS, which takes advantage of the special characteristics of iterative algorithms involving repeated use of non-negativity constraints. Another class of tools is the class of projected gradient algorithms [Rosen 1960, Calamai 1987, Barzilai 1988]. They are based on successive projections on the feasible region. In [Lin 2007b], Lin used this kind of algorithms for NMF problems. Low memory requirements

and simplicity make algorithms in this class attractive for large scale problems. Nevertheless, they are characterized by slow convergence rate if not combined with appropriate step size selection. The class of multiplicative algorithms is very popular for dealing with NMF problems [Lee 2001, Lin 2007a]. Particularly efficient updates were derived in this way for a large number of problems involving non-negativity constraints [Lantéri 2001]. These algorithms however require batch processing, which is not suitable for online system identification problems.

In this chapter, we consider the problem of system identification under non-negativity constraints on the parameters to estimate. The Karush-Kuhn-Tucker (KKT) conditions are established for any convex cost function, and a fixed-point iteration strategy is then applied in order to derive a gradient descent algorithm. Considering the square-error criterion as a particular case, a stochastic gradient scheme is presented. A convergence analysis of this algorithm is proposed. The resulting model accurately predicts the algorithm behavior for both transient and steady-state conditions. Finally, experiments are conducted to evaluate the algorithm performance and its consistency with the analysis.

2.2 System identification with non-negativity constraints

2.2.1 A fixed-point iteration scheme

In this part, non-negativity is a desired constraint that is imposed on the coefficient vector $\boldsymbol{\alpha}$. Therefore, the problem of identifying the optimum model can be formalized as follows

$$\begin{aligned} \boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} & J(\boldsymbol{\alpha}) \\ \text{subject to } & \alpha_i \geq 0, \quad \forall i \end{aligned} \tag{2.3}$$

with $J(\boldsymbol{\alpha})$ a convex cost function, and $\boldsymbol{\alpha}^o$ the solution to the constrained optimization problem. In order to solve the problem (2.3), let us consider its Lagrangian function $Q(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ given by [Boyd 2004]

$$Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}^\top \boldsymbol{\alpha},$$

where $\boldsymbol{\lambda}$ is the vector of non-negative Lagrange multipliers. The Karush-Kuhn-Tucker conditions must necessarily be satisfied at the optimum defined by $\boldsymbol{\alpha}^o$, $\boldsymbol{\lambda}^o$, namely,

$$\begin{aligned} \nabla_\alpha Q(\boldsymbol{\alpha}^o, \boldsymbol{\lambda}^o) &= 0 \\ \alpha_i^o \left[\boldsymbol{\lambda}^o\right]_i &= 0, \ \forall i \end{aligned}$$

where the symbol ∇_α stands for the gradient operator with respect to $\boldsymbol{\alpha}$. Using $\nabla_\alpha Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \nabla_\alpha J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}$, these equations can be combined into the following expression

$$\alpha_i^o \left[-\nabla_\alpha J(\boldsymbol{\alpha}^o)\right]_i = 0 \tag{2.4}$$

where the extra minus sign is just used to make a gradient descent of $J(\boldsymbol{\alpha})$ apparent. To solve equation (2.4) iteratively, two important points have to be noticed. The first point is that $\boldsymbol{D}(-\nabla_\alpha J(\boldsymbol{\alpha}))$ is also a gradient descent of $J(\boldsymbol{\alpha})$ if $\boldsymbol{D}$ is a symmetric positive definite matrix. The second point is that equations of the form $\varphi(u) = 0$ can be solved with a fixed-point iteration algorithm, under some conditions on function φ, by considering the problem $u = u + \varphi(u)$. Implementing this strategy with equation (2.4) leads us to the component-wise gradient descent algorithm

$$\alpha_i(n+1) = \alpha_i(n) + \eta_i(n) f_i(\boldsymbol{\alpha}(n))\, \alpha_i(n)[-\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i \tag{2.5}$$

with $\eta_i(n)$ a positive step size required to get a contraction scheme and to control the convergence rate. Arbitrary function $f_i(\boldsymbol{\alpha}) > 0$ in (2.5) is the i-th entry of a diagonal matrix $\boldsymbol{D}$. Some criteria $J(\boldsymbol{\alpha})$ are defined only for inputs $\boldsymbol{\alpha}$ with positive entries, e.g., Itakura-Saito distance, Kullback-Leibler divergence. If necessary, this condition can be managed by an appropriate choice of the step size parameter. Let us assume that $\alpha_i(n) \geq 0$. Non-negativity of $\alpha_i(n+1)$ is guaranteed if

$$1 + \eta_i(n) f_i(\boldsymbol{\alpha}(n))[-\nabla_\alpha J(\boldsymbol{\alpha}(n)]_i \geq 0. \tag{2.6}$$

If $[\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i \leq 0$, condition (2.6) is clearly satisfied and non-negativity does not impose any restriction on the step size. Conversely, if $[\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i > 0$, non-negativity of $\alpha_i(n+1)$ holds if

$$0 \leq \eta_i(n) \leq \frac{1}{f_i(\boldsymbol{\alpha}(n))\,[\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i}. \tag{2.7}$$

Using a single step size $\eta(n)$ in $]0, \eta_{\max}(n)]$ for all the entries of $\boldsymbol{\alpha}$ so that

$$\eta_{\max}(n) = \min_i \frac{1}{f_i(\boldsymbol{\alpha}(n))\,[\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i}, \quad i = 1, \ldots, N \tag{2.8}$$

the update equation can be written in vector form as

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n)\, \boldsymbol{d}(n), \tag{2.9}$$

where the weight adjustment direction $\boldsymbol{d}(n)$, whose i-th entry is defined as follows

$$[\boldsymbol{d}(n)]_i = f_i(\boldsymbol{\alpha}(n))\, \alpha_i(n)[-\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i \tag{2.10}$$

is a gradient descent direction because $f_i[\boldsymbol{\alpha}(n)]\, \alpha_i(n) \geq 0$. It should be noted that condition (2.8) on the step size $\eta(n)$ guarantees the non-negativity of $\boldsymbol{\alpha}(n)$ for any time index n, but does not insure the stability of the algorithm.

2.2.2 The non-negative least-mean-square algorithm

Let us now consider the mean-square error criterion $J_{\mathrm{MSE}}(\boldsymbol{\alpha})$ to be minimized with respect to $\boldsymbol{\alpha}$, that is,

$$\begin{aligned} \boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} & E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\} \\ & \text{subject to } \alpha_i^o \geq 0, \quad \forall i \end{aligned} \tag{2.11}$$

where we have included the non-negativity constraint only on the optimum solution because $J_{\mathrm{MSE}}(\boldsymbol{\alpha})$ is defined for all $\boldsymbol{\alpha}$, that is, for all positive and negative entries α_i. The gradient of $J_{\mathrm{MSE}}(\boldsymbol{\alpha})$ can be easily computed as

$$\nabla_\alpha J(\boldsymbol{\alpha}) = 2\,(\boldsymbol{R}_x\boldsymbol{\alpha} - \boldsymbol{r}_{xy}) \tag{2.12}$$

with $\boldsymbol{R}_x$ the autocorrelation matrix of $\boldsymbol{x}(n)$ and $\boldsymbol{r}_{xy}$ the correlation vector between $\boldsymbol{x}(n)$ and $y(n)$. Using (2.9) and (2.10) with $f_i(\boldsymbol{\alpha}) = \frac{1}{2}$ for all i, the update rule for minimizing the mean-square error under non-negativity constraints is given by

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n)\,\boldsymbol{D}_\alpha(n)\,(\boldsymbol{r}_{xy} - \boldsymbol{R}_x\,\boldsymbol{\alpha}(n)) \tag{2.13}$$

where $\boldsymbol{D}_\alpha(n)$ is the diagonal matrix with diagonal entries given by $\boldsymbol{\alpha}(n)$. Following a stochastic gradient approach, the second-order moments $\boldsymbol{R}_x$ and $\boldsymbol{r}_{xy}$ are replaced in (2.13) by the instantaneous estimates $\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)$ and $y(n)\,\boldsymbol{x}(n)$, respectively. This leads to the stochastic approximation of (2.13) given by [1]

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta(n)\,e(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}(n), \qquad \eta(n) > 0 \tag{2.14}$$

where $\boldsymbol{D}_x(n)$ stands for the diagonal matrix with diagonal entries given by $\boldsymbol{x}(n)$, and $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\,\boldsymbol{x}(n)$.

It is interesting to notice how the term $\boldsymbol{\alpha}(n)$ in the update term on the right -hand side (r.h.s.) of (2.14) affects the dynamics of the coefficient update when compared with the well-known LMS algorithm (I.5). Note that the extra multiplying factor $\alpha_i(n)$ in the update term of the i-th row of (2.14), which is not present in the LMS update, provides extra control of both magnitude and the direction of the weight update, as compared to LMS. For a fixed step size η, the update term for the i-th component of $\boldsymbol{\alpha}(n)$ is proportional to $-\alpha_i\,e(n)x_i(n)$, which corresponds to the stochastic gradient component $-e(n)x_i(n)$, multiplied by $\alpha_i(n)$. A negative $\alpha_i(n)$ will then change the sign of LMS adjustment, which on average tends to avoid convergence to negative coefficients of the unconstrained solution. Thus, coefficients that would normally converge, on average, to negative values using unconstrained LMS will tend to converge to zero using the constrained algorithm. In addition, $\alpha_i(n)$ close to zero will tend to slow its own convergence unless the magnitude of $e(n)x_i(n)$ is very large. As shown in following, $\alpha_i(n) = 0$ is clearly a stationary point of (2.14).

In what follows, the adaptive weight behavior of the adaptive algorithm (2.14), called non-negative LMS is studied in the mean and mean-square senses for a time-invariant step size η.

2.3 Mean behavior analysis

We shall now propose a model to characterize the stochastic behavior of the non-negative LMS algorithm. Figure I.1 shows a block diagram of the problem studied

1. Note that $\boldsymbol{D}_\alpha(n)\,\boldsymbol{x}(n) = \boldsymbol{D}_x(n)\,\boldsymbol{\alpha}(n)$.

in this part. The input signal $x(n)$ and the desired output signal $y(n)$ are assumed stationary and zero-mean. Let $\boldsymbol{R}_x$ denote the correlation matrix of $\boldsymbol{x}(n)$, and let $\boldsymbol{r}_{xy}$ be the correlation vector of $\boldsymbol{x}(n)$ and $y(n)$. Let us denote by $\boldsymbol{\alpha}^*$ the solution of the unconstrained least-mean-square problem (I.2). The residual signal $z(n) = y(n) - \boldsymbol{x}^\top(n)\,\boldsymbol{\alpha}^*$ in Figure I.1 accounts for measurement noise and modeling errors. It is assumed that $z(n)$ is stationary, zero-mean with variance σ_z^2 and statistically independent of any other signal. Thus, $E\{z(n)\,\boldsymbol{D}_x(n\} = 0$.

The adaptive algorithm (2.14) attempts to estimate the optimum $\boldsymbol{\alpha}^o$ for the constrained problem (2.11). The analytical determination of the optimal solution $\boldsymbol{\alpha}^o$ is not trivial in general. In the particular case of independent and identically distributed (i.i.d.) input samples, however, $\boldsymbol{R}_x = \sigma_x^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. In this case, the Karush-Kuhn-Tucker conditions imply that $\boldsymbol{\alpha}^o$ is obtained by turning the negative entries of $\boldsymbol{\alpha}^*$ to zero

$$\boldsymbol{\alpha}^o = \{\boldsymbol{\alpha}^*\}_+ \tag{2.15}$$

where $\{u\}_+ = \max\{0, u\}$. The minimum mean-square error produced by solution $\boldsymbol{\alpha}^o$ is then

$$J_{ms_{min}} = \sigma_y^2 - 2\,\boldsymbol{r}_{xy}\{\boldsymbol{\alpha}^*\}_+ + \sigma_x^2\,\{\boldsymbol{\alpha}^*\}_+^\top\{\boldsymbol{\alpha}^*\}_+ \tag{2.16}$$

with σ_y^2 the variance of $y(n)$.

2.3.1 Mean weight behavior model

Defining the weight-error vector as follows

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^* = [v_1(n), v_2(n), \ldots, v_N(n)]^\top \tag{2.17}$$

the update equation (2.14) can be written as

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta\,e(n)\,\boldsymbol{D}_x(n)\,\big(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*\big). \tag{2.18}$$

Using $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\,\boldsymbol{x}(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$ leads us to the following expression

$$\begin{aligned}\boldsymbol{v}(n+1) = \boldsymbol{v}(n) &+ \eta\,z(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{v}(n) + \eta\,z(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^* \\ &-\eta\,\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n) - \eta\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n).\end{aligned} \tag{2.19}$$

Taking the expectation of expression (2.19), neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$,[2] and using that $E\{z(n)\,\boldsymbol{D}_x(n)\} = 0$ yields

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\}\big)\,E\{\boldsymbol{v}(n)\} \\ - \eta\,E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\}.\end{aligned} \tag{2.20}$$

2. This assumption is less restrictive than the well-known independence assumption, as it does not require $x(n)$ be Gaussian [Sayed 2008].

The first expectation on the r.h.s. of (2.20) is given by

$$E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\} = E\{\boldsymbol{D}_{\alpha^*}\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\} = \boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x. \tag{2.21}$$

In order to evaluate the second expectation on the r.h.s. of (2.20), let us compute the i-th component of the vector $\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$. We have

$$[\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)]_i = \sum_{j=1}^{N} x(n-i+1)\,v_i(n)\,v_j(n)\,x(n-j+1) \tag{2.22}$$

Taking the expectation of this expression, defining $\boldsymbol{K}(n) = E\{\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\}$, and neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$, we obtain

$$\begin{aligned}[E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\}]_i &\approx \sum_{j=1}^{N} r_x(j-i)\,[\boldsymbol{K}(n)]_{ij} \\ &= [\boldsymbol{R}_x\,\boldsymbol{K}(n)]_{ii}\end{aligned} \tag{2.23}$$

This implies that $E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\} = \mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}$, where $\mathrm{Diag}\{\boldsymbol{A}\}$ denotes the vector whose i-th entry is defined by $[\boldsymbol{A}]_{ii}$. Using these results with equation (2.20) yields the following mean weight-error vector update equation

$$E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} - \eta\,\mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} \tag{2.24}$$

This equation requires second-order moments defined by $\boldsymbol{K}(n)$ in order to update the first-order moment provided by $E\{\boldsymbol{v}(n)\}$. A recursive model will be derived for $\boldsymbol{K}(n)$ in Section 2.4, see (2.38). That model can be used along with (2.24) to predict the mean weight behavior of the algorithm. Nevertheless, we have found that a sufficiently accurate and more intuitive mean behavior model can be obtained using the following separation approximation

$$\boldsymbol{K}(n) \approx E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\} \tag{2.25}$$

Using (2.25) in (2.24) we obtain the following result

$$E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} - \eta\,\mathrm{Diag}\{\boldsymbol{R}_x\,E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}\} \tag{2.26}$$

Approximation (2.25) assumes that

$$\mathrm{Cov}\{v_i(n), v_j(n)\} \ll E\{v_i(n)\}E\{v_j(n)\} \tag{2.27}$$

In general, (2.27) is valid when the adaptive weights are far from convergence, as the mean weight-error component tends to be much larger than the weight-error fluctuation about the mean. For correlated $x(n)$, the level of the fluctuations at convergence tends to be much less than the values of the nonzero optimal weights. For white input signals $E\{v_i(n)\}$ tends to zero for those indexes corresponding to the positive coefficients of $\boldsymbol{\alpha}^o$. In this case, approximation (2.27) will in fact tend to eliminate the weight estimation error at convergence. Extensive simulation have shown that the simplified model in (2.26) yields a prediction of the mean weight behavior which is sufficient for design purposes. Furthermore, this simplification allows the more detailed analytical study of the mean weight behavior shown in the next section.

2.3.2 Special case of a white input signal

In general, problem (2.26) can become very complex to be studied analytically [May 1976]. In order to obtain analytical results that allow some understanding of mean weight behavior, we study here the particular case of $x(n)$ i.i.d. and drawn from a zero-mean distribution. Unit variance σ_x^2 is also assumed without loss of generality. Using $\boldsymbol{R}_x = \boldsymbol{I}$ in (2.26) yields the component-wise expression

$$E\{v_i(n+1)\} = (1 - \eta\,\alpha_i^*)\,E\{v_i(n)\} - \eta\,E\{v_i(n)\}^2. \tag{2.28}$$

Function $E\{v_i(n+1)\}$ in equation (2.28) is a parabola in $E\{v_i(n)\}$ with roots at $E\{v_i(n)\} = 0$ and $E\{v_i(n)\} = (1 - \eta\,\alpha_i^*)/\eta$. It reaches its maximum value $(1 - \eta\,\alpha_i^*)^2/4\eta$ at $E\{v_i(n)\} = (1 - \eta\,\alpha_i^*)/2\eta$. Fixed points are found by solving $E\{v_i(n+1)\} = E\{v_i(n)\}$, which yields $E\{v_i(n)\} = 0$ or $E\{v_i(n)\} = -\alpha_i^*$. This result is consistent with solution (2.15) where

$$v_i^o = \begin{cases} 0 & \text{if } \alpha_i^* \geq 0 \\ -\alpha_i^* & \text{otherwise} \end{cases} \tag{2.29}$$

with v_i^o the i-th entry of $\boldsymbol{v}^o = \boldsymbol{\alpha}^o - \boldsymbol{\alpha}^*$.

Let us now derive conditions ensuring convergence of (2.28) to 0 and $-\alpha_i^*$. Writing $u(n) = \eta\,E\{v_i(n)\}/(1 - \eta\,\alpha_i^*)$, where the index i has been dropped to simplify the notation, we obtain the following difference equation known as the *logistic map* [May 1976, Alligood 1997, Perrin 2008]

$$u(n+1) = \rho\,u(n)\,(1 - u(n)) \tag{2.30}$$

with $\rho = 1 - \eta\,\alpha_i^*$, which is assumed nonzero. Fixed points defined in (2.29) now correspond to $u = 0$ and $u = \frac{\rho-1}{\rho}$, respectively. Convergence of the logistic map to these values depends on the parameter ρ and on the initial condition $u(0)$ as follows. See [May 1976, Alligood 1997, Perrin 2008] for details and Figure 2.1 for illustration.

Case 1 : $0 < \rho < 1$

An illustration of this case is shown in Figure 2.1 (left). The fixed point $u = 0$ attracts all the trajectories originating in the interval $\left]\frac{\rho-1}{\rho}; \frac{1}{\rho}\right[$. The logistic map $u(n)$ is identically equal to $\frac{\rho-1}{\rho}$ for $n \geq 1$ if $u(0) = \frac{\rho-1}{\rho}$ or $u(0) = \frac{1}{\rho}$. Outside this interval, it diverges to $-\infty$.

Case 2 : $\rho = 1$

The fixed point $u = 0$ attracts all the trajectories originating in the interval $[0; 1]$. The logistic map $u(n)$ is identically equal to 0 for $n \geq 1$ if $u(0) = 0$ or 1. It diverges to $-\infty$ if $u(n) \notin [0; 1]$.

Case 3 : $1 < \rho \leq 3$

An illustration of this case is shown in Figure 2.1 (right). The fixed point $u = \frac{\rho-1}{\rho}$ attracts all the trajectories originating in $]0; 1[$. With the initial conditions $u(0) = 0$ or $u(0) = 1$, we have $u(n) = 0$ for all n. It can been shown that the logistic map diverges to $-\infty$ if $u(n) \notin [0; 1]$.

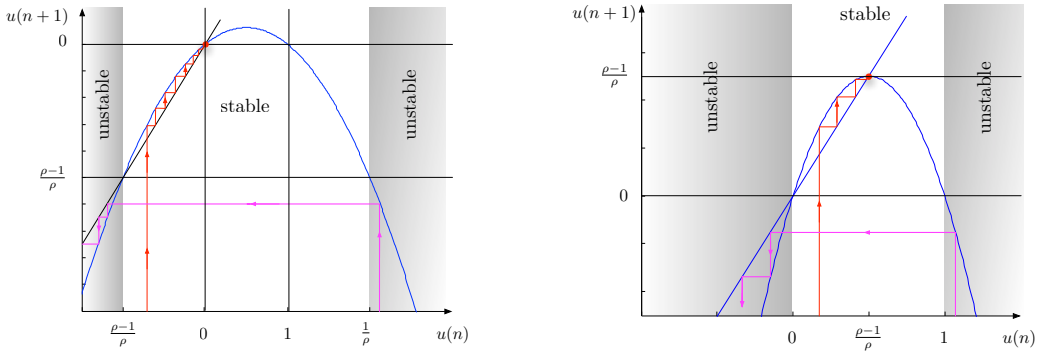


FIGURE 2.1 – Convergence of the logistic map, in the Case 1 (left) and in the Case 3 (right).

Case 4 : $\rho > 3$

Fixed points become unstable. New ones appear between which the system alternates in stable cycles of period 2^k, with k tending to infinity as ρ increases. This case may even lead to a chaotic behavior, and falls out of the scope of our study.

To derive conditions for convergence of the difference equation (2.28) to 0 or $-\alpha_i^*$, we must consider separately components of $E\{v_i(n)\}$ associated with positive or negative unconstrained optimum α_i^*, respectively. On the one hand, based on the analysis of the logistic map (2.30), convergence of (2.28) to 0 corresponds to the conditions on ρ and $u(0)$ satisfying Case 1 and Case 2 above. This yields

$$0 < \eta \leq \frac{1}{\alpha_i^*} \qquad -\alpha_i^* < v_i(0) < \frac{1}{\eta} \tag{2.31}$$

in the case where $\alpha_i^* > 0$. If $\alpha_i^* = 0$, these two conditions become $\eta > 0$ and $0 \leq v_i(0) \leq \frac{1}{\eta}$. On the other hand, ρ and $u(0)$ must obey the conditions presented in Case 3 for convergence of equation (2.28) to $-\alpha_i^*$. This leads to

$$0 < \eta \leq -\frac{2}{\alpha_i^*} \qquad 0 < v_i(0) < \frac{1}{\eta} - \alpha_i^* \tag{2.32}$$

in the case where $\alpha_i^* < 0$. Finally, combining these inequalities leads to the following theoretical conditions for convergence of $E\{\boldsymbol{v}(n)\}$:

$$0 < \eta \leq \min_i \frac{1}{|\alpha_i^*|} \quad \text{and} \quad 0 < v_i(0) < \frac{1}{\eta} \quad \text{for all } i \tag{2.33}$$

or, using also equations (2.31) and (2.32), for convergence of $E\{\boldsymbol{\alpha}(n)\}$:

$$0 < \eta \leq \min_i \frac{1}{|\alpha_i^*|} \quad \text{and} \quad 0 < \alpha_i(0) < \frac{1}{\eta} \quad \text{for all } i. \tag{2.34}$$

Conditions (2.33) and (2.34) on $v_i(0)$ and $\alpha_i(0)$ show that there is more freedom in choosing $\alpha_i(0)$ for small values of η. They guarantee convergence of the difference (2.28).

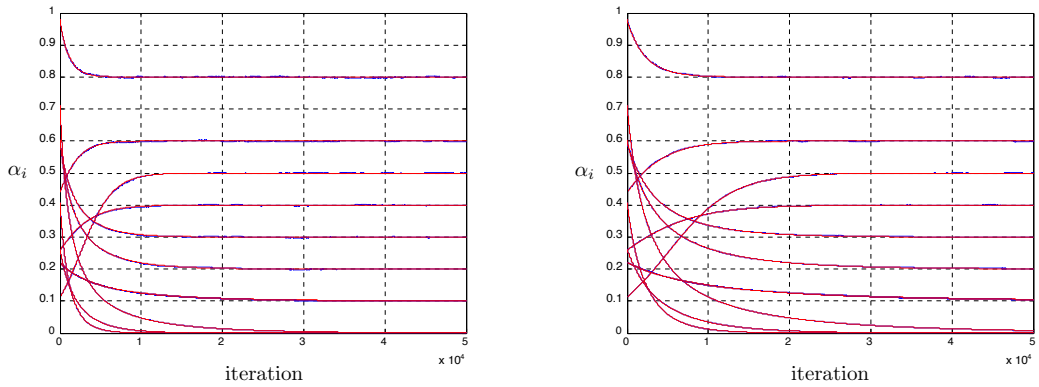


FIGURE 2.2 – Convergence of the coefficients $\alpha_i(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered : $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \cdot 10^{-4}$ on the right figure. The theoretical curves (red line) obtained from the model (2.28) and simulation curves (blue line) are perfectly superimposed.

2.3.3 Simulation examples for the first-order moment analysis

This section presents simulation examples to verify the validity of the first-order moment analysis of the non-negative LMS algorithm. We illustrate the accuracy of the model (2.28) through a first example where inputs $x(n)$ and noise $z(n)$ are i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_z^2 = 10^{-2}$, respectively. The impulse response $\boldsymbol{\alpha}^*$ is given by

$$\boldsymbol{\alpha}^* = [0.8 \quad 0.6 \quad 0.5 \quad 0.4 \quad 0.3 \quad 0.2 \quad 0.1 \quad -0.1 \quad -0.3 \quad -0.6]^\top \tag{2.35}$$

The initial impulse response $\boldsymbol{\alpha}(0)$ is drawn from the uniform distribution $\mathcal{U}([0;1])$, and kept unchanged for all the simulations. The algorithm's stability limit was determined experimentally to be $\eta_{\max} \approx 5 \times 10^{-3}$. As usually happens with adaptive algorithms, this limit is more restrictive than the mean weight convergence limit given by (2.34), as stability is determined by the weight fluctuations [Sayed 2008]. The mean value $E\{\alpha_i(n)\}$ of each coefficient is shown in Figure 2.2 for $\eta = 10^{-3} = \frac{\eta_{\max}}{5}$ and $\eta = 5 \times 10^{-5} = \eta_{\max}/10$. The simulation curves (blue line) were obtained from Monte Carlo simulation averaged over 100 realizations. The theoretical curves (red line) were obtained from model (2.28). One can notice that all the curves are perfectly superimposed and, as predicted by the result (2.15), each coefficient $\alpha_i(n)$ tends to $\{\alpha_i^*\}_+$ as n goes to infinity.

It is interesting to note that convergence towards the solution $\{\boldsymbol{\alpha}^*\}_+$ agrees with the theoretically predicted behavior of (2.30). For each positive entry α_i^* of $\boldsymbol{\alpha}^*$, the corresponding value of $\rho = 1 - \eta\,\alpha_i^*$ is in $]0;1[$. This corresponds to Case 1 in Section 2.3.2, where the fixed point $u = 0$ attracts all the trajectories and $v_i(n)$ converges to zero. It can also be verified that each ρ associated with a negative entry α_i^* is in $]1;3]$. This corresponds to Case 3 where $u = (\rho - 1)/\rho$ attracts all the trajectories and $\lim_{n\to\infty} v_i(n) = -\alpha_i^*$.

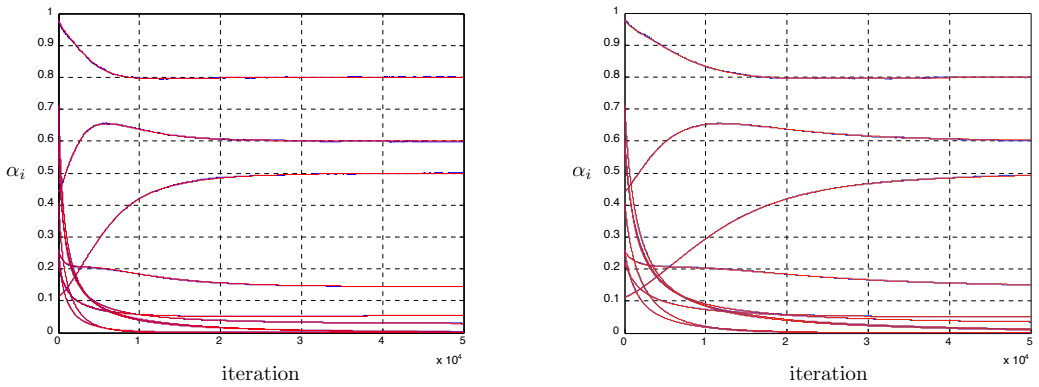


FIGURE 2.3 – Same experiment as in Figure 2.2 except that input sequence $x(n)$ is generated by a first-order AR process.

The second simulation example illustrates the accuracy of the model (2.28) for inputs $x(n)$ correlated in time. We consider a first-order AR model given by

$$x(n) = a\,x(n-1) + w(n) \tag{2.36}$$

with $a = \frac{1}{2}$. The noise $w(n)$ is i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_w^2 = 1 - \frac{1}{4}$, so that $\sigma_x^2 = 1$ as in the first example. The other parameters of the initial experimental setup remain unchanged, except for the step size values. In order to verify the model's accuracy also for large step sizes we performed the simulations of $\eta = 2.5\times10^{-3} = \frac{\eta_{\max}}{2}$ and $\eta = 5\times10^{-5} = \frac{\eta_{\max}}{10}$. The mean value $E\{\alpha_i(n)\}$ of each coefficient is shown in Figure 2.3. As before, the simulation curves (blue line) and the theoretical curves (red line) are superimposed. It can be noticed that $\boldsymbol{\alpha}(n)$ no longer converges to $\{\boldsymbol{\alpha}^*\}_+$ since the input samples $x(n)$ are now correlated. We can easily verify that $E\{e^2(n)\} = 4.97$ dB using $\{\boldsymbol{\alpha}^*\}_+$, and $E\{e^2(n)\} = 3.82$ dB at convergence of the non-negative LMS algorithm.

2.4 Second-order moment analysis

We now present a model for the behavior of the second-order moments of the adaptive weights. To allow further analysis progress, we assume in this section that the input $x(n)$ is Gaussian.

2.4.1 Second moment behavior model

Using $e(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$, neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$, and using the properties assumed for $z(n)$ yields and expression for the mean-square estimation error (MSE) :

$$\begin{aligned} E\{e^2(n)\} &= E\{(z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n))(z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n))\} \\ &= \sigma_z^2 + E\{\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\} \\ &= \sigma_z^2 + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}. \end{aligned} \tag{2.37}$$

Equation (2.24) clearly shows that the mean behavior of each coefficient is a function of a signal diagonal entry of the matrix $\boldsymbol{R}_x\boldsymbol{K}(n)$. In this case, approximation (2.26) could be used without compromising the accuracy of the resulting mean behavior model. This accuracy has been verified through Monte Carlo simulations in Section 2.3.3. The MSE (2.37), however, is a function of trace of $\boldsymbol{R}_x\boldsymbol{K}(n)$. Thus, the effect of the second order moments of the weight-error vector entries on the MSE behavior becomes more significant than in (2.24), and in general cannot be neglected. Thus we determine a recursion for $\boldsymbol{K}(n)$ starting again from the weight error update equation (2.19).

Post-multiplying equation (2.19) by its transpose, taking the expected value, and using the statistical properties of $z(n)$,[3] yields

$$\begin{aligned}\boldsymbol{K}(n+1) = \boldsymbol{K}(n) &- \eta\,\boldsymbol{P}_1(n)\,\boldsymbol{K}(n) - \eta\,\boldsymbol{K}(n)\,\boldsymbol{P}_1^{\top}(n) + \eta^2\sigma_z^2\,\boldsymbol{P}_2(n) \\ &+ \eta^2\sigma_z^2\,[\boldsymbol{P}_3(n) + \boldsymbol{P}_3^{\top}(n)] + \eta^2\sigma_z^2\,\boldsymbol{P}_4(n) - \eta\,[\boldsymbol{P}_5(n) + \boldsymbol{P}_5^{\top}(n)] \\ &+ \eta^2\,\boldsymbol{P}_6(n) + \eta^2\,\boldsymbol{P}_7(n) + \eta^2\,\boldsymbol{P}_7^{\top}(n) + \eta^2\,\boldsymbol{P}_8(n)\end{aligned} \tag{2.38}$$

where matrices $\boldsymbol{P}_1$ to $\boldsymbol{P}_8$ are defined by

$$\boldsymbol{P}_1 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^{\top}(n)\} \tag{2.39}$$

$$\boldsymbol{P}_2 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}\,\boldsymbol{D}_x(n)\} \tag{2.40}$$

$$\boldsymbol{P}_3 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{\alpha}^{*\top}\,\boldsymbol{D}_x(n)\} \tag{2.41}$$

$$\boldsymbol{P}_4 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} \tag{2.42}$$

$$\boldsymbol{P}_5 = E\{\boldsymbol{v}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} \tag{2.43}$$

$$\boldsymbol{P}_6 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{\alpha}^{*\top}\,\boldsymbol{D}_x(n)\} \tag{2.44}$$

$$\boldsymbol{P}_7 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} \tag{2.45}$$

$$\boldsymbol{P}_8 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^{\top}(n)\,\boldsymbol{D}_x(n)\} \tag{2.46}$$

The expected values in (2.39)–(2.46) can be calculated in the following. In order to keep the calculations mathematically tractable, the following statistical assumptions are employed :

A1 : The input vector $\boldsymbol{x}(n)$ is zero-mean Gaussian.

A2 : The weight-error vector $\boldsymbol{v}(n)$ is statistically independent of $\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)$. The reasoning for this approximation has been discussed in detail in [Minkoff 2001].

A3 : THe statistical dependence of $\boldsymbol{v}(n)\boldsymbol{v}^{\top}(n)$ and $\boldsymbol{v}(n)$ is neglected. This assumption follows the same reasoning valid for A2, see [Minkoff 2001].

A4 : $\boldsymbol{v}(n)$ and $(\boldsymbol{x}(n)\boldsymbol{v}^{\top}(n))^2$ are statistically independent given A2. This is because $(\boldsymbol{x}^{\top}(n)\boldsymbol{v}(n))^2$ is linear combination of entries of $\boldsymbol{v}(n)\boldsymbol{v}^{\top}(n)$. Thus this approximation follows basically the same reasoning discussed in [Minkoff 2001] to justify A2.

3. The two important properties of $z(n)$ used in evaluating (2.38) are its independence of any other signal and its zero-mean.

Expected value $\boldsymbol{P}_1$: This expected value has been already calculated in (2.21). Remember that

$$\boldsymbol{P}_1 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\boldsymbol{x}(n)\} = E\{\boldsymbol{D}_{\alpha^*}\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\} = \boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x. \tag{2.47}$$

Expected value $\boldsymbol{P}_2$: Basic linear algebra gives

$$\boldsymbol{P}_2 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\} = E\{\boldsymbol{D}_{\alpha^*}\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{D}_{\alpha^*}\} = \boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha^*}. \tag{2.48}$$

Expected value $\boldsymbol{P}_3$: Neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$ yields

$$\boldsymbol{P}_3 = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\} \approx E\{\boldsymbol{D}_v(n)\}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha^*}. \tag{2.49}$$

Expected value $\boldsymbol{P}_4$: The (i,j) th entry of the matrix within the expectation in $\boldsymbol{P}_4$ is given by

$$[\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_x(n)]_{ij} = x(n-i+1)\,v_i(n)\,v_j(n)\,x(n-j+1). \tag{2.50}$$

Using **A2**, $E\{x(n-i+1)\,v_i(n)\,v_j(n)\,x(n-j+1)\} \approx E\{x(n-i+1)\,x(n-j+1)\}\,E\{v_i(n)\,v_j(n)\}$ and

$$\boldsymbol{P}_4 \approx \boldsymbol{R}_x \circ \boldsymbol{K}(n) \tag{2.51}$$

where $\circ$ denotes the so-called Hadamard entry-wise product.

Expected value $\boldsymbol{P}_5$: Defining $\boldsymbol{D}_v(n)$ as the diagonal matrix with diagonal entries given by $\boldsymbol{v}(n)$, we first note that

$$E\{\boldsymbol{v}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_x(n)\} = E\{\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{D}_v(n)\} \tag{2.52}$$

Now, using **A2** and **A3**, the expectation can be approximated as

$$E\{\boldsymbol{v}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_x(n)\} \approx E\{\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\}\,E\{\boldsymbol{D}_v(n)\}. \tag{2.53}$$

Finally, using the statistical independence of $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$, we obtain

$$\boldsymbol{P}_5 \approx \boldsymbol{K}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_v(n)\}. \tag{2.54}$$

Expected value $\boldsymbol{P}_6$: Basic linear algebra gives

$$\begin{aligned}\boldsymbol{P}_6 &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{\alpha}^{*\top}\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{D}_{\alpha^*}\,E\{\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\}\,\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{2.55}$$

Under **A1** and applying the same methodology used to derive [Hubscher 2003, equation (29)],

$$\begin{aligned}\boldsymbol{P}_6 &= \boldsymbol{D}_{\alpha^*}\left(2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + E\{\boldsymbol{v}^\top(n)\,\boldsymbol{R}_x\,\boldsymbol{v}(n)\}\,\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}\\ &= \boldsymbol{D}_{\alpha^*}\left(2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + E\{\mathrm{trace}\{\boldsymbol{v}^\top(n)\,\boldsymbol{R}_x\,\boldsymbol{v}(n)\}\}\,\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}\\ &= \boldsymbol{D}_{\alpha^*}\left(2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x\right)\boldsymbol{D}_{\alpha^*}.\end{aligned} \tag{2.56}$$

Expected value $\boldsymbol{P}_7$: Using basic algebra, **A2** and **A3** as done to obtain (2.53), we have

$$\begin{aligned}\boldsymbol{P}_7 &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{D}_{\alpha^*}\,E\{\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\}\,E\{\boldsymbol{D}_v(n)\}.\end{aligned} \tag{2.57}$$

Finally, under **A1** and applying the same methodology as in [Hubscher 2003, equation (29)], yields

$$\boldsymbol{P}_7 \approx \boldsymbol{D}_{\alpha^*}\,(2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x)\;E\{\boldsymbol{D}_v(n)\}. \tag{2.58}$$

Expected value $\boldsymbol{P}_8$: Computing the (i,j)-th entry of matrix $\boldsymbol{P}_8$ within the expectation, and using **A2**, yields

$$\begin{aligned}[\boldsymbol{P}_8]_{ij} &= \sum_\ell \sum_k E\{x(n-i+1)\,[\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)]_{ik}\,[\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)]_{k\ell}\,[\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)]_{\ell j}\\ &\qquad\qquad \times x(n-j+1)\}\\ &= \sum_\ell \sum_k E\{x(n-i+1)\,x(n-k+1)\,x(n-\ell+1)\,x(n-j+1)\}\\ &\qquad\qquad \times E\{v_i(n)\,v_k(n)\,v_\ell(n)\,v_j(n)\}\end{aligned} \tag{2.59}$$

for $\boldsymbol{x}(n)$ a zero-mean Gaussian signal (**A1**), we know that

$$\begin{aligned}&E\{x(n-i+1)\,x(n-k+1)\,x(n-\ell+1)\,x(n-j+1)\}\\ &= r_x(k-i)\,r_x(j-\ell) + r_x(\ell-i)\,r_x(j-k) + r_x(j-i)\,r_x(\ell-k).\end{aligned} \tag{2.60}$$

The expectation $E\{v_i(n)v_j(n)v_k(n)v_\ell(n)\}$ cannot be evaluated directly, as the statistics of $\boldsymbol{v}(n)$ is unknown. Approximate expression can be obtained using numerous different approaches. We have chosen to use the following approximation which preserves relevant information about the second moment behavior of the adaptive weights while keeping the mathematical problem tractable. We write

$$\begin{aligned}E\{v_i(n)\,v_k(n)\,v_\ell(n)\,v_j(n)\} \approx{}& E\{v_k(n)\,v_\ell(n)\}\,E\{v_i(n)\,v_j(n)\}\\ &+ \mathrm{Cov}\{v_i(n)v_j(n), v_k(n)v_l(n)\}\end{aligned} \tag{2.61}$$

Now, writing

$$v_i(n+1)v_j(n+1) = (v_i(n) + \eta\Delta v_i(n))\,(v_j(n) + \eta\Delta v_j(n))) \tag{2.62}$$

we see that the fluctuations in $v_i(n+1)v_j(n+1)$ are proportional to η. Using the same reasoning for $v_k(n)v_\ell(n)$ we note that the covariance in (2.61) is proportional to η^2. The higher order moments of the entries of $\boldsymbol{v}(n)$ in (2.61) will be then proportional to η^p with $p \geq 2$. Thus, for sufficiently small values of η, neglecting these terms yields the approximation

$$E\{v_i(n)\,v_k(n)\,v_\ell(n)\,v_j(n)\} \approx E\{v_k(n)\,v_\ell(n)\}\,E\{v_i(n)\,v_j(n)\}. \tag{2.63}$$

This approximation is supported by the simulation results presented in Section 2.4.2. Substituting the two equations above into the expression of $[\boldsymbol{P}_8]_{ij}$ leads to

$$\begin{aligned}[\boldsymbol{P}_8]_{ij} = & \, r_x(j-i) \sum_{\ell} \sum_{k} r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_{\ell} \sum_{k} r_x(k-i)\,r_x(j-\ell)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ & + \sum_{\ell} \sum_{k} r_x(\ell-i)\,r_x(j-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij}.\end{aligned} \tag{2.64}$$

The first right-hand term of equation (2.64) can be expressed as follows

$$\begin{aligned} & r_x(j-i) \sum_{\ell} \sum_{k} r_x(\ell-k)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ =& [\boldsymbol{R}_x]_{ij} \sum_{k} \Big(\sum_{\ell} [\boldsymbol{R}_x]_{k\ell} [\boldsymbol{K}(n)]_{k\ell} \Big) [\boldsymbol{K}(n)]_{ij} \\ =& [\text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x]_{ij}\,[\boldsymbol{K}(n)\}]_{ij}. \end{aligned} \tag{2.65}$$

The second and third right-hand terms write

$$\begin{aligned} & \sum_{\ell} \sum_{k} r_x(k-i)\,r_x(j-\ell)\,[\boldsymbol{K}(n)]_{k\ell}\,[\boldsymbol{K}(n)]_{ij} \\ =& \left(\sum_{\ell} \sum_{k} [\boldsymbol{R}_x]_{ik} [\boldsymbol{K}(n)]_{k\ell} [\boldsymbol{R}_x]_{\ell j} \right) [\boldsymbol{K}(n)]_{ij} \\ =& [\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x]_{ij}\,[\boldsymbol{K}(n)\}]_{ij}. \end{aligned} \tag{2.66}$$

This leads to the following close-form expression

$$\boldsymbol{P}_8 = (\text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}\,\boldsymbol{R}_x + 2\,\boldsymbol{R}_x\,\boldsymbol{K}(n)\,\boldsymbol{R}_x) \circ \boldsymbol{K}(n). \tag{2.67}$$

Using the expected values $\boldsymbol{P}_1$ to $\boldsymbol{P}_8$ in equation (2.38), we finally obtain a recursive analytical model for the behavior of $\boldsymbol{K}(n)$. This result can be used to study the convergence properties of $E\{e^2(n)\}$, and can be applied for design. The next section illustrates the model accuracy in predicting the non-negative LMS algorithm behavior.

2.4.2 Simulation examples for the second-order moment analysis

This section presents simulation examples to verify the accuracy of the model (2.38). Figs. 2.4 and 2.5 show the behavior of the excess MSE $J_{\text{MSE}}(n) = \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}$ corresponding to the experiments conducted in Section 2.3.3. The simulation curves (blue line) were obtained from Monte Carlo simulations averaged over 100 realizations. The theoretical curves (red line) were obtained from model (2.38). Note the model's accuracy even for step sizes as large as $\eta_{\max}/2$ (left side of Figure 2.5). Also note that the theoretical value of the minimum excess mean-square

error $J_{\mathrm{MSE}_{min}}$ is represented in Figure 2.4. [4] It can be observed that $J_{\mathrm{MSE}}(n)$ tends to $J_{\mathrm{MSE}_{min}}$ as n goes to infinity. Figure 2.6 highlights the performance of the model for uncorrelated and correlated input signals $x(n)$ through the same experimental setup as described before, except that the noise variance σ_z^2 is now set to 1. All these experiments illustrate the accuracy of the model, which can provide important guidelines for the use of the non-negative LMS algorithm in practical applications.

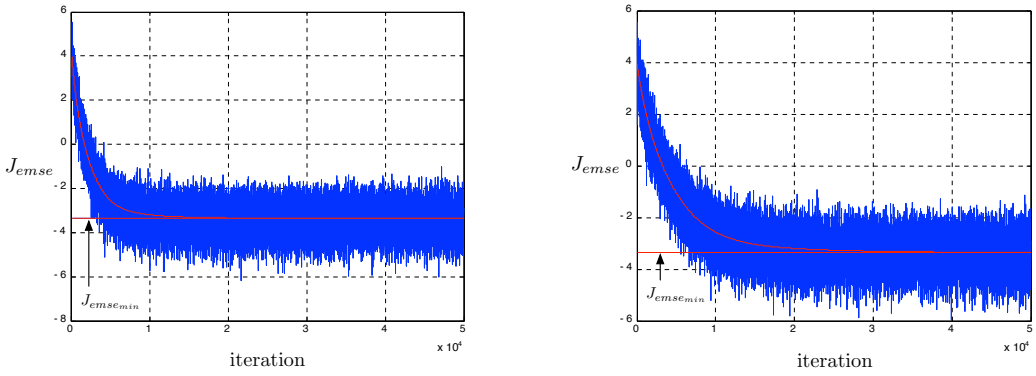


FIGURE 2.4 – Convergence of $J_{\mathrm{MSE}}(n)$ in the case where input $x(n)$ and noise $z(n)$ are i.i.d. Two different step sizes are considered : $\eta = 10^{-3}$ on the left figure, and $\eta = 5 \cdot 10^{-4}$ on the right figure. The theoretical curves (red line) obtained from the model (2.28) and simulation curves (blue line) are perfectly superimposed.

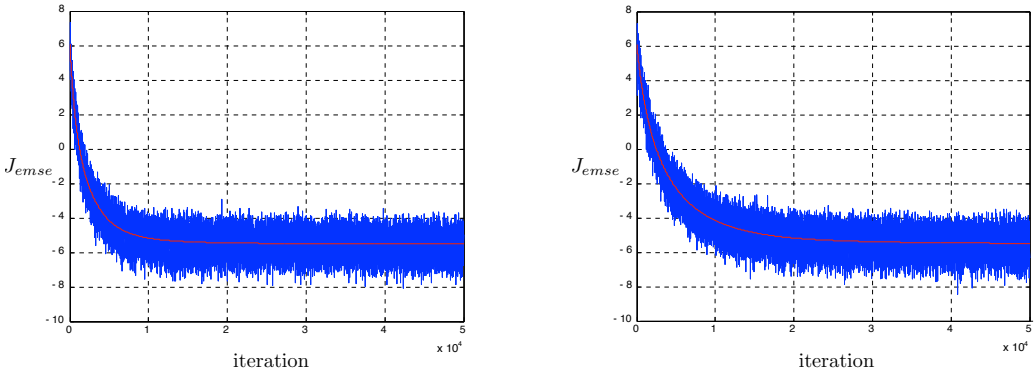


FIGURE 2.5 – Same experiment as in Figure 2.4 except that input sequence $x(n)$ is generated by a first-order AR process.

2.5 Variants of NNLMS algorithm

In previous sections, the problem of online system identification under non-negativity constraints on the parameters to estimate was investigated. We pro-

4. It can be easily shown, from equations (2.15)–(2.16), that $J_{\mathrm{MSE}_{min}} = \|\boldsymbol{\alpha}^* - (\boldsymbol{\alpha}^*)_+\|^2$ in the case where $\boldsymbol{R}_x = \boldsymbol{I}$.

posed an LMS-type adaptive algorithm, called Non-Negative Least-Mean-Square (NNLMS) to solve the Wiener problem under the constraint that the resulting weights need to be non-negative. It is based on the stochastic gradient descent approach combined with a fixed point iteration which converges to a solution satisfying the Karush-Kuhn-Tucker conditions. The stochastic behavior of this algorithm was also analyzed. In the following sections, we shall extend this framework and derive useful variants of the NNLMS algorithm. Each of these variants is derived to improve the NNLMS properties in some sense. A normalized algorithm is proposed to reduce the NNLMS performance sensitivity to the input power value. An exponential algorithm is proposed to improve the balance of weight convergence rates. Compared to NNLMS, the new algorithm leads to faster convergence of the weights in the active set (weights for which the inequality constraint is satisfied with the equal sign). Finally, a sign-based algorithm is proposed to reduce implementation cost in critical real-time applications.

2.5.1 Normalized NNLMS

A direct extension of the original algorithm is the Normalized NNLMS. Conditioned on $\boldsymbol{\alpha}(n)$, the product $e(n)\boldsymbol{D}_x(n)$ in (2.14) has dimension of signal power. Thus, η is inversely proportional to signal power. Hence, setting a constant value for η leads to different weight updates for different signal power levels. This is the same sensitivity to signal power verified in the LMS algorithm. A popular way to address this limitation is to normalize the weight update by squared ℓ_2-norm of the input vector, which yields the Normalized NNLMS update equation

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)}\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{N}}(n) \tag{2.68}$$

Like in Normalized LMS (NLMS) algorithm, adding a small positive regularization parameter ε to the denominator $\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)$ may be necessary to avoid numeri-

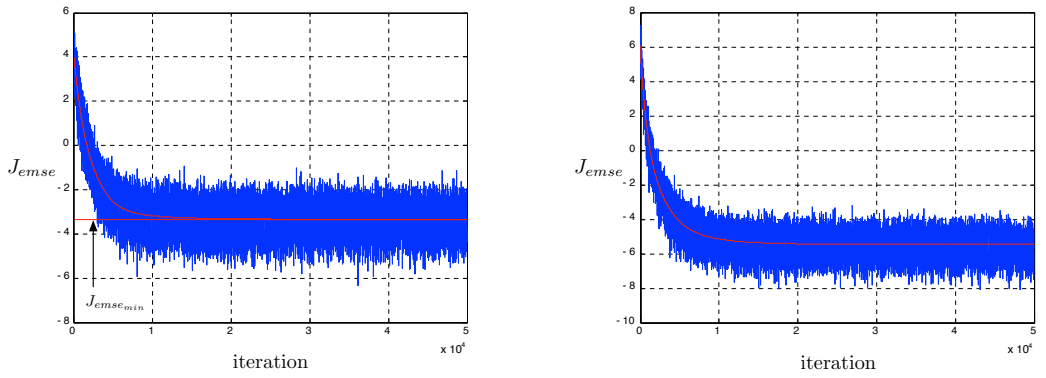


FIGURE 2.6 – Convergence of $J_{\mathrm{MSE}}(n)$ with step size $\eta = 10^{-3}$, in the case where input $x(n)$ is i.i.d. on the left figure, and generated by a first-order AR process on the right figure. Compared to Figure 2.4 (left) and 2.5 (left), the variance of the noise $z(n)$ has been increased from 10^{-2} to 1.

cal difficulties when $\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n)$ becomes very small. The resulting ε-Normalized NNLMS will then be

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n) + \varepsilon}\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{N}}(n) \tag{2.69}$$

where we maintained the notation $\boldsymbol{\alpha}_{\mathrm{N}}(n)$ because (2.68) is a particular case of (2.69) for $\varepsilon = 0$. From now on, we refer to (2.69) simply as the Normalized NNLMS algorithm.

2.5.2 Exponential NNLMS

Each component $\alpha_i(n)$ in the update term of (2.14) can be viewed as a distinct variable step size adjustment along the i-th axis. Hence, each component of $\boldsymbol{\alpha}(n)$ will have a different convergence rate in general. Specifically in the case of weights in the active set (those that tend to zero in steady-state), the convergence rate will progressively reduce in time, becoming very small near steady-state. To alleviate this convergence rate unbalance, we introduce the Exponential NNLMS algorithm.

To achieve a faster convergence for the adaptive coefficients as they get close to zero we propose the use of $f_i(\boldsymbol{\alpha}_E(n)) = \alpha_{\mathrm{E}_i}^{\gamma-1}(n)$ in (2.10), with parameter γ chosen in order to attract small values of $\alpha_{\mathrm{E}_i}(n)$ towards zero. This leads to the i-th weight update equation

$$\alpha_{\mathrm{E}_i}(n+1) = \alpha_{\mathrm{E}_i}(n) + \eta\, e(n)\, x_i(n)\, \alpha_{\mathrm{E}_i}^{\gamma}(n). \tag{2.70}$$

For $0 < \gamma < 1$, the i-th weight update in (2.70) becomes larger than that in (2.14) when $|\alpha_{\mathrm{E}_i}(n)| < 1$, thus accelerating convergence towards a null steady-state coefficient value.

The condition for $\alpha_{\mathrm{E}_i}(n+1) \geq 0$ given $\alpha_{\mathrm{E}_i}(n) \geq 0$ can be easily determined from (2.70) as

$$\eta \leq \frac{1}{e(n)\, \alpha_{\mathrm{E}_i}^{\gamma-1}(n)}, \quad \forall\, i, n. \tag{2.71}$$

This condition, however, is not useful for design purposes, since it requires *a priori* knowledge of the algorithm behavior. We then propose a modified version of the update equation (2.70) that allows for instantaneous negative values of $\alpha_{\mathrm{E}_i}(n)$. The problem with real and negative instantaneous values of $\alpha_{\mathrm{E}_i}(n)$ is that it may lead to a complex value for $\alpha_{\mathrm{E}_i}^{\gamma}$ for $0 < \gamma < 1$. To obtain always real values for $\alpha_{\mathrm{E}_i}^{\gamma}$ we propose to use $\gamma = p/q$ with p and q odd integers and $0 < p < q$. The oddness of p and q guarantees that $\mathrm{sgn}(\alpha_{\mathrm{E}_i}^{\gamma}(n)) = \mathrm{sgn}(\alpha_{\mathrm{E}_i}(n))$. Then, the real solution for $\alpha_{\mathrm{E}_i}^{\gamma}$ can be obtained by calculating $\mathrm{sgn}\{\alpha_{\mathrm{E}_i}\}\,|\alpha_{\mathrm{E}_i}|^{\gamma}$. This leads to the following weight update equation for the Exponential NNLMS algorithm in vector form :

$$\boldsymbol{\alpha}_{\mathrm{E}}(n+1) = \boldsymbol{\alpha}_{\mathrm{E}}(n) + \eta\, e(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n) \tag{2.72a}$$

with the i-th component of $\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)$ defined as

$$[\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)]_i = \mathrm{sign}\{\alpha_{\mathrm{E}_i}(n)\}\,|\alpha_{\mathrm{E}_i}(n)|^{\gamma}. \tag{2.72b}$$

TABLE 2.1 – Computational Complexity

Algorithm	Recursion	Computational cost per iter.				Main property
		+	×	sgn	$(\cdot)^\gamma$	
NNLMS	Eq. (2.14)	$2N$	$3N+1$			Original one, simplicity
Norm. NNLMS	Eq. (2.69)	$3N$	$4N+1$			Insensitivity to input power
Exp. NNLMS	Eq. (2.72)	$2N$	$3N+1$		N	Balance on weight convergence
S-S NNLMS	Eq. (2.73)	N	$2N$	N		Reduced computational cost

As in the gamma correction used in image processing, an exponent in the range $0 < \gamma < 1$ reduces the dynamic range of each $\alpha_{\mathrm{E}_i}(n)$. Large values of $\alpha_{\mathrm{E}_i}(n)$ will be compressed towards 1 and small values of $\alpha_{\mathrm{E}_i}(n)$ increased to prevent from stalling convergence. When $\gamma = 1$, the update equation degenerates into the NNLMS algorithm (2.14). Using $\gamma > 1$ is generally not recommended, as it tends to spread the vector component values.

2.5.3 Sign-Sign NNLMS

Like Sign-Sign LMS, which has been included in the CCITT standard for adaptive differential pulse code modulation [G72 1994], the motivation for introducing a Sign-Sign NNLMS algorithm is its computational simplicity and its robustness against disturbances [Koike 1998]. Replacing the input regressor vector and the estimation error in the update term by their signs reduces computation time and dynamic range requirements by replacing multiplications with shifts in real-time implementations. The Sign-Sign NNLMS algorithm is given by

$$\boldsymbol{\alpha}_{\mathrm{S}}(n+1) = \boldsymbol{\alpha}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{e(n)\}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\}\,\boldsymbol{\alpha}_{\mathrm{S}}(n) \tag{2.73}$$

After the two signs are evaluated, the i-th component update is given by

$$\alpha_{\mathrm{S}_i}(n+1) = \alpha_{\mathrm{S}_i}(n) \pm \eta\,\alpha_{\mathrm{S}_i}(n) \tag{2.74}$$

where the sign before η is determined by $\mathrm{sgn}\{e(n)x_i(n)\}$. The step-size η is usually selected as a power of 2^{-1}, say $\eta = 2^{-m}$ for some integer $m > 0$. In this case, Equation (2.74) can be efficiently implemented using shift-add operations. Moreover, the non-negativity constraint will be always satisfied if $\boldsymbol{\alpha}_{\mathrm{S}}$ is initialized with a positive vector and $0 < \eta < 1$. Table 2.1 compares the computation complexities of NNLMS and its three variants described above. The rightmost column describes the anticipated property of each algorithm, to be verified in the following.

2.6 Mean weight behavior for the variants

Convergence in the mean sense of the NNLMS algorithm (2.14) has been studied for a stationary environment. We now study the stochastic behavior of the

NNLMS variants introduced in Section 2.5 for fixed step sizes and for a time variant unconstrained solution given by

$$\boldsymbol{\alpha}^*(n) = \boldsymbol{\alpha}_{\mathrm{o}}^*(n) + \boldsymbol{\xi}(n) \tag{2.75}$$

where $\boldsymbol{\alpha}_{\mathrm{o}}^*(n)$ is a deterministic time-variant mean and $\boldsymbol{\xi}(n)$ is zero-mean, Gaussian, with covariance matrix $\boldsymbol{\Xi} = \sigma_\xi^2 \boldsymbol{I}$ and independent of any other signal. This simple model provides some information on how the performance of the proposed algorithms is affected by a time variant optimal solution which consists of a deterministic trajectory and a random perturbation. The model (2.75) leads to a tractable analysis and permits inferences about the behavior of the algorithms in time variant environments by varying the mean value $\boldsymbol{\alpha}_{\mathrm{o}}^*(n)$ and the power σ_ξ^2 of $\boldsymbol{\xi}(n)$.

To conserve space and to simplify notation without ambiguity, from now on we use the generic notations $\boldsymbol{\alpha}(n)$ and $\alpha_i(n)$ whenever the given expression is valid for all the algorithms under study. Notations $\boldsymbol{\alpha}_{\mathrm{N}}$, $\boldsymbol{\alpha}_{\mathrm{E}}$ and $\boldsymbol{\alpha}_{\mathrm{S}}$ will be used only for expressions which are specific to the corresponding algorithm. The same notational observation applies to the any vector or matrix when referring to specific algorithm.

For the analyses that follow, we shall define the weight error vector with respect to the unconstrained solution $\boldsymbol{\alpha}^*(n)$ as

$$\tilde{\boldsymbol{v}}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^*(n) \tag{2.76}$$

and the weight error vector with respect to the mean unconstrained solution $\boldsymbol{\alpha}_{\mathrm{o}}^*(n)$ as

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}_{\mathrm{o}}^*(n) \tag{2.77}$$

The two vectors are related by $\tilde{\boldsymbol{v}}(n) = \boldsymbol{v}(n) - \boldsymbol{\xi}(n)$.

2.6.1 Normalized NNLMS algorithm

Using (2.76) with the appropriate subscript in (2.69) and $e(n) = z(n) - (\boldsymbol{v}_{\mathrm{N}}(n) - \boldsymbol{\xi}(n))^\top \boldsymbol{x}(n)$ yields

$$\begin{aligned}
&\boldsymbol{v}_{\mathrm{N}}(n+1) \\
&= \boldsymbol{v}_{\mathrm{N}}(n) + \frac{\eta\, z(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{v}_{\mathrm{N}}(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} + \frac{\eta\, z(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{o}}^*(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} \\
&\quad - \frac{\eta\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}_{\mathrm{N}}(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} - \frac{\eta\, \boldsymbol{D}_x(n)\, \boldsymbol{v}_{\mathrm{N}}(n)\, \boldsymbol{v}_{\mathrm{N}}^\top(n)\, \boldsymbol{x}(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} \\
&\quad + \frac{\eta\, \boldsymbol{D}_x(n)\, \boldsymbol{v}_{\mathrm{N}}(n)\, \boldsymbol{\xi}^\top(n)\, \boldsymbol{x}(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} + \frac{\eta\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\, \boldsymbol{\xi}^\top(n)\, \boldsymbol{x}(n)}{\boldsymbol{x}^\top(n)\, \boldsymbol{x}(n) + \varepsilon} - \boldsymbol{\Delta}_{\mathrm{N}}(n)
\end{aligned} \tag{2.78}$$

where $\boldsymbol{\Delta}_{\mathrm{N}}(n) = \boldsymbol{\alpha}_{\mathrm{o}}^*(n+1) - \boldsymbol{\alpha}_{\mathrm{o}}^*(n)$ is a deterministic vector proportional to the derivative of the mean unconstrained optimal solution. Taking the expected value of (2.78) and noting that the expectations of the second, third, sixth and seventh

terms on the r.h.s. are equal to zero by virtue of the natures of $z(n)$ and $\boldsymbol{\xi}(n)$, and that $\boldsymbol{\Delta}_{\mathrm{N}}(n)$ is deterministic, yields

$$E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} = E\{\boldsymbol{v}_{\mathrm{N}}(n)\} - \eta E\left\{\frac{1}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)+\varepsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\} - \eta E\left\{\frac{1}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)+\varepsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\,\boldsymbol{x}(n)\right\} - \boldsymbol{\Delta}_{\mathrm{N}}(n). \tag{2.79}$$

Using the independence assumption, the second expectation in the r.h.s. of (2.79) can be written as

$$\begin{aligned} &E\left\{\frac{1}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)+\varepsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\} \\ &= \boldsymbol{D}_{\alpha_{\mathrm{o}}^{*}}(n)\,E\left\{\frac{\boldsymbol{x}(n)\,\boldsymbol{x}^{\top}(n)}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)+\varepsilon}\right\}E\{\boldsymbol{v}_{\mathrm{N}}(n)\} \end{aligned} \tag{2.80}$$

Evaluation of the first expected value in the r.h.s. of (2.80) requires approximations. Each numerator element is given by $x(n-i)x(n-j)$. The random part of the denominator is given by $\sum_{k=0}^{N-1} x^2(n-k)$. A common approximation that works well for reasonably large N is to neglect the correlation between these two variables, as the latter tends to vary much slower than the former [Samson 1983, Almeida 2005]. Moreover, given its slow variation we approximate $\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)$ by its mean value $N\sigma_x^2$, which is reasonable for large values of N. Using these approximations yields

$$E\left\{\frac{1}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)+\varepsilon}\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\right\} \approx \frac{1}{N\sigma_x^2+\varepsilon}\boldsymbol{D}_{\alpha_{\mathrm{o}}^{*}}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{v}_{\mathrm{N}}(n)\}. \tag{2.81}$$

Using again $\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n) \approx N\sigma_x^2$ and removing it from the expected value, the i-th component of the second expectation in the r.h.s. of (2.79) is

$$\Big[\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\,\boldsymbol{x}(n)\Big]_i = \sum_{j=1}^{N} x(n-i+1)\,v_{\mathrm{N}_i}(n)\,v_{\mathrm{N}_j}(n)\,x(n-j+1). \tag{2.82}$$

Taking the expectation, using the independence assumption and defining $\boldsymbol{K}_{\mathrm{N}}(n) = E\{\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\}$ we obtain

$$\begin{aligned} \Big[E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\,\boldsymbol{x}(n)\}\Big]_i &= \sum_{j=1}^{N} r_x(j-i)\,[\boldsymbol{K}_{\mathrm{N}}(n)]_{ij} \\ &= [\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)]_{ii} \end{aligned} \tag{2.83}$$

which yields $E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\,\boldsymbol{x}(n)\} = \mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\}$, where $\mathrm{Diag}\{\cdot\}$ denotes the vector of diagonal entries in the matrix. Hence, (2.79) becomes

$$\begin{aligned} E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} = &\left(\boldsymbol{I} - \frac{\eta}{N\sigma_x^2+\varepsilon}\,\boldsymbol{D}_{\alpha_{\mathrm{o}}^{*}}(n)\,\boldsymbol{R}_x\right)E\{\boldsymbol{v}_{\mathrm{N}}(n)\} \\ &- \frac{\eta}{N\sigma_x^2+\varepsilon}\,\mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\} - \boldsymbol{\Delta}_{\mathrm{N}}(n) \end{aligned} \tag{2.84}$$

This recursion for $E\{\boldsymbol{v}_{\mathrm{N}}(n)\}$ requires a model for $\boldsymbol{K}_{\mathrm{N}}(n)$. A recursive model will be derived for $\boldsymbol{K}_{\mathrm{N}}(n)$ in Section 2.7, see (2.100). That model can be used along with (2.84) to predict the mean weight behavior of the Normalized NNLMS algorithm. Nevertheless, we have found that a sufficiently accurate and more intuitive mean behavior model can be obtained by neglecting the weight error fluctuations and using the following separation approximation

$$\boldsymbol{K}_{\mathrm{N}}(n) \approx E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\, E\{\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\}. \tag{2.85}$$

This approximation has been successfully used in the analysis of NNLMS algorithm. Extensive simulation results have shown that this approximation achieves adequate accuracy in modeling the mean behavior of the adaptive weights. We thus obtain the following model

$$\begin{aligned} E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} = &\left(\boldsymbol{I} - \frac{\eta}{N\sigma_x^2+\varepsilon}\, \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\, \boldsymbol{R}_x\right) E\{\boldsymbol{v}_{\mathrm{N}}(n)\} \\ &- \frac{\eta}{N\sigma_x^2+\varepsilon}\, \mathrm{Diag}\left\{\boldsymbol{R}_x\, E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\, E\{\boldsymbol{v}_{\mathrm{N}}^{\top}(n)\}\right\} - \boldsymbol{\Delta}_{\mathrm{N}}(n). \end{aligned} \tag{2.86}$$

2.6.2 Exponential NNLMS algorithm

Using (2.76) with the appropriate subscript in (2.72), and

$$\begin{aligned} e(n) &= y(n) - \boldsymbol{\alpha}_{\mathrm{E}}^{\top}(n)\, \boldsymbol{x}(n) \\ &= z(n) - (\boldsymbol{v}_{\mathrm{E}}(n) - \boldsymbol{\xi}(n))^{\top}\, \boldsymbol{x}(n) \end{aligned} \tag{2.87}$$

and considering that $\mathrm{sign}\{\alpha_{\mathrm{E}_i}\}\, |\alpha_{\mathrm{E}_i}|^{\gamma}$ is equal to the real solution of $\alpha_{\mathrm{E}_i}^{\gamma}$, the Exponential NNLMS weight error update equation can be written as

$$\begin{aligned} \boldsymbol{v}_{\mathrm{E}}(n+1) &= \boldsymbol{v}_{\mathrm{E}}(n) + \eta\, e(n)\, \boldsymbol{D}_x(n)\, \left(\boldsymbol{v}_{\mathrm{E}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)} \\ &= \boldsymbol{v}_{\mathrm{E}}(n) + \eta\, z(n)\, \boldsymbol{D}_x(n)\, \left(\boldsymbol{v}_{\mathrm{E}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)} \\ &\quad - \eta\, \boldsymbol{D}_x(n)\, \left(\boldsymbol{v}_{\mathrm{E}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)}\, \boldsymbol{v}_{\mathrm{E}}^{\top}(n)\, \boldsymbol{x}(n) \\ &\quad + \eta\, \boldsymbol{\xi}^{\top}(n)\boldsymbol{x}(n)\, \left(\boldsymbol{v}_{\mathrm{E}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)} - \boldsymbol{\Delta}_{\mathrm{E}}(n). \end{aligned} \tag{2.88}$$

where $\left(\boldsymbol{v}_{\mathrm{E}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)}$ is a real vector.

The nonlinear term $(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}_{\mathrm{o}}^*(n)(n))^{(\gamma)}$ on the r.h.s. complicates the evaluation of the expected value of (2.88) because the statistics of the weight error vector are unknown. We have again found out that using a zero-th order approximation of $\boldsymbol{v}_{\mathrm{E}}(n)$ is sufficient to provide a reasonably good model for the mean weight error behavior. Thus, we make

$$(v_{\mathrm{E}_i}(n) + \alpha_{\mathrm{o}_i}^*(n))^{\gamma} \approx (E\,\{v_{\mathrm{E}_i}(n)\} + \alpha_{\mathrm{o}_i}^*(n))^{\gamma}. \tag{2.89}$$

Using (2.89) in (2.88), taking the expected value and considering the statistical properties of $z(n)$ and $\boldsymbol{\xi}(n)$ yields

$$\begin{aligned} & E\{\boldsymbol{v}_{\mathrm{E}}(n+1)\} \\ \approx & E\{\boldsymbol{v}_{\mathrm{E}}(n)\} - \eta\, \boldsymbol{D}_x(n) \left(E\{\boldsymbol{v}_{\mathrm{E}}(n)\} + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)\right)^{(\gamma)} E\{\boldsymbol{v}_{\mathrm{E}}^\top(n)\}\, \boldsymbol{x}(n) - \boldsymbol{\Delta}_{\mathrm{E}}(n) \\ = & \Big(\boldsymbol{I}_N - \eta\, \boldsymbol{D}_r(n) \boldsymbol{R}_x\Big) E\{\boldsymbol{v}_{\mathrm{E}}(n)\} - \boldsymbol{\Delta}_{\mathrm{E}}(n) \end{aligned} \tag{2.90}$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix and $\boldsymbol{D}_r(n)$ is an $N \times N$ diagonal matrix defined as $\boldsymbol{D}_r(n) = \mathrm{Diag}\{\boldsymbol{r}(n)\}$ with $\boldsymbol{r}(n)$ being the $N \times 1$ vector whose i-th component is $r_i(n) = \left(E\left\{v_{\mathrm{E}_i}(n)\right\} + \alpha_{\mathrm{o}_i}^*(n)\right)^\gamma$. It is simple to verify that this model collapses to the NNLMS model derived in Section 2.3.1 for $p = q = 1$.

2.6.3 Sign-Sign NNLMS algorithm

The statistical analysis of the Sign-Sign NNLMS algorithm behavior is complicated by the fact that the weight update term is discontinuous in both the input vector $\boldsymbol{x}(n)$ and the error $e(n)$ [Eweda 1999]. To make that analysis tractable, we consider the case of input signal $x(n)$ zero-mean and Gaussian [Koike 1998, Eweda 1999].

Using (2.76) with the appropriate subscript in (2.73) and $e(n) = z(n) - (\boldsymbol{v}_{\mathrm{S}}(n) - \boldsymbol{\xi}(n))^\top \boldsymbol{x}(n)$, the Sign-Sign NNLMS weight error update equation can be written as

$$\begin{aligned} \boldsymbol{v}_{\mathrm{S}}(n+1) = & \boldsymbol{v}_{\mathrm{S}}(n) + \eta\, \mathrm{sgn}\{z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\} \\ & \cdot \mathrm{sgn}\{\boldsymbol{D}_x(n)\}\, (\boldsymbol{v}_{\mathrm{S}}(n) + \boldsymbol{\alpha}_{\mathrm{o}}^*(n)) - \boldsymbol{\Delta}_{\mathrm{S}}(n). \end{aligned} \tag{2.91}$$

Note that, unlike the former two variants, the non-stationarity effect appears in the weight error update equation (2.91) as a nonlinear function of $\boldsymbol{\xi}(n)$. The i-th component of (2.91) is given by

$$\begin{aligned} v_{\mathrm{S}_i}(n+1) = & v_{\mathrm{S}_i}(n) + \eta\, \mathrm{sgn}\big\{z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\big\} \\ & \cdot \mathrm{sgn}\big\{x_i(n)\big\}\, (v_{\mathrm{S}_i}(n) + \alpha_{\mathrm{o}_i}^*(n)) - \Delta_{\mathrm{S}_i}(n) \end{aligned} \tag{2.92}$$

To determine the expected value of (2.92), we first note that it has been demonstrated in [Eweda 1999, Dasgupta 1990] using Price's theorem [Price 1958] that

$$E\Big\{\mathrm{sgn}\{\theta_1\}\, \mathrm{sgn}\{\theta_2\}\Big\} = \frac{2}{\pi}\, \sin^{-1}\left(\frac{E\{\theta_1\theta_2\}}{\sigma_{\theta_1}\sigma_{\theta_2}}\right) \tag{2.93}$$

for θ_1 and θ_2 two zero-mean jointly Gaussian variables with variances $\sigma_{\theta_1}^2$ and $\sigma_{\theta_2}^2$, respectively. Then, noting that $z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)$ and $x_i(n)$ are zero-mean Gaussian when conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$, they are jointly Gaussian[5]. Using the result in (2.93) we obtain

$$\begin{aligned} & E\Big\{\mathrm{sgn}\{z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\, \boldsymbol{x}(n)\}\, \mathrm{sgn}\{x_i(n)\} | \boldsymbol{v}_{\mathrm{S}}(n), \boldsymbol{\xi}(n)\Big\} \\ \approx & \frac{2}{\pi}\, \sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\, \boldsymbol{v}_{\mathrm{S}}(n) - \boldsymbol{R}_i^\top\, \boldsymbol{\xi}(n)}{\sigma_x\, \sigma_{e|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)}}\right) \end{aligned} \tag{2.94}$$

5. As $\boldsymbol{x}(n)$ and $z(n)$ are independent and both Gaussian, $[\boldsymbol{x}(n), z(n)]$ is jointly Gaussian. When conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$, $[\boldsymbol{x}^\top(n), z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\boldsymbol{x}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)]$ is jointly Gaussian as a linear transformation of $[\boldsymbol{x}^\top(n), z(n)]$.

where $\boldsymbol{R}_i$ the i-th column of $\boldsymbol{R}$ and $\sigma^2_{e|\boldsymbol{v}_{\mathrm{S}}(n),\boldsymbol{\xi}(n)}$ is the variance of $e(n)$ when conditioned on $\boldsymbol{v}_{\mathrm{S}}(n)$ and $\boldsymbol{\xi}(n)$.

Now, since $\sin^{-1}(\cdot)$ is a nonlinear function and the distribution of its argument is unknown, we proceed as we did for the Exponential NNLMS algorithm and replace the nonlinear function by its zero-th order approximation

$$E\Big\{\mathrm{sgn}\{z(n)-\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\boldsymbol{x}(n)+\boldsymbol{\xi}^{\top}(n)\boldsymbol{x}(n)\}\,\mathrm{sgn}\{x_i(n)\}\Big\} \approx \frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^{\top}\,E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right) \tag{2.95}$$

with

$$\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}} = \sqrt{\sigma_z^2 + \mathrm{tr}\Big\{\boldsymbol{R}_x\,E\{\boldsymbol{v}_{\mathrm{S}}(n)\}E\{\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\}\Big\} + \mathrm{trace}\{\boldsymbol{R}_x\boldsymbol{\Xi}\}} \tag{2.96}$$

Taking the expected value of (2.92), using the results (2.94) and (2.95) and expressing the result in vector form yields the mean weight error vector behavior model

$$E\{\boldsymbol{v}_{\mathrm{S}}(n+1)\} = \Big(\boldsymbol{I}_N + \eta\,\boldsymbol{D}_p(n)\Big)E\{\boldsymbol{v}_{\mathrm{S}}(n)\} + \eta\,\boldsymbol{D}_p(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n) - \boldsymbol{\Delta}_{\mathrm{S}}(n) \tag{2.97}$$

where $\boldsymbol{D}_p(n)$ is the $N\times N$ diagonal matrix $\boldsymbol{D}_p(n) = \mathrm{Diag}\{\boldsymbol{p}(n)\}$ with $\boldsymbol{p}(n)$ being the $N\times 1$ vector whose i-th entry is given by (2.95).

2.7 Second-order moment analysis for the variants

For mathematical feasibility, in this section the four statistical assumptions in Section 2.4 are likewise used. In what follows, the second-order model for these variants are studied separately.

The excess means square estimation error (EMSE) is given by

$$J_{\mathrm{EMSE}}(n) = E\left\{\tilde{\boldsymbol{v}}^{\top}(n)\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)\tilde{\boldsymbol{v}}(n)\right\} \tag{2.98}$$

Using the relation between $\tilde{\boldsymbol{v}}(n)$ and $\boldsymbol{v}(n)$, the properties of $\boldsymbol{\xi}(n)$, and noting from (2.78), (2.88) and (2.91) that $\boldsymbol{v}(n)$ and $\boldsymbol{\xi}(n)$ are independent, we can write $\zeta(n)$ as

$$\begin{aligned} J_{\mathrm{EMSE}}(n) &= E\left\{(\boldsymbol{v}(n)-\boldsymbol{\xi}(n))^{\top}\boldsymbol{x}(n)\boldsymbol{x}^{\top}(n)(\boldsymbol{v}(n)-\boldsymbol{\xi}(n))\right\} \\ &= \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\} \end{aligned} \tag{2.99}$$

with $\boldsymbol{K}(n) = E\left\{\boldsymbol{v}(n)\boldsymbol{v}^{\top}(n)\right\}$. The term $\mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\}$ is the contribution of the non-stationarity of the system to the EMSE caused by random perturbation due to the weight lag. In the following, we derive recursive models for $\boldsymbol{K}(n)$ for each of the algorithms.

2.7.1 Normalized NNLMS algorithm

Post-multiplying (2.78) by its transpose, taking the expectation, using the approximation $\boldsymbol{x}^\top(n)\,\boldsymbol{x}(n) \approx N\sigma_x^2$ in (2.78), defining $\tilde{\eta} = \eta/(N\sigma_x^2 + \varepsilon)$ and using assumptions A1–A4 leads to

$$\begin{aligned}\boldsymbol{K}_{\mathrm{N}}(n+1) = \boldsymbol{K}_{\mathrm{N}}(n) &- \tilde{\eta}\left(\boldsymbol{P}_{1_{\mathrm{N}}}(n)\,\boldsymbol{K}_{\mathrm{N}}(n) + \boldsymbol{K}_{\mathrm{N}}(n)\,\boldsymbol{P}_{1_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{5_{\mathrm{N}}}(n) + \boldsymbol{P}_{5_{\mathrm{N}}}^\top(n)\right)\\ &+ \tilde{\eta}^2\left(\boldsymbol{P}_{6_{\mathrm{N}}}(n) + \boldsymbol{P}_{7_{\mathrm{N}}}(n) + \boldsymbol{P}_{7_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{8_{\mathrm{N}}}(n)\right)\\ &+ \tilde{\eta}^2\sigma_z^2\left(\boldsymbol{P}_{2_{\mathrm{N}}}(n) + \boldsymbol{P}_{3_{\mathrm{N}}}(n) + \boldsymbol{P}_{3_{\mathrm{N}}}^\top(n) + \boldsymbol{P}_{4_{\mathrm{N}}}(n)\right)\\ &+ \tilde{\eta}^2\left(\boldsymbol{P}_{9_{\mathrm{N}}}(n) + \boldsymbol{P}_{10_{\mathrm{N}}}(n) + \boldsymbol{P}_{11_{\mathrm{N}}}(n) + \boldsymbol{P}_{11_{\mathrm{N}}}^\top(n)\right) + \boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{N}}}(n)\end{aligned} \tag{2.100}$$

with

$$\boldsymbol{P}_{1_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\,\boldsymbol{x}(n)\} = \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\,\boldsymbol{R}_x. \tag{2.101}$$

$$\boldsymbol{P}_{2_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)^\top\boldsymbol{D}_x(n)\} = \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n). \tag{2.102}$$

$$\boldsymbol{P}_{3_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)^\top\boldsymbol{D}_x(n)\} \approx E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}\,\boldsymbol{R}_x\,\boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n). \tag{2.103}$$

$$\boldsymbol{P}_{4_{\mathrm{N}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\} \approx \boldsymbol{R}_x \circ \boldsymbol{K}_{\mathrm{N}}(n) \tag{2.104}$$

where $\circ$ denotes the so-called Hadamard entry-wise product,

$$\boldsymbol{P}_{5_{\mathrm{N}}}(n) = E\{\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{x}^\top(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\} \approx \boldsymbol{K}_{\mathrm{N}}(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}. \tag{2.105}$$

$$\begin{aligned}\boldsymbol{P}_{6_{\mathrm{N}}}(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)^\top\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\,\boldsymbol{Q}_{\mathrm{N}}(n)\,\boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n).\end{aligned} \tag{2.106}$$

where the matrix $\boldsymbol{Q}_{\mathrm{N}}(n)$ is defined by

$$\boldsymbol{Q}_{\mathrm{N}}(n) = \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\,(2\,\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\,\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{N}}(n)\}\,\boldsymbol{R}_x)$$

$$\begin{aligned}\boldsymbol{P}_{7_{\mathrm{N}}}(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\}\\ &\approx \boldsymbol{D}_{\alpha_{\mathrm{o}}^*}(n)\,\boldsymbol{Q}_{\mathrm{N}}(n)\,E\{\boldsymbol{D}_{v_{\mathrm{N}}}(n)\}.\end{aligned} \tag{2.107}$$

$$\begin{aligned}\boldsymbol{P}_{8_{\mathrm{N}}}(n) &= E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}_{\mathrm{N}}(n)\,\boldsymbol{v}_{\mathrm{N}}^\top(n)\,\boldsymbol{D}_x(n)\}\\ &= \boldsymbol{Q}_{\mathrm{N}}(n) \circ \boldsymbol{K}_{\mathrm{N}}(n).\end{aligned} \tag{2.108}$$

$$\boldsymbol{P}_{9_{\mathrm{N}}}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{v}_{\mathrm{N}}^\top(n)\boldsymbol{D}_x(n)\} \tag{2.109}$$

$$\boldsymbol{P}_{10_{\mathrm{N}}}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\boldsymbol{D}_x(n)\} \tag{2.110}$$

and

$$\boldsymbol{P}_{11_{\mathrm{N}}}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_x(n)\boldsymbol{v}_{\mathrm{N}}(n)\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\boldsymbol{D}_x(n)\} \tag{2.111}$$

In obtaining (2.100), it was considered that the products of the last two terms of (2.78) by the other terms lead to zero mean values due to the properties of $\boldsymbol{\xi}(n)$.

Expected values $\boldsymbol{P}_{1_{\mathrm{N}}}(n)$ through $\boldsymbol{P}_{8_{\mathrm{N}}}(n)$ correspond to the terms of the weight error vector recursive equation derived for the NNLMS algorithm in Section 2.4 with $\tilde{\eta}$ substituted for η. Thus, we use the results from Section 2.4 and indicate their values

directly in (2.101) through (2.108). We now derive expressions for $\boldsymbol{P}_{9_\mathrm{N}}(n)$ through $\boldsymbol{P}_{11_\mathrm{N}}(n)$. These terms convey the effect of the environment non-stationarity due to the random variations of system weights.

Computing (i,j)-th entry of $\boldsymbol{P}_{9_\mathrm{N}}(n)$ yields

$$\begin{aligned}[\boldsymbol{P}_{9_\mathrm{N}}]_{ij}(n) &= E\left\{\sum_k\sum_l \xi_k(n)\xi_l(n)x_k(n)x_l(n)x_i(n)v_{\mathrm{N}_i}(n)v_{\mathrm{N}_j}(n)x_j(n)\right\}\\ &= \sum_k\sum_l E\left\{\xi_k(n)\xi_l(n)\right\}\, E\left\{v_{\mathrm{N}_i}(n)v_{\mathrm{N}_j}(n)\right\}\, E\left\{x_k(n)x_l(n)x_i(n)x_j(n)\right\}\end{aligned} \tag{2.112}$$

As $E\left\{\xi_k(n)\xi_l(n)\right\} \neq 0$ only for $k=l$,

$$[\boldsymbol{P}_{9_\mathrm{N}}]_{ij}(n) = \sigma_\xi^2 \sum_k [\boldsymbol{K}_\mathrm{N}(n)]_{ij}\, E\left\{x_k^2(n)x_i(n)x_j(n)\right\}$$

Using the Gaussian moment factorizing theorem yields

$$\begin{aligned}\sum_k E\left\{x_k^2(n)x_i(n)x_j(n)\right\} &= \sum_k \left([\boldsymbol{R}_x]_{ij}[\boldsymbol{R}_x]_{kk} + [\boldsymbol{R}_x]_{ik}[\boldsymbol{R}_x]_{jk}\right)\\ &= \left[\boldsymbol{R}_x\,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x\right]_{ij}.\end{aligned} \tag{2.113}$$

This enables us to write the result in matrix form

$$\boldsymbol{P}_{9_\mathrm{N}}(n) = \sigma_\xi^2\, \boldsymbol{K}_\mathrm{N}(n) \circ (\boldsymbol{R}_x\,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \tag{2.114}$$

Similarly, we have

$$\boldsymbol{P}_{10_\mathrm{N}}(n) = \sigma_\xi^2 \left(\boldsymbol{\alpha}_\mathrm{o}^*(n)\,\boldsymbol{\alpha}_\mathrm{o}^{*\top}(n)\right) \circ (\boldsymbol{R}_x\,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \tag{2.115}$$

$$\boldsymbol{P}_{11_\mathrm{N}}(n) = \sigma_\xi^2 \left(E\left\{\boldsymbol{v}_\mathrm{N}(n)\right\}\boldsymbol{\alpha}_\mathrm{o}^{*\top}(n)\right) \circ (\boldsymbol{R}_x\,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}_x\,\boldsymbol{R}_x) \tag{2.116}$$

The last term $\boldsymbol{K}_{\boldsymbol{\Delta}_\mathrm{N}}(n)$ conveys the effect of deterministic variation of the mean of system weights. Observing the terms multiplied with $\boldsymbol{\Delta}_\mathrm{N}(n)$, we have

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{\Delta}_\mathrm{N}}(n) = &-\boldsymbol{\Delta}_\mathrm{N}(n)\boldsymbol{\Delta}_\mathrm{N}^\top(n) - \boldsymbol{\Delta}_\mathrm{N}(n)E\left\{\boldsymbol{v}_\mathrm{N}(n+1) + \boldsymbol{\Delta}_\mathrm{N}(n)\right\}^\top\\ &- E\left\{\boldsymbol{v}_\mathrm{N}(n+1) + \boldsymbol{\Delta}_\mathrm{N}(n)\right\}\boldsymbol{\Delta}_\mathrm{N}^\top(n)\end{aligned} \tag{2.117}$$

2.7.2 Exponential NNLMS algorithm

The second order moment analysis of the Exponential NNLMS algorithm requires an improvement on approximation (2.89) for $(\boldsymbol{v}_\mathrm{E}(n) + \boldsymbol{\alpha}_\mathrm{o}^*(n))^{(\gamma)}$ in (2.88). We use instead the following first order approximation for the real-valued solution of $(v_{\mathrm{E}_i}(n) + \alpha_{\mathrm{o}_i}^*(n))^\gamma$:

$$\begin{aligned}(v_{\mathrm{E}_i}(n) + \alpha_{\mathrm{o}_i}^*(n))^\gamma \approx &(E\left\{v_{\mathrm{E}_i}(n)\right\} + \alpha_{\mathrm{o}_i}^*(n))^\gamma + \gamma\, g(E\left\{v_{\mathrm{E}_i}(n)\right\})\\ &\cdot (E\left\{v_{\mathrm{E}_i}(n)\right\} + \alpha_{\mathrm{o}_i}^*(n))^{\gamma-1}\,(v_{\mathrm{E}_i}(n) - E\left\{v_{\mathrm{E}_i}(n)\right\})\end{aligned} \tag{2.118}$$

where

$$g(E\{v_{\mathrm{E}_i}(n)\}) = 1 - \Big[u\big(E\{v_{\mathrm{E}_i}(n)\} + \alpha^*_{\mathrm{o}_i}(n) + \varepsilon\big) - u\big(E\{v_{\mathrm{E}_i}(n)\} + \alpha^*_{\mathrm{o}_i}(n) - \varepsilon\big)\Big] \tag{2.119}$$

with $u(\cdot)$ being the unit step function and ε a small constant. The reason to include the gate function g about $E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n)$ in the regular Taylor series is that the derivative of $(v_{\mathrm{E}_i}(n)+\alpha^*_{\mathrm{o}_i}(n))^{\gamma}$ tends to infinity if $v_{\mathrm{E}_i}(n)$ approaches $-\alpha^*_{\mathrm{o}_i}(n)$. It is simple to verify that $\lim_{v_{\mathrm{E}_i}(n)\to-\alpha^*_{\mathrm{o}_i}(n)} g(E\{v_{\mathrm{E}_i}(n)\})\,(E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^{\gamma-1} = 0$. The zero-th order approximation is sufficient about the point where the function is equal to zero. With this new approximation, the term on $v_{\mathrm{E}_i}(n)$ in (2.118) will include moments of the weight error vector which are necessary to proper modeling its fluctuations.

To use vector notation, we define two deterministic vectors $\boldsymbol{r}(n)$ and $\boldsymbol{s}(n)$ whose i-th entries are respectively

$$r_i = (E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^{\gamma} - \gamma\, g(E\{v_{\mathrm{E}_i}(n)\})\,(E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^{\gamma-1}\,E\{v_{\mathrm{E}_i}(n)\}$$

$$s_i = \gamma\, g(E\{v_{\mathrm{E}_i}(n)\})\,(E\{v_{\mathrm{E}_i}(n)\}+\alpha^*_{\mathrm{o}_i}(n))^{\gamma-1}$$

respectively. We define also the corresponding diagonal matrices $\boldsymbol{D}_r(n) = \mathrm{Diag}\{\boldsymbol{r}(n)\}$ and $\boldsymbol{D}_s(n) = \mathrm{Diag}\{\boldsymbol{s}(n)\}$. With these new definitions, the linear approximation can be written in vector form as

$$\big(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}^*_{\mathrm{o}}(n)\big)^{(\gamma)} \approx \boldsymbol{r}(n) + \boldsymbol{D}_s(n)\,\boldsymbol{v}_{\mathrm{E}}(n) \tag{2.120}$$

Post-multiplying (2.88) by its transpose, using (2.120), taking the expected value, using assumptions A1–A4 and defining matrix $\boldsymbol{D}_{v_{\mathrm{E}}}(n) = \mathrm{Diag}\{\boldsymbol{v}_{\mathrm{E}}(n)\}$ yields, after simple algebraic manipulations

$$\begin{aligned}\boldsymbol{K}_{\mathrm{E}}(n+1) =& \boldsymbol{K}_{\mathrm{E}}(n) - \eta\left(\boldsymbol{P}_{1_{\mathrm{E}}}(n)\,\boldsymbol{K}_{\mathrm{E}}(n) + \boldsymbol{K}_{\mathrm{E}}(n)\,\boldsymbol{P}^{\top}_{1_{\mathrm{E}}}(n)\right)\\ &- \eta\left(\boldsymbol{P}_{5_{\mathrm{E}}}(n)\boldsymbol{K}_{\mathrm{E}}(n) + \boldsymbol{K}_{\mathrm{E}}(n)\boldsymbol{P}^{\top}_{5_{\mathrm{E}}}(n)\right) + \eta^2\left(\boldsymbol{P}_{6_{\mathrm{E}}}(n) + \boldsymbol{P}_{7_{\mathrm{E}}}(n)\right.\\ &\left.+ \boldsymbol{P}^{\top}_{7_{\mathrm{E}}}(n) + \boldsymbol{P}_{8_{\mathrm{E}}}(n)\right) + \eta^2\sigma_z^2\left(\boldsymbol{P}_{2_{\mathrm{E}}}(n) + \boldsymbol{P}_{3_{\mathrm{E}}}(n) + \boldsymbol{P}^{\top}_{3_{\mathrm{E}}}(n)\right.\\ &\left.+ \boldsymbol{P}_{4_{\mathrm{E}}}(n)\right) + \eta^2\boldsymbol{P}_{9_{\mathrm{E}}}(n) + \boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{E}}}(n)\end{aligned} \tag{2.121}$$

with the eight moments $\boldsymbol{P}_{1_{\mathrm{E}}}(n) - \boldsymbol{P}_{8_{\mathrm{E}}}(n)$ given by

$$\boldsymbol{P}_{1_{\mathrm{E}}}(n) = E\Big\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{x}^{\top}(n)\Big\} = \boldsymbol{D}_r(n)\,\boldsymbol{R}_x. \tag{2.122}$$

$$\boldsymbol{P}_{2_{\mathrm{E}}}(n) = E\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{r}^{\top}(n)\,\boldsymbol{D}_x(n)\} \approx \boldsymbol{D}_r(n)\,\boldsymbol{R}_x\,\boldsymbol{D}_r(n). \tag{2.123}$$

$$\begin{aligned}\boldsymbol{P}_{3_{\mathrm{E}}}(n) =& E\{\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{v}^{\top}_{\mathrm{E}}(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\}\\ \approx& \boldsymbol{D}_r(n)\,\boldsymbol{R}_x\,E\{\boldsymbol{D}_{v_{\mathrm{E}}}(n)\}\,\boldsymbol{D}_s(n)\end{aligned} \tag{2.124}$$

$$\begin{aligned}\boldsymbol{P}_{4_{\mathrm{E}}}(n) =& E\{\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{v}_{\mathrm{E}}(n)\,\boldsymbol{v}^{\top}_{\mathrm{E}}(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\}\\ \approx& \boldsymbol{D}_s(n)\big(\boldsymbol{R}_x \circ \boldsymbol{K}_{\mathrm{E}}(n)\big)\,\boldsymbol{D}_s(n).\end{aligned} \tag{2.125}$$

$$\boldsymbol{P}_{5_{\mathrm{E}}}(n) = \big\{\boldsymbol{v}^{\top}_{\mathrm{E}}(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\big\} = \mathrm{Diag}\Big\{\boldsymbol{R}_x\,E\{\boldsymbol{v}_{\mathrm{E}}(n)\}\Big\}\boldsymbol{D}_s(n) \tag{2.126}$$

$$\begin{aligned}\boldsymbol{P}_{6_{\mathrm{E}}}(n) =\ & E\{\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{r}^{\top}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{E}}(n)\}\\ \approx\ & \boldsymbol{D}_r(n)\,\boldsymbol{Q}_{\mathrm{E}}(n)\,\boldsymbol{D}_r(n).\end{aligned} \tag{2.127}$$

where the matrix $\boldsymbol{Q}_{\mathrm{E}}(n)$ in above equations is defined by $\boldsymbol{Q}_{\mathrm{E}}(n) = 2\,\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{E}}(n)\,\boldsymbol{R}_x + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}_{\mathrm{E}}(n)\}\,\boldsymbol{R}_x$

$$\begin{aligned}\boldsymbol{P}_{7_{\mathrm{E}}}(n) &= E\{\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{r}(n)\,\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{E}}(n)\}\\ &\approx \boldsymbol{D}_r(n)\,\boldsymbol{Q}_{\mathrm{E}}(n)\,E\{\boldsymbol{D}_{v_{\mathrm{E}}}(n)\}\,\boldsymbol{D}_s(n).\end{aligned} \tag{2.128}$$

and

$$\begin{aligned}\boldsymbol{P}_{8_{\mathrm{E}}}(n) &= E\{\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\,\boldsymbol{x}(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{v}_{\mathrm{E}}(n)\,\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\,\boldsymbol{D}_s(n)\,\boldsymbol{D}_x(n)\boldsymbol{x}^{\top}(n)\,\boldsymbol{v}_{\mathrm{E}}(n)\}\\ &\approx \boldsymbol{D}_s(n)\left(\boldsymbol{Q}_{\mathrm{E}}(n)\circ\boldsymbol{K}_{\mathrm{E}}(n)\right)\boldsymbol{D}_s(n).\end{aligned} \tag{2.129}$$

The last expectation conveys the non-stationarity effects and is given by

$$\boldsymbol{P}_{9_{\mathrm{E}}}(n) = E\left\{(\boldsymbol{\xi}^{\top}(n)\boldsymbol{x}(n))^2\left(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\right)^{(\gamma)}\left(\boldsymbol{v}_{\mathrm{E}}(n)+\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\right)^{(\gamma)\top}\right\} \tag{2.130}$$

Using the first order approximation (2.120) and simple manipulations yields

$$\begin{aligned}\boldsymbol{P}_{9_{\mathrm{E}}}(n) = \mathrm{trace}\{\boldsymbol{\Xi}\,\boldsymbol{R}_x\}\Big\{&\boldsymbol{r}(n)\boldsymbol{r}^{\top}(n) + \boldsymbol{D}_s(n)\,\boldsymbol{K}_{\mathrm{E}}(n)\boldsymbol{D}_s(n)\\ &+\boldsymbol{r}(n)\,E\big\{\boldsymbol{v}_{\mathrm{E}}^{\top}(n)\big\}\boldsymbol{D}_s(n) + \boldsymbol{D}_s(n)\,E\left\{\boldsymbol{v}_{\mathrm{E}}(n)\right\}\,\boldsymbol{r}^{\top}(n)\Big\}\end{aligned} \tag{2.131}$$

The last term $\boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{E}}}(n)$ is obtained in the same form of (2.117)

$$\begin{aligned}\boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{E}}}(n) =& -\boldsymbol{\Delta}_{\mathrm{E}}(n)\boldsymbol{\Delta}_{\mathrm{E}}^{\top}(n) - \boldsymbol{\Delta}_{\mathrm{E}}(n)E\left\{\boldsymbol{v}_{\mathrm{E}}(n+1)+\boldsymbol{\Delta}_{\mathrm{E}}(n)\right\}^{\top}\\ &- E\left\{\boldsymbol{v}_{\mathrm{E}}(n+1)+\boldsymbol{\Delta}_{\mathrm{E}}(n)\right\}\boldsymbol{\Delta}_{\mathrm{E}}^{\top}(n)\end{aligned} \tag{2.132}$$

2.7.3 Sign-Sign NNLMS algorithm

Using the weight error vector definition $\boldsymbol{v}_{\mathrm{S}}(n) = \boldsymbol{\alpha}_{\mathrm{S}}(n) - \boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)$ in (2.73) and $\mathrm{sgn}\{e(n)\}\mathrm{sgn}\{\boldsymbol{D}_x(n)\} = \mathrm{sgn}\{\boldsymbol{D}_x(n)e(n)\}$ yields

$$\boldsymbol{v}_{\mathrm{S}}(n+1) = \boldsymbol{v}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{v}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n) - \boldsymbol{\Delta}_{\mathrm{S}}(n) \tag{2.133}$$

Post-multiplying (2.133) by its transpose, taking the expected value and rearranging the terms leads to

$$\begin{aligned}\boldsymbol{K}_{\mathrm{S}}(n+1) =& \boldsymbol{K}_{\mathrm{S}}(n) + \eta\left(\boldsymbol{P}_{1_{\mathrm{S}}}(n) + \boldsymbol{P}_{1_{\mathrm{S}}}^{\top}(n)\right) + \eta\left(\boldsymbol{P}_{2_{\mathrm{S}}}(n) + \boldsymbol{P}_{2_{\mathrm{S}}}^{\top}(n)\right)\\ &+\eta^2\left(\boldsymbol{P}_{3_{\mathrm{S}}}(n) + \boldsymbol{P}_{4_{\mathrm{S}}}(n) + \boldsymbol{P}_{4_{\mathrm{S}}}^{\top}(n) + \boldsymbol{P}_{5_{\mathrm{S}}}(n)\right) + \boldsymbol{K}_{\boldsymbol{\Delta}_{\mathrm{S}}}(n)\end{aligned} \tag{2.134}$$

where

$$\boldsymbol{P}_{1_{\mathrm{S}}}(n) = E\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \tag{2.135}$$

$$\boldsymbol{P}_{2_{\mathrm{S}}}(n) = E\{\boldsymbol{v}_{\mathrm{S}}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \tag{2.136}$$

$$\boldsymbol{P}_{3_{\mathrm{S}}}(n) = E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \tag{2.137}$$

$$\boldsymbol{P}_{4_{\mathrm{S}}}(n) = E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \tag{2.138}$$

$$\boldsymbol{P}_{5_{\mathrm{S}}}(n) = E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\,\boldsymbol{v}(n)\,\boldsymbol{v}_{\mathrm{S}}^{\top}(n)\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)\}\} \tag{2.139}$$

These expected values are calculated in the following for $\boldsymbol{x}(n)$ Gaussian.

Expected value $\boldsymbol{P}_{1_S}(n)$: Using the properties of statistical expectation $\boldsymbol{P}_{1_S}(n)$ can be written as

$$\boldsymbol{P}_{1_S}(n) = E_v\Big\{\boldsymbol{v}_S(n)\,\boldsymbol{\alpha}_o^{*\top}(n)\,E\big\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)|\boldsymbol{v}_S(n),\boldsymbol{\xi}(n)\}\big\}\Big\}. \tag{2.140}$$

The conditional expectation in (2.140) is given by (2.94), which must be approximated. Approximation (2.95) for the i-th element of (2.94) is too simple to predict the weight error fluctuations. A more suitable approximation is given by a first order Taylor series expansion :

$$\begin{aligned}
&\frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\,\boldsymbol{v}_S(n)}{\sigma_x\sigma_{e|\boldsymbol{v}_S(n),\boldsymbol{\xi}(n)}}\right)\\
&\approx \frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}}\right)\\
&\qquad -\frac{2}{\pi}\,\frac{\boldsymbol{R}_i^\top}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}\sqrt{1-\left(\dfrac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\})}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}}\right)^2}}\,(\boldsymbol{v}_S(n)-E\{\boldsymbol{v}_S(n)\})\\
&= q_i(n) + \boldsymbol{s}_i^\top(n)\boldsymbol{v}_S(n).
\end{aligned} \tag{2.141}$$

where the scalar $q_i(n)$ and the vector $\boldsymbol{s}_i(n)$ are deterministic variables defined respectively as

$$q_i(n) = \frac{2}{\pi}\sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}}\right)+\frac{2}{\pi}\,\frac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\}}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}})\sqrt{1-\left(\dfrac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\})}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}}\right)^2}} \tag{2.142}$$

$$\boldsymbol{s}_i(n) = -\frac{2}{\pi}\,\frac{\boldsymbol{R}_i}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}\sqrt{1-\left(\dfrac{\boldsymbol{R}_i^\top\,E\{\boldsymbol{v}_S(n)\})}{\sigma_x\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}}\right)^2}} \tag{2.143}$$

with $\sigma_{e|E\{\boldsymbol{v}_S(n)\},\boldsymbol{\Xi}}$ defined in (2.96). Using (2.141) in (2.140) and defining the $N\times 1$ vector $\boldsymbol{q}(n)$ with i-th element $q_i(n)$ and the $N\times N$ matrix $\boldsymbol{S}(n)$ with i-th column $\boldsymbol{s}_i(n)$, (2.140) becomes, after simple manipulations,

$$\boldsymbol{P}_{1_S}(n) \approx E\{\boldsymbol{v}_S(n)\}\,\boldsymbol{\alpha}^{*\top}\ \mathrm{Diag}\{\boldsymbol{q}(n)\} + \boldsymbol{K}_S(n)\,\boldsymbol{S}(n)\,\boldsymbol{D}_{\alpha^*} \tag{2.144}$$

Expected value $\boldsymbol{P}_{2_S}(n)$: Similar to $\boldsymbol{P}_{1_S}(n)$, we first express $\boldsymbol{P}_{2_S}(n)$ in the form

$$\boldsymbol{P}_{2_S}(n) = E_v\{\boldsymbol{v}_S(n)\,\boldsymbol{v}_S^\top(n)\,E\{\mathrm{sgn}\{\boldsymbol{D}_x(n)\,e(n)|\boldsymbol{v}_S(n)\}\}\} \tag{2.145}$$

Then, using (2.94) and (2.141) we obtain

$$\boldsymbol{P}_{2_S}(n) \approx \boldsymbol{K}_S(n)\ \mathrm{Diag}\{\boldsymbol{q}(n)\} + E\{\boldsymbol{v}_S(n)\,\boldsymbol{v}_S^\top(n)\ \mathrm{Diag}\{\boldsymbol{S}^\top(n)\boldsymbol{v}_S(n)\}\} \tag{2.146}$$

The (i,j)-th element of the expectation in (2.146) is given by

$$E\{[\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\,\mathrm{Diag}\{\boldsymbol{S}^\top(n)\boldsymbol{v}(n)\}]_{ij}\} = \sum_{k=1}^{N} S_{kj}\,E\{v_k(n)v_i(n)v_j(n)\}. \tag{2.147}$$

Evaluation of the third order moment in (2.147) requires further approximation, as the distribution of $\boldsymbol{v}_{\mathrm{S}}(n)$ is unknown. We assume that the distribution of $\boldsymbol{v}(n)$ can be approximated by a Gaussian distribution about its mean value. Then, using the properties of Gaussian variables [Miller 1964] and defining the centered variable $\bar{v}_p(n) = v_p(n) - E\{v_p(n)\}$ for $p = i, j, k$, we have

$$\begin{aligned} E\{v_i(n)v_j(n)v_k(n)\} &\approx E\{\bar{v}_i(n)\bar{v}_j(n)\}\,E\{v_k(n)\} + E\{\bar{v}_i(n)\bar{v}_k(n)\}\,E\{v_j(n)\} \\ &\quad + E\{\bar{v}_j(n)\bar{v}_k(n)\}\,E\{v_i(n)\} + E\{v_i(n)\}E\{v_j(n)\}E\{v_k(n)\} \\ &= K_{ij}(n)\,E\{v_k(n)\} + K_{ik}(n)\,E\{v_j(n)\} + K_{jk}(n)\,E\{v_i(n)\} \\ &\quad - 2\,E\{v_i(n)\}E\{v_j(n)\}E\{v_k(n)\} \end{aligned} \tag{2.148}$$

which completes the derivation of (2.146).

Expected value $\boldsymbol{P}_{3_{\mathrm{S}}}$: The (i,j)-th entry of matrix $\boldsymbol{P}_{3_{\mathrm{S}}}(n)$ is given by

$$\begin{aligned} [\boldsymbol{P}_{3_{\mathrm{S}}}(n)]_{ij} &= E\{\mathrm{sgn}\{x_i(n)\,e(n)\}\,[\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}]_{ij}\,\mathrm{sgn}\{x_j(n)\,e(n)\}^\top\} \\ &= E\{[\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}]_{ij}\,\mathrm{sgn}\{x_i(n)\,x_j(n)\}\} \end{aligned} \tag{2.149}$$

Using (2.93), $E\{\mathrm{sgn}\{x_i(n)\,x_j(n)\}\} = (2/\pi)\sin^{-1}([\boldsymbol{R}_x]_{ij}/\sigma_x^2)$ and

$$[\boldsymbol{P}_{3_{\mathrm{S}}}(n)]_{ij} = [\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}]_{ij}\,\frac{2}{\pi}\sin^{-1}\left(\frac{[\boldsymbol{R}_x]_{ij}}{\sigma_x^2}\right) \tag{2.150}$$

Finally, expressing the result in the matrix form yields

$$\boldsymbol{P}_{3_{\mathrm{S}}}(n) = \frac{2}{\pi}\,(\boldsymbol{\alpha}^*\,\boldsymbol{\alpha}^{*\top}) \circ \boldsymbol{T} \tag{2.151}$$

where the (i,j)-th elements of the $N \times N$ matrix $\boldsymbol{T}$ is given by

$$[\boldsymbol{T}]_{ij} = \sin^{-1}([\boldsymbol{R}_x]_{ij}/\sigma_x^2)$$

Expected values $\boldsymbol{P}_{4_{\mathrm{S}}}(n)$ and $\boldsymbol{P}_{5_{\mathrm{S}}}(n)$: Using the same reasoning and approximations used to evaluate $\boldsymbol{P}_{1_{\mathrm{S}}}(n)$ to $\boldsymbol{P}_{3_{\mathrm{S}}}(n)$ yields

$$\boldsymbol{P}_{4_{\mathrm{S}}}(n) = \frac{2}{\pi}\,(\boldsymbol{\alpha}^*\,E\{\boldsymbol{v}_{\mathrm{S}}^\top(n)\}) \circ \boldsymbol{T} \tag{2.152}$$

$$\boldsymbol{P}_{5_{\mathrm{S}}}(n) = \frac{2}{\pi}\,(\boldsymbol{K}_{\mathrm{S}}(n)) \circ \boldsymbol{T} \tag{2.153}$$

which concludes the determination of the recursive model (2.134) for $\boldsymbol{K}_{\mathrm{S}}(n)$.

2.8 Simulation results and discussion

We now present simulation examples to illustrate the properties of the three algorithms and the accuracy of the derived models. The parameters for these examples

were chosen to illustrate several properties of the three algorithms while conserving space. Similar results have been obtained using a variety of parameter sets. For all examples, $N = 31$. The unknown stationary system is defined as

$$\alpha_{o_i}^{*\,(\text{stat.})} = \begin{cases} 0.9 - 0.05\,i, & i = 0, \ldots, 18 \\ -0.01\,(i - 18) & i = 19, \ldots, 31 \end{cases} \tag{2.154}$$

For the non-stationary case, we consider an unknown response defined by

$$\alpha_{o_i}^{*\,(\text{nonstat.})}(n) = \boldsymbol{\alpha}_{o_i}^{*\,(\text{stat.})} + \frac{|\boldsymbol{\alpha}_{o_i}^{*\,(\text{stat.})}|}{10} \sin\left(\frac{2\,\pi}{T}\,n + 2\,\pi\frac{i-1}{N}\right) + \xi_i(n) \tag{2.155}$$

where the period T of the deterministic sinusoidal component was set to 2500. $\boldsymbol{\xi}(n)$ is a zero-mean Gaussian random vector with correlation matrix $\sigma_\xi^2 \boldsymbol{I}$ with $\sigma_\xi^2 = 5{\times}10^{-4}$. The input signal is a first-order AR process given by $x(n) = 0.5\,x(n-1) + w(n)$, with $w(n)$ i.i.d. zero-mean Gaussian with variance σ_w^2, adjusted to obtain the desired input power $\sigma_x^2 = 1$. The noise $z(n)$ is zero-mean i.i.d. Gaussian with variance $\sigma_z^2 = 10^{-2}$. The adaptive weights in $\boldsymbol{\alpha}_i(0)$ were all initialized at $10/N$ for all realizations. The step size was always set to $\eta = 0.005$ for all but the normalized variant. For the latter we used $\eta = 0.005\,N\sigma_x^2$, which leads to an equivalent step size $\tilde{\eta} = 0.005$. Monte Carlo simulations were obtained by averaging 100 runs.

2.8.1 Mean behavior

2.8.1.1 Example 1

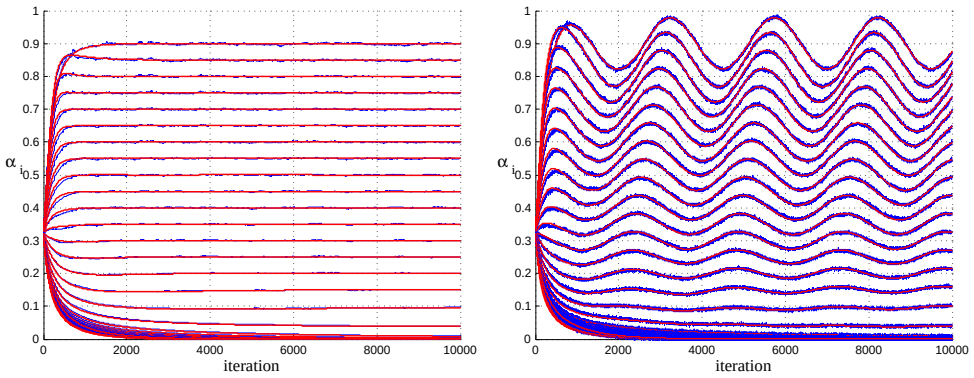
Figs. 2.7 and 2.8 show the results for the Normalized NNLMS. The parameter ε was set to 0. Blue curves show simulation results and red curves show the theoretical predictions from (2.86). Figure 2.7 is for $\sigma_x^2 = 1$ and Figure 2.8 is for $\sigma_x^2 = 0.5$. It can be verified that the model (2.86) accurately predicts the algorithm behavior, and that normalization has made the algorithm performance basically independent of the input power.

2.8.1.2 Example 2

Figure 2.9 illustrates the results for the Exponential NNLMS algorithm. The parameter $(p, q) = (5, 7)$ was used. Compared with Figure 2.7, these figures clearly show that the coefficients that tend to zero in steady-state had their convergence rate significantly improved by the Exponential NNLMS algorithm. Also, the accuracy of the theoretical model (2.90) can be verified.

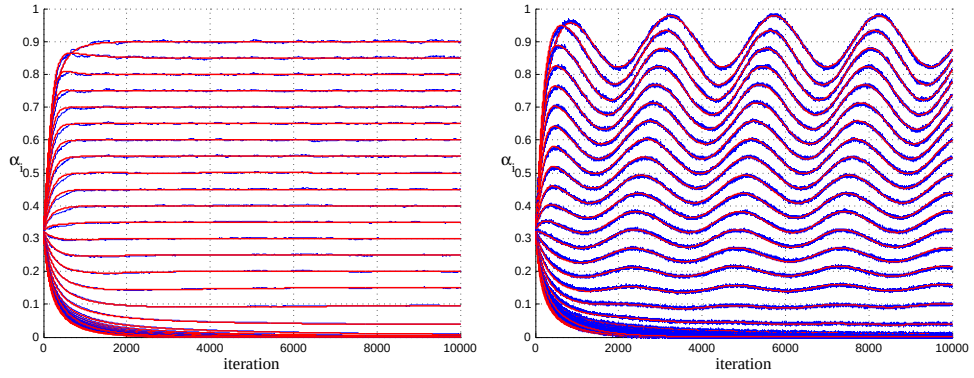
2.8.1.3 Example 3

Figure 2.10 illustrate the result of the Sign-Sign NNLMS under stationary and nonstationary environment. These figures illustrate the accuracy of the model (2.97). It is also clear that the Sign-Sign NNLMS coefficients converge much slower than those for the NNLMS algorithm as expected.



(a) stationary case, $\sigma_x^2 = 1$ (b) nonstationary case, $\sigma_x^2 = 1$

FIGURE 2.7 – Evolution of the coefficients $\alpha_i(n)$ for the normalized NNLMS algorithm in stationary and nonstationary environments for $\sigma_x^2 = 1$. Theoretical curves are from (2.86).



(a) stationary case, $\sigma_x^2 = 0.5$ (b) nonstationary case, $\sigma_x^2 = 0.5$

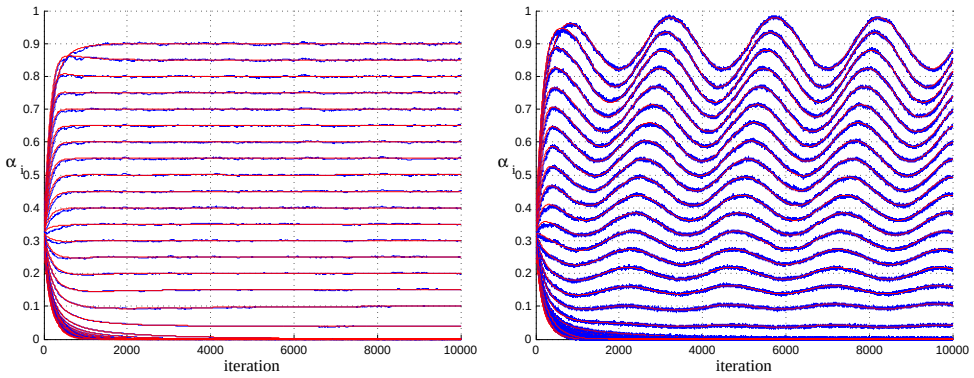
FIGURE 2.8 – Evolution of the coefficients $\alpha_i(n)$ for the normalized NNLMS algorithm in stationary and nonstationary environments for $\sigma_x^2 = 0.5$. Theoretical curves are from (2.86).

2.8.2 Second moment behavior

We now illustrate the EMSE behavior of the NNLMS variants. Again, blue curves were obtained from Monte Carlo simulation and red curves show the theoretical predictions.

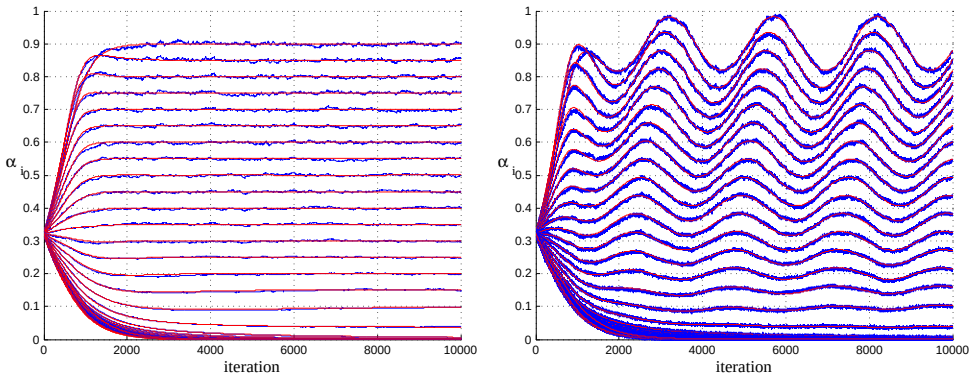
2.8.2.1 Example 4

Figure 2.11 illustrates the EMSE behavior of the Normalized NNLMS algorithm. The accuracy of model (2.100) can be easily verified. Two more curves are added to



(a) stationary case (b) nonstationary case

FIGURE 2.9 – Evolution of the coefficients $\alpha_i(n)$ for the Exponential NNLMS algorithm in stationary and nonstationary environments. Theoretical curves are from (2.90).

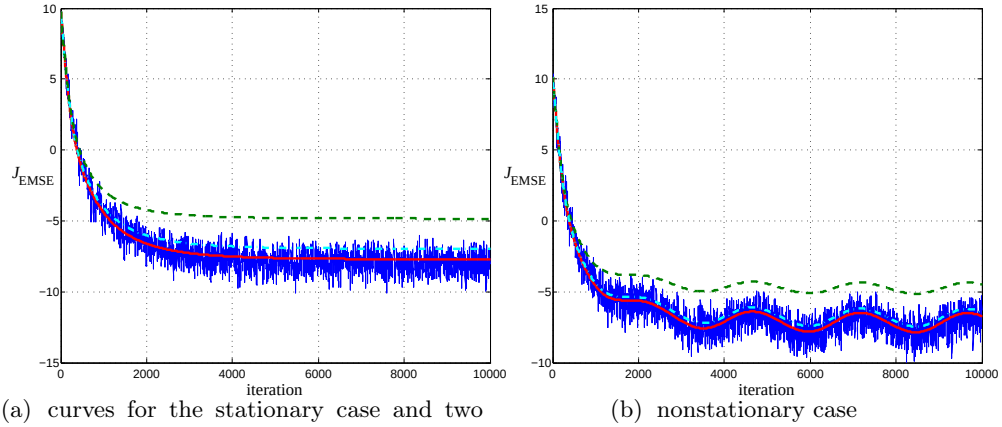


(a) stationary case (b) nonstationary case

FIGURE 2.10 – Evolution of the coefficients $\alpha_i(n)$ for the Sign-Sign NNLMS algorithm in stationary and nonstationary environments. Theoretical curves are from (2.97).

each plot to illustrate the effect of the random non-stationarity parameter σ_ξ^2.

Random perturbations with different variances were also added to $\alpha_{o_i}^{*(\text{stat.})}$ (Figure 2.11(a)) and to the nonstationary case (Figure 2.11(b)). The light blue (dash-dot) lines show the theoretical EMSE for $\sigma_\xi^2 = 10^{-3}$, while the green (dash) lines show the theoretical EMSE for $\sigma_\xi^2 = 5 \times 10^{-3}$. These curves illustrate the expected extra EMSE due to tracking of the random optimal solution variations. Simulation curves coincide with the theoretical ones, but are not shown to preserve the visibility of the other curves.



(a) curves for the stationary case and two theoretical curves with added $\boldsymbol{\xi}(n)$

(b) nonstationary case

FIGURE 2.11 – Evolution of the EMSE for the Normalized NNLMS algorithm in stationary and nonstationary environment. Light blue dash-dot line and green dashed line show the theoretical results for $\sigma_\xi^2 = 10^{-3}$ and 5×10^{-3}, respectively.

2.8.2.2 Example 5

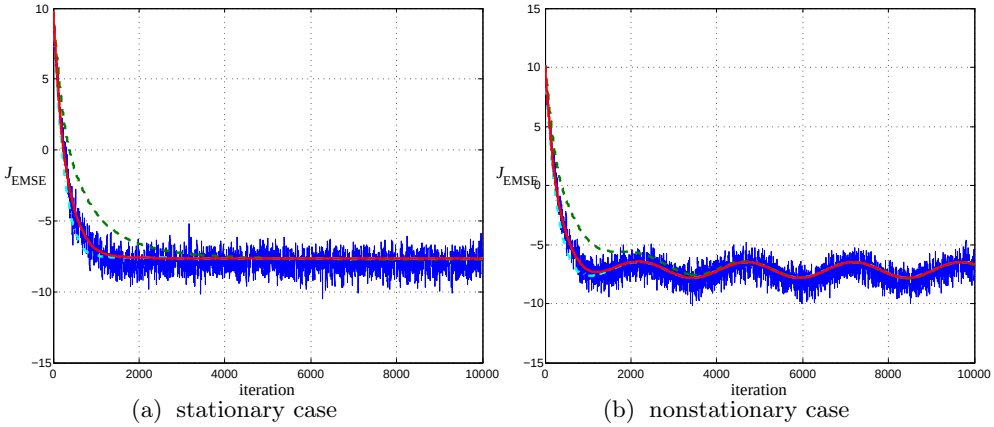
Figure 2.12 illustrates the EMSE behavior of the Exponential NNLMS algorithm. The blue and red curves show again the simulation results and the accurate theoretical predictions using (2.121) for $(p, q) = (5, 7)$. The light blue (dash-dot) and the green (dash) curves show the theoretical predictions of the EMSE behavior for $(p, q) = (3, 5)$ and $(p, q) = (1, 1)$ (original NNLMS), respectively. The simulation results agree with these curves but are not shown for clarity. These results confirm that the Exponential NNLMS algorithm accelerates the convergence of the adaptive weights when compared to NNLMS.

2.8.2.3 Example 6

Figure 2.13 illustrates EMSE behavior of the Sign-Sign NNLMS algorithm. Once more, the red curves and blue curves illustrate the accuracy of the model (2.134). The green (dashed) curves show the performance of the original NNLMS in the same conditions. These curves illustrate the slower convergence rate of Sign-Sign NNLMS when compared to NNLMS, the price paid for a reduced computational complexity.

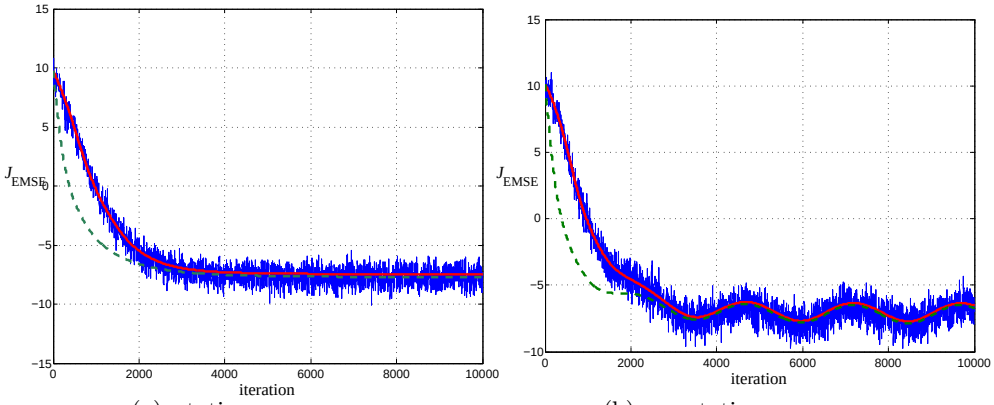
2.8.3 A comparative example

This example compares the performance of the NNLMS algorithm and its variants with that of unconstrained algorithms in solving the unconstrained solution problem of identifying an unknown weight vector $\boldsymbol{\alpha}^*$ with non-negative coefficients. This is an interesting application, as in this case the unconstrained algorithm will converge in the mean to the optimal solution. Though the problem description may



(a) stationary case (b) nonstationary case

FIGURE 2.12 – Evolution of the EMSE for the Exponential NNLMS algorithm in stationary and nonstationary environment. Light blue dash-dot line and green dashed line show the theoretical results for $(p, q) = (3, 5)$ and $(p, q) = (1, 1)$ (original NNLMS), respectively.



(a) stationary case (b) nonstationary case

FIGURE 2.13 – Evolution of the EMSE for the Sign-Sign NNLMS algorithm in stationary and nonstationary environment. Theoretical evolution of original NNLMS is represented by the green dashed line.

guarantee that the optimal weights are positive, often in practice one do not have accurate information about the number of coefficients in the optimal solution. A common approach is to set the adaptive filter with a sufficient number of coefficients, usually larger than the actual unknown number. This examples illustrates the performance of the different algorithms in this case.

Consider a non-negative unknown optimal solution

$$\boldsymbol{\alpha}_i^* = \exp(-0.6\, i) \tag{2.156}$$

with $i = 1, \ldots, 10$ and adaptive filters with $N = 30$ coefficients. Five algorithms

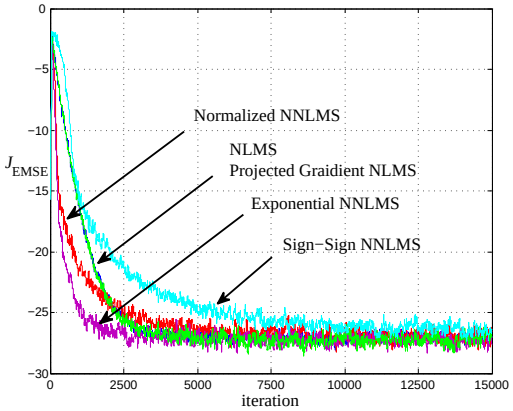
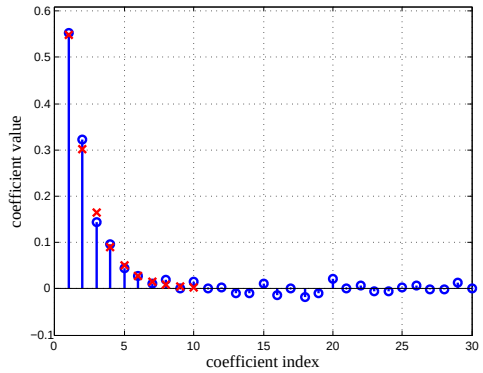
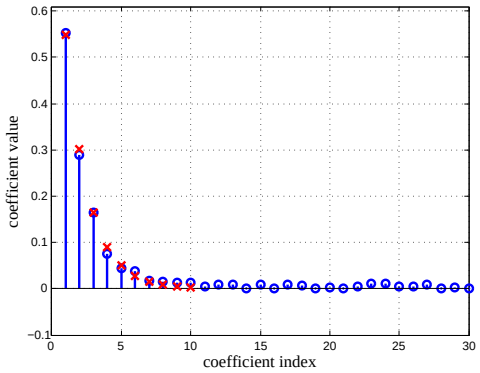


FIGURE 2.14 – EMSE (db) for the four algorithms compared.

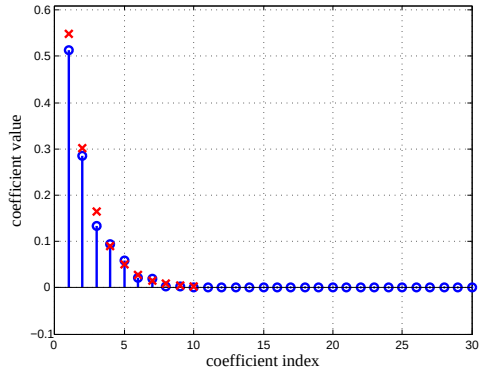
were tested : NLMS [Sayed 2008], Projected Gradient NLMS [Theodoridis 2011], Normalized NNLMS, Exponential NNLMS and Sign-Sign NNLMS. In Projected Gradient NLMS, the coefficients which activate the non-negativity constraints are projected into the feasible region, i.e. set to 0, at each iteration. The input signal was given by $x(n) = 0.5\,x(n-1) + w(n)$ with $\sigma_w^2 = 3/4$ so that $\sigma_x^2 = 1$. The initial weights $\boldsymbol{\alpha}(0)$ were drawn from the uniform distribution $\mathcal{U}(0,1)$. The additive noise $z(n)$ was i.i.d. Gaussian with $\sigma_z^2 = 0.1$. The step sizes were chosen for each algorithm by experimentation so that all would reach approximately the same steady-state EMSE with the value of 2×10^{-3}. The step sizes were $\eta = 0.035$ for both NLMS and Projected Gradient NLMS, $\eta = 0.8750$ for Normalized NNLMS, $\eta = 0.022$ for Exponential NNLMS and $\eta = 0.007$ for Sign-Sign NNLMS. Figure 2.14 shows the EMSE evolution for the five algorithms (Monte Carlo simulation averaged over 100 realizations). Figure 2.15 shows the estimated weights for a single realization of the input signal at $n = 15000$. Although the unconstrained NLMS algorithm is able to converge to the optimal solution in the mean sense, it does not provide a good estimation of the zero-valued coefficients in a single realization. NNLMS-type algorithms, including the Sign-Sign algorithm (which has not even converged to the steady-state at $n = 15000$) do a better job in determining the support of the actual response.



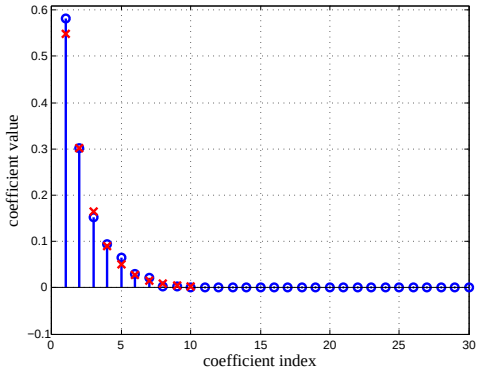
(a) NLMS algorithm.



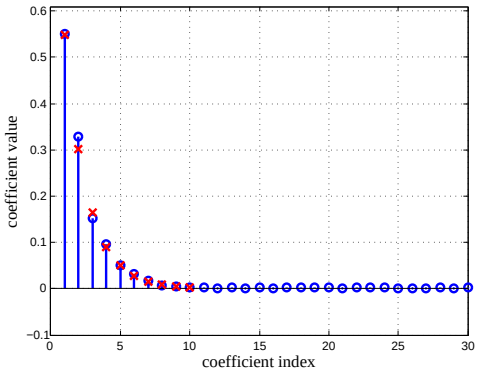
(b) Projected NLMS algorithm.



(c) Normalized NNLMS algorithm.



(d) Exponential NNLMS algorithm.



(e) Sign-Sign NNLMS algorithm.

FIGURE 2.15 – Weights estimated by NLMS, Projected NLMS, Normalized NNLMS, and Exponential NNLMS at $n = 15000$ for a single realization. Real weights are marked by $\times$. The NNLMS variants determine clearly the support of the response.

Chapter 3

Adaptive system identification with ℓ_1-type constraints

Contents

3.1 Introduction

Information processing with ℓ_1-norm constraint has been a topic of high interest during the last five years. Many recent methods in statistical learning and related fields can be formulated as convex optimization problems subject to ℓ_1-type constraint on the set of parameters to estimate. One motivation for using ℓ_1-constraint is that it encourages sparse solutions [Candès 2008].

This chapter is a continuation and extension to our work in the previous chapter. We shall investigate an extension of our proposed NNLMS to address ℓ_1-type constraints/regularization. Firstly, we shall briefly discuss a direct extension of NNLMS to address the identification problem under non-negativity and constant ℓ_1-norm constraints, i.e., the constraint set $\Theta_{\ell_1}^{+}$ of (1.6). After that we shall discuss in detail an online system identification algorithm with ℓ_1-constraint to promote sparsity, by extending the proposed NNLMS algorithm.

3.2 NNLMS algorithm with constant ℓ_1-norm constraint

In this part we discuss the online identification problem under the non-negativity constraint combined with constant ℓ_1-norm constraint. Note that in this case constant ℓ_1-norm constraint is trivial as ℓ_1-norm becomes the sum of the elements. However we keep the nomination constant ℓ_1-norm constraint because a general case can be converted to this simple one by splitting the coefficient vector as (1.8). This constraint plays an important role in several problems. Especially when the unit constant is considered, which means that each component of the parameter vector represents a proportion of the associated mode. For instance, in the multi-kernel learning problem, combination of candidate kernels is used instead of a single kernel to improve the classification/regression accuracy [Gönen 2011]. In order to guarantee the non-negativity of the constructed kernel and avoid trivial solutions, the weight coefficients to be estimated must satisfy these constraints. Another example is the hyperspectral data unmixing problem, where the fractional abundances associated to different materials are to be determined. To be physically interpretable, the driving abundances are often required to satisfy two constraints : all abundances must be nonnegative, and their sum must be equal to one [Keshava 2002].

In this section, we examine the task of minimizing a convex function subject to non-negativity constraint and constant ℓ_1-norm constraint on the solution in the context of system identification. Popular ways to deal with such constraints are projected-gradient type methods or regularization methods. The former project the results on the feasible region at each iteration. And the latter minimize the objective function with additional penalty term that is related to constraints. In [Duchi 2008], the authors studied the projections onto both the positive orthant of ℓ_1-ball and the whole ℓ_1-ball. In [Rao 2005], the authors studied a regularization method based on augmented Lagrangian method. These methods require additional calculations which makes the algorithm more complex and sometimes reduces the accuracy of the result.

We shall propose a method which differs from projected-gradient methods or regularization methods in that it integrates the constraints into the coefficient update process. No extra process such as projection is required. In order to deal with non-stationary systems and to reduce computational requirements, we also develop a stochastic gradient algorithm that updates the parameters in an online way.

3.2.1 Formulation of problem

Consider the model described in Figure I.1. The system identification problem under non-negativity and constant ℓ_1-norm constraints can be formalized as

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \tag{3.1}$$

$$\text{subject to} \quad \boldsymbol{\alpha} \succeq \mathbf{0} \tag{3.2}$$

$$\|\boldsymbol{\alpha}\|_1 = \varepsilon_0 \tag{3.3}$$

with $J(\alpha)$ a convex cost function, $\|\cdot\|_1$ the ℓ_1-norm, and ε_0 a given constant. In this case, ℓ_1-norm constraint becomes the trivial constant-sum constraint $\sum_i^N \alpha_i = \varepsilon_0$.

3.2.2 Weight update algorithm

If non-negativity of the parameters is guaranteed at each iteration, we can make the following variable change to ensure that the constant-sum constraint (3.3) is satisfied

$$\alpha_j = \frac{w_j}{\sum_{\ell=1}^N w_\ell}\varepsilon_0 \tag{3.4}$$

With $w_i \geq 0$, the problem becomes unconstrained with respect to constraint (3.3). The partial derivative of the cost function J with respect to the new variables w_i can be expressed as follows

$$\frac{\partial J}{\partial w_i} = \sum_{j=1}^N \frac{\partial J}{\partial \alpha_j} \times \frac{\partial \alpha_j}{\partial w_i} \tag{3.5}$$

where

$$\begin{aligned}\frac{\partial \alpha_j}{\partial w_i} &= \frac{\frac{\partial w_j}{\partial w_i}\sum_{\ell=1}^N w_\ell - \frac{\partial \sum_{\ell=1}^N w_\ell}{\partial w_i} w_j}{\left(\sum_{\ell=1}^N w_\ell\right)^2}\varepsilon_0 \\ &= \frac{\varepsilon_0 \delta_{ij} - \alpha_j}{\sum_{\ell=1}^N w_\ell}\end{aligned} \tag{3.6}$$

The Kronecker symbol δ_{ij} results from the derivative $\partial w_j / \partial w_i$. Replacing (3.6) into (3.5), the negative partial derivative of J with respect to w_i can now be written as

$$-\frac{\partial J}{\partial w_i} = \frac{1}{\sum_{\ell=1}^N w_\ell}\big(-\varepsilon_0[\nabla_{\boldsymbol{\alpha}} J]_i + \sum_{j=1}^N \alpha_j \nabla_{\boldsymbol{\alpha}} J]_j\big) \tag{3.7}$$

Let us now use the same rule as (2.5) for updating the non-negative entries $w_i(k)$. The component-wise update equation is given by

$$w_i(k+1) = w_i(k) + \eta \frac{w_i(k)}{\sum_{\ell=1}^M w_\ell(k)}\big(-\varepsilon_0[\nabla_{\boldsymbol{\alpha}} J]_i + \sum_{j=1}^N \alpha_j \nabla_{\boldsymbol{\alpha}} J]_j\big) \tag{3.8}$$

It can be easily found that $\sum_{i=1}^N w_i(k+1) = \sum_{i=1}^N w_i(k)$ for all η. The factor $(\sum_{\ell=1}^M w_\ell(k))^{-1}$ is thus constant and can be absorbed into η. This yields

$$w_i(k+1) = w_i(k) + \eta\, w_i(k)\big(-\varepsilon_0[\nabla_{\boldsymbol{\alpha}} J]_i + \sum_{j=1}^N \alpha_j \nabla_{\boldsymbol{\alpha}} J]_j\big) \tag{3.9}$$

Dividing by $\sum_{i=1}^N w_i(k+1)$ and $\sum_{i=1}^N w_i(k)$ the left and right sides of (3.9), respectively, and considering the variable change defined by (3.4), we obtain

$$\alpha_i(k+1) = \alpha_i(k) + \eta\, \alpha_i(k)\big(-\varepsilon_0[\nabla_{\boldsymbol{\alpha}} J]_i + \sum_{j=1}^N \alpha_j \nabla_{\boldsymbol{\alpha}} J]_j\big) \tag{3.10}$$

We can verify that $\sum_{i=1}^{N} \alpha_i(k+1) = \sum_{i=1}^{N} \alpha_i(k)$, which means that the algorithm satisfies the constant-sum constraint as long as the weight vector is initialized by any vector $\boldsymbol{\alpha}(0)$ such that $\sum_{i=1}^{N} \alpha_i(0) = \varepsilon_0$.

3.2.3 Application to the mean square error cost function

A general update rule with respect to $J(\boldsymbol{\alpha})$ was derived in the previous section. We shall now consider the usual situation where the mean square error is considered, namely

$$J(\boldsymbol{\alpha}) = E\{[\boldsymbol{\alpha}^\top \boldsymbol{x}(n) - y(n)]^2\}$$

In this case the update rule (3.10) simply writes

$$\begin{aligned} \alpha_i(k+1) =& \alpha_i(k) + \eta\alpha_i(k)\big(\varepsilon_0(\boldsymbol{r}_{xy} - \boldsymbol{R}_x\boldsymbol{\alpha}(k))_i \\ & - \sum_{j=1}^{N} \alpha_j(k)\left(\boldsymbol{r}_{xy} - \boldsymbol{R}_x\boldsymbol{\alpha}(k)\right)_j\big) \end{aligned} \tag{3.11}$$

where $\boldsymbol{R}_x$ is the correlation matrix of input data $\boldsymbol{x}(n)$ and $\boldsymbol{r}_{xy}$ is the intercorrelation vector between $\boldsymbol{x}(n)$ and $y(n)$. It can be noticed that this update rule requires the second-order moments of the input signal, which is unknown in most applications. In order to enable online computation, we shall now present a LMS-algorithm based on stochastic gradient

$$-\nabla J(\boldsymbol{\alpha}) \approx \boldsymbol{x}(n)e(n)$$

with $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n)$. Substituting this expression into equation (3.10) yields the following update rule

$$\alpha_i(n+1) = \alpha_i(n) + \eta\alpha_i(n)\big(\varepsilon_0\boldsymbol{x}_i(n)e(n) - \sum_{j=1}^{N} \alpha_j(n)\left(\boldsymbol{x}_j(n)e(n)\right)\big)$$

which can be written in vectorial form as

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta\boldsymbol{D}_{\boldsymbol{\alpha}}(n)\big(\varepsilon_0\boldsymbol{x}(n)e(n) - \mathbf{1}\,\boldsymbol{\alpha}^\top(n)x(n)e(n)\big) \tag{3.12}$$

where $\mathbf{1}$ is the all-one vector, and $\boldsymbol{D}_{\boldsymbol{\alpha}}(n)$ is the diagonal matrix with $\boldsymbol{\alpha}(n)$ as diagonal elements.

3.2.4 Examples

3.2.4.1 Algorithm behavior illustration

To illustrate the behavior of the proposed algorithm, the impulse response of the system of reference $\boldsymbol{\alpha}^*$, i.e., the solution of the unconstrained problem (3.1), was set to

$$\boldsymbol{\alpha}^* = [0.4 \;\; -0.1 \;\; 0.3 \;\; 0.05 \;\; 0 \;\; 0.01 \;\; 0.2 \;\; -0.01 \;\; 0.03 \;\; 0.1]^\top$$

Note that $\boldsymbol{\alpha}^*$ has two negative entries used to activate the non-negativity constraints, and $\|\boldsymbol{\alpha}^*\|_1 = 1.2$. In all the experiments, the same initial vector $\boldsymbol{\alpha}(0) > 0$ was used,

normalized so that the sum of its entries is equal to 1. In the first experiment, the inputs $x(n)$ and the noise $z(n)$ were chosen i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_z^2 = 0.01$. The non-negativity of the weight coefficients was imposed, and their sum to be equal to 1. The update rule (3.12) was tested with the step-size values $\eta = 0.05$ and $\eta = 0.15$. The results were averaged over 100 Monte-Carlo runs. Figure 3.1, first column, shows the learning curves and the mean-weight behavior of the algorithm [1]. One can observe that the algorithm exhibits the very usual behavior of LMS-style approaches, for which small step-size usually means slower convergence but better asymptotic performance. One can easily check that the non-negativity and the full-additivity constraints are satisfied.

In the second experiment, our algorithm was tested with correlated inputs $x(n)$. We considered an AR model given by $x(n) = ax(n-1) + w(n)$, with $a = 0.5$. The noise $w(n)$ was chosen i.i.d and zero-mean Gaussian with variance $\sigma_w^2 = 1 - 0.5^2$, so that $\sigma_x^2 = 1$ as in the first example. The other parameters remained unchanged. As shown in Figure 3.1, second column, the same conclusions as above can be repeated.

3.2.4.2 Comparison with projected gradient method

The performance of the proposed algorithm was also analyzed in the case where $\boldsymbol{\alpha}^*$ is sparse, and compared with the popular projected-gradient scheme. The same experimental setup as above was considered, except that

$$\boldsymbol{\alpha}^* = [0.4 \;\; -0.1 \;\; \boldsymbol{O}_{20} \;\; 0.3 \;\; 0.05 \;\; 0 \;\; \boldsymbol{O}_{30} \;\; 0.01 \;\; 0.2 \;\; -0.01 \;\; \boldsymbol{O}_{40} \;\; 0.03 \;\; 0.1]^\top$$

where $\boldsymbol{O}_M$ is an all-zero vector of length M. Vector $\boldsymbol{\alpha}^*$ was thus containing 100 entries, among which 90 were null. The projected-gradient method described in [Duchi 2008] was considered for comparison, where the problem (3.1)-(3.3) is solved using the update rule

$$\boldsymbol{\alpha}(n+1) = \Pi(\boldsymbol{\alpha}(n) - \eta \nabla J(\boldsymbol{\alpha}(n)))$$

where Π is the Euclidean projection onto the non-negative ℓ_1 ball. Figure 3.2(a) compares the learning curves of these two online methods for uncorrelated inputs. Clearly, the proposed method outperformed the projected-gradient method, both in convergence speed and steady-state error. Figures 3.2(b)-(c) represent the coefficient vector $\boldsymbol{\alpha}(n)$ estimated by the two methods. One can notice that the zeros in $\boldsymbol{\alpha}^*$ were more accurately recovered by our algorithm, because zero is a fixed-point for the update equation (3.12), at considerably lower computational complexity. Similar results were obtained with correlated inputs.

1. It should be noticed that the analytical results for weight behavior, which can be found in [Chen 2011], are also presented in the figures. As this algorithm is not the major objective of this chapter, we do not detail them in the thesis. Interested reader can refer to [Chen 2011]

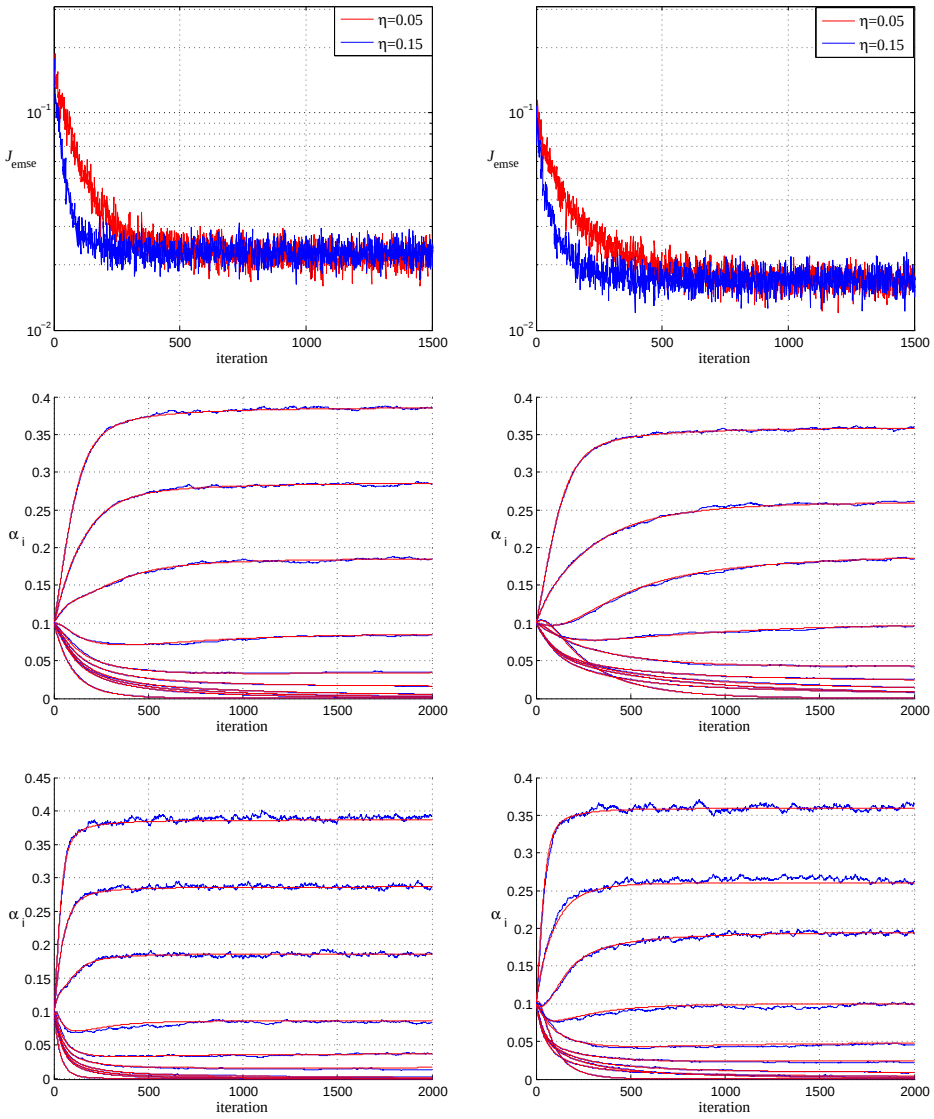


FIGURE 3.1 – First column : algorithm behavior with white input. Second column : algorithm behavior with correlated input. From top to bottom : EMSE convergence curves, weight behavior with $\eta = 0.05$, weight behavior with $\eta = 0.15$. The red curves for weight behavior are analytical results obtained with the model described in [Chen 2011]

3.3 Online system identification with ℓ_1-norm regularization

Several applications have recently shown the need for online sparse identification techniques. Particularly, a driving force behind the development of such algorithms is echo cancellation. Network echo path is sparse in nature. Although the number of coefficients of the impulse is large, only a small portion has significant values (active coefficients). The others are just zero or unnoticeably small (inactive coefficients).

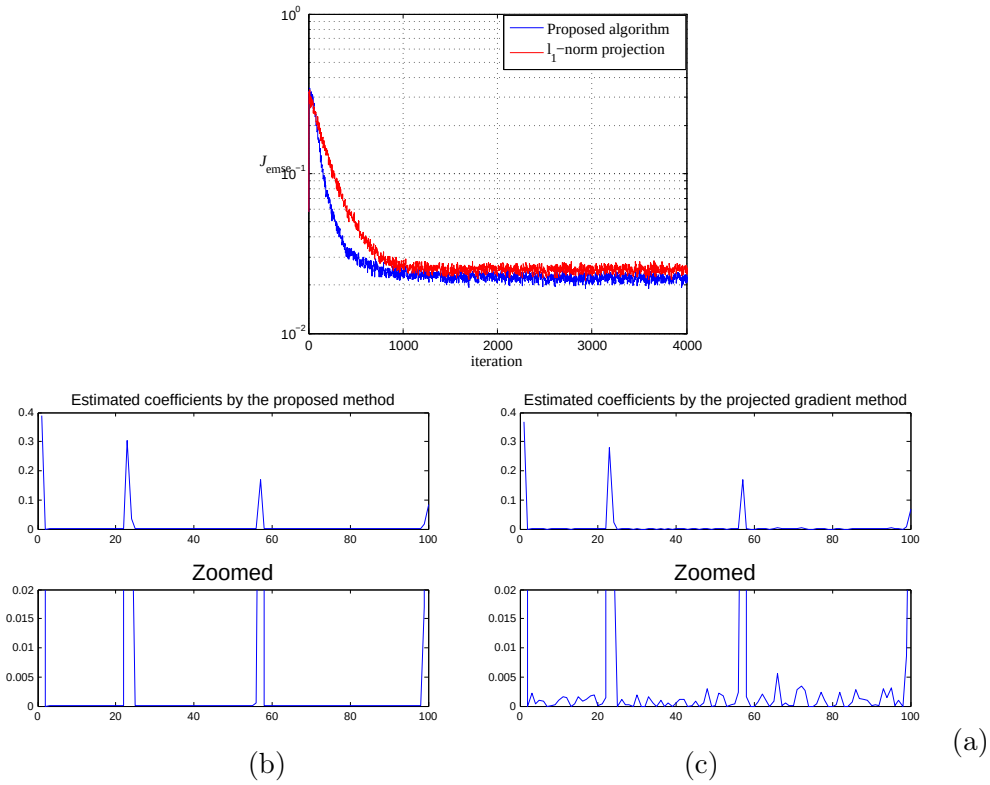


FIGURE 3.2 – Comparison between the proposed method and the projected-gradient method. (a) Learning curves. (b) Estimated coefficients $\alpha_i(n)$ by the proposed method. (c) Estimated coefficients by the projected-gradient method.

Compressive sensing theory provides a robust framework to estimate sparse signals. Instead of the accurate count of non-zero elements by ℓ_0-norm, which leads to NP hard optimization problems, other sparse-inducing norms are used to overcome the difficulty caused by the ℓ_0-norm. The problem using an explicit ℓ_1-norm constraint

$$\begin{aligned} \boldsymbol{\theta} &= \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 \\ &\text{subject to } \|\boldsymbol{\theta}\|_1 \leq \tau \end{aligned} \tag{3.13}$$

is often called the LASSO (Least Absolute Shrinkage and Selection Operator) problem. It can be proved that the LASSO problem is equivalent to the following regularized least-square problem with a proper choice of the regularization parameter λ

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \tag{3.14}$$

Many approaches to solve this problem have been described in the literature. Interior-point methods transfer these problems to a convex quadratic problem [Nesterov 1994]. Other recent developed computational methods include coordinate-wise descent methods [Friedman 2007], fixed point continuation method [Hale 2007], sequential subspace optimization methods [Narkiss 2005],

bound optimization methods [Figueiredo 2005], iterated shrinkage methods [Daubechies 2004], gradient methods [Nesterov 2007], gradient projection for sparse reconstruction algorithm (GPSR) [Figueiredo 2007], sparse reconstruction by separable approximation (SpaRSA) [Wright 2009], and Bregman iterative method [Cai 2009, Yin 2008]. Some of these methods, such as the GPSR, SpaRSA and Bregman iterative method, can efficiently handle large-scale problems.

However, the above methods all operate in batch mode. In order to online identify the sparse system, several adaptive algorithms have also been proposed, including proportionate adaptive filters which incorporate the importance of the individual components by weights, ℓ_0-norm constraint LMS algorithm which approximates ℓ_0-norm by a differentiable function [Gu 2009], sparse LMS which uses the sign function as the subgradient of ℓ_1 term [Chen 2009b, Chen 2010b]. The last one of these three algorithms has the direct connection to the problem (3.14).

Considering that the ℓ_1-norm can be rewritten by the sum of an extended non-negative vector, we shall now investigate a new method to solve the ℓ_1-regularized system identification problem based on the NNLMS algorithm. The mean weight behavior and second-order behavior of the proposed algorithm will be also presented.

3.3.1 Problem formulation

Consider the system represented in Figure I.1 parameterized by $\boldsymbol{\alpha}$, with input $x(n)$ and desired reference $y(n)$. We intend to determine the system coefficients $\boldsymbol{\alpha}$ by minimizing the mean-square error with the sparsity-inducing ℓ_1-norm

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} E\{[\boldsymbol{\alpha}^\top \boldsymbol{x}(n) - y(n)]^2\} + \lambda \|\boldsymbol{\alpha}\|_1 \tag{3.15}$$

where the parameter λ provides a tradeoff between data fidelity and solution sparsity. This ℓ_1-regularized problem can be easily rewritten as a standard non-negative least-square problem, by introducing two $N \times 1$ non-negative vectors $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ which satisfy the following relations

$$\begin{aligned} \boldsymbol{\alpha} &= \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \\ \boldsymbol{\alpha}^+ &\succeq 0 \\ \boldsymbol{\alpha}^- &\succeq 0 \end{aligned} \tag{3.16}$$

For simplicity, let us define a new vector $\tilde{\boldsymbol{\alpha}}$ of the size $2N \times 1$ by connecting $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$, i.e.

$$\tilde{\boldsymbol{\alpha}} = \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix} \tag{3.17}$$

The problem (3.15) with respect to $\boldsymbol{\alpha}$ is thus reformulated as

$$\begin{aligned} \tilde{\boldsymbol{\alpha}}^o = \arg\min_{\tilde{\boldsymbol{\alpha}}} \frac{1}{2} E\left\{ \left(\begin{bmatrix} \boldsymbol{x}(n) \\ -\boldsymbol{x}(n) \end{bmatrix}^\top \tilde{\boldsymbol{\alpha}} - y(n) \right)^2 \right\} + \lambda \mathbf{1}_{2N}^\top \tilde{\boldsymbol{\alpha}} \\ \text{subject to } \tilde{\boldsymbol{\alpha}} \succeq 0 \end{aligned} \tag{3.18}$$

with $\mathbf{1}_{2N}$ an all-one vector with $2N$ elements. As the problem has been reformulated as an optimization problem under the non-negativity constraint in the form of (2.3), we can now solve it in an online manner based on the proposed NNLMS algorithm.

3.3.2 NNLMS-based algorithm for ℓ_1-norm constraint

Considering that $E\{\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\} = \boldsymbol{R}_x$ and $E\{\boldsymbol{x}(n)y(n)\} = \boldsymbol{r}_{xy}$, the problem (3.18) is equivalent to the following expression by dropping the $\boldsymbol{\alpha}$-independent term $E\{y^2(n)\}$

$$\begin{aligned}\tilde{\boldsymbol{\alpha}}^o = \arg\min_{\tilde{\boldsymbol{\alpha}}} \frac{1}{2}\tilde{\boldsymbol{\alpha}}^\top \begin{bmatrix} \boldsymbol{R}_x & -\boldsymbol{R}_x \\ -\boldsymbol{R}_x & \boldsymbol{R}_x \end{bmatrix} \tilde{\boldsymbol{\alpha}} + \left(\lambda \mathbf{1}_{2N} - \begin{bmatrix} \boldsymbol{r}_{xy} \\ -\boldsymbol{r}_{xy} \end{bmatrix}\right)^\top \tilde{\boldsymbol{\alpha}} \\ \text{subject to } \tilde{\boldsymbol{\alpha}} \succeq 0\end{aligned} \tag{3.19}$$

Denoting the cost function in (3.19) by $J(\boldsymbol{\alpha})$, its gradient, or equivalently the gradient of (3.18), is expressed by

$$\nabla J(\tilde{\boldsymbol{\alpha}}) = \begin{bmatrix} \boldsymbol{R}_x & -\boldsymbol{R}_x \\ -\boldsymbol{R}_x & \boldsymbol{R}_x \end{bmatrix} \tilde{\boldsymbol{\alpha}} + \left(\lambda \mathbf{1}_{2N} - \begin{bmatrix} \boldsymbol{r}_{xy} \\ -\boldsymbol{r}_{xy} \end{bmatrix}\right) \tag{3.20}$$

Using the condition (2.4) for the non-negative constrained problem, we can obtain the following fixed-point update to solve this constrained problem

$$\tilde{\boldsymbol{\alpha}}(n+1) = \tilde{\boldsymbol{\alpha}}(n) + \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}}(n) \left(- \begin{bmatrix} \boldsymbol{R}_x & -\boldsymbol{R}_x \\ -\boldsymbol{R}_x & \boldsymbol{R}_x \end{bmatrix} \tilde{\boldsymbol{\alpha}}(n) - \left(\lambda \mathbf{1}_{2N} - \begin{bmatrix} \boldsymbol{r}_{xy} \\ -\boldsymbol{r}_{xy} \end{bmatrix}\right)\right) \tag{3.21}$$

In order to have an online algorithm, the second-order moments $\boldsymbol{R}_x$ and $\boldsymbol{r}_{xy}$ are replaced by their instantaneous estimates $\boldsymbol{x}(n)\boldsymbol{x}^\top(n)$ and $\boldsymbol{x}(n)y(n)$ following a stochastic gradient approach. This leads to the stochastic approximation of (3.21) given by

$$\tilde{\boldsymbol{\alpha}}(n+1) = (1 - \eta\,\lambda)\tilde{\boldsymbol{\alpha}}(n) + \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}}(n) e(n) \tilde{\boldsymbol{x}}(n) \tag{3.22}$$

where we define $\tilde{x}(n) = \begin{bmatrix} \boldsymbol{x}(n) \\ -\boldsymbol{x}(n) \end{bmatrix}$ and the estimation error as

$$\begin{aligned} e(n) &= y(n) - \boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n) \\ &= y(n) - \tilde{\boldsymbol{\alpha}}^\top(n)\tilde{\boldsymbol{x}}(n) \end{aligned} \tag{3.23}$$

Using the relation in (3.16) and (3.17), the system coefficients $\boldsymbol{\alpha}(n)$ is obtained by

$$\boldsymbol{\alpha}(n) = \boldsymbol{\alpha}^+(n) + \boldsymbol{\alpha}^-(n) \tag{3.24}$$

Equations (3.22) and (3.24) enable us to solve the ℓ_1-regularized problem (3.15) in an online manner. In order to give a further inspection into this update rule, we write (3.22) into two parts

$$\boldsymbol{\alpha}^+(n+1) = (1 - \eta\,\lambda)\boldsymbol{\alpha}^+(n) + \eta\, \boldsymbol{D}_{\boldsymbol{\alpha}^+}(n) e(n) \boldsymbol{x}(n) \tag{3.25a}$$

$$\boldsymbol{\alpha}^{-}(n+1) = (1-\eta\,\lambda)\boldsymbol{\alpha}^{-}(n) + \eta\,\boldsymbol{D}_{\boldsymbol{\alpha}^{-}}(n)e(n)\boldsymbol{x}(n) \tag{3.25b}$$

This adaptive algorithm (3.22) can be viewed as a convex combination of two non-negative adaptive filters which are updated by the above two NNLMS algorithms.

The ℓ_1-NNLMS algorithm (3.22) is derived based on original NNLMS algorithm (2.14). In Section 2.5, several variants of NNLMS were proposed in order to improve the performance in some sense. These variants can be transposed directly to the ℓ_1-NNLMS algorithm. Moreover, weighted ℓ_1-norm can also be used for ℓ_1-NNLMS.

3.4 Mean behavior analysis of the ℓ_1-NNLMS

We now study the mean behavior of this ℓ_1-NNLMS algorithm (3.22). As for NNLMS, the input signal $x(n)$ and the desired output signal $y(n)$ are assumed stationary and zero-mean. The signal $z(n) = y(n) - \boldsymbol{x}^{\top}(n)\boldsymbol{\alpha}^{*}$ accounts for measurement noise and modeling errors. It is assumed that $z(n)$ be stationary, zero-mean with the variance σ_z^2 and statistically independent of any other signal. The behavior is studied in non-stationary environments with the same model as in the previous chapter, such as

$$\boldsymbol{\alpha}^{*}(n) = \boldsymbol{\alpha}_{\mathrm{o}}^{*}(n) + \boldsymbol{\xi}(n) \tag{3.26}$$

where $\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)$ is a deterministic time-variant mean and $\boldsymbol{\xi}(n)$ is zero-mean, Gaussian, with covariance matrix $\boldsymbol{\Xi} = \sigma_{\xi}^2\boldsymbol{I}$ and independent of any other signal.

For the analyses that follow, we shall define the weight error vector with respect to the mean unconstrained solution $\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)$ as

$$\begin{aligned}\tilde{\boldsymbol{v}}(n) &= \tilde{\boldsymbol{\alpha}}(n) - \tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n)\\ &= \begin{bmatrix}\boldsymbol{\alpha}^{+}(n)\\ \boldsymbol{\alpha}^{-}(n)\end{bmatrix} - \begin{bmatrix}\boldsymbol{\alpha}_{\mathrm{o}}^{+*}(n)\\ \boldsymbol{\alpha}_{\mathrm{o}}^{-*}(n)\end{bmatrix}\end{aligned} \tag{3.27}$$

3.4.1 Mean weight behavior model

Using the relation (3.27) in the update equation (3.22), we have an update equation for the weight error

$$\tilde{\boldsymbol{v}}(n+1) = (1-\eta\,\lambda)\tilde{\boldsymbol{v}}(n) - \boldsymbol{\Delta}(n) + \eta\,\boldsymbol{D}_{\tilde{x}}(n)\tilde{\boldsymbol{\alpha}}(n)e(n) \tag{3.28}$$

where $\boldsymbol{\Delta}(n) = \tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n+1) - (1-\eta\,\lambda)\,\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n)$. Now we express the estimation error at instant n by

$$\begin{aligned}e(n) &= y(n) - \boldsymbol{\alpha}^{\top}(n)\boldsymbol{x}(n)\\ &= z(n) - \tilde{\boldsymbol{v}}^{\top}(n)\tilde{\boldsymbol{x}}(n) + \boldsymbol{\xi}^{\top}(n)\boldsymbol{x}(n)\end{aligned} \tag{3.29}$$

Note that we do not decompose $\boldsymbol{\xi}(n)$ into two vectors for reason that $\boldsymbol{\xi}(n)$ is a random perturbation and it is not easy to consider its distribution if split. Hence

$\boldsymbol{x}(n)$ is used in the last term instead of $\tilde{\boldsymbol{x}}$(n). Using the expression of $e(n)$, the weight error update equation (3.28) is now written as

$$\begin{aligned}\tilde{\boldsymbol{v}}(n+1) =& (1-\eta\,\lambda)\tilde{\boldsymbol{v}}(n) - \boldsymbol{\Delta}(n) + \eta\, z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{v}}(n) + \eta\, z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n) \\ & - \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}(n) - \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n) \\ & + \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{x}^{\top}(n)\boldsymbol{\xi}(n)\tilde{\boldsymbol{v}}(n) + \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{x}^{\top}(n)\boldsymbol{\xi}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n)\end{aligned} \tag{3.30}$$

Taking the expectation of (3.30), neglecting the statistical dependence of $\tilde{\boldsymbol{x}}(n)$ and $\boldsymbol{v}(n)$, noting the $\boldsymbol{\xi}(n)$ is zero mean and independent of the other signals, and using that $E\left\{z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\right\} = 0$, yields

$$\begin{aligned}E\left\{\tilde{\boldsymbol{v}}(n+1)\right\} =& (1-\eta\,\lambda)E\left\{\tilde{\boldsymbol{v}}(n)\right\} - \boldsymbol{\Delta}(n) - \eta\, E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}(n)\right\} \\ & - \eta\, E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}(n)\right\}\end{aligned} \tag{3.31}$$

Following the calculus that leads to (2.23), the third term of r.h.s. of (3.31) is expressed by

$$E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}(n)\right\} = \mathrm{diag}\{\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\} \tag{3.32}$$

where $\tilde{\boldsymbol{R}}_x$ is the correlation matrix of $\tilde{\boldsymbol{x}}(n)$ expressed with

$$\tilde{\boldsymbol{R}}_x = \begin{bmatrix} \boldsymbol{R}_x & -\boldsymbol{R}_x \\ -\boldsymbol{R}_x & \boldsymbol{R}_x \end{bmatrix} \tag{3.33}$$

and $\tilde{\boldsymbol{K}}(n) = E\left\{\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}^{\top}(n)\right\}$ is the covariance matrix of $\tilde{\boldsymbol{v}}(n)$. Following the calculation of (2.21), the last term of the r.h.s. of (3.31) is expressed by

$$E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^{\top}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}\right\} = \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}}(n)\,\tilde{\boldsymbol{R}}_x\,E\left\{\tilde{\boldsymbol{v}}(n)\right\} \tag{3.34}$$

Using these results with (3.31) yields the following mean weight-error vector update equation

$$\begin{aligned}E\left\{\tilde{\boldsymbol{v}}(n+1)\right\} =& \left((1-\eta\,\lambda)\boldsymbol{I} - \eta\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}}(n)\,\tilde{\boldsymbol{R}}_x\right)E\left\{\boldsymbol{v}(n)\right\} \\ & - \boldsymbol{\Delta}(n) - \eta\,\mathrm{diag}\{\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\}\end{aligned} \tag{3.35}$$

As for the analysis of the NNLMS algorithm, second-order moments defined by the matrix $\tilde{\boldsymbol{K}}(n)$ is needed in order to update the first-order one provided by $E\left\{\tilde{\boldsymbol{v}}(n)\right\}$. For analysis trackability we simplify the model by approximating $\tilde{\boldsymbol{K}}(n) \approx E\left\{\tilde{\boldsymbol{v}}(n)\right\}E\left\{\tilde{\boldsymbol{v}}^{\top}(n)\right\}$, which leads us to the following expression

$$\begin{aligned}E\left\{\tilde{\boldsymbol{v}}(n+1)\right\} =& \left((1-\eta\,\lambda)\boldsymbol{I} - \eta\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^{*}}(n)\tilde{\boldsymbol{R}}_x\right)E\left\{\tilde{\boldsymbol{v}}(n)\right\} - \boldsymbol{\Delta}(n) \\ & - \eta\,\mathrm{diag}\{\tilde{\boldsymbol{R}}_x\,E\left\{\tilde{\boldsymbol{v}}(n)\right\}E\{\tilde{\boldsymbol{v}}^{\top}(n)\}\}\end{aligned} \tag{3.36}$$

This recursion has only expectations containing the first-order moment of $\tilde{\boldsymbol{v}}(n)$ and allows us to analyze the behavior of the algorithm. Extensive simulations will permit us to verify the accuracy of this simplified model.

3.4.2 Special case of a white input signal in stationary environment

In general the behavior of (3.36) can be too complex to be studied analytically. In order to obtain analytical results that allow some understanding of the mean weight behavior, let us now consider a special case where the system is stationary so that $\tilde{\boldsymbol{\alpha}}^*(n) = \boldsymbol{\alpha}^*$, and the input signal $x(n)$ is i.i.d. and drawn from a zero-mean distribution. Unit variance is also assumed without loss of generality, i.e., $\boldsymbol{R}_x = \boldsymbol{I}$. In this case the component-wise recursion is

$$E\{\tilde{v}_i(n+1)\} = (1 - \eta\lambda - \eta\tilde{\alpha}_i^*)\, E\{\tilde{v}_i(n)\} - \eta\,\lambda\,\tilde{\alpha}_i^* - \eta\, E\{\tilde{v}_i(n)\}^2 \tag{3.37}$$

Compared to (2.28), this is a logistic map with an offset $-\eta\lambda\tilde{\alpha}_i$. In what follows, we shall firstly check the stationary points of this relation. Considering the condition $E\{\tilde{v}_i(n+1)\} = E\{\tilde{v}_i(n)\}$ as $n \to \infty$ for the steady state, we have the following equation with respect to $E\{\tilde{v}_i(\infty)\}$

$$E\{\tilde{v}_i(\infty)\}^2 + (\lambda + \tilde{\alpha}_i)\, E\{\tilde{v}_i(\infty)\} + \lambda\,\tilde{\alpha}_i^* = 0 \tag{3.38}$$

The solution of this equation can be obtained without difficulties

$$E\{\tilde{v}_i(\infty)\} = \begin{cases} -\lambda \\ -\tilde{\alpha}_i^* \end{cases} \tag{3.39}$$

This steady state solution can equivalently be written for $\tilde{\alpha}_i$ by adding $\tilde{\alpha}_i$ to $E\{\tilde{v}_i(\infty)\}$

$$E\{\tilde{\alpha}_i(\infty)\} = \begin{cases} \tilde{\alpha}_i^* - \lambda \\ 0 \end{cases} \tag{3.40}$$

If we can check that $E\{\tilde{\alpha}_i(\infty)\}$ converges to $\tilde{\alpha}_i^* - \lambda$ for $\tilde{\alpha}_i \geq \lambda$, and converges to 0 for $\tilde{\alpha}_i < \lambda$, the result (3.40) thus corresponds to the well-known soft-threshold result, which is the solution of the LASSO problem with orthogonal basis. This allows us to certify the validity of the algorithm, in the mean sense, for white input signals. In what follows we shall check the conditions in order that the algorithm is able to converge to this soft-threshold result.

Let us define new variables $\gamma_i(n) = \tilde{v}_i(n) + \lambda$, (3.37) can be rewritten as

$$E\{\gamma_i(n+1)\} = (1 - \eta\,(\tilde{\alpha}_i^* - \lambda))\, E\{\gamma_i(n)\} - \eta\, E\{\gamma_i(n)\}^2 \tag{3.41}$$

Considering (3.39) and (3.40), this recursion is supposed to converge to 0 for $\tilde{\alpha}_i^* \geq \lambda$, and to $-\tilde{\alpha}_i^* + \lambda$ for $\tilde{\alpha}_i^* < \lambda$. We now derive conditions ensuring such convergence. Writing $u(n) = \frac{E\{\gamma_i(n)\}}{1-\eta\,(\tilde{\alpha}_i^* - \lambda)}$, where the index i have been dropped to simplify the notation, we obtain the logistic map

$$u(n+1) = \rho\, u(n)\,(1 - u(n)) \tag{3.42}$$

with $\rho = 1 - \eta\,(\tilde{\alpha}_i^* - \lambda)$, which is assumed nonzero. Desired fixed points now correspond to $u = 0$ and $u = \frac{\rho-1}{\rho}$, respectively. This relation is the same as that in (2.30),

except for the definition of ρ. Considering the convergence scenarios for the logistic map described in Section 2.3.2, yields

$$0 < \eta < \frac{1}{\tilde{\alpha}_i^* - \lambda}, \qquad -\tilde{\alpha}_i^* + \lambda < \gamma_i(0) < \frac{1}{\eta} \tag{3.43}$$

in the case where $\tilde{\alpha}_i^* > \lambda$. If $\tilde{\alpha}_i^* = \lambda$, these two conditions become $\eta > 0$ and $0 < \gamma_i(0) < \frac{1}{\eta}$. In the case where $\tilde{\alpha}_i^* < \lambda$, η and $\gamma_i(0)$ must obey the conditions

$$0 < \eta < -\frac{2}{\tilde{\alpha}_i^* - \lambda}, \qquad 0 < \gamma_i(0) < \frac{1}{\eta} - \tilde{\alpha}_i^* + \lambda \tag{3.44}$$

Finally combining these inequalities and writing the condition with respect to η and $\alpha_i(0)$, we have

$$0 < \eta \leq \min_i \frac{1}{|\tilde{\alpha}_i^* - \lambda|} \qquad \text{and} \qquad 0 < \alpha_i(0) < \frac{1}{\eta} \text{ for all } i \tag{3.45}$$

3.5 Second-order behavior analysis

Using $e(n) = z(n) - \tilde{\boldsymbol{v}}^\top(n)\tilde{\boldsymbol{x}}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)$, neglecting the statistical dependence of $\boldsymbol{x}(n)$ and $\tilde{\boldsymbol{v}}(n)$, and using the properties assumed for $z(n)$ and $\boldsymbol{\xi}(n)$ yields an expression for the mean-square estimation error

$$\begin{aligned} E\left\{e^2(n)\right\} &= E\left\{(z(n) - \tilde{\boldsymbol{v}}^\top(n)\tilde{\boldsymbol{x}}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n))(z(n) - \tilde{\boldsymbol{v}}^\top(n)\tilde{\boldsymbol{x}}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n))\right\} \\ &= \sigma_z^2 + E\left\{\tilde{\boldsymbol{v}}^\top(n)\tilde{\boldsymbol{x}}(n)\tilde{\boldsymbol{x}}^\top(n)\tilde{\boldsymbol{v}}(n)\right\} + E\left\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{x}^\top(n)\boldsymbol{\xi}(n)\right\} \\ &= \sigma_z^2 + \text{trace}\{\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\} + \text{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\} \end{aligned} \tag{3.46}$$

The term $\text{trace}\{\boldsymbol{R}_x\,\boldsymbol{\Xi}\}$ is the contribution of the non-stationarity of the system to the EMSE caused by random perturbation due to the weight lag. In order to determine the excess mean square error due to $\text{trace}\{\tilde{\boldsymbol{R}}_x\tilde{\boldsymbol{K}}(n)\}$, we shall determine a recursion for $\tilde{\boldsymbol{K}}(n)$. Post-multiplying (3.36) by its transpose, taking the expected value, and using the statistical properties of $z(n)$ and $\boldsymbol{\xi}(n)$, yields

$$\begin{aligned} \tilde{\boldsymbol{K}}(n+1) = &(1-\eta\lambda)^2\,\tilde{\boldsymbol{K}}(n) - (\boldsymbol{P}_1(n) + \boldsymbol{P}_1^\top(n)) - \eta\,(\boldsymbol{P}_2(n) + \boldsymbol{P}_2^\top(n)) \\ &- \eta\,(\boldsymbol{P}_3(n) + \boldsymbol{P}_3^\top(n)) + \boldsymbol{P}_4(n) + (\boldsymbol{P}_5(n) + \boldsymbol{P}_5^\top(n)) \\ &+ (\boldsymbol{P}_6(n) + \boldsymbol{P}_6^\top(n)) + \eta^2\,\boldsymbol{P}_7(n) + +\eta\,(\boldsymbol{P}_8(n) + \boldsymbol{P}_8^\top(n)) \\ &+ \eta^2\,\boldsymbol{P}_9(n) + \eta^2\,\boldsymbol{P}_{10}(n) + \eta\,(\boldsymbol{P}_{11}(n) + \boldsymbol{P}_{11}^\top(n)) + \eta\,(\boldsymbol{P}_{12}(n) + \boldsymbol{P}_{12}^\top(n)) \\ &+ \eta^2(\boldsymbol{P}_{13}(n) + \boldsymbol{P}_{14}(n) + \boldsymbol{P}_{15}(n) + \boldsymbol{P}_{15}^\top(n)) \end{aligned} \tag{3.47}$$

where matrices $\boldsymbol{P}_1(n)$ to $\boldsymbol{P}_{12}(n)$ are defined by

$$\boldsymbol{P}_1(n) = E\left\{(1-\eta\lambda)\boldsymbol{v}(n)\boldsymbol{\Delta}(n)^\top\right\} \tag{3.48}$$

$$\boldsymbol{P}_2(n) = E\left\{(1-\eta\lambda)\boldsymbol{v}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}(n)\right)^\top\right\} \tag{3.49}$$

$$\boldsymbol{P}_3(n) = E\left\{(1-\eta\lambda)\boldsymbol{v}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.50}$$

$$\boldsymbol{P}_4(n) = E\left\{\boldsymbol{\Delta}(n)\,\boldsymbol{\Delta}^\top(n)\right\} \tag{3.51}$$

$$\boldsymbol{P}_5(n) = E\left\{\boldsymbol{\Delta}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}(n)\right)^\top\right\} \tag{3.52}$$

$$\boldsymbol{P}_6(n) = E\left\{\boldsymbol{\Delta}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.53}$$

$$\boldsymbol{P}_7(n) = E\left\{z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{v}(n)\left(z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{v}(n)\right)^\top\right\} \tag{3.54}$$

$$\boldsymbol{P}_8(n) = E\left\{z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{v}(n)\left(z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.55}$$

$$\boldsymbol{P}_9(n) = E\left\{z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\left(z(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.56}$$

$$\boldsymbol{P}_{10}(n) = E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}(n)\right)^\top\right\} \tag{3.57}$$

$$\boldsymbol{P}_{11}(n) = E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\boldsymbol{v}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.58}$$

$$\boldsymbol{P}_{12}(n) = E\left\{\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\boldsymbol{v}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \tag{3.59}$$

These terms do not involve the random perturbation $\boldsymbol{\xi}(n)$ on the weights. Their expected values are calculated in the following under the same statistical assumptions as those used for NNLMS algorithm.

The expected value $\boldsymbol{P}_1(n)$ is given by basic linear algebra

$$\begin{aligned}\boldsymbol{P}_1(n) &= E\left\{(1-\eta\lambda)\boldsymbol{v}(n)\boldsymbol{\Delta}^\top\right\}\\ &= (1-\eta\lambda)E\left\{\boldsymbol{v}(n)\right\}\,\boldsymbol{\Delta}^\top\end{aligned} \tag{3.60}$$

According to (2.54) and (2.47) in the analysis of NNLMS algorithm, $\boldsymbol{P}_2(n)$ and $\boldsymbol{P}_3(n)$ are expressed respectively by

$$\boldsymbol{P}_2(n) = (1-\eta\lambda)\,\tilde{\boldsymbol{K}}(n)\,\tilde{\boldsymbol{R}}_x\,\boldsymbol{D}_{\tilde{\boldsymbol{v}}}(n) \tag{3.61}$$

and

$$\boldsymbol{P}_3(n) = (1-\eta\lambda)\,\tilde{\boldsymbol{K}}(n)\,\tilde{\boldsymbol{R}}_x\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \tag{3.62}$$

The expected value $\boldsymbol{P}_5(n)$ is calculated by

$$\begin{aligned}\boldsymbol{P}_5(n) &= E\left\{\boldsymbol{\Delta}(n)\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}(n)\right)^\top\right\}\\ &= \boldsymbol{\Delta}(n)\,E\left\{\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}(n)\right)^\top\right\}\\ &= \boldsymbol{\Delta}(n)\left(\mathrm{diag}\{\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\}\right)^\top\end{aligned} \tag{3.63}$$

The last line of this calculation is obtained as (2.23).
The expected value $\boldsymbol{P}_6$ is written by

$$\begin{aligned}\boldsymbol{P}_6(n) &= E\left\{\boldsymbol{\Delta}\left(\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{x}}^\top(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*(n)\right)^\top\right\} \\ &= \boldsymbol{\Delta}\, E\left\{\tilde{\boldsymbol{v}}^\top(n)\,\tilde{\boldsymbol{x}}(n)\tilde{\boldsymbol{x}}^\top(n)\right\}\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \\ &= \boldsymbol{\Delta}\, E\{\tilde{\boldsymbol{v}}^\top(n)\}\,\tilde{\boldsymbol{R}}_x\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n)\end{aligned} \tag{3.64}$$

The other terms can be obtained directly using the results in analysis of NNLMS. Using (2.50) to (2.51) ,

$$\boldsymbol{P}_7(n) = \sigma_z^2\,(\tilde{\boldsymbol{R}}_x \circ \tilde{\boldsymbol{K}}(n)) \tag{3.65}$$

Using (2.49),

$$\boldsymbol{P}_8(n) = \sigma_z^2\, E\,\{\boldsymbol{D}_{\tilde{\boldsymbol{v}}}(n)\}\;\tilde{\boldsymbol{R}}_x\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \tag{3.66}$$

Using (2.48),

$$\boldsymbol{P}_9(n) = \sigma_z^2\;\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n)\,\tilde{\boldsymbol{R}}_x\,\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \tag{3.67}$$

Using (2.59) to (2.67),

$$\boldsymbol{P}_{10}(n) = \boldsymbol{Q}(n) \circ \tilde{\boldsymbol{K}}(n) \tag{3.68}$$

with $\boldsymbol{Q}(n) = \mathrm{trace}\{\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\}\,\tilde{\boldsymbol{R}}_x + 2\,\tilde{\boldsymbol{R}}_x\,\tilde{\boldsymbol{K}}(n)\,\tilde{\boldsymbol{R}}_x$.
Using (2.57) to (2.58),

$$\boldsymbol{P}_{11}(n) = E\,\{\boldsymbol{D}_{\tilde{\boldsymbol{v}}}(n)\}\;\boldsymbol{Q}(n)\;\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \tag{3.69}$$

Using (2.55) to (2.56),

$$\boldsymbol{P}_{12}(n) = \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n)\,\boldsymbol{Q}(n)\;\boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}_{\mathrm{o}}^*}(n) \tag{3.70}$$

We now derive expressions for $\boldsymbol{P}_{13}(n)$ through $\boldsymbol{P}_{15}(n)$. These terms convey the effect of the environment non-stationarity due to the random variations of system weights. These terms are expressed by

$$\boldsymbol{P}_{13}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{v}}(n)\tilde{\boldsymbol{v}}^\top(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\} \tag{3.71}$$

$$\boldsymbol{P}_{14}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\boldsymbol{\alpha}_{\mathrm{o}}^*(n)\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\} \tag{3.72}$$

$$\boldsymbol{P}_{15}(n) = E\{\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\tilde{\boldsymbol{v}}(n)\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\boldsymbol{D}_{\tilde{\boldsymbol{x}}}(n)\} \tag{3.73}$$

Computing (i,j)-th entry of $\boldsymbol{P}_{13}(n)$ yields

$$\begin{aligned}[\boldsymbol{P}_{13}]_{ij}(n) &= E\left\{\sum_k\sum_l \xi_k(n)\xi_l(n)x_k(n)x_l(n)\tilde{x}_i(n)\tilde{v}_i(n)\tilde{v}_j(n)\tilde{x}_j(n)\right\} \\ &= \sum_k\sum_l E\,\{\xi_k(n)\xi_l(n)\}\;E\,\{\tilde{v}_i(n)\tilde{v}_j(n)\}\;E\,\{x_k(n)x_l(n)\tilde{x}_i(n)\tilde{x}_j(n)\}\end{aligned} \tag{3.74}$$

As $E\,\{\xi_k(n)\xi_l(n)\} \neq 0$ only for $k = l$,

$$[\boldsymbol{P}_{13}]_{ij}(n) = \sigma_\xi^2\sum_k[\tilde{\boldsymbol{K}}(n)]_{ij}\,E\left\{x_k^2(n)\tilde{x}_i(n)\tilde{x}_j(n)\right\} \tag{3.75}$$

Using the Gaussian moment factorizing theorem yields

$$\begin{aligned}\sum_k E\left\{x_k^2(n)\tilde{x}_i(n)\tilde{x}_j(n)\right\} &= \sum_k \left([\tilde{\boldsymbol{R}}_x]_{ij}[\boldsymbol{R}_x]_{kk} + [\boldsymbol{R}'_x]_{ik}[\boldsymbol{R}'_x]_{jk}\right) \\ &= \left[\tilde{\boldsymbol{R}}_x \,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}'_x\,\boldsymbol{R}'^{\top}_x\right]_{ij}\end{aligned} \tag{3.76}$$

with $\boldsymbol{R}'_x = [\boldsymbol{R}_x, -\boldsymbol{R}_x]^\top$. This enables us to write the result in matrix form

$$\boldsymbol{P}_{13}(n) = \sigma_\xi^2\,\tilde{\boldsymbol{K}}(n) \circ \left(\tilde{\boldsymbol{R}}_x \,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}'_x\,\boldsymbol{R}'^{\top}_x\right) \tag{3.77}$$

Similarly, we have

$$\boldsymbol{P}_{14}(n) = \sigma_\xi^2\,\left(\boldsymbol{\alpha}_{\mathrm{o}}^{*}(n)\,\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\right) \circ \left(\tilde{\boldsymbol{R}}_x \,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}'_x\,\boldsymbol{R}'^{\top}_x\right) \tag{3.78}$$

$$\boldsymbol{P}_{15}(n) = \sigma_\xi^2\,\left(E\left\{\tilde{\boldsymbol{v}}(n)\right\}\boldsymbol{\alpha}_{\mathrm{o}}^{*\top}(n)\right) \circ \left(\tilde{\boldsymbol{R}}_x \,\mathrm{trace}\{\boldsymbol{R}_x\} + 2\,\boldsymbol{R}'_x\,\boldsymbol{R}'^{\top}_x\right) \tag{3.79}$$

Using the expected values $\boldsymbol{P}_1(n)$ to $\boldsymbol{P}_{15}(n)$in (3.47), we finally obtain a recursive analytical model for the behavior of $\tilde{\boldsymbol{K}}(n)$. This result can be used to study the convergence properties of $E\left\{e^2(n)\right\}$.

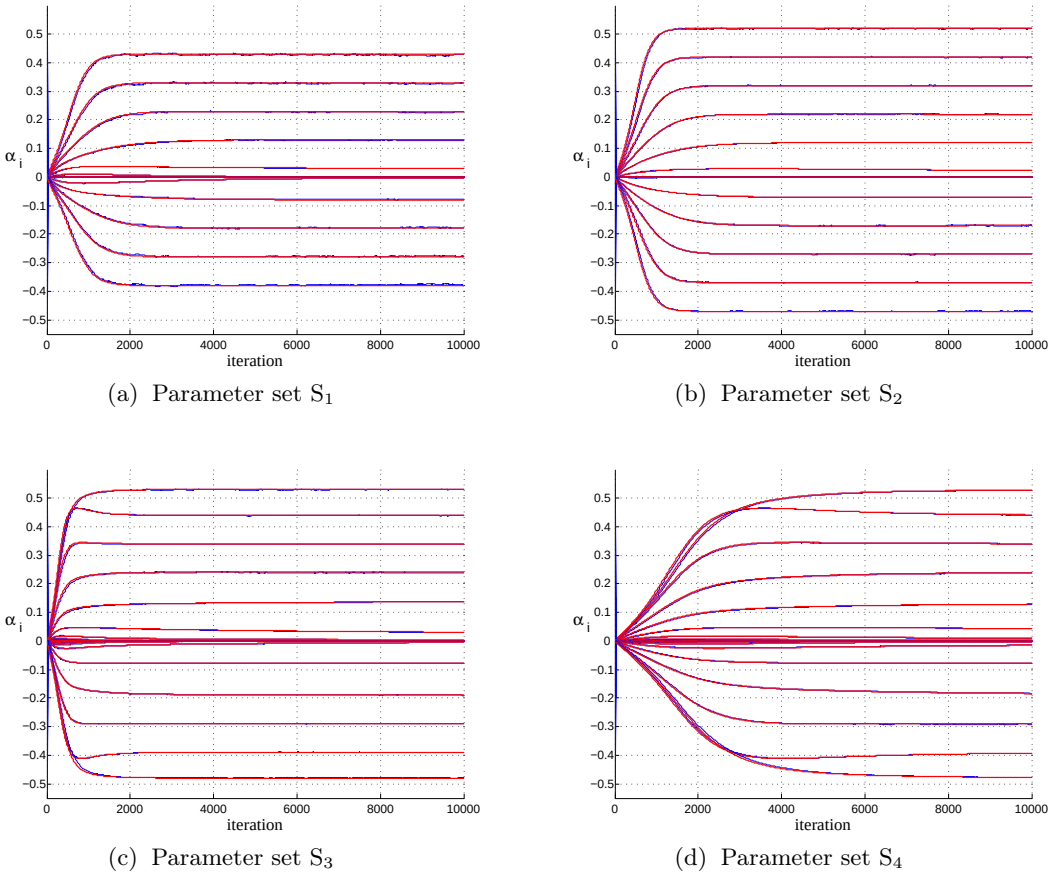
3.6 Simulation results and discussion

3.6.1 Stationary environment

This section presents simulation examples to verify the validity of the analytical models of the NNLMS-based ℓ_1-regularized algorithm. We will firstly illustrate the accuracy of the models through examples with the 30-th order impulse response given by

$$\alpha_{\mathrm{o}_i}^{*(\mathrm{stat.})} = \begin{cases} 0.55 - 0.1\,i & i = 1,\ldots,5 \\ 0 & i = 6,\ldots,25 \\ -0.1\,(i-25) & i = 26,\ldots,30 \end{cases} \tag{3.80}$$

In order to verify the validity of model, and to understand how the parameters affect the performance of the algorithm, we shall conduct the simulation with the following parameter sets : $\mathrm{S}_1 = \{\boldsymbol{R} = \boldsymbol{I}, \eta = 0.01, \lambda = 0.12\}$, $\mathrm{S}_2 = \{\boldsymbol{R} = \boldsymbol{I}, \eta = 0.01, \lambda = 0.03\}$, $\mathrm{S}_3 = \{\text{correlated input}, \eta = 0.01, \lambda = 0.03\}$ and $\mathrm{S}_4 = \{\text{correlated input}, \eta = 0.003, \lambda = 0.03\}$, where the correlated input is generated by the first-order AR model $x(n) = 0.5\,x(n-1) + w(n)$. Signal $w(n)$ is a white noise whose variance was adjusted to ensure $\sigma_x^2 = 1$. Comparison between S_1 and S_2 will enable us to illustrate the effect of the parameter λ. Comparison between S_2 and S_3 will enable us to verify the model precision for different input statistical properties. Comparison between S_3 and S_4 will show the effect of the step size. The noise variance remained $\sigma_z^2 = 1$ through all the simulations. The adaptive weights in $\alpha(0)$ were all initialized at $1/N$ for each realization. Red curves and blue curves were obtained using the theoretical mode and Monte Carlo simulations. Theoretical curves are determined by (3.36) and (3.47).



(a) Parameter set S_1

(b) Parameter set S_2

(c) Parameter set S_3

(d) Parameter set S_4

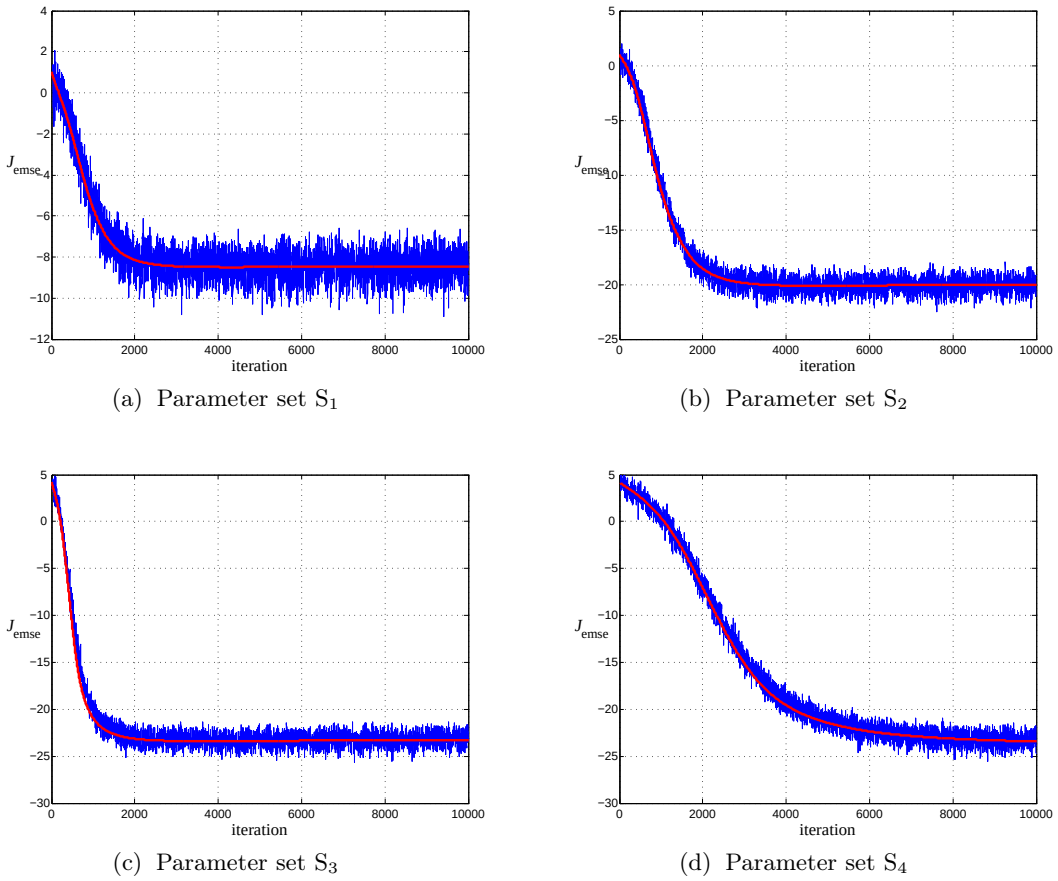
FIGURE 3.3 – Evolution of the coefficients $\alpha_i(n)$ for the NNLMS-based ℓ_1-regularized algorithm with different parameter settings.

Firstly we illustrate the mean behavior of the algorithm in Figure 3.3.

In all these cases, it can be observed that our model has a good match with the simulation results. In Figure 3.3(a), all the weights are reduced by $\lambda = 0.12$ and those smaller than λ shrank to 0. This result is consistent with the soft-threshold solution. In Figure 3.3(b), all the weights are less biased than in Figure 3.3(a) due to the smaller value for λ. In the case where the input is correlated, a different solution is obtained in Figure 3.3(c) compared with Figure 3.3(b). In Figure 3.3(d) a smaller step size leads to a slower convergence rate of the algorithm.

Let us illustrate now the EMSE behavior of the proposed algorithm. Red curves and blue curves are obtained using the theoretical mode and Monte Carlo simulations respectively. The results are shown in Figure 3.4.

These figures show the accuracy of the derived model in different cases. The accuracy of these analytical models enable us to use them as important tools for adaptive filter design.



(a) Parameter set S_1 (b) Parameter set S_2

(c) Parameter set S_3 (d) Parameter set S_4

FIGURE 3.4 – EMSE behavior of the NNLMS-based ℓ_1-regularized algorithm with different parameter settings.

3.6.2 Non-stationary environment

After examining the proposed models in the stationary environment, we shall now consider non-stationary environments defined by

$$\alpha_{o_i}^{*\,(\text{nonstat.1})}(n) = \boldsymbol{\alpha}_{o_i}^{*\,(\text{stat.})} + \xi_i(n) \tag{3.81}$$

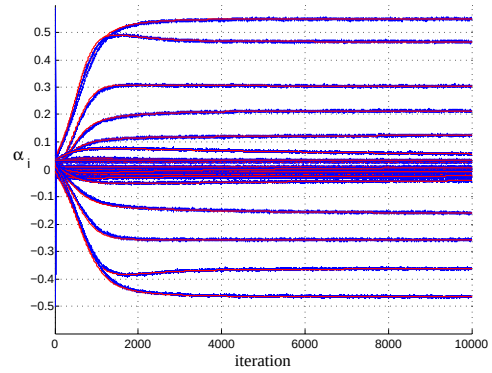
and

$$\alpha_{o_i}^{*\,(\text{nonstat.2})}(n) = \boldsymbol{\alpha}_{o_i}^{*\,(\text{stat.})} + \frac{|\boldsymbol{\alpha}_{o_i}^{*\,(\text{stat.})}|}{10} \sin\left(\frac{2\,\pi}{T}\,n + 2\,\pi\frac{i-1}{N}\right) + \xi_i(n) \tag{3.82}$$

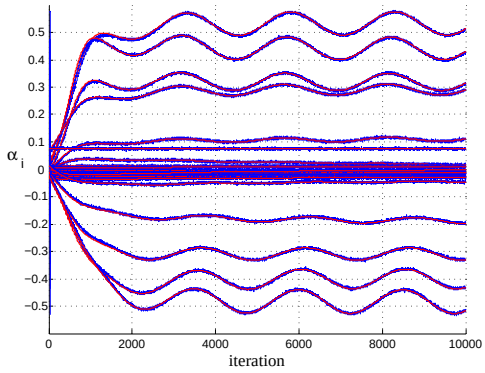
In this first case, there is only one random perturbation added to the stationary unconstrained solutions. Whereas a deterministic sinusoidal time-varying trajectory is also added in the second case. The period T of sinusoidal components was set to 2500. $\boldsymbol{\xi}(n)$ is a zero-mean Gaussian random vector with correlation matrix $\sigma_\xi^2\boldsymbol{I}$ with $\sigma_\xi^2 = 5 \times 10^{-4}$. The input signal is a first-order AR process given by $x(n) = 0.5\,x(n-1) + w(n)$, with $w(n)$ i.i.d. zero-mean Gaussian with variance σ_w^2, adjusted

to obtain the desired input power $\sigma_x^2 = 1$. The step size was set to $\eta = 0.005$ and regularization parameter was set to $\lambda = 0.06$. The variance of modeling noise error $z(n)$ remained $\sigma_z^2 = 0.01$.

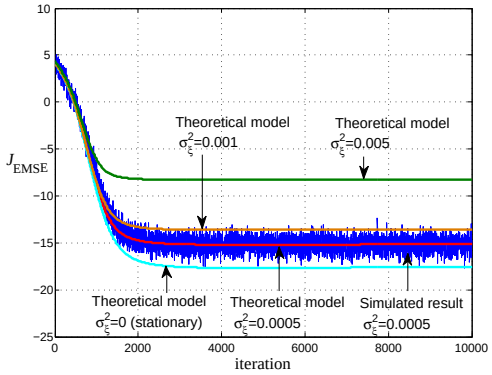
The blue and red curves show the simulation results and the accurate theoretical predictions. In addition to these curves, the variance σ_ξ^2 was varied using the values in $\{0, 0.001, 0.005\}$. The second-order EMSE curves obtained from the theoretical model are illustrated in Figure 3.5(c) and 3.5(d). The simulation results conform with these curves but are not shown for clarity. Effects of the deterministic time-varying trajectory and random perturbations can be clearly observed in this figures. Extra EMSE arises due to tracking of the varying optimal solution variations.



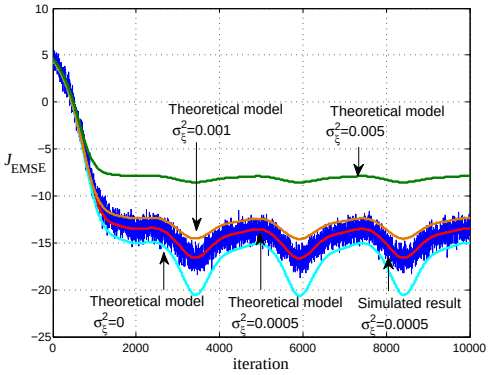
(a) mean behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.1})}$



(b) mean behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.2})}$



(c) 2nd-order behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.1})}$



(d) 2nd-order behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.2})}$

FIGURE 3.5 – Algorithm behavior in non-stationary environments.

Part II

Constrained nonlinear system identification : the problem of hyperspectral image unmixing

Context

Hyperspectral imaging has become a quickly growing area of research. It has received considerable attention in the last decade. It makes use of hundreds of contiguous spectral bands to expand the capacity of multispectral sensors that use tens of discrete spectral bands. These characteristics are suitable for detection and classification of surfaces and chemical elements in the observed images. Applications include land use analysis, pollution monitoring, wide-area reconnaissance, and field surveillance, to cite a few.

The number and variety of processing tasks in hyperspectral remote sensing are large. One of the most popular and important ones is unmixing. Due to multiple factors, including the possible low spatial resolution of some hyperspectral-imaging devices, the diversity of materials in the observed scene, the reflections of photons onto several objects, etc., mixed pixel problems can occur and be critical for proper interpretation of images. Under such circumstances, substance information can only be obtained at subpixel level. The unmixing task involves the decomposition of each mixed pixel into its pure endmember spectra, and the estimation of the abundance value for each endmember [Keshava 2002]. Several approaches have been developed for endmember extraction [Bioucas-Dias 2012]. On the one hand, methods with pure pixel assumption have been proposed to extract the endmembers from pixels in the scene, such as the pixel purity index algorithm [Boardman 1993], the vertex component analysis (VCA) [Nascimento 2005], and the N-FINDR algorithm [Winter 1999], among others [Plaza 2011b, Honeine 2012]. On the other hand, some methods have been proposed to overcome the absence of pure pixels, by generating virtual endmembers, such as the minimum volume simplex analysis (MVSA) [Li 2008], the minimum volume enclosing simplex algorithm (MVES) [Chan 2009], and the minimum volume constrained nonnegative matrix factorization (MVC-NMF) [Miao 2007]. Endmember identification and abundance estimation can be conducted either in a sequential or collaborative manner. Under the assumption that the endmembers have been identified, hyperspectral image unmixing then reduces to estimating the fractional abundances. The term unmixing in the paper represents the abundance estimation step, which is referred to as the *supervised unmixing* in some literature.

The classical unmixing problem usually considers an observation model where it is assumed that each observed pixel is a linear mixture of endmembers, weighted by the fractional abundances. This linear mixture model has been widely used to identify and quantify pure components in remotely sensed images due to its simple physical interpretation and tractable estimation process. To be physically interpretable, the driving abundances are often required to satisfy two constraints : all the abundances must be nonnegative, and their sum must be equal to one. Nevertheless, there are many situations in which this model may not be appropriate (e.g., involving multiple light scattering effects) and could be advantageously replaced by a nonlinear one. For instance, multiple scattering effects can be observed on complex vegetated surfaces [Ray 1996] where it is assumed that incident solar radiation

is scattered by the scene through multiple bounces involving several endmembers. Some nonlinear mixture models, such as the *generalized bilinear model* studied in [Halimi 2011], account for presence of multi-photon interactions by introducing additional interaction terms in the linear model. Another typical situation is the case where the components of interest are in an intimate association, and the photons interact with all the materials simultaneously as they are multiply scattered. A bidirectional reflectance model based on the fundamental principles of radiative transfer theory was proposed in [Hapke 1981] to describe these interactions. It is usually referred to as the *intimate mixture model*. Obviously, the mixture mechanism in a real scene may be much more complex than the above models and often relies on scene parameters that are difficult to obtain.

The general unmixing problem can thus be considered as a nonlinear system identification problem with respect to the parameters under the non-negativity and the sum-to-one constraints. Nonlinear unmixing algorithms will be explored in this part of the thesis. We shall study kernel-based algorithms formulated on a linear mixture/nonlinear fluctuation model (Chapter 4), and extend this model by incorporating spatial information in nonlinear unmixing to improve the performance (Chapter 5).

Specific notations

Suppose that the hyperspectral image under study has w pixels in each row and h pixels in each column. Each pixel consists of a reflectance vector in L contiguous spectral bands. In order to facilitate the presentation, we transform this three dimensional image into an $L \times N$ matrix, with $N = w \times h$ the total number of pixels. Then let

- $n \in \{1, \ldots, N\}$ be the sequential index of pixels in the image.
- $\boldsymbol{r}_n = [r_{n1}, r_{n2}, \ldots, r_{nL}]^\top$ be the $(L \times 1)$ observed reflectance vector for the pixel n, which consists of a mixture of R endmember spectra.
- $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_R]$ be the $L \times R$ endmember target matrix, where each column $\boldsymbol{m}_i$ is an endmember spectral signature. For expository convenience, we denote by $\boldsymbol{m}_{\lambda_\ell}^\top$ the ℓ-th row of $\boldsymbol{M}$, namely, the vector of the R endmember signatures at the ℓ-th wavelength band.
- $\boldsymbol{\alpha}_n = [\alpha_{n1}, \alpha_{n2}, \ldots, \alpha_{nR}]^\top$ be the $(R \times 1)$ abundance vector associated to the pixel n.
- $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ be the matrix composed of all the abundance vectors.
- $\boldsymbol{1}_N$ be the $(N \times 1)$ all-one vector, and $\boldsymbol{I}_N$ the $(N \times N)$ identity matrix.

Chapter 4

Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model

Contents

4.1 Introduction

The linear mixture model has been widely used to identify and quantify pure components in remotely sensed images due to its simple physical interpretation and tractable estimation process. To be physically interpretable, the driving abundances are often required to satisfy two constraints : all abundances must be nonnegative, and their sum must be equal to one. At least two classes of approaches can be distinguished to determine abundances. On the one hand, there are estimation methods that lead to an optimization problem which must be solved subject to non-negativity and sum-to-one constraints [Heinz 2001]. On the other hand, following the principles of Bayesian inference, there are simulation techniques that define prior distributions for abundances, and estimate unknown parameters based on the

resulting joint posterior distribution [Moussaoui 2008, Dobigeon 2009, Eches 2010, Themelis 2012]. Some recent works also take sparsity constraints into account in the unmixing process [Bioucas-Dias 2012, Themelis 2012, Guo 2009, Bioucas-Dias 2010, Iordache 2011].

Although the linear mixture model has obvious practical advantages, there are many situations in which it may not be appropriate (e.g., involving multiple light scattering effects) and could be advantageously replaced by a nonlinear one. Therefore nonlinear spectral unmixing has received intensive attention during the last five years, as one may encounter situations where the linear mixing model is not adequate for describing specific mixing effects. Examples of these nonlinear effects usually result from multiple light scattering, intimate association among of substance components and other distortion. In these cases, the hyperspectral data are no longer confined in a simplex. Simple use of linear method may lead to incorrect estimation results.

To the best of our knowledge, we can classify recent nonlinear unmixing approaches into the following classes, with an arbitrary naming.

1) *Direct methods.* Some methods explore the nonlinear relation/distortion between the material spectra and received hyperspectral data in an explicit way, in order to design unmixing algorithms with these deterministic relations. For instance, explicitly considering the second-order photon interaction caused by multiple light scattering, in [Raksuntorn 2010, Nascimento 2009], the authors extended the collection of endmembers by adding artificial cross-terms of pure signatures to model light scattering effects on different materials. The constrained least square problems were consequently solved as an extended linear FCLS problem, in order to estimate abundances, and proportion factors of cross-terms if necessary. In [Nascimento 2010], the authors addressed the nonlinear unmixing problem with an intimate mixture model. The proposed method directly converts observed reflectances into albedo using a look-up table established by the inverse of the Hapke's reflectance model [Hapke 1981] with known observation parameters. After, a linear algorithm estimates the endmember albedos and the mass fractions for each sample. These approaches are based on explicit hypothesis of the full knowledge of the mixing process. They are characterized by the simplicity and the efficiency as advantages. The limitation of the methods of this class is that unmixing is highly depend on determination of the underlying nonlinear transfer function and lack of flexibility.

2) *Manifold learning methods.* Manifold learning is a well-known concept for nonlinear dimensionality reduction. Algorithms for this task are based on the idea that the high dimensional data are perhaps embedded in a low dimensional manifold. The most popular methods include Isomap, LLE, Laplacian eigenmaps, etc. In [Bachmann 2005] the authors firstly studied the manifold learning problem within the context of hyperspectral imaging. Algorithms, based on Isomap and LLE for exploiting the nonlinear structure of hyperspectral imagery were developed and compared to the linear mixing. In [Heylen 2011], the authors proposed a simplex volume maximization based on the geodesic distances calculated by Isomap for extracting endmembers. The abundances were then estimated by the proportions among vo-

lumes on the manifold. Besides direct use of manifold dimensionality reduction for unmixing, manifold related terms, such as Laplacian eigenmap were also used in nonnegative matrix factorization as regularization terms to improve unmixing performance in nonlinear cases [Cai 2011, Guan 2011]. Manifold learning based algorithms can also be found in the hyperspectral image classification problem [Ma 2010]. These approaches take advantage of manifold learning to transfer the nonlinearity of spectral space spanned by observed data, to low dimensional linear representation. However, these types of algorithms lead to a very high computational burden and an insufficient number of data may lead to an inaccurate estimation of the manifold due to the very high dimension of spectral vectors.

3) *Kernel based methods.* Nonlinear algorithms operating in reproducing kernel Hilbert spaces (RKHS) have been a topic of considerable interest in the machine learning community, and have proved their efficiency in solving nonlinear problems. Kernel-based methods have already been considered for detection and classification in hyperspectral images [Camps-Valls 2005, Heesung 2005]. Kernel-based nonlinear unmixing approaches have also been investigated [Broadwater 2007, Broadwater 2009]. These algorithms were mainly obtained by replacing each inner product between endmember, in the cost functions to be optimized, by a kernel function. This can be viewed as a nonlinear distortion map applied to the spectral signature of each material, independently of their interactions. This principle may be extremely efficient in solving detection and classification problems as a proper distortion can increase the detectability or separability of some patterns. It is however of little physical interest in solving the unmixing problem because the nonlinear nature of mixing is not only governed by individual spectral distortions, but also by nonlinear interactions of the materials. To conquer this problem, in [Chen 2013], we formulated the problem of estimating abundances of a nonlinear mixture of hyperspectral data. This new kernel-based paradigm, which has a clear physical interpretation, allows us to take nonlinear interactions of the endmembers into account. The abundances are then determined by solving an appropriate kernel-based regression problem under constraints. The complexity of these methods is usually moderate.

4) *Bayesian methods.* In this class of methods, Bayesian inference algorithms were studied to estimate the abundance coefficients and some other parameters. Metropolis-within-Gibbs sampling was often used to generate samples distributed according to the posterior distribution of the unknown model parameters. In [Halimi 2011], a nonlinear unmixing algorithm for the general bilinear mixture model was proposed. In [Altmann 2011], the same strategy was applied to a post-nonlinear mixing model to estimate the abundances. If the parameter prior distributions are properly defined, in addition to the estimated values of the parameters, these algorithms are able to provide also their distributions, but at the price of a high computational cost.

5) *Supervised methods.* Supervised methods refer to neural network type methods, which intends to establish an input-output relation between observed spectral data and the abundances by a black-box way. Using such approaches can bypass difficulties with unknown mixing mechanism and parameters. In [Guilfoyle 2001],

a radial basis function neural network was used to unmix intimate mixtures. In [Plaza 2004], the authors designed a multi-layer perceptron neural network combined with a Hopfield neural network to deal with nonlinear mixtures. However, these supervised methods require their networks to be trained, and the quality of the training data may affect the performance notably. Moreover, for a new set of spectra in a scene, or different embedded parameters, a new neural network should be trained again before unmixing can be performed. As our work focuses on the unsupervised unmixing, this class of methods are beyond the discussion in the rest of the paper.

In this chapter, we formulate the problem of estimating abundances of a nonlinear mixture of hyperspectral data. This new kernel-based paradigm allows to take nonlinear interactions of the endmembers into account. It leads to a more meaningful interpretation of the unmixing mechanism than existing kernel based methods. The abundances are determined by solving an appropriate kernel-based regression problem under constraints.

4.2 A kernel-based nonlinear unmixing paradigm

We first consider the linear mixing model where any observed pixel is a linear combination of the endmembers, weighted by the fractional abundances, that is,

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{4.1}$$

where $\boldsymbol{n}$ is a noise vector. Under the assumption that the endmember matrix $\boldsymbol{M}$ is known, the vector $\boldsymbol{\alpha}$ of fractional abundances is usually determined by minimizing a cost function of the form

$$\begin{aligned} J(\boldsymbol{\alpha}) &= J_{\text{reg}}(\boldsymbol{\alpha}) + \frac{1}{2\mu}\|\boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha}\|^2 \\ &= J_{\text{reg}}(\boldsymbol{\alpha}) + \frac{1}{2\mu}\sum_{\ell=1}^{L}(r_\ell - \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell})^2 \end{aligned} \tag{4.2}$$

under the non-negativity and sum-to-one constraints [1]

$$\begin{aligned} &\alpha_i \geq 0, \qquad \forall i \in 1, \ldots, R \\ &\sum_{i=1}^{R} \alpha_i = 1, \end{aligned} \tag{4.3}$$

where $J_{\text{reg}}(\cdot)$ is a regularization function, and μ is a small positive parameter that controls the trade-off between regularization and fitting. The above analysis assumes that the relationship between $\boldsymbol{m}_{\lambda_\ell}$ and r_ℓ is dominated by a linear function. There are however many situations, involving multiple scattering effects, in which model (4.1) may be inappropriate and could be advantageously replaced by a nonlinear one.

1. For ease of notation, these two constraints will be denoted by $\boldsymbol{\alpha} \succeq \mathbf{0}$ and $\mathbf{1}^\top\boldsymbol{\alpha} = 1$, where $\mathbf{1}$ is a vector of ones.

Consider the general mixing mechanism

$$\boldsymbol{r} = \boldsymbol{\Psi}(\boldsymbol{M}) + \boldsymbol{n} \tag{4.4}$$

with $\boldsymbol{\Psi}$ an unknown nonlinear function that defines the interactions between the endmembers in matrix $\boldsymbol{M}$. This requires us to consider a more general problem of the form

$$\psi^* = \arg\min_{\psi \in \mathcal{H}} \frac{1}{2}\|\psi\|_{\mathcal{H}}^2 + \frac{1}{2\mu}\sum_{\ell=1}^{L}(r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}))^2 \tag{4.5}$$

with $\mathcal{H}$ a given functional space, and μ a positive parameter that controls the trade-off between regularity of the function ψ and fitting. Clearly, this basic strategy may fail if the functional ψ of $\mathcal{H}$ cannot be adequately and finitely parameterized. Kernel-based methods rely on mapping data from the original input space into a feature space by means of a nonlinear function, and then solving a linear problem in that new space. They lead to efficient and accurate resolution of the inverse problem (4.5), as it has been shown in the literature. See, e.g., [Schölkopf 1999, Vapnik 1995]. We exploit the central idea of this research area, known as the *kernel trick*, to investigate new nonlinear unmixing algorithms. We shall now review the main definitions and properties related to reproducing kernel Hilbert spaces [Aronszajn 1950] and Mercer kernels [Mercer 1909].

Let $\mathcal{H}$ denote a Hilbert space of real-valued functions ψ on a compact $\mathcal{M}$, and let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ be the inner product in the space $\mathcal{H}$. Suppose that the evaluation functional δ_{m_λ} defined by $\delta_{m_\lambda}[\psi] = \psi(\boldsymbol{m}_\lambda)$ is linear with respect to ψ and bounded, for all $\boldsymbol{m}_\lambda$ in $\mathcal{M}$. By virtue of the Riesz representation theorem, there exists a unique positive definite function $\boldsymbol{m}_\lambda \mapsto \kappa(\boldsymbol{m}_\lambda, \boldsymbol{m}_{\lambda_j})$ in $\mathcal{H}$, denoted by $\kappa(\cdot, \boldsymbol{m}_{\lambda_j})$ and called *representer of evaluation* at $\boldsymbol{m}_{\lambda_j}$, which satisfies [Aronszajn 1950]

$$\psi(\boldsymbol{m}_{\lambda_j}) = \langle \psi, \kappa(\cdot, \boldsymbol{m}_{\lambda_j}) \rangle_{\mathcal{H}}, \quad \forall \psi \in \mathcal{H} \tag{4.6}$$

for every fixed $\boldsymbol{m}_{\lambda_j} \in \mathcal{M}$. A proof of this may be found in [Aronszajn 1950]. Replacing ψ by $\kappa(\cdot, \boldsymbol{m}_{\lambda_i})$ in (4.6) yields

$$\kappa(\boldsymbol{m}_{\lambda_i}, \boldsymbol{m}_{\lambda_j}) = \langle \kappa(\cdot, \boldsymbol{m}_{\lambda_i}), \kappa(\cdot, \boldsymbol{m}_{\lambda_j}) \rangle_{\mathcal{H}} \tag{4.7}$$

for all $\boldsymbol{m}_{\lambda_i}, \boldsymbol{m}_{\lambda_j} \in \mathcal{M}$. Equation (4.7) is the origin of the generic term *reproducing kernel* to refer to κ. Denoting by $\boldsymbol{\Phi}$ the map that assigns the kernel function $\kappa(\cdot, \boldsymbol{m}_{\lambda_j})$ to each input data $\boldsymbol{m}_{\lambda_j}$, equation (4.7) immediately implies that $\kappa(\boldsymbol{m}_{\lambda_i}, \boldsymbol{m}_{\lambda_j}) = \langle \boldsymbol{\Phi}(\boldsymbol{m}_{\lambda_i}), \boldsymbol{\Phi}(\boldsymbol{m}_{\lambda_j}) \rangle_{\mathcal{H}}$. The kernel thus evaluates the inner product of any pair of elements of $\mathcal{M}$ mapped to the space $\mathcal{H}$ without any explicit knowledge of $\boldsymbol{\Phi}$ and $\mathcal{H}$. Within the machine learning area, this key idea is known as the *kernel trick*.

The kernel trick has been widely used to transform linear algorithms expressed only in terms of inner products into nonlinear ones. Considering again (4.5), the optimum function ψ^* can be obtained by solving the following least squares support

vector machines (LS-SVM) problem [Suykens 2002]

$$\begin{aligned} \psi^* = \arg\min_{\psi\in\mathcal{H}} \frac{1}{2}\|\psi\|_{\mathcal{H}}^2 + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\ \text{subject to} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}), \quad \ell \in \{1,\dots,L\} \end{aligned} \tag{4.8}$$

We introduce the Lagrange multipliers β_ℓ, and consider the Lagrange function

$$\mathcal{L} = \frac{1}{2}\|\psi\|_{\mathcal{H}}^2 + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 - \sum_{\ell=1}^{L} \beta_\ell \left(e_\ell - r_\ell + \psi(\boldsymbol{m}_{\lambda_\ell})\right). \tag{4.9}$$

The conditions for optimality with respect to the primal variables are given by

$$\begin{cases} \psi^* = \sum_{\ell=1}^{L} \beta_\ell^* \, \kappa(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_\ell^* = \mu \, \beta_\ell^* \end{cases} \tag{4.10}$$

We then derive the dual optimization problem

$$\boldsymbol{\beta}^* = \arg\max_{\boldsymbol{\beta}} -\frac{1}{2}\,\boldsymbol{\beta}^\top (\boldsymbol{K} + \mu \boldsymbol{I})\,\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{r}, \tag{4.11}$$

where $\boldsymbol{K}$ is the so-called Gram matrix whose (ℓ, p)-th entry is defined by $\kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p})$. Classic examples of kernels are the radially Gaussian kernel

$$\kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \exp\left(-\|\boldsymbol{m}_{\lambda_\ell} - \boldsymbol{m}_{\lambda_p}\|^2 / 2\sigma^2\right)$$

and the Laplacian kernel

$$\kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \exp\left(-\|\boldsymbol{m}_{\lambda_\ell} - \boldsymbol{m}_{\lambda_p}\| / \sigma\right)$$

with $\sigma \geq 0$ the kernel bandwidth. Another example of interest is the q-th degree polynomial kernel

$$\kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = (1 + \boldsymbol{m}_{\lambda_\ell}^\top \boldsymbol{m}_{\lambda_p})^q$$

with $q \in \mathbb{N}^*$.

The kernel function κ maps $\boldsymbol{m}_{\lambda_\ell}$ into a very high, even infinite, dimensional space $\mathcal{H}$ without any explicit knowledge of the associated nonlinear function. The vector $\boldsymbol{\beta}^*$ and κ then describe the relation between the endmembers and the observation. The goal of the analysis is however to estimate the abundance vector, and there is no direct relation between $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ in the general case. In what follows, we shall focus attention on the design of specific kernels that enable us to determine abundance fractions within this context.

4.3 Kernel design and unmixing algorithms

The aim of this section is to propose kernel design methods and the corresponding algorithms to estimate abundances. The two approaches described hereafter are flexible enough to capture wide classes of nonlinear relationships, and to reliably interpret a variety of experimental measurements. Both have clear interpretation.

4.3.1 A preliminary approach for kernel-based hyperspectral unmixing : the K-Hype algorithm

In order to extract the mixing ratios of the endmembers, we define the function ψ in (4.5) by a linear trend parameterized by the abundance vector $\boldsymbol{\alpha}$, combined with a nonlinear fluctuation term, namely,

$$\begin{aligned} \psi(\boldsymbol{m}_{\lambda_\ell}) &= \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} + \psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}) \\ &\text{subject to} \quad \boldsymbol{\alpha} \succeq \mathbf{0} \quad \text{and} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1 \end{aligned} \tag{4.12}$$

where ψ_{nlin} can be any real-valued functions on a compact $\mathcal{M}$, of a reproducing kernel Hilbert space denoted by $\mathcal{H}_{\text{nlin}}$. Let κ_{nlin} be its reproducing kernel. It can be shown [Haussler 1999] that, as the direct sum $\mathcal{H}_{\text{lin}} \oplus \mathcal{H}_{\text{nlin}}$ of the RKHS of kernels $\kappa_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \boldsymbol{m}_{\lambda_\ell}^\top \boldsymbol{m}_{\lambda_p}$ and $\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p})$ defined on $\mathcal{M}$, the space $\mathcal{H}$ of functions of the form (4.12) is also a RKHS with kernel function

$$\begin{aligned} \kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) &= (\kappa_{\text{lin}} \oplus \kappa_{\text{nlin}})(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) \\ &= \boldsymbol{m}_{\lambda_\ell}^\top \boldsymbol{m}_{\lambda_p} + \kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}). \end{aligned} \tag{4.13}$$

The corresponding Gram matrix $\boldsymbol{K}$ is given by

$$\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top + \boldsymbol{K}_{\text{nlin}} \tag{4.14}$$

where $\boldsymbol{K}_{\text{nlin}}$ is the Gram matrix associated with the nonlinear map ψ_{nlin}, with (ℓ, p)-th entry $\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p})$.

We propose to conduct hyperspectral data unmixing by solving the following convex optimization problem

$$\begin{aligned} \psi^* = \arg\min_{\psi} \frac{1}{2} \left(\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} + \|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}} \right) + \frac{1}{2\mu} \sum_{\ell=1}^{L} e_\ell^2 \\ \text{where} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \quad \text{with} \quad \psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} \\ \text{subject to} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \\ \boldsymbol{\alpha} \succeq \mathbf{0} \quad \text{and} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1 \end{aligned} \tag{4.15}$$

By the strong duality property, we can derive a dual problem that has the same solution as the above primal problem. Let us introduce the Lagrange multipliers β_ℓ, γ_r and λ. The Lagrange function associated with the problem (4.15) can be written as

$$\begin{aligned} G = &\frac{1}{2} \left(\|\boldsymbol{\alpha}\|^2 + \|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}} \right) + \frac{1}{2\mu} \sum_{\ell=1}^{L} e_\ell^2 - \sum_{\ell=1}^{L} \beta_\ell \left(e_\ell - r_\ell + \psi(\boldsymbol{m}_{\lambda_\ell}) \right) \\ &- \sum_{r=1}^{R} \gamma_r \, \alpha_r + \lambda(\mathbf{1}^\top \boldsymbol{\alpha} - 1) \end{aligned} \tag{4.16}$$

with $\gamma_r \geq 0$. We have used that $\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} = \|\boldsymbol{\alpha}\|^2$ because the functional space $\mathcal{H}_{\text{lin}}$, parametrized by $\boldsymbol{\alpha}$, contains all the function of the variable $\boldsymbol{m}_{\lambda_\ell}$ of the form $\psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell}$. It is characterized by the norm

$$\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} = \langle \kappa_{\text{lin}}(\boldsymbol{\alpha}, \cdot), \kappa_{\text{lin}}(\boldsymbol{\alpha}, \cdot)\rangle_{\mathcal{H}} = \kappa_{\text{lin}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|^2. \tag{4.17}$$

The conditions for optimality of G with respect to the primal variables are given by

$$\begin{cases} \boldsymbol{\alpha}^* = \sum_{\ell=1}^{L} \beta_\ell^* \, \boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}^* - \lambda^* \mathbf{1} \\ \psi^*_{\text{nlin}} = \sum_{\ell=1}^{L} \beta_\ell^* \, \kappa_{\text{nlin}}(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_\ell^* = \mu \, \beta_\ell^* \end{cases} \tag{4.18}$$

By substituting (4.18) into (4.16), we get the following dual problem

$$\begin{aligned} \max_{\boldsymbol{\beta},\boldsymbol{\gamma},\lambda} G'(\boldsymbol{\beta},\boldsymbol{\gamma},\lambda) = -\frac{1}{2} \left(\begin{array}{c} \boldsymbol{\beta} \\ \hline \boldsymbol{\gamma} \\ \hline \lambda \end{array}\right)^\top \left(\begin{array}{c|c|c} \boldsymbol{K} + \mu \boldsymbol{I} & \boldsymbol{M} & -\boldsymbol{M}\mathbf{1} \\ \hline \boldsymbol{M}^\top & \boldsymbol{I} & -\mathbf{1} \\ \hline -\mathbf{1}^\top \boldsymbol{M}^\top & -\mathbf{1}^\top & R \end{array}\right) \left(\begin{array}{c} \boldsymbol{\beta} \\ \hline \boldsymbol{\gamma} \\ \hline \lambda \end{array}\right) + \left(\begin{array}{c} \boldsymbol{r} \\ \hline \mathbf{0} \\ \hline -1 \end{array}\right)^\top \left(\begin{array}{c} \boldsymbol{\beta} \\ \hline \boldsymbol{\gamma} \\ \hline \lambda \end{array}\right) \\ \text{subject to} \quad \boldsymbol{\gamma} \succeq \mathbf{0} \end{aligned} \tag{4.19}$$

with $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top + \boldsymbol{K}_{\text{nlin}}$. Provided that the coefficient vector $\boldsymbol{\beta}^*$ has been determined, the measured pixel can be reconstructed using

$$\begin{aligned} \boldsymbol{r}^* &= [\psi^*(\boldsymbol{m}_{\lambda_1}), \ldots, \psi^*(\boldsymbol{m}_{\lambda_L})]^\top \\ &= \boldsymbol{M}(\boldsymbol{M}^\top \boldsymbol{\beta}^* + \boldsymbol{\gamma}^* - \lambda^* \mathbf{1}) + \boldsymbol{K}_{\text{nlin}} \, \boldsymbol{\beta}^* \end{aligned} \tag{4.20}$$

as indicated by (4.18). Comparing the above expression with (4.12), we observe that the first and the second term of the r.h.s. of equation (4.20) correspond to the linear trend and the nonlinear fluctuations, respectively. Finally, the abundance vector $\boldsymbol{\alpha}^*$ can be estimated as follows

$$\boldsymbol{\alpha}^* = \boldsymbol{M}^\top \boldsymbol{\beta}^* + \boldsymbol{\gamma}^* - \lambda^* \mathbf{1} \tag{4.21}$$

Problem (4.19) is a quadratic program (QP). Numerous candidate methods exist to solve it, such as interior point, active set and projected gradient, as presented in [Bertsekas 1999, Luenberger 2008]. These well known numerical procedures lie beyond the scope of this work.

4.3.2 Some remarks on kernel selection

Selecting an appropriate kernel is of primary importance as it captures the nonlinearity of the mixture model. Though an infinite variety of possible kernels exists, it is always desirable to select a kernel that is closely related to the application context. The following example justifies the combination (4.12), which associates a linear model with a nonlinear fluctuation term. It also allows us to define a possible family of appropriate kernels for data unmixing.

Consider the generalized bilinear mixing model presented in [Halimi 2011], at first, limited to three endmember spectra for the sake of clarity

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{M}\,\boldsymbol{\alpha} + \gamma_{12}\,\alpha_1\,\alpha_2\,(\boldsymbol{m}_1 \otimes \boldsymbol{m}_2) + \gamma_{13}\,\alpha_1\,\alpha_3\,(\boldsymbol{m}_1 \otimes \boldsymbol{m}_3) + \gamma_{23}\,\alpha_2\,\alpha_3\,(\boldsymbol{m}_2 \otimes \boldsymbol{m}_3) + \boldsymbol{n} \\ &\text{with} \quad \alpha_1, \alpha_2, \alpha_3 \geq 0 \\ &\qquad \sum_{i=1}^{3} \alpha_i = 1 \end{aligned} \tag{4.22}$$

where γ_{12}, γ_{13} and γ_{23} are attenuation parameters, and $\otimes$ the Hadamard product. It can be observed that the nonlinear term with respect to $\boldsymbol{\alpha}$, in the r.h.s. of (4.22), is closely related to the homogeneous polynomial kernel of degree 2, that is, $\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = (\boldsymbol{m}_{\lambda_\ell}^\top\, \boldsymbol{m}_{\lambda_p})^2$. Indeed, with a slight abuse of notation, the latter can be written in an inner product form as follows

$$\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \boldsymbol{\Phi}_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell})^\top \boldsymbol{\Phi}_{\text{nlin}}(\boldsymbol{m}_{\lambda_p}) \tag{4.23}$$

with

$$\boldsymbol{\Phi}_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}) = (m_{\lambda_{\ell,1}}^2,\, m_{\lambda_{\ell,2}}^2,\, m_{\lambda_{\ell,3}}^2,\, \sqrt{2} m_{\lambda_{\ell,1}} m_{\lambda_{v,2}},\, \sqrt{2} m_{\lambda_{\ell,1}} m_{\lambda_{\ell,3}},\, \sqrt{2} m_{\lambda_{\ell,2}} m_{\lambda_{\ell,3}})^\top \tag{4.24}$$

where $m_{\lambda_{\ell,i}}$ is the i-th entry of $\boldsymbol{m}_{\lambda_\ell}$. This means that, in addition to the linear mixture term $\boldsymbol{M}\boldsymbol{\alpha}$, the auto and interaction terms considered by the kernel-based model are of the form $\boldsymbol{m}_i \otimes \boldsymbol{m}_j$ for all $i, j = 1, \ldots, R$.

By virtue of the reproducing kernel machinery, endmember spectra do not need to be explicitly mapped into the feature space. This allows to consider complex interaction mechanisms by changing the kernel κ_{nlin}, without having to modify the optimization algorithm described in the previous subsection. As an illustration, consider the polynomial kernel $\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = (1 + \boldsymbol{m}_{\lambda_\ell}^\top\, \boldsymbol{m}_{\lambda_p})^q$. Making use of the binomial theorem yields

$$\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \sum_{k=0}^{q} \binom{q}{k} (\boldsymbol{m}_{\lambda_\ell}^\top\, \boldsymbol{m}_{\lambda_p})^k. \tag{4.25}$$

We observe that each component $(\boldsymbol{m}_{\lambda_\ell}^\top\, \boldsymbol{m}_{\lambda_p})^k = (m_{\lambda_{\ell,1}}\, m_{\lambda_{p,1}} + \ldots + m_{\lambda_{\ell,R}}\, m_{\lambda_{p,R}})^k$ of the above expression can be expanded into a weighted sum of k-th degree monomials of the form

$$(m_{\lambda_{\ell,1}}\, m_{\lambda_{p,1}})^{k_1} (m_{\lambda_{\ell,2}}\, m_{\lambda_{p,2}})^{k_2} \ \ldots\ (m_{\lambda_{\ell,R}}\, m_{\lambda_{p,R}})^{k_R} \tag{4.26}$$

with $\sum_{r=1}^{R} k_r = k$. This means that, in addition to the linear mixture term $\boldsymbol{M}\boldsymbol{\alpha}$, the auto and interaction terms considered by the kernel-based model are of the form $\boldsymbol{m}_1^{k_1} \otimes \boldsymbol{m}_2^{k_2} \otimes \ldots \otimes \boldsymbol{m}_R^{k_R}$ for every set of exponents in the Hadamard sense satisfying $0 \leq \sum_{r=1}^{R} k_r \leq q$. Note that it would be computationally expensive to explicitly form these interaction terms. Their number is indeed very large : there are p^k monomials (4.26) of degree k, and then $\frac{1-R^{q+1}}{1-R}$ components in the entire q-th order representation. Compared with the methods introduced in

[Raksuntorn 2010, Nascimento 2009], which insert products of pure material signatures as new endmembers, we do not need to extend the endmember matrix by adding such terms. The kernel trick makes the computation much more tractable. In the experimentations reported hereafter, the following 2-nd degree polynomial kernel was used

$$\kappa_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p}) = \left(1 + \frac{1}{R^2}(\boldsymbol{m}_{\lambda_\ell} - 1/2)^\top (\boldsymbol{m}_{\lambda_p} - 1/2)\right)^2 \tag{4.27}$$

where the constants $1/R^2$ and $1/2$ serve the purpose of normalization.

4.3.3 Nonlinear unmixing by multiple kernel learning : the SK-Hype algorithm

The proposed model relies on the assumption that the mixing mechanism can be described by a linear mixture of endmember spectra, with additive nonlinear fluctuations ψ_{nlin} defined in a RKHS. This justifies the use of a Gram matrix of the form $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top + \boldsymbol{K}_{\text{nlin}}$ in the algorithm presented previously. Model (4.12) however has some limitations in that the balance between the linear component $\boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell}$ and the nonlinear component $\psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell})$ cannot be tuned. This should however be made possible as recommended by physically-inspired models such as model (4.22). In addition, kernels κ_{nlin} with embedded linear component such as the inhomogeneous polynomial kernel (4.25) introduces a bias into the estimation of $\boldsymbol{\alpha}$, unless correctly estimated and removed. Another difficulty is that the model (4.12) cannot capture the dynamic of the mixture, which requires that $\boldsymbol{r}$ or the $\boldsymbol{m}_{\lambda_\ell}$'s be locally normalized. This unlikely situation occurs, e.g., if a library of reflectance signatures is used for the unmixing process. To address problems such as the above, it should be interesting to consider Gram matrices of the form

$$\boldsymbol{K}_u = u\boldsymbol{M}\boldsymbol{M}^\top + (1-u)\boldsymbol{K}_{\text{nlin}} \tag{4.28}$$

with u in $[0, 1]$ in order to ensure positiveness of $\boldsymbol{K}_u$. The intuition for equation (4.28) is as follows. The performance of kernel-based methods, such as SVM, Gaussian Processes, etc., strongly relies on kernel selection. There is a large body of literature addressing various aspects of this problem, including the use of training data to select or combine the most suitable kernels out of a specified family of kernels. The great majority of theoretical and algorithmic results focus on learning convex combinations of kernels as originally considered by [Lanckriet 2004]. Learning both the parameter u and the mixing coefficients $\boldsymbol{\beta}$ in a single optimization problem is known as the multiple kernel learning problem. See [Rakotomamonjy 2008] and references therein. The rest of this section is devoted to the formalization of this intuition, which will lead us to formulate and solve a convex optimization problem.

4.3.3.1 Primal problem formulation

In order to tune the balance between ψ_{lin} and ψ_{nlin}, it might seem tempting to substitute matrix $\boldsymbol{K}$ with $\boldsymbol{K}_u$ in the dual problem (4.19). Unfortunately, a primal

problem must be first formulated in order to identify, in the spirit of equation (4.18), explicit expressions for ψ_{lin} and ψ_{nlin}. We propose to conduct hyperspectral data unmixing by solving the following primal problem

$$\begin{aligned} \psi^*, u^* = \arg\min_{\psi,u} \frac{1}{2}\left(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}'_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}'_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\ \text{subject to} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \quad \text{and} \quad 0 \leq u \leq 1 \\ \text{where} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \end{aligned} \tag{4.29}$$

where u allows to adjust the balance between ψ_{lin} and ψ_{nlin} via their norms. The spaces $\mathcal{H}'_{\text{lin}}$ and $\mathcal{H}'_{\text{nlin}}$ are RKHS of the general form

$$\mathcal{H}'_{\text{lin/nlin}} = \left\{ \psi \in \mathcal{H}_{\text{lin/nlin}} : \frac{\|\psi\|_{\mathcal{H}_{\text{lin/nlin}}}}{u} < \infty \right\} \tag{4.30}$$

with the convention $\frac{x}{0} = 0$ if $x = 0$, and ∞ otherwise. This implies that, if $u = 0$, then ψ belongs to space $\mathcal{H}'_{\text{lin/nlin}}$ if and only if $\psi = 0$. By continuity consideration via this convention, it can be shown that the problem (4.29) is a convex optimization problem by virtue of the convexity of the so-called perspective function

$$f(u, \psi) = \frac{\|\psi\|^2_{\mathcal{H}'_{\text{lin/nlin}}}}{u}$$

over $\mathbb{R}_+ \times \mathcal{H}'_{\text{lin/nlin}}$. This has been shown in [Boyd 2004, Chapter 3] in the finite-dimensional case, and extended in [Rakotomamonjy 2008] to the infinite-dimensional case. This allows to formulate the two-stage optimization procedure, with respect to ψ and u successively, in order to solve problem (4.29).

$$\min_u J(u) \qquad \text{subject to} \qquad 0 \leq u \leq 1 \tag{4.31}$$

where

$$\begin{aligned} J(u) = \Bigg\{ & \min_{\psi} F(u, \psi) = \frac{1}{2}\left(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}'_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}'_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\ & \text{subject to} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \quad \text{with} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \\ & \text{and} \quad \psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{h}^\top \boldsymbol{m}_{\lambda_\ell} \quad \text{with} \quad \boldsymbol{h} \succeq \boldsymbol{0} \end{aligned} \tag{4.32}$$

The connection between (4.29) and this problem is as follows. We have [Boyd 2004, p. 133]

$$\min_{u,\psi} F(u, \psi) = \min_u J(u) \tag{4.33}$$

where $J(u) = \min_\psi F(u, \psi)$, subject to all the constraints over u and ψ defined in (4.31)-(4.32). In addition, as proven in textbooks [Boyd 2004, p. 87], because F is convex in (u, ψ) subject to convex constraints over ψ, it turns out that $J(u)$ is

convex in u and, as a consequence, that the constrained optimization problem (4.31) is convex.

Compared to the preliminary algorithm described in Section 4.3.1, it is important to note that the sum-to-one constraint $\mathbf{1}^\top \boldsymbol{h} = 1$ has been given up. Indeed, relaxing the ℓ_1-norm of the weight vector $\boldsymbol{h}$ acts as an additional degree of freedom for the minimization of the regularized reconstruction error $F(u, \psi)$. This mechanism operates in conjunction with parameter u setting, which is adjusted to achieve the best balance between ψ_{lin} and ψ_{nlin}. The effectiveness of this strategy has been confirmed by experiments, which have revealed a significant improvement in performance. Note that the resulting vector $\boldsymbol{h}$ cannot be directly interpreted as a vector of fractional abundances. It is normalized afterwards by writing $\boldsymbol{h} = \theta\,\boldsymbol{\alpha}$, with $\boldsymbol{\alpha}$ the vector of fractional abundances, and $\theta = \mathbf{1}^\top \boldsymbol{h}$. The reader would certainly have been more pleased if the scaling factor θ had been explicitly included in the optimization process as follows

$$\begin{aligned}
\psi^*, \theta^*, u^* = \underset{\psi,\theta,u}{\arg\min}\; & \frac{1}{2}\left(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}'_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}'_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\
& \text{subject to} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \quad \text{and} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \\
& \text{where} \quad \psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \theta\,\boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} \\
& \qquad\quad \boldsymbol{\alpha} \succeq \mathbf{0} \qquad \mathbf{1}^\top\boldsymbol{\alpha} = 1 \qquad \theta \in \mathbb{R}_+^* \qquad 0 \le u \le 1
\end{aligned} \tag{4.34}$$

This problem is not convex, and is difficult to solve as formulated. Fortunately, as indicated hereafter, it is equivalent to the problem (4.31)-(4.32) in the sense of [Boyd 2004, p. 130]. Consider the change of variable $\boldsymbol{h} = \theta\,\boldsymbol{\alpha}$. The cost function can be directly reformulated as a function of $\boldsymbol{h}$. The two constraints over $\boldsymbol{\alpha}$ become

$$(\boldsymbol{h} \succeq \mathbf{0}) \qquad \text{and} \qquad (\mathbf{1}^\top\boldsymbol{h} = \theta \quad \text{with} \quad \theta \in \mathbb{R}_+^*)$$

Eliminating the second constraint, which is trivial because of the first constraint, leads us to the problem (4.31)-(4.32). Because $\boldsymbol{h} = \theta\,\boldsymbol{\alpha}$ and $\mathbf{1}^\top\boldsymbol{\alpha} = 1$, we have $\boldsymbol{\alpha}^* = \frac{\boldsymbol{h}^*}{\theta^*}$ with $\theta^* = \mathbf{1}^\top\boldsymbol{h}^*$. This result is consistent with the normalization of $\boldsymbol{h}$ proposed above.

4.3.3.2 Dual problem formulation and algorithm

By the strong duality property, we shall now derive a dual problem that has the same solution $J(u) = F(u, \psi^*)$ as the primal problem (4.32). Let us introduce the Lagrange multipliers β_ℓ and γ_r. The Lagrange function associated with the problem (4.32) can be written as

$$G = \frac{1}{2}\left(\frac{1}{u}\|\boldsymbol{h}\|^2 + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 - \sum_{\ell=1}^{L}\beta_\ell\,(e_\ell - r_\ell + \psi(\boldsymbol{m}_{\lambda_\ell})) - \sum_{r=1}^{R}\gamma_r\,h_r \tag{4.35}$$

with $\gamma_r \geq 0$, where we have used that $\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} = \|\boldsymbol{h}\|^2$. The conditions for optimality of G with respect to the primal variables are given by

$$\begin{cases} \boldsymbol{h}^* = u\left(\sum_{\ell=1}^{L} \beta_\ell^* \, \boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}^*\right) \\ \psi^*_{\text{nlin}} = (1-u)\sum_{\ell=1}^{L} \beta_\ell^* \, \kappa_{\text{nlin}}(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_\ell^* = \mu \, \beta_\ell^* \end{cases} \tag{4.36}$$

By substituting (4.36) into (4.35), we get the following dual problem

$$J(u) = \left\{ \begin{array}{l} \max\limits_{\boldsymbol{\beta},\boldsymbol{\gamma}} G'(u,\boldsymbol{\beta},\boldsymbol{\gamma}) = -\frac{1}{2}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix}^\top \left(\begin{array}{c|c} \boldsymbol{K}_u + \mu\boldsymbol{I} & u\boldsymbol{M} \\ \hline u\boldsymbol{M}^\top & u\boldsymbol{I}\end{array}\right)\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix} + \begin{pmatrix}\boldsymbol{r}\\ \boldsymbol{0}\end{pmatrix}^\top \begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix} \\ \qquad\qquad \text{subject to} \quad \boldsymbol{\gamma} \succeq \boldsymbol{0} \end{array} \right. \tag{4.37}$$

with $\boldsymbol{K}_u = u\boldsymbol{M}\boldsymbol{M}^\top + (1-u)\boldsymbol{K}_{\text{nlin}}$. Pixel reconstruction can be performed using $\boldsymbol{r}^* = [\psi^*(\boldsymbol{m}_{\lambda_1}), \ldots, \psi^*(\boldsymbol{m}_{\lambda_L})]^\top$ with $\psi^*(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{m}_{\lambda_\ell}^\top \boldsymbol{h}^* + \psi^*_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell})$ defined in equation (4.36). Finally, the estimated abundance vector is given by

$$\boldsymbol{\alpha}^* = \frac{\boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^*}{\boldsymbol{1}^\top(\boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^*)} \tag{4.38}$$

4.3.3.3 Solving with respect to u

Update with gradient : Let us briefly address the differentiability issue of the problem (4.31)-(4.37). The existence and computation of the derivatives of supremum functions such as $J(u)$ have been largely discussed in the literature. As pointed out in [Rakotomamonjy 2008, Bonnans 1998], the differentiability of J at any point u_0 is ensured by the unicity of the corresponding minimizer $(\boldsymbol{\beta}_0^*, \boldsymbol{\gamma}_0^*)$, and by the differentiability of the cost function $F(u,\psi)$ in (4.32). The derivative of J at u_0 can be calculated as if the minimizer $(\boldsymbol{\beta}_0^*, \boldsymbol{\gamma}_0^*)$ was independent of u_0, namely, $\frac{dJ(u)}{du}\big|_{u=u_0} = \frac{\partial G'(u,\boldsymbol{\beta}_0^*,\boldsymbol{\gamma}_0^*)}{\partial u}\big|_{u=u_0}$. This yields

$$\left.\frac{dJ(u)}{du}\right|_{u=u_0} = -\frac{1}{2}\left(\|\boldsymbol{M}^\top\boldsymbol{\beta}_0^* + \boldsymbol{\gamma}_0^*\|^2 - \boldsymbol{\beta}_0^{*\top}\boldsymbol{K}_{\text{nlin}}\,\boldsymbol{\beta}_0^*\right) \tag{4.39}$$

Table 4.1 summarizes the proposed algorithm. Note that (4.31) is a very small-size problem. Indeed, it involves a one-dimension optimization variable and can thus be solved with an ad-hoc procedure. Using a gradient projection method, e.g., based on Armijo rule along the feasible direction, makes practical sense in this case [Bertsekas 1999, Chapter 2]. Moreover, both problems can benefit of warm-starting between successive solutions to speed-up the optimization procedure. The algorithm can be stopped based on conditions for optimality in convex optimization framework. In particular, the KKT conditions and the duality gap should be equal to zero, within a numerical error tolerance specified by the user. The variation of the cost $J(u)$ between two successive iterations should also be considered as a potential stopping criterion.

TABLE 4.1 – Summary of SK-Hype algorithm

Initialization

Choose the kernel κ_{nlin} and the regularization constant $\mu \geq 0$.

Calculate the kernel matrix $\boldsymbol{K}_{\text{nlin}}$.

Repeat

Calculate $J(u)$ in (4.37) by using a generic QP solver

Solve the optimization problem

$$\min_u J(u) \quad \text{subject to} \quad 0 \leq u \leq 1$$

by performing one iteration of the gradient projection algorithm, using (4.39) or (4.42).

until stopping criterion is satisfied

Estimate the abundances by (4.38)

Direct update : Instead of updating u with a gradient descent algorithm as in [Rakotomamonjy 2008], we found that u can be optimized directly for this specific problem.

Notice that the function

$$f_{p,q}(u) = \frac{p}{u} + \frac{q}{1-u} \quad \text{with } p \geq 0, q \geq 0 \tag{4.40}$$

is convex over the interval $u \in [0,1]$. The optimum value $f_{p,q}(u^*) = (\sqrt{p} + \sqrt{q})^2$ at the point $u^* = \frac{1}{1+\sqrt{q/p}}$. Now consider the problem (4.31), we have

$$u^* = \frac{1}{1 + \|\psi_{\text{nlin}}\|_{\mathcal{H}_{\text{nlin}}} / \|\psi_{\text{nlin}}\|_{\mathcal{H}_{\text{lin}}}} \tag{4.41}$$

Using the stationary conditions (4.36), at each step with respect to u, its optimum value is given by

$$u^* = \frac{1}{1 + (1-u^*_{-1})\sqrt{\frac{\boldsymbol{\beta}^{*\top}\boldsymbol{K}_{\text{nlin}}\boldsymbol{\beta}^*}{\boldsymbol{h}^{*\top}\boldsymbol{h}^*}}} \tag{4.42}$$

where u^*_{-1} denotes the optimal u used in the previous step when optimizing ψ.

Before testing our algorithms, and comparing their performance with state-of-the-art approaches, we shall now explain how they differ from existing kernel-based techniques for hyperspectral data processing.

4.3.4 Comparison with existing kernel-based methods in hyperspectral imagery

Some kernel-based methods have already been proposed to process hyperspectral images, with application to classification, supervised or unsupervised unmixing, etc.

By taking advantage of capturing nonlinear data dependences, some of them have been shown to achieve better performance than their linear counterpart. Let us now briefly discuss the main difference between our kernel-based model and those presently existing. The central idea underlying most of state-of-the-art methods is to nonlinearly transform hyperspectral pixel-vectors prior to applying a linear algorithm, simply by replacing inner products with kernels in the cost function. This basic principle is fully justified in detection/classification problems because a proper nonlinear distortion of spectral signatures can increase the detectability/separability of materials. Within the context of hyperspectral unmixing, this leads to consider mixtures of the form

$$\Phi(\boldsymbol{r}) = [\Phi(\boldsymbol{m}_1), \Phi(\boldsymbol{m}_2), \ldots, \Phi(\boldsymbol{m}_R)]\,\boldsymbol{\alpha} + \boldsymbol{n}. \tag{4.43}$$

This model is inherent in the KFCLS algorithm [Broadwater 2007, Broadwater 2009], which optimizes the following mean-square error criterion where all the inner products have been replaced by kernels

$$J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \boldsymbol{K}_{\boldsymbol{m}}\,\boldsymbol{\alpha} - 2\,\boldsymbol{\alpha}^\top \boldsymbol{\kappa}_{rm} + \kappa(\boldsymbol{r},\boldsymbol{r}), \tag{4.44}$$

where $\boldsymbol{K}_{\boldsymbol{m}}$ is the Gram matrix with (i,j)-th entry $\kappa(\boldsymbol{m}_i,\boldsymbol{m}_j)$, and $\boldsymbol{\kappa}_{rm}$ is a vector with i-th entry $\kappa(\boldsymbol{r},\boldsymbol{m}_i)$. Unfortunately, even though model (4.43) allows distortions of spectral signatures, it does not explicitly include nonlinear interactions of the endmember spectra. The analysis in Section 4.3.2 has shown strong connections between our kernel-based model and well-characterized models, e.g., the generalized bilinear mixture model. The experimental comparison on simulated and real data reported in the next section confirms this view.

4.4 Experimental results

We shall now conduct some simulations to validate the proposed unmixing algorithms, and to compare them with state-of-the-art methods, using both synthetic and real images.

4.4.1 Experiments on synthetic images

Let us first report some experimental results on synthetic images, generated by linear and nonlinear mixing of several endmember signatures. The materials we have considered are alunite, calcite, epidote, kaolinite, buddingtonite, almandine, jarosite and lepidolite. They were selected from the ENVI software library. These spectra consist of 420 contiguous bands, covering wavelengths ranging from 0.3951 to 2.56 micrometers.

In the first scene, only three materials were selected to generate images : epidote, kaolinite, buddingtonite. In the second scene, five materials were used : alunite, calcite, epidote, kaolinite, buddingtonite. In the third scene, the eight materials were used. For each scene, three 50-by-50 hyperspectral images were generated with

different mixture models, each providing $N = 2500$ pixels for evaluating and comparing the performance of several algorithms. These three models were the linear model (4.1), the bilinear mixture model defined as

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{\alpha} + \sum_{i=1}^{R-1} \sum_{j=i+1}^{R} \alpha_i \, \alpha_j \, (\boldsymbol{m}_i \otimes \boldsymbol{m}_j) + \boldsymbol{n}, \tag{4.45}$$

and a post-nonlinear mixing model (PNMM) [Jutten 2003] defined by

$$\boldsymbol{r} = (\boldsymbol{M}\boldsymbol{\alpha})^{\xi} + \boldsymbol{n} \tag{4.46}$$

where $(\,\cdot\,)^{\xi}$ denotes the exponential value ξ applied to each entry of the input vector. Parameter ξ was set to 0.7. The abundance vectors $\boldsymbol{\alpha}_n$, with $n = 1, \ldots, 2500$, were uniformly generated in the simplex defined by non-negative and sum-to-one constraints. Finally, all these images were corrupted with an additive white Gaussian noise $\boldsymbol{n}$ with two levels of SNR, 30 dB and of 15 dB.

The following algorithms were considered

- **The so-called Fully Constrained Least Square method (FCLS)**, [Heinz 2001] : This technique was derived based on linear mixture model. It provides the optimal solution in the least-mean-square sense, subject to non-negativity and sum-to-one constraints. A relaxation parameter ν has to be tuned to specify a compromise between the residual error and the sum-to-one constraint.
- **The extended endmember-matrix method (ExtM)**, [Raksuntorn 2010] : This method consists of extending the endmember matrix $\boldsymbol{M}$ artificially with cross-spectra of pure materials in order to model light scatter effects. In the experiments, all the second-order cross terms $\boldsymbol{m}_i \otimes \boldsymbol{m}_j$ were inserted so that it would correspond to the generalized bilinear model. This approach also has a relaxation parameter ν for the sum-to-one constraint.
- **The so-called Kernel Fully Constrained Least Square method (KFCLS)**, [Broadwater 2007] : This is a kernel method, directly derived from FCLS, in which all the inner products are replaced by kernel functions. As for all the other kernel-based algorithms considered in this paper, the Gaussian kernel was used for simulations. This algorithm has two parameters, the bandwidth σ of the Gaussian kernel, and a relaxation parameter ν for the sum-to-one constraint.
- **The Bayesian algorithm derived for generalized bilinear model (BilBay)**, [Halimi 2011] : This method is based on appropriate prior distributions for the unknown parameters, which must satisfy the non-negativity and sum-to-one constraints, and then derives joint posterior distribution of these parameters. A Metropolis-within-Gibbs algorithm is used to estimate the unknown model parameters. The MMSE estimates of the abundances were computed by averaging the 2500 generated samples obtained after 500 burn-in iterations.
- **The first algorithm proposed in this chapter (K-Hype)** : This is the preliminary algorithm described in Section 4.3.1. The Gaussian kernel (G) with

bandwidth σ, and the polynomial kernel (P) defined by (4.27) were considered. The Matlab optimization function Quadprog was used to solve the QP problem.

– **The second algorithm proposed in this chapter (SK-Hype)** : This is the main algorithm described in Section 4.3.3 and Table 4.1. As for K-Hype, the Gaussian kernel and the polynomial kernel were considered. In order to simplify the experiments, the weight vector $\boldsymbol{u}$ was constrained to be of the form $\boldsymbol{u} = [u, 1-u]^\top$ with $0 \leq u \leq 1$. Obviously, this allows to reduce the number of variables but does not change the structure of the algorithm. The variable u was initially set to $\frac{1}{2}$. A gradient projection method, based on the Armijo rule to compute the optimal step size along the feasible direction, was used to determined u. The algorithm was stopped when the relative variation of u between two successive iterations became less than $\zeta_{\max} = 10^{-3}$, or the maximum number of iterations $\mathrm{It}_{\max} = 10$ was reached. The Matlab optimization function Quadprog was used to solve the QP problem.

The root mean square error defined by

$$\mathrm{RMSE} = \sqrt{\frac{1}{NR}\sum_{n=1}^{N}\|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n^*\|^2} \tag{4.47}$$

was used to compare these six algorithms. In order to tune their parameters, preliminary runs were performed on 100 independent test pixels for each experiment. The bandwidth σ of the Gaussian kernel in the algorithms ExtM, K-Hype and SK-Hype was varied within $\{1, \ldots, 3\}$ with increment of $\frac{1}{2}$. The parameter μ of K-Hype and SK-Hype algorithms was varied within $\{1, 10^{-1}, 10^{-2}, 5 \cdot 10^{-3}\}$. The parameter ν in algorithms FCLS, ExtM, KFCLS was chosen within $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. Detailed parameter settings can be found in [Chen 2013].

Results for Scene 1 to Scene 3 unmixing, with three, five and eight endmember materials, are reported in Table 4.2, Table 4.3 and Table 4.4 respectively. Because the FCLS method was initially derived for the linear mixing model, it achieves a very low RMSE for linearly-mixed images, and produces a relatively large RMSE with nonlinearly-mixed images. With second-order cross terms that extend the endmember matrix $\boldsymbol{M}$, the ExtM algorithm notably reduces the RMSE when dealing with bilinearly-mixed images when compared with FCLS. However, it marginally improves the performance in PNMM image ummixing. BilBay algorithm was derived for the bilinear mixing model, and thus achieves very good performance with bilinearly-mixed images. Nevertheless, the performance of BilBay clearly degrades when dealing with a nonlinear mixing model for which it was not originally designed. KFCLS with Gaussian kernel performs worse than FCLS, even with nonlinearly-mixed images as it does not clearly investigate nonlinear interactions between materials.

For the less noisy scenes (30 dB), our algorithms K-Hype and SK-Hype exhibit significantly reduced RMSE when dealing with nonlinearly-mixed images. In the case of the bilinear model, K-Hype and SK-Hype achieve very good performance

TABLE 4.2 – Scene 1 (three materials) : RMSE comparison

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	$0.0037\pm2\times10^{-5}$	0.0758±0.0019	0.0604±0.0017
ExtM	0.0079±0.0001	0.0312±0.0013	0.0601±0.0016
KFCLS	$0.0054\pm3\times10^{-5}$	0.2711±0.0516	0.2371±0.0197
BilBay	0.0384±0.0013	0.0285±0.0006	0.1158±0.0058
K-Hype (G)	0.0208±0.0004	0.0349±0.0013	0.0446±0.0020
K-Hype (P)	0.0346±0.0011	0.0281±0.0011	0.0569±0.0031
SK-Hype (G)	0.0104±0.0001	0.0315±0.0012	0.0230±0.0007
SK-Hype (P)	0.0106±0.0002	0.0310±0.0011	0.0245±0.0007
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0212±0.0005	0.0960±0.0060	0.0886±0.0063
ExtM	0.0404±0.0031	0.0991±0.096	0.0869±0.0066
KFCLS	0.0296±0.0009	0.2694±0.0498	0.2372±0.0235
BilBay	0.1135±0.0098	0.1059±0.0085	0.1191±0.0091
K-Hype (G)	0.0562±0.0041	0.0611±0.0048	0.0786±0.0067
K-Hype (P)	0.0589±0.0041	0.0628±0.0053	0.0794±0.0066
SK-Hype (G)	0.0562±0.0044	0.0598±0.0048	0.0757±0.0073
SK-Hype (P)	0.0561±0.0043	0.0602±0.0048	0.0742±0.0075

compared to the other algorithms. Indeed, they are the best performers except in a few cases. In the case of the PNMM model, they outperform all the other algorithms, and it can be observed that SK-Hype outperforms K-Hype in several scenarios. For the noisiest scenes (15 dB), although the increase in the noise level significantly degrades the performance of all the algorithms, K-Hype and SK-Hype still maintain an advantage. Last but not least, the margin of performance over the other approaches becomes larger as the number of endmembers increases.

To give a more meaningful comparison of the performance of these algorithms, one-tailed Welch's t-tests with significance level 0.05 were used to test the hypothesis

$$\mathcal{H}_0 : \quad \mathrm{RMSE}_{\mathrm{proposed}} < \mathrm{RMSE}_{\mathrm{literature}}$$

where $\mathrm{RMSE}_{\mathrm{proposed}}$ denotes the RMSE of the K-Hype and SK-Hype algorithms, with Gaussian and polynomial kernels, and $\mathrm{RMSE}_{\mathrm{literature}}$ is the RMSE of the algorithms of the literature selected in this paper. The results for Scene 2 and the SNR level 30 dB are reported here, in Table 4.5 to 4.7. The letter $\mathcal{A}$ means that the hypothesis $\mathcal{H}_0$ is accepted. Without ambiguity, these results confirm the advantage of our algorithms.

The computational time of these algorithms mainly depends on the constrained

TABLE 4.3 – Scene 2 (five materials) : RMSE comparison

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	0.0134±0.0002	0.1137±0.0032	0.1428±0.0039
ExtM	0.0157±0.0003	0.0575±0.0024	0.1427±0.0040
KFCLS	0.0200±0.0004	0.2051±0.0148	0.1955±0.0115
BilBay	0.0585 ±0.0017	0.0441±0.0010	0.1741±0.0082
K-Hype (G)	0.0231±0.0004	0.0307±0.0008	0.0398±0.0012
K-Hype (P)	0.0218±0.0004	0.0465±0.0012	0.0386±0.0011
SK-Hype (G)	0.0196±0.0004	0.0288±0.0007	0.0346±0.0010
SK-Hype (P)	0.0195±0.0004	0.0349±0.0008	0.0346±0.0010
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0657±0.0047	0.1444±0.0116	0.1611±0.0134
ExtM	0.0761±0.0060	0.1207±0.0160	0.1678±0.0139
KFCLS	0.0890±0.0080	0.1884±0.0113	0.1572±0.0114
BilBay	0.1465±0.0109	0.1007±0.0063	0.1609±0.0124
K-Hype (G)	0.1076±0.0093	0.0748±0.0046	0.0823±0.0053
K-Hype (P)	0.0738±0.0043	0.0847±0.0052	0.0828±0.0054
SK-Hype (G)	0.0675±0.0040	0.0778±0.0043	0.0942±0.0065
SK-Hype (P)	0.0673±0.0040	0.0830±0.0046	0.0965±0.0071

optimization problem to be solved. FCLS and KFLCS minimize a quadratic cost function of dimension R, under inequality constraints of the same dimension. ExtM solves a similar problem but with an increased dimension due to the cross-spectra that are artificially inserted. In the case where only the second-order cross spectra are added, the dimension of the optimization problem is $R + \binom{R}{2}$ with $R = 3$, 5 and 8 in this study. BilBay has to generate numerous samples to estimate the model parameters, and suffers from the large computational cost of this sampling strategy. K-Hype solves a quadratic programming problem of dimension $L + R + 1$. It is interesting to note that the computational cost is independent of the complexity of the unmixing model. A sparsification strategy as described in [Richard 2009] should be advantageously used to greatly reduce the computational complexity with negligible effect on the quality of the results. SK-Hype has similar advantages as K-Hype except that the alternating optimization scheme requires more time [2]. The average computational times per pixel of all these algorithms are listed in Table 4.8. [3]

2. Time counted by gradient update of u. Time counted by direct update up u can be found in the next Chapter

3. Using Matlab R2008a on a iMac with 3.06GHz Intel Core 2 Duo and 4 Go Memory

TABLE 4.4 – Scene 3 (eight materials) : RMSE comparison

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	0.0148±0.0002	0.0930±0.0024	0.1079±0.0018
ExtM	0.0173±0.0003	0.0560±0.0017	0.1126±0.0019
KFCLS	0.0216±0.0004	0.1431±0.0059	0.1274±0.0039
BilBay	0.0448±0.0007	0.0369±0.0004	0.1159±0.0029
K-Hype (G)	0.0203±0.0003	0.0202±0.0003	0.0300±0.0006
K-Hype (P)	0.0195±0.0003	0.0330±0.0006	0.0297±0.0006
SK-Hype (G)	0.0185±0.0003	0.0221±0.0003	0.0291±0.0006
SK-Hype (P)	0.0184±0.0002	0.0247±0.0004	0.0313±0.0007
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0652±0.0031	0.1177±0.0068	0.1252±0.0065
ExtM	0.0743±0.0038	0.1066±0.0062	0.1322±0.0063
KFCLS	0.0647±0.0032	0.1270±0.0038	0.2250±0.0220
BilBay	0.0745±0.0020	0.0792±0.0026	0.1040±0.0430
K-Hype (G)	0.0562±0.0020	0.0548±0.0018	0.0642±0.0024
K-Hype (P)	0.0585±0.0021	0.0646±0.0024	0.0657±0.0026
SK-Hype (G)	0.0561±0.0019	0.0573±0.0020	0.0696±0.0027
SK-Hype (P)	0.0571±0.0021	0.0620±0.0021	0.0736±0.0031

4.4.2 Experiment with AVIRIS image

This section illustrates the performance of the proposed algorithms, and several other algorithms, when applied to real hyperspectral data. The scene that was used for our experiment is the well-known image captured on the Cuprite mining district (NV, USA) by AVIRIS. A sub-image of 250×191 pixels was chosen to evaluate the algorithms. This area of interest has $L = 188$ spectral bands. The number of endmembers was first estimated via the virtual dimensionality, and R was accordingly set to 12 [Nascimento 2005]. VCA algorithm was then used to extract the endmembers. Both our algorithms were compared with all the state-of-the-art algorithms considered previously. After preliminary experiments, the regularization parameters of FCLS and ExtM algorithms were set to $\nu = 0.01$. K-Hype algorithm and SK-Hype algorithm were run with the polynomial kernel (4.27), and the Gaussian kernel. The bandwidth of the Gaussian kernel was set to $\sigma = 2$. The regularization parameter μ was fixed to $2 \cdot 10^{-3}$. To evaluate the performance, the averaged spectral angle

TABLE 4.5 – Welsh's t-tests for scene 2 with SNR = 30 dB (linear model)

	FCLS	ExtM	KFCLS	BilBay
K-Hype (G)				$\mathcal{A}$
K-Hype (P)				$\mathcal{A}$
SK-Hype (G)				$\mathcal{A}$
SK-Hype (P)				$\mathcal{A}$

TABLE 4.6 – Welsh's t-tests for scene 2 with SNR = 30 dB (bilinear model)

	FCLS	ExtM	KFCLS	BilBay
K-Hype (G)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
K-Hype (P)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	
SK-Hype (G)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
SK-Hype (P)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$

between original $\boldsymbol{r}$ and reconstructed $\boldsymbol{r}^*$ pixel vectors was used

$$\Theta = \frac{1}{N}\sum_{n=1}^{N}\theta(\boldsymbol{r}_n, \boldsymbol{r}_n^*)$$

where N is the number of processed pixels and $\theta(\boldsymbol{r}, \boldsymbol{r}^*) = \cos^{-1}\left(\frac{\langle \boldsymbol{r}, \boldsymbol{r}^* \rangle}{\|\boldsymbol{r}\|\|\boldsymbol{r}^*\|}\right)$. It is important to note that the quality of reconstruction, estimated by the averaged spectral angle or mean-square error for instance, is not necessarily in proportion to the the quality of unmixing, especially for real images where the nonlinear mixing mechanism can be complex. In particular, more complicated model may better fit the data. Parameter Θ is only reported here as complementary information. The averaged spectral angle of each approach is reported in Table 4.9. Note that KFCLS was not considered in these tests as there is no possible direct reconstruction of pixels. Clearly, our algorithms have much lower reconstruction errors than the other approaches. Six typical estimated abundance maps out of twelve available are shown in Figure 4.1. It can be observed that the estimated locations of the different materials are quite similar for the four methods, except the US Highway 95 in the last column which is much more accurately depicted by our methods. Finally, the distributions of reconstruction errors $\|\boldsymbol{r}_n - \boldsymbol{r}_n^*\|/L$ associated to these methods are shown in Figure 4.2.

TABLE 4.7 – Welsh's t-tests for scene 2 with SNR = 30 dB (PNMM)

	FCLS	ExtM	KFCLS	BilBay
K-Hype (G)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
K-Hype (P)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
SK-Hype (G)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$
SK-Hype (P)	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$	$\mathcal{A}$

TABLE 4.8 – Averaged computational time per pixel (in seconds)

	$L = 420$			$L = 210$		
	$R = 3$	$R = 5$	$R = 8$	$R = 3$	$R = 5$	$R = 8$
FCLS	7.6×10^{-4}	1.2×10^{-3}	2.4×10^{-3}	6.4×10^{-4}	1.1×10^{-3}	2.1×10^{-3}
ExtM	1.6×10^{-3}	5.5×10^{-3}	0.020	1.4×10^{-3}	4.0×10^{-3}	0.014
KFCLS	6.0×10^{-4}	1.3×10^{-3}	2.5×10^{-3}	5.6×10^{-4}	8.4×10^{-4}	1×10^{-3}
BilBay	6.6	15	40	6.4	14	33
K-Hype	0.17	0.29	0.47	0.038	0.065	0.11
SK-Hype	1.6	2.4	3.4	0.4	0.5	0.7

TABLE 4.9 – Spectral angles comparison

Algorithms	Θ	Algorithms	Θ
FCLS	0.0136	K-Hype (G)	0.0070
ExtM	0.0123	K-Hype (P)	0.0098
BilBay	0.0182	SK-Hype (G)	0.0078
		SK-Hype (P)	0.0104

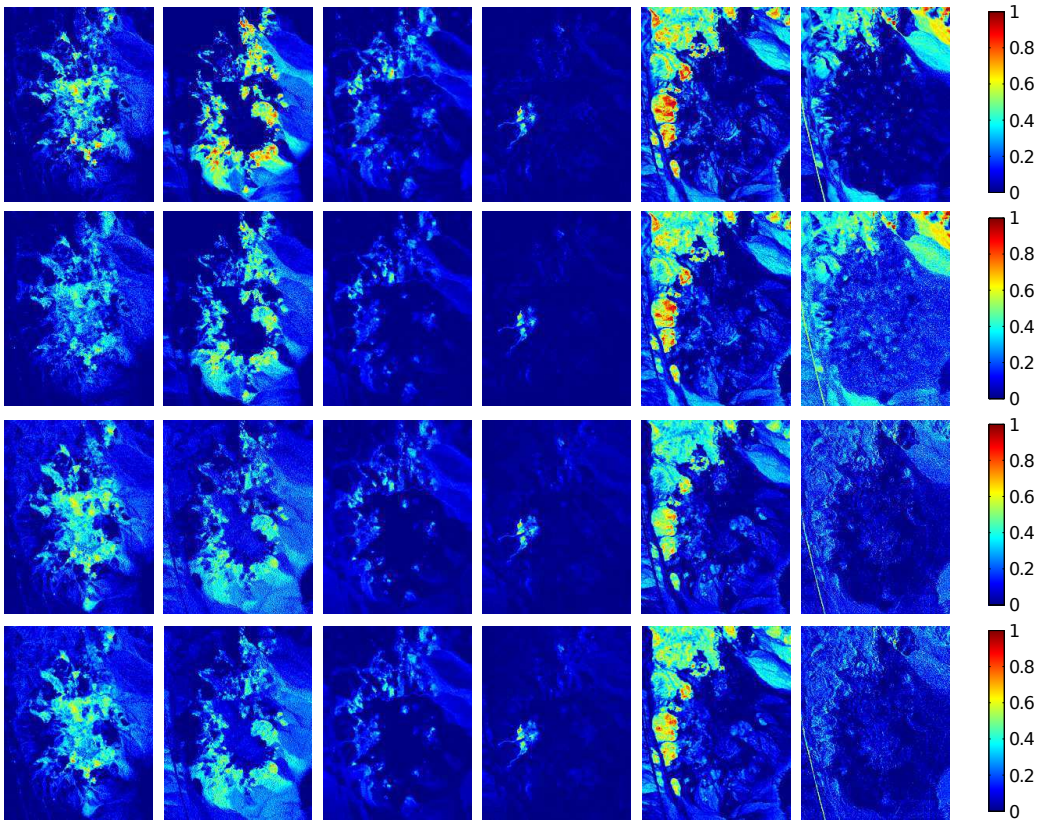


FIGURE 4.1 – Abundances maps of selected materials. From top to bottom : FCLS, BilBay, K-Hype (G), SK-Hype (G). From left to right : chalcedony, alunite, kaolinite, buddingtonite, sphene, US highway 95.

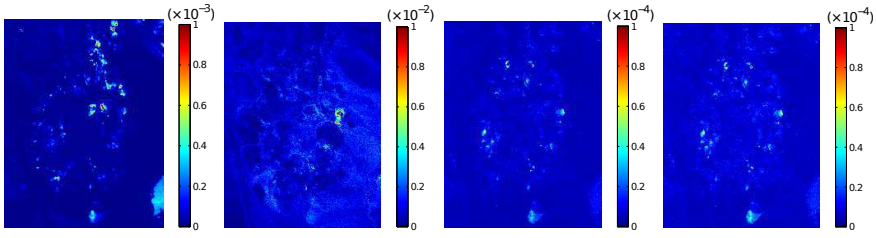


FIGURE 4.2 – Maps of reconstruction error. From left to right : FCLS, BilBay, K-Hype (G), SK-Hype (G).

Chapter 5

Nonlinear unmixing with spatial regularization

Contents

In the previous chapter we have investigated the problem of nonlinear unmixing hyperspectral data and proposed efficient algorithms to address the mixing prpblem based on linear-mixture/nonlinear-fluctuation model. It is expected that exploring and incorporating other information embedded in the image or structure of data should further improve the performance of unmixing algorithms. In this chapter, we shall benefit from the spatial correlation in hyperspectral images to improve our nonlinear unmixing algorithm.

5.1 Introduction

Several linear and nonlinear state-of-the-art unmixing techniques have shown interesting performance for the hyperspectral unmixing problem, however, they have been mostly focused on exploiting spectral information available in the hyperspectral space. These approaches consist of considering pixel vectors as if they were independent from their neighboring pixels. One of the distinguishing properties of remotely sensed data is that they convey multivariate information into a 2D pictorial representation [Plaza 2011a]. Subsequently, instead of disregarding spatial contextual information, hyperspectral analysis techniques should benefit from the inherent spatial-spectral duality in hyperspectral scene. Following this idea, researchers have attempted to exploit spatial information in hyperspectral image analysis.

Spatial preprocessing techniques were investigated in the endmember determination [Rogge 2007, Zortea 2009, Martin 2011]. Spatial correlation have been incorporated into hyperspectral image classification algorithms [Fauvel 2012, Li 2011]. Concerning the unmixing problem, an NMF-type problem, regularized by the ℓ_1-norm of differences between neighboring pixels, was proposed in [Zymnis 2007]. A projected subgradient method was used to solve this problem. In [Eches 2011], a Markov random field (MRF) was proposed to model the spatial dependency of the pixels within classes. Bayesian inference was then used to estimate the model parameters. In [Iordache 2012], total variation was employed for spatial regularization to enhance the unmixing performance. The alternating direction method of multipliers was used to solve the regularized problem. Some other works also showed that incorporating spatial information has a positive impact in hyperspectral unmixing [Jia 2007, Zare 2011].

To the best of our knowledge, these two popular recent trends, nonlinear unmixing and incorporating spatial information, have not yet been studied in an integrated manner for hyperspectral unmixing problem. As nonlinear unmixing is itself an important but challenging issue, it appears difficult to address these two problems simultaneously. In the previous chapter a new nonlinear model was proposed in our work, where we have assumed that a nonlinear mixture can be decomposed into a linear trend, and an additive nonlinear fluctuation term in a reproducing kernel Hilbert space to model nonlinear effects. Based on this promising advance within the area of nonlinear unmixing, in this chapter, we take spatial information into account in the unmixing process using ℓ_1-norm spatial regularization. An optimization method based on split Bregman iteration is proposed to deal with this problem that suffers from the nonlinearity of the model and the non-smoothness of the regularization term. Experiments with both synthetic and real data are conducted to validate the proposed method.

5.2 Formulation of the nonlinear unmixing problem with spatial regularization

Suppose that the materials in a scene have been determined by some endmember extraction algorithm. The unmixing problem boils down to estimating the abundance vector associated to each pixel. One way to exploit spatial information is to define an appropriate criterion to be optimized, e.g., by considering extra penalty terms to promote similarity of fractional abundances between neighboring pixels. The rationale is that homogeneous regions within which correlation among neighboring pixels is high potentially exist in real images. This suggests that increasing spatial homogeneity should tend to increase the accuracy of the representation of spectral objects, and to suppress high-spatial-frequency artifacts [Warner 1997] although there are risks to remove small features.

To take the spatial relationship among pixels into consideration, the unmixing problem can then be solved by minimizing a general cost function with respect to

the abundance matrix $\boldsymbol{A}$

$$J(\boldsymbol{A}) = J_{\text{err}}(\boldsymbol{A}) + \eta\, J_{\text{sp}}(\boldsymbol{A}) \tag{5.1}$$

subject to the non-negativity constraint imposed on each entry of $\boldsymbol{A}$ and the sum-to-one constraint imposed on each column of $\boldsymbol{A}$, namely, on each $\boldsymbol{\alpha}_n$. For ease of notation, these two physical constraints will be expressed by

$$\begin{aligned} &\boldsymbol{A} \succeq \boldsymbol{0} \\ &\boldsymbol{A}^\top \boldsymbol{1}_R = \boldsymbol{1}_N \end{aligned} \tag{5.2}$$

Recent works have raised the question of relaxing the sum-to-one constraint. Indeed, poor estimates of the endmember signatures may affect the performance of the unmixing process. This constraint is maintained in the following as a baseline for comparison with existing approaches. In the general expression (5.1), the function J_{err} represents the modeling error and J_{sp} is a regularization term to promote similarity of the fractional abundances within neighboring pixels. The non-negative parameter η controls the trade-off between data fidelity and pixel similarity. In [Zymnis 2007] for instance, the ℓ_1-norm of the differences of abundance vectors in the neighborhood of each pixel was used as the spatial regularizer. In [Iordache 2012], anisotropic total variation norm was considered for linear sparse unmixing. In [Zare 2011], a so-called fuzzy local information proportion was used for incorporating spatial information. In [Li 2011, Eches 2011], a Markov random field was proposed to model spatial dependency. All these approaches have shown a substantial advantage of using spatial information for hyperspectral data unmixing, although there are risks to remove small and significant features. Preprocessing can be conducted to alleviate such hazards by separating these features in advance [Du 2009].

Rarely, if ever, have nonlinear mixing models incorporating spatial information been considered in the literature. In this paper, we intend to formulate J_{err} as in our recent work [Chen 2013] because this approach has shown excellent performance and low computational cost. For self-containedness, let us briefly review part of this work. Consider the general mixing process acting on the entries of the reflectance vector $\boldsymbol{r}_n$, defined as

$$r_{n,\ell} = \psi_{\boldsymbol{\alpha}_n}(\boldsymbol{m}_{\lambda_\ell}) + e_{n,\ell} \tag{5.3}$$

with $\psi_{\boldsymbol{\alpha}_n}$ an unknown nonlinear function to be estimated that defines the interaction between the endmember spectra, in the proportion $\boldsymbol{\alpha}_n$, and with $\boldsymbol{e}_n$ an additive noise. This requires us to consider a general unmixing problem of the form

$$\psi^*_{\boldsymbol{\alpha}_n} = \arg\min_{\psi_{\boldsymbol{\alpha}_n}} \frac{1}{2}\|\psi_{\boldsymbol{\alpha}_n}\|^2_{\mathcal{H}} + \frac{1}{2\mu}\sum_{\ell=1}^{L}(r_{n,\ell} - \psi_{\boldsymbol{\alpha}_n}(\boldsymbol{m}_{\lambda_\ell}))^2 \tag{5.4}$$

with μ a positive parameter that controls the trade-off between structural error and misadjustment error. Clearly, this basic strategy may fail if the functionals $\psi_{\boldsymbol{\alpha}_n}$ cannot be adequately and finitely parameterized. In [Chen 2013], we defined

them by a linear trend parameterized by the abundance vector $\boldsymbol{\alpha}_n$, combined with a nonlinear fluctuation function ψ_n, namely,

$$\psi_{\boldsymbol{\alpha}_n}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{\alpha}_n^\top \boldsymbol{m}_{\lambda_\ell} + \psi_n(\boldsymbol{m}_{\lambda_\ell}) \tag{5.5}$$

where ψ_n can be any real-valued function of a reproducing kernel Hilbert space $\mathcal{H}$, endowed with the reproducing kernel κ such that $\psi_n(\boldsymbol{m}_{\lambda_\ell}) = \langle \psi_n, \kappa(\cdot, \boldsymbol{m}_{\lambda_\ell})\rangle$. Indeed, kernel-based methods lead to efficient and accurate resolution of inverse problems of the form (5.4) by exploiting the central idea of this research area, known as the *kernel trick* [Schölkopf 1999, Vapnik 1995]. We proposed in [Chen 2013] to conduct data unmixing (5.4)-(5.5) by solving the following least-square support vector regression (LS-SVR) problem

$$\begin{aligned} \boldsymbol{\alpha}_n^*, \psi_n^* = &\arg\min_{\boldsymbol{\alpha}_n, \psi_n} \frac{1}{2}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2\right) \\ &\text{subject to} \quad \boldsymbol{\alpha}_n \succeq \mathbf{0} \quad \text{and} \quad \mathbf{1}^\top \boldsymbol{\alpha}_n = 1 \end{aligned} \tag{5.6}$$

where $\boldsymbol{e}_n$ is the $(L \times 1)$ misadjustment error vector with ℓ-th entry $e_{n,\ell} = r_{n,\ell} - (\boldsymbol{\alpha}_n^\top \boldsymbol{m}_{\lambda_\ell} + \psi_n(\boldsymbol{m}_{\lambda_\ell}))$ as defined in (5.4). It can be shown that problem (5.6) is convex so that it can be solved exactly by the duality theory. This so-called K-Hype method was introduced in the previous section. A very efficient extension, called SK-Hype, consisting of relaxing the sum-to-one constraint, and automatically adjusting the balance between $\boldsymbol{\alpha}_n^\top \boldsymbol{m}_{\lambda_\ell}$ and $\psi_n(\boldsymbol{m}_{\lambda_\ell})$ via an appropriate parameterization, was also presented. In the following, we restrict our attention to K-Hype and we maintain the sum-to-one constraint as a baseline for comparison with existing approaches. Finally, considering all the pixels of the image to process, the modeling error to be optimized writes

$$J_{\text{err}}(\boldsymbol{A}, \boldsymbol{\psi}) = \frac{1}{2}\sum_{n=1}^{N}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2\right) \tag{5.7}$$

subject to the non-negativity and sum-to-ones constraints over the abundance vectors. In this expression, $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ and $\boldsymbol{\psi} = \{\psi_n \in \mathcal{H} : n = 1, \ldots, N\}$.

In order to take into account spatial correlation between pixels, we shall use ℓ_1-type regularizers of the form [Zymnis 2007, Iordache 2012] to promote piecewise constant transitions in the fractional abundance of each endmember among neighboring pixels. The regularization function is expressed as

$$J_{\text{sp}}(\boldsymbol{A}) = \sum_{n=1}^{N}\sum_{m \in \mathcal{N}(n)} \|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_m\|_1 \tag{5.8}$$

where $\|\,\|_1$ denotes the vector ℓ_1-norm, and $\mathcal{N}(n)$ the set of neighbors of the pixel n. Without any loss of generality, in this paper, we define the neighborhood of the pixel n by taking the 4 nearest pixels $n-1$ and $n+1$ (row adjacency), $n-w$ and $n+w$ (column adjacency). In this case, let us define the $(N \times N)$ matrices

$\boldsymbol{H}_{\leftarrow}$ and $\boldsymbol{H}_{\rightarrow}$ as the two linear operators that compute the difference between any abundance vector and its left-hand neighbor, and right-hand neighbor, respectively. Similarly, let $\boldsymbol{H}_{\uparrow}$ and $\boldsymbol{H}_{\downarrow}$ be the linear operators that compute that difference with the top neighbor and the down neighbor, respectively. With these notations, the regularization function (5.8) can be rewritten in matrix form as

$$J_{\text{sp}}(\boldsymbol{A}) = \|\boldsymbol{A}\boldsymbol{H}\|_{1,1} \tag{5.9}$$

with $\boldsymbol{H}$ the $(N \times 4N)$ matrix $\left(\boldsymbol{H}_{\leftarrow}\ \boldsymbol{H}_{\rightarrow}\ \boldsymbol{H}_{\uparrow}\ \boldsymbol{H}_{\downarrow}\right)$ and $\|\ \|_{1,1}$ the sum of the ℓ_1-norms of the columns of a matrix. Obviously, other boundaries for the neighborhood $\mathcal{N}(n)$ may be used by simply defining the appropriate matrix $\boldsymbol{H}$. Finally, note that this regularization function is convex but non-smooth.

Now considering both the modeling error J_{err} and the regularization term J_{sp}, the optimization problem becomes

$$\begin{aligned} \boldsymbol{A}^*, \boldsymbol{\psi}^* = \underset{\boldsymbol{A},\boldsymbol{\psi}}{\arg\min} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 \right) + \eta \|\boldsymbol{A}\boldsymbol{H}\|_{1,1} \\ \text{subject to} \quad \boldsymbol{A} \succeq 0 \quad \text{and} \quad \boldsymbol{A}^\top \mathbf{1}_R = \mathbf{1}_N \end{aligned} \tag{5.10}$$

The constraints over $\boldsymbol{A}$ define a convex set $\mathcal{S}_A$. For ease of exposition, in the formulation of optimization problems, we will write $\boldsymbol{A} \in \mathcal{S}_A$.

5.3 Solving the problem with split-Bregman method

Although the optimization problem (5.10) is convex, it cannot be solved easily because it combines an LS-SVM problem with a huge-dimensional non-smooth regularization term. In order to overcome this drawback, we rewrite (5.10) in the following equivalent form

$$\begin{aligned} \min_{\boldsymbol{A}\in\mathcal{S}_A,\boldsymbol{\psi}} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 \right) + \eta \|\boldsymbol{U}\|_{1,1} \\ \text{subject to} \quad \boldsymbol{V} = \boldsymbol{A} \quad \text{and} \quad \boldsymbol{U} = \boldsymbol{V}\boldsymbol{H} \end{aligned} \tag{5.11}$$

where two new matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and two additional constraints, have been introduced. This variable-splitting approach was initially introduced in [Goldstein 2009]. Matrix $\boldsymbol{U}$ will allow us to decouple the non-smooth ℓ_1-norm regularization functional from the constrained LS-SVR problem. Matrix $\boldsymbol{V}$ will make the LS-SVM problem tractable by relaxing connections between pixels.

As studied in [Goldstein 2009], the split Bregman iteration algorithm is an efficient method to deal with a broad class of ℓ_1-regularized problems. By applying this framework to (5.10), the following formulation is obtained

$$\begin{aligned} \boldsymbol{A}^{(k+1)},\boldsymbol{\psi}^{(k+1)},\boldsymbol{V}^{(k+1)},\boldsymbol{U}^{(k+1)} = \underset{\boldsymbol{A}\in\mathcal{S}_A,\boldsymbol{\psi},\boldsymbol{V},\boldsymbol{U}}{\arg\min} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 \right) \\ + \eta \|\boldsymbol{U}\|_{1,1} + \frac{\zeta}{2}\|\boldsymbol{A} - \boldsymbol{V} - \boldsymbol{D}_1^{(k)}\|_F^2 + \frac{\zeta}{2}\|\boldsymbol{U} - \boldsymbol{V}\boldsymbol{H} - \boldsymbol{D}_2^{(k)}\|_F^2 \end{aligned} \tag{5.12}$$

with

$$\begin{aligned} \boldsymbol{D}_1^{(k+1)} &= \boldsymbol{D}_1^{(k)} + \left(\boldsymbol{V}^{(k+1)} - \boldsymbol{A}^{(k+1)}\right) \\ \boldsymbol{D}_2^{(k+1)} &= \boldsymbol{D}_2^{(k)} + \left(\boldsymbol{V}^{(k+1)}\boldsymbol{H} - \boldsymbol{U}^{(k+1)}\right) \end{aligned} \tag{5.13}$$

where $\| \|_F^2$ denotes the matrix Frobenius norm, and ζ is a positive parameter. Because of the way we have split the components of the cost function, we can now perform the above minimization efficiently by iteratively minimizing with respect to $(\boldsymbol{A}, \boldsymbol{\psi})$, $\boldsymbol{V}$ and $\boldsymbol{U}$ separately. The three steps we have to perform are :

5.3.1 Optimization with respect to $\boldsymbol{A}$ and $\boldsymbol{\psi}$

The optimization problem (5.12) reduces to

$$\boldsymbol{A}^{(k+1)}, \boldsymbol{\psi}^{(k+1)} = \underset{\boldsymbol{A}\in\mathcal{S}_A, \boldsymbol{\psi}}{\arg\min} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 + \zeta\|\boldsymbol{\alpha}_n - \boldsymbol{\xi}_n^{(k)}\|^2 \right) \tag{5.14}$$

where $\boldsymbol{\xi}_n^{(k)} = \boldsymbol{V}_n^{(k)} + \boldsymbol{D}_{1,n}^{(k)}$. Here, $\boldsymbol{V}_n$ and $\boldsymbol{D}_{1,n}$ denote the n-th column of $\boldsymbol{V}$ and $\boldsymbol{D}_1$, respectively. It can be observed that this problem can be decomposed into subproblems, each one involving an abundance vector $\boldsymbol{\alpha}_n$. This results from the use of the matrix $\boldsymbol{V}$ in the split iteration algorithm (5.12).

Let us now solve the local optimization problem

$$\begin{aligned} \boldsymbol{\alpha}_n^{(k+1)}, \psi_n^{(k+1)} = \underset{\boldsymbol{\alpha}_n, \psi_n, \boldsymbol{e}_n}{\arg\min}\ & \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\sum_{\ell=1}^{L} e_{n,\ell}^2 + \zeta\|\boldsymbol{\alpha}_n - \boldsymbol{\xi}_n^{(k)}\|^2 \right) \\ \text{subject to}\quad & e_{n,\ell} = r_{n,\ell} - (\boldsymbol{\alpha}_n^\top \boldsymbol{m}_{\lambda_\ell} + \psi_n(\boldsymbol{m}_{\lambda_\ell})) \\ & \boldsymbol{\alpha}_n \succeq 0 \\ & \boldsymbol{\alpha}_n^\top \mathbf{1}_R = 1 \end{aligned} \tag{5.15}$$

By introducing the Lagrange multipliers $\beta_{n,\ell}$, $\gamma_{n,\ell}$ and λ_n, where the superscript (k) of these variables has been omitted for simplicity of notation, the Lagrange function associated to (5.15) is written as

$$\begin{aligned} \mathcal{L}_n =& \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\sum_{\ell=1}^{L} e_{n,\ell}^2 + \zeta\|\boldsymbol{\alpha}_n - \boldsymbol{\xi}_n^{(k)}\|^2 \right) \\ & - \sum_{\ell=1}^{L} \beta_\ell (e_{n,\ell} - r_{n,\ell} + \boldsymbol{\alpha}_n^\top \boldsymbol{m}_{\lambda_\ell} + \psi_n(\boldsymbol{m}_{\lambda_\ell})) - \sum_{r=1}^{R} \gamma_r \alpha_{n,r} + \lambda_n(\boldsymbol{\alpha}_n^\top \mathbf{1}_R - 1) \end{aligned} \tag{5.16}$$

with $\gamma_{n,r} \geq 0$. The conditions for optimality of $\mathcal{L}_n$ with respect to the primal variables lead us to

$$\begin{cases} \boldsymbol{\alpha}_n^* = \frac{1}{\zeta+1}\left(\sum_{\ell=1}^{L} \beta_{n,\ell}^* \boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}_n^* - \lambda_n^* \mathbf{1} + \zeta\boldsymbol{\xi}_n^{(k)}\right) \\ \psi_n^* = \sum_{\ell=1}^{L} \beta_{n,\ell}^* \kappa(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_{n,\ell}^* = \mu\, \beta_{n,\ell}^* \end{cases} \tag{5.17}$$

$$\max_{\boldsymbol{\beta}_n,\boldsymbol{\gamma}_n,\lambda_n} \mathcal{L}'_n(\boldsymbol{\beta}_n,\boldsymbol{\gamma}_n,\lambda_n) = -\frac{\rho}{2\zeta}\left(\begin{array}{c}\boldsymbol{\beta}_n\\\hline\boldsymbol{\gamma}_n\\\hline\lambda_n\end{array}\right)^\top \left(\begin{array}{c|c|c} \boldsymbol{K}_\psi & \boldsymbol{M} & -\boldsymbol{M}\mathbf{1}_R \\\hline \boldsymbol{M}^\top & \boldsymbol{I} & -\mathbf{1}_R \\\hline -\mathbf{1}_R^\top\boldsymbol{M}^\top & -\mathbf{1}_R^\top & R \end{array}\right)\left(\begin{array}{c}\boldsymbol{\beta}_n\\\hline\boldsymbol{\gamma}_n\\\hline\lambda_n\end{array}\right) + \left(\begin{array}{c}\boldsymbol{r}_n - \rho\,\boldsymbol{M}\,\boldsymbol{\xi}_n^{(k)}\\\hline -\rho\,\boldsymbol{\xi}_n^{(k)}\\\hline \rho\,\boldsymbol{\xi}_n^{(k)\top}\mathbf{1}_R - 1\end{array}\right)^\top \left(\begin{array}{c}\boldsymbol{\beta}_n\\\hline\boldsymbol{\gamma}_n\\\hline\lambda_n\end{array}\right)$$
$$\text{subject to} \quad \boldsymbol{\gamma}_n \succeq \mathbf{0}$$
$$\text{with} \quad \boldsymbol{K}_\psi = \frac{1}{\zeta}(\boldsymbol{K} + \mu\,\boldsymbol{I}) + \boldsymbol{M}\boldsymbol{M}^\top \text{ and } \rho = \frac{\zeta}{1+\zeta} \tag{5.18}$$

where κ is the reproducing kernel of $\mathcal{H}$. By substituting (5.17) into (5.16), we get the dual problem (5.18), where $\boldsymbol{K}$ is the Gram matrix defined as $[\boldsymbol{K}]_{\ell p} = \kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_p})$. The problem (5.18) is a convex quadratic programming problem with respect to the dual variables. Finally, provided that the optimal dual variables $\boldsymbol{\beta}_n^*$, $\boldsymbol{\gamma}_n^*$ and λ_n^* have been determined, the vector of fractional abundances $\boldsymbol{\alpha}_n^{(k+1)}$ is estimated by

$$\boldsymbol{\alpha}_n^* = \frac{1}{\zeta+1}\left(\boldsymbol{M}^\top\boldsymbol{\beta}_n^* + \boldsymbol{\gamma}_n^* - \lambda_n^*\mathbf{1} + \zeta\boldsymbol{\xi}_n^{(k)}\right) \tag{5.19}$$

This process has to be repeated for $n = 1, \ldots, N$ in order to get $\boldsymbol{A}^{(k+1)}$. Matrices $\boldsymbol{A}^{(k+1)}$ and $\boldsymbol{V}^{(k+1)}$, whose calculation is presented hereafter, allow to evaluate $\boldsymbol{D}_1^{(k+1)}$ using equation (5.13).

5.3.2 Optimization with respect to $\boldsymbol{V}$

The optimization problem (5.12) now reduces to

$$\boldsymbol{V}^{(k+1)} = \arg\min_{\boldsymbol{V}} \|\boldsymbol{A}^{(k+1)} - \boldsymbol{V} - \boldsymbol{D}_1^{(k)}\|_F^2 + \|\boldsymbol{U}^{(k)} - \boldsymbol{V}\boldsymbol{H} - \boldsymbol{D}_2^{(k)}\|_F^2 \tag{5.20}$$

Equating to zero the derivative of (5.20) with respect to $\boldsymbol{V}$ leads to

$$\left(\boldsymbol{A}^{(k+1)} - \boldsymbol{V} - \boldsymbol{D}_1^{(k)}\right) - \left(\boldsymbol{U}^{(k)} - \boldsymbol{V}\boldsymbol{H} - \boldsymbol{D}_2^{(k)}\right)\boldsymbol{H}^\top = 0 \tag{5.21}$$

whose solution is then given by

$$\boldsymbol{V}^{(k+1)} = \left(\boldsymbol{A}^{(k+1)} - \boldsymbol{D}_1^{(k)} + (\boldsymbol{U}^{(k)} - \boldsymbol{D}_2^{(k)})\boldsymbol{H}^\top\right)(\boldsymbol{I} + \boldsymbol{H}\boldsymbol{H}^\top)^{-1} \tag{5.22}$$

5.3.3 Optimization with respect to $\boldsymbol{U}$

Finally, the optimization problem we have to consider is as follows

$$\boldsymbol{U}^{(k+1)} = \arg\min_{\boldsymbol{U}} \eta\|\boldsymbol{U}\|_{1,1} + \frac{\zeta}{2}\|\boldsymbol{U} - \boldsymbol{V}^{(k+1)}\boldsymbol{H} - \boldsymbol{D}_2^{(k)}\|_F^2 \tag{5.23}$$

Its solution can be expressed via the well-known soft threshold function

$$\boldsymbol{U}^{(k+1)} = \text{Thresh}\left(\boldsymbol{V}^{(k+1)}\boldsymbol{H} + \boldsymbol{D}_2^{(k)}, \frac{\eta}{\zeta}\right) \tag{5.24}$$

where $\text{Thresh}(\cdot, \tau)$ denotes the component-wise application of the soft threshold function defined as [Tibshirani 1996]

$$\text{Thresh}(x, \tau) = \text{sign}(x)\ \max(|x| - \tau, 0) \tag{5.25}$$

To conclude, the problem (5.11) is solved by iteratively applying (5.12) and (5.13), where the optimization of (5.12) can be performed by applying Steps 1 to 3. These iterations continue until some stopping criterion is satisfied. It can be shown that, if the problem (5.12) has a solution $\boldsymbol{A}^*$ given any $\zeta > 0$, then the generated sequence $\boldsymbol{A}^{(k)}$ converges to the optimum $\boldsymbol{A}^*$ [Eckstein 1992].

5.4 Experiment Results

In this section, experiments on spatially correlated images are reported to validate the proposed algorithm.

5.4.1 Experiments with synthetic images

5.4.1.1 Simulation scenario settings

Two spatially correlated abundance maps were generated for the following experiments. The endmembers were randomly selected from the spectral library ASTER [Baldridge 2009]. Each signature of this library has reflectance values measured over 224 spectral bands, uniformly distributed in the interval $3 - 12$ μm. Two synthetic images identical to [Iordache 2012] were used.

The first data cube, denoted by IM1, and containing 75×75 pixels, was generated as using five signatures randomly selected from the ASTER library. Pure regions and mixed regions involving between 2 and 5 endmembers, distributed spatially in the form of square regions, were generated. The background pixels were defined as mixtures of the same 5 endmembers with the abundance vector $[0.1149, 0.0741, 0.2003, 0.2055, 0.4051]^\top$. The first row in Figure 5.1 shows the true fractional abundances for each endmember. The reflectance samples were generated with the bilinear mixing model, based on the 5 endmembers, and corrupted by a zero-mean white Gaussian noise $\boldsymbol{v}_n$ with a SNR of 20 dB, namely,

$$\boldsymbol{r}_n = \boldsymbol{M}\boldsymbol{\alpha}_n + \sum_{i=1}^{R}\sum_{j=i+1}^{R} \alpha_{n,i}\,\alpha_{n,j}\,\boldsymbol{m}_i \otimes \boldsymbol{m}_j + \boldsymbol{v}_n \tag{5.26}$$

with $\otimes$ the Hadamard product. At the end of this series of experiments with synthetic images, note that we will also consider a signal-dependent noise $\boldsymbol{v}_n$.

The second data cube, denoted by IM2 and containing 100×100 mixed pixels, was generated using 9 endmember signatures. The abundance maps of the endmembers are the same as for the image DC2 in [Iordache 2012]. Among these 9 materials, only the 1st, 3rd, 5th, 8th and 9th abundances are considered for pictorial illustration in Figure 5.2. The first row of this figure depicts the true distribution of these 5 materials. Spatially homogeneous areas with sharp transitions can be clearly observed. Based on these abundance maps, an hyperspectral data cube was generated with the bilinear model (5.26) applied to the 9 endmember spectral signatures. The scene was also corrupted by a zero-mean white Gaussian noise $\boldsymbol{v}_n$ with a SNR of 20 dB.

5.4.1.2 Comparative simulations

Several algorithms were tested in order to compare their unmixing performance on these two images. Their tuning parameters were set during preliminary experiments on independent data, via a simple search over the grids defined hereafter.

- The linear unmixing method FCLS [Heinz 2001] : The fully constrained least-square method was run with the sum-to-one constraint strictly satisfied for comparability among algorithms.
- The spatially-regularized FCLS : In order to compare linear and nonlinear algorithms, we added the spatial regularization term (5.9) to the FCLS algorithm, and we conducted the split-Bregman iterations to solve the problem.
- The nonlinear unmixing algorithm K-Hype [Chen 2013] : Unmixing is performed in this case by solving problem (4.15). As in [Chen 2013], the polynomial kernel defined by

$$\kappa(\boldsymbol{m}_{\lambda_\ell}, \boldsymbol{m}_{\lambda_\ell}) = \left(1 + \frac{1}{R^2}\left(\boldsymbol{m}_{\lambda_\ell} - 1/2\right)^\top \left(\boldsymbol{m}_{\lambda_\ell} - 1/2\right)\right)^2 \tag{5.27}$$

  was used. The parameter μ that controls the tradeoff between the misadjustment error and the regularization error was varied in the set $\{0.001, 0.005, 0.01, 0.05, 0.1\}$.
- The proposed nonlinear algorithm incorporating spatial regularization : The polynomial kernel (5.27) was used, and the regularization parameter μ was varied in the same set as above. Parameter ζ was adjusted in an adaptive way based on primal and dual residual norms at each iteration, see [Boyd 2011]. We varied the spatial regularization parameter η from 0.25 to 1 at an increment 0.25. Finally, the optimization algorithm was stopped when both $\frac{\|\boldsymbol{V}-\boldsymbol{A}\|_F}{N\times R}$ and $\frac{\|\boldsymbol{U}-\boldsymbol{V}\boldsymbol{H}\|_F}{4N\times R}$ became smaller than 10^{-5}, or the number of iterations exceeded 10.

The above tests were performed on training images IM1 and IM2 to estimate the best parameter values in the sense that they minimize the estimation error (RMSE) defined as

$$E = \sqrt{\frac{1}{NR}\sum_{n=1}^{N} \|\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_n^*\|^2}. \tag{5.28}$$

TABLE 5.1 – Parameter settings for the comparative simulations on IM1 and IM2

	IM1	IM2
Reg-FCLS	$\eta = 0.25$	$\eta = 0.25$
K-Hype	$\mu = 0.1$	$\mu = 0.01$
Proposed	$\mu = 0.005$, $\eta = 0.5$	$\mu = 0.005$, $\eta = 0.5$

These preliminary experiments led to the parameter settings shown in Table 5.1. The performance, with standard deviation, on independent test images IM1 and IM2 are reported in Table 5.2, and the estimated fractional abundances are represented in Figures 5.1 and 5.2. For both images, it can be observed that when applied to nonlinearly mixed data, the linear method FCLS has large estimation errors. The abundance maps appear quite correct visually, but they are uniformly severely biased due to the nonlinearity of the mixing model. The spatially-regularized FCLS algorithm does not allow to improve the performance. The nonlinear K-Hype method notably reduces this error in the mean sense, but the estimated abundance maps are corrupted by a noise that partially masks spatial structures of the materials. Finally, the proposed spatially-regularized method has lower reconstruction error and clearer abundance maps. The use of spatial information provides a clear advantage to this approach.

5.4.1.3 Influence of the parameter η

The penalty term η controls the trade-off between data fitting and similarity among neighboring pixels. In the case $\eta = 0$, the algorithm reduces to the original K-Hype that only considers spectral information at each pixel. The larger η, the flatter the image is. In order to illustrate this intuition, we varied parameter η from 0.25 to 1 for IM1 and IM2, with $\mu = 0.005$. Note that, according to Table 5.3, the optimal value of η is 0.5 in both cases. To illustrate this experiment, the results are represented for IM1 in Figure 5.3.

5.4.1.4 Influence of the neighborhood

In the above experiments, we have used the four nearest neighbors to construct the difference matrix $\boldsymbol{H}_4$, with the subscript (4) to specify the size of the neighborhood. Any other neighborhood could be considered, provided the matrix $\boldsymbol{H}$ is properly defined. For illustration purpose, we also considered the larger matrix of the eight nearest neighbors of each pixel defined as $\boldsymbol{H}_8 = \left(\boldsymbol{H}_{\nwarrow}\ \boldsymbol{H}_{\nearrow}\ \boldsymbol{H}_4\ \boldsymbol{H}_{\swarrow}\ \boldsymbol{H}_{\searrow}\right)$, with $\boldsymbol{H}_{\nwarrow}$, $\boldsymbol{H}_{\nearrow}$, $\boldsymbol{H}_{\swarrow}$, $\boldsymbol{H}_{\searrow}$ the four diagonal adjacency matrices. Estimation error of the abundance fractions are reported in Table 5.2. Abundance maps provided by the proposed algorithm are shown in the last row of Figures 5.1 and 5.2. No significant improvement can be observed in these two cases but, obviously, the proper definition of a neighborhood is closely related to the structure of the images.

TABLE 5.2 – RMSE comparison

	scenario	FCLS	FCLS (regularized)	K-Hype	Proposed
IM1	s.i. noise	0.1730±0.0092	0. 1741±0.0091	0.0784±0.0051	0.0444±0.0016
	s.d. noise	0.1730±0.0091	0.1739±0.0089	0.0785±0.0051	0.0441±0.0017
	8-neighbor	-	0.1777±0.0110	-	0.0509±0.0032
IM2	s.i. noise	0.1680±0.0265	0.1764±0.0291	0.0755±0.0080	0.0521±0.0033
	s.d. noise	0.1681±0.0266	0.1762±0.0292	0.0742±0.0075	0.0522±0.0033
	8-neighbor	-	0.1796±0.0298	-	0.0557±0.0036
Comp. time per-pixel		1.08×10^{-4} s	1.10×10^{-3} s	5.34×10^{-3}s	6.43×10^{-2} s

TABLE 5.3 – RMSE of the proposed method a function of η

η	0.25	0.50	0.75	1.00
IM1	0.0548±0.0027	0.0456±0.0015	0.0479±0.0024	0.0508±0.0037
IM2	0.0576±0.0040	0.0521±0.0033	0.0530±0.0032	0.0567±0.0037

5.4.1.5 Test with a signal dependent noise

Unmixing algorithms proposed in the literature have usually been tested on images corrupted by an i.i.d. additive Gaussian noise. Due to improved sensitivity of electronic components, this assumption may not be appropriate for data collected with new-generation sensors. Noise modeling and identification in hyperspectral images has recently become an active subject of research. It is admitted that cameras provide images corrupted by two independent noises, a signal-dependent one and a signal-independent one [Acito 2011, Meola 2011, Alparone 2009, Aiazzi 2006].

The former results from the stochastic nature of the photon arrival/detection process. The latter results from sensor electronics and quantization process. We compared the unmixing algorithms on images IM1 and IM2 corrupted by the signal-dependent noise defined as

$$v_{n,\ell} = \tilde{r}_{n,\ell}^{\gamma}\, v_{n,\ell}^{(1)} + v_{n,\ell}^{(2)} \tag{5.29}$$

where $\tilde{r}_{n,\ell}$ is the ℓ-th wavelength band of the noise-free reflectance vector, $v_{n,\ell}^{(1)}$ and $v_{n,\ell}^{(2)}$ are two i.i.d. zero-mean Gaussian noises. Parameter γ was set to $\frac{1}{2}$, and the noise variances $\sigma^2_{v^{(1)}} = \sigma^2_{v^{(2)}}$ were set so that the SNR is 20 dB. Table 5.2 provides the performance of the algorithms. It can be noticed that they were not notably affected by this noise setting. The proposed algorithm still exhibits the best performance.

5.4.2 Experiments with AVIRIS data

This part provides unmixing results for the proposed algorithm when applied on real hyperspectral data. Supervised classification of hyperspectral images is a very challenging but important goal, because it generally involves a limited number of training data with unusually high-dimensional patterns. Several feature extraction techniques have been recommended though out the literature, including PCA and

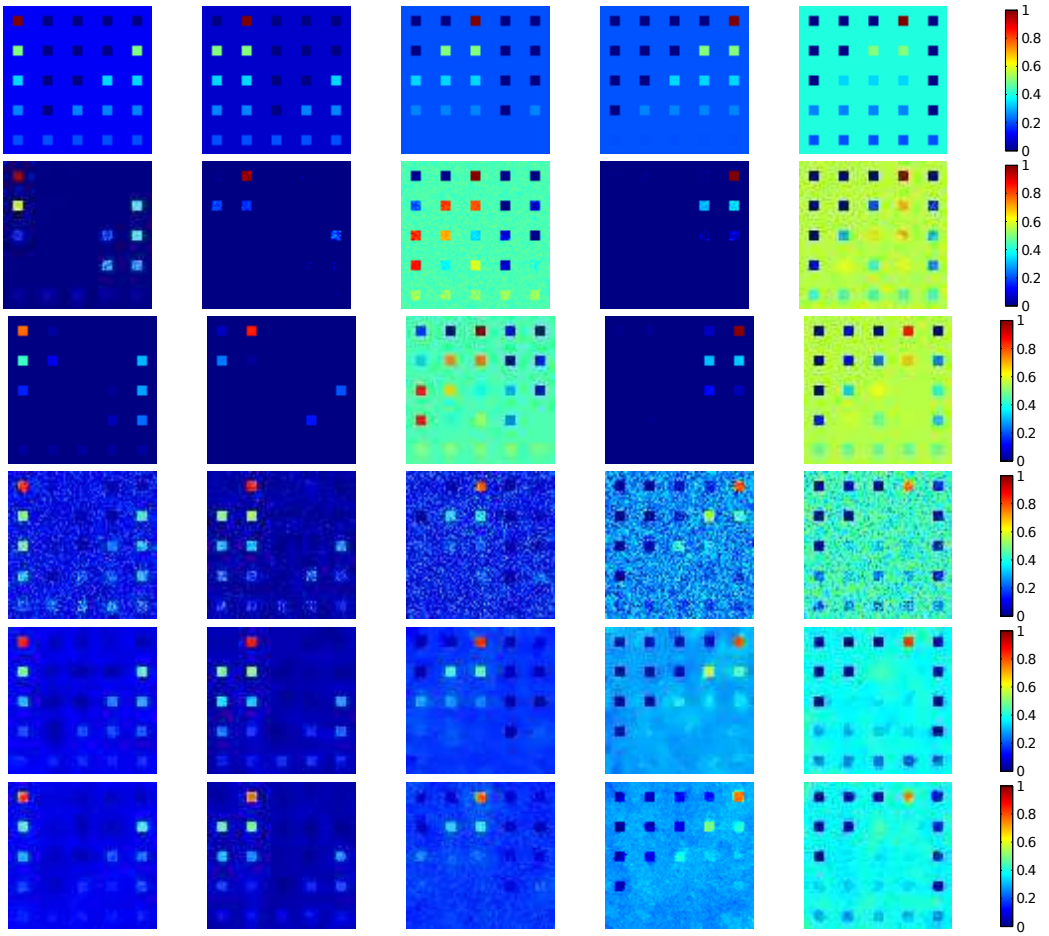


FIGURE 5.1 – Estimated abundance maps for IM1 image. For each row, from top to bottom : true abundance maps, FCLS, spatially-regularized FCLS, K-Hype, proposed algorithm with four neighbors, proposed algorithm with eight neighbors.

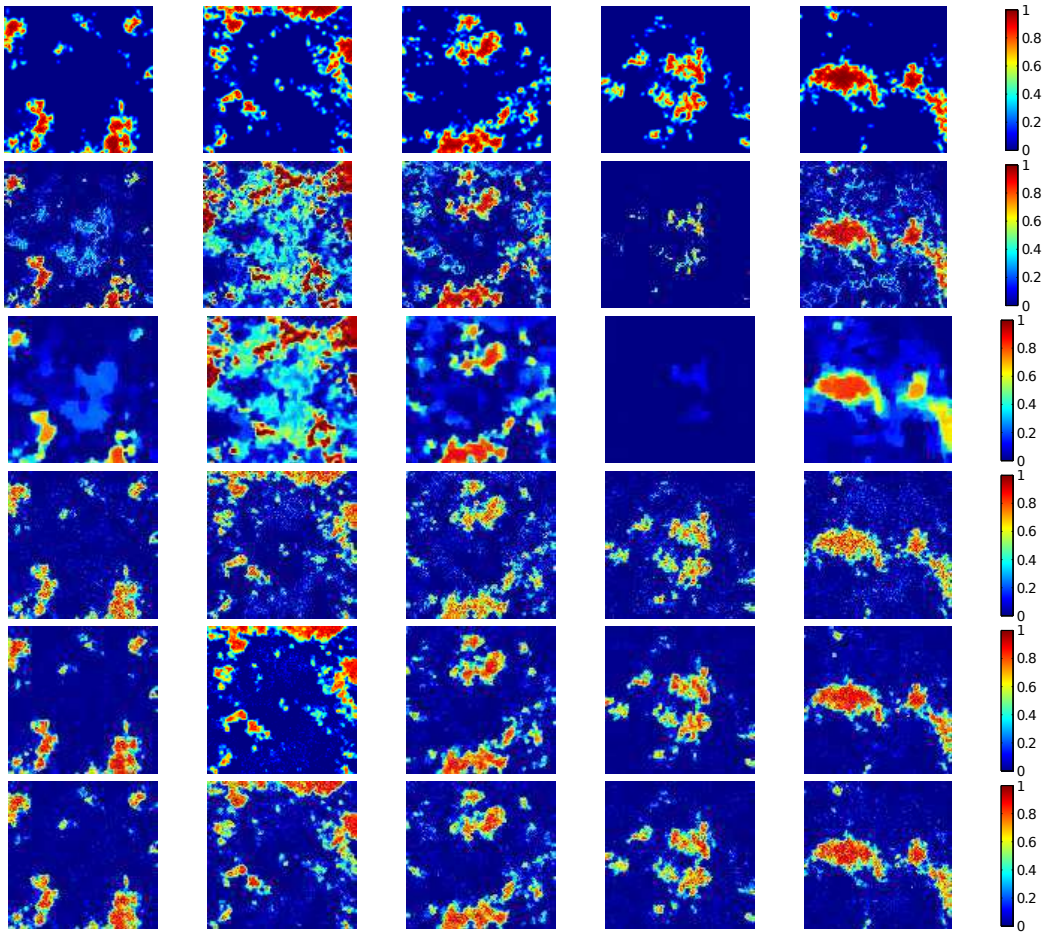


FIGURE 5.2 – Estimated abundance maps for IM2 image. For each row, from top to bottom : true abundance maps, FCLS, spatially-regularized FCLS, K-Hype, proposed algorithm with four neighbors, proposed algorithm with eight neighbors.

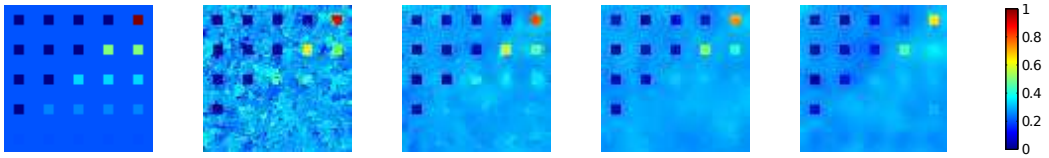


FIGURE 5.3 – Influence of spatial regularization parameter η. From left to right : true, $\eta = 0.25$, $\eta = 0.5$ (optimum), $\eta = 0.75$, $\eta = 1$.

ICA. In [Dòpido 2011], the authors explored an alternative strategy consisting of using spectral unmixing for feature extraction, prior to classification. They considered different unmixing-based processes to evaluate the feasibility of this strategy, and to perceive the necessity of extracting pure spectral endmembers for classification purposes. The so-called unmixing chain #4 in [Dòpido 2011] was found to be the most efficient one. It simply consists of averaging the training samples in each labeled class, and use these spectral signatures to unmix the original image. The features resulting from the unmixing of training samples are used to train an SVM classifier. The latter is tested using the remaining labeled samples.

The scene used in our experiment is the well-known data set captured on the Indian Pines region by AVIRIS. The scene comprises 145×145 samples, consisting of 220 contiguous spectral bands that cover wavelengths ranging from 0.40 to 2.5 μm, with spectral resolution approximately equal to 0.01 μm. Prior to analysis, noisy and water absorption bands were removed, yielding a total of 202 available spectral bands. The ground-truth data contains 16 mutually exclusive classes. The number of pixels in the smallest class is 20, while it is equal to 2468 pixels in the largest class. This widely used benchmark data set is known to be dominated by mixed pixels, even if ground-truth information assigns each pixel a unique class. In this experiment, we used FCLS, K-Hype, and the proposed algorithm with 4-nearest neighbors, for unmixing-based feature extraction. A one-against-all multi-class SVM with Gaussian kernel was applied to these data. We constructed five training sets by randomly selecting 5%, 10%, and 15% of the ground-truth pixels. All the required parameters were optimized by a grid search procedure and five-fold cross-validation. The regularization parameter μ was set to 0.05 for K-Hype and for the proposed algorithm. In addition, for the latter, the spatial regularization parameter η was set to 0.5.

Table 5.4 summarizes the classification accuracies of SVM operating on features extracted with FCLS, K-Hype and the proposed algorithm. Figure 5.4 presents these results in the case of SVM trained with 10% of the available samples per class. It appears that the two nonlinear unmixing algorithms are more effective than the linear one for feature extraction. This clearly means that our nonlinear unmixing model provides less confusing features between the heavily mixed-pixel classes that characterizes the Indian Pines benchmark. Finally, we observe that spatial regularization allows to greatly improve the classification accuracy. Spatial homogeneity is a significant prior information for this problem, which allows to substantially improve the quality of the unmixing process.

5.5 Discussion on the sum-to-one constraint

Recent works have raised the question of relaxing the sum-to-one constraint, because poor estimates of the endmember signatures or misadjustment of the model may affect the performance of the unmixing process. In this thesis, we maintained this constraint for comparison purpose with existing approaches. We shall now pro-

TABLE 5.4 – Classification accuracies after applying SVM to three different types of features (FCLS, K-Hype, proposed)

	5%	10%	15%
FCLS	56.41	61.36	62.32
K-Hype	67.67	71.39	74.68
Proposed	93.82	96.80	97.02

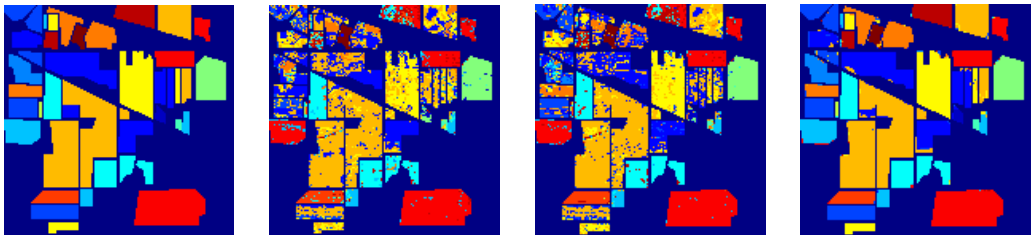


FIGURE 5.4 – Indian Pines classification map. From left to right : ground-truth, FCLS (61.36%), K-Hype (71.39%), Proposed (96.80%)

vide the main result in the case where this constraint over $\boldsymbol{\alpha}_n$ is relaxed in problem (5.11). For clarity, let us denote by $\boldsymbol{h}_n$ the non-normalized vector of abundances. The Lagrange function (5.16) becomes

$$\begin{aligned}\mathcal{L}_n = \frac{1}{2}\left(\|\boldsymbol{h}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\sum_{\ell=1}^{L} e_{n,\ell}^2 + \zeta\|\boldsymbol{h}_n - \boldsymbol{\xi}_n^{(k)}\|^2\right) \\ - \sum_{\ell=1}^{L}\beta_\ell(e_{n,\ell} - r_{n,\ell} + \boldsymbol{h}_n^\top \boldsymbol{m}_{\lambda_\ell} + \psi_n(\boldsymbol{m}_{\lambda_\ell})) - \sum_{r=1}^{R}\gamma_r h_{n,r}\end{aligned} \tag{5.30}$$

with $\gamma_{n,r} \geq 0$. The conditions for optimality of $\mathcal{L}_n$ with respect to the primal variables lead us to

$$\begin{cases} \boldsymbol{h}_n^* = \frac{1}{\zeta+1}\left(\sum_{\ell=1}^{L}\beta_{n,\ell}^*\,\boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}_n^* + \zeta\boldsymbol{\xi}_n^{(k)}\right) \\ \psi_n^* = \sum_{\ell=1}^{L}\beta_{n,\ell}^*\,\kappa(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_{n,\ell}^* = \mu\,\beta_{n,\ell}^* \end{cases} \tag{5.31}$$

By substituting these conditions into the primal problem, we get the following dual problem

$$\begin{aligned}\max_{\boldsymbol{\beta}_n,\boldsymbol{\gamma}_n,\lambda_n} \mathcal{L}'_n(\boldsymbol{\beta}_n,\boldsymbol{\gamma}_n,\lambda_n) = &-\frac{\rho}{2\zeta}\begin{pmatrix}\boldsymbol{\beta}_n\\ \boldsymbol{\gamma}_n\end{pmatrix}^\top \left(\begin{array}{c|c}\boldsymbol{K}_\psi & \boldsymbol{M}\\ \hline \boldsymbol{M}^\top & \boldsymbol{I}\end{array}\right)\begin{pmatrix}\boldsymbol{\beta}_n\\ \boldsymbol{\gamma}_n\end{pmatrix}\\ &+\begin{pmatrix}\boldsymbol{r}_n-\rho\,\boldsymbol{M}\,\boldsymbol{\xi}_n^{(k)}\\ -\rho\,\boldsymbol{\xi}_n^{(k)}\end{pmatrix}^\top\begin{pmatrix}\boldsymbol{\beta}_n\\ \boldsymbol{\gamma}_n\end{pmatrix} \qquad (5.32)\\ &\text{subject to}\quad \boldsymbol{\gamma}_n \succeq \mathbf{0}\\ &\text{with}\quad \boldsymbol{K}_\psi = \frac{1}{\zeta}\left(\boldsymbol{K}+\mu\,\boldsymbol{I}\right)+\boldsymbol{M}\boldsymbol{M}^\top \text{ and } \rho=\frac{\zeta}{1+\zeta}\end{aligned}$$

Provided that the optimal dual variables $\boldsymbol{\beta}_n^*$ and $\boldsymbol{\gamma}_n^*$ have been determined, the solution $\boldsymbol{h}_n^*$ is given by

$$\boldsymbol{h}_n^* = \frac{1}{\zeta+1}\left(\boldsymbol{M}^\top\boldsymbol{\beta}_n^*+\boldsymbol{\gamma}_n^*+\zeta\boldsymbol{\xi}_n^{(k)}\right) \qquad (5.33)$$

If necessary, note that $\boldsymbol{h}_n^*$ can be normalized afterwards by writing $\boldsymbol{h}_n^* = \theta_n^*\,\boldsymbol{\alpha}_n^*$, with $\boldsymbol{\alpha}_n^*$ the vector of what would correspond to fractional abundances, and $\theta_n^* = \mathbf{1}^\top\boldsymbol{h}_n^*$ the scaling factor.

Chapter 6

Conclusion and perspectives

Contents

6.1 Thesis summary

This thesis presented investigation for some problems with nonnegativity and ℓ_1-type constraints. These two constraints have attracted considerable attention during the last decade, for the reason that they allow to incorporate prior information in the identification process, and to enhance interpretability of the obtained results. Several revolutionary techniques, such as nonnegative matrix factorization, compressed sensing, are closely related to these two constraints. The work of this thesis explored problems related to these two constraints under two important signal processing contexts, adaptive filtering and hyperspectral image analysis, within the framework of constrained optimization.

The first part of the thesis concerned the development and the analysis of constrained adaptive filters. In Chapter 2, we started by analyzing the stationarity conditions of the nonnegativity constrained optimization problem and solving the problem via a fixed-point iteration scheme. In order that the algorithm is able to address the online system identification problem, the stochastic gradient was used to obtain an instant update relation. This online algorithm was called NNLMS in the case where the least-mean-square cost was considered. After that, we provided analytical model to characterize the first and the second order moments behavior of the algorithm, in a stationary environment. After, three variants of NNLMS algorithm were proposed to improve its performance in some sense. Moreover, we extended the theoretical analysis conducted for NNLMS for these variants to a nonstationary environment, where the system coefficients fluctuate along a deterministic mean trajectory with random perturbations. Chapter 3 investigated solutions of the least-mean-square problems with ℓ_1-type constraint in an online way by extending

NNLMS algorithm. A direct extension with the constant ℓ_1-norm constraint was firstly presented in brief. The ℓ_1-regularized problem was then discussed with analytical modeling of the algorithm behavior.

The second part of the thesis concerned the problem of unmixing hyperspectral data. When formulated in the form of an optimization problem, due to the physical interpretation of material fractions, the abundance vector of each pixel should be nonnegative and possibly sum-to-one. The unmixing problem based on the linear model has been largely discussed in literature. In Chapter 4, we firstly proposed a linear mixture/nonlinear fluctuation model to mimic the inherent nonlinear mixture mechanism. The nonlinear fluctuation function was restricted to be a functional of reproducing kernel Hilbert space, which allowed us to conduct the unmixing by formulating a support vector regression problem. In order to further control the balance between the linear component and the nonlinear component, a multi-kernel learning strategy was considered. It is interesting to note that many spectral unmixing algorithms, use only spectral information but do not benefit from the spatial information embedded in these 3D images. To improve the performance of the method, Chapter 5 incorporated the spatial correlation into the nonlinear unmixing by introducing the ℓ_1-type regularization for neighboring abundances. Split-Bregman method was used to solve this convex but non-smooth optimization problem. Experiments with spatial correlated images showed that this regularization bring additional benefits to the nonlinear unmixing.

6.2 Perspectives

In this thesis, we discussed theories and algorithms in an systematic and comprehensive way. However, there are still some interesting problems to be further studied. We list these problems hereafter. We will consider them as further research directions.

6.2.1 NNLMS algorithm and its variants

We studied analytically the performance of NNLMS and its variants by giving the first-order and the second-order transient models. These models enable us to evaluate how algorithms evolve with respect to specific parameters, and to estimate steady-state errors by taking values of excess mean square error by solving transient equations when the number of iteration is sufficiently large. However we have not been able to provide closed-form analytical expressions for their steady-state errors. This is mainly due to the following reasons : 1) the recursive equations for EMSE are highly nonlinear. It is impossible to estimate the steady state error when iteration n tends to ∞. 2) the so-called energy conservation relation [Sayed 2008] does not hold for these algorithms as the update terms are not linear with respect to the estimated weights. This makes a direct use of energy conservation relation to estimate EMSE impossible. Thus, to derive expressions for these EMSE with reasonable approximation is still an open question and will be valuable to complete the

theoretical analysis of these algorithms.

6.2.2 Online ℓ_1-constrained algorithm

We considered two types of ℓ_1-constrained problem : constant ℓ_1-norm and ℓ_1-regularization. We derived a simple and efficient algorithm for the first problem at the beginning of Chapter 3. The mean weight behavior model was presented in [Chen 2011]. However, the analytical model for the second-order behavior is extremely complicated and has not been successfully derived. For the coherence of the presentation, we did not present the analytical analysis of this algorithm, but it should be no-doubt further investigated. For the ℓ_1-regularized problem, extension with variants of NNLMS and using re-weighted ℓ_1-norm regularization can be further studied.

6.2.3 Kernel-based nonlinear unmixing of hyperspectral images

The proposed nonlinear unmixing algorithm is a supervised unmixing algorithm, which means that it needs to use endmember spectral signature as prior information. To the best of our knowledge, there is an increasing interest in nonlinear estimation of abundance maps, but few algorithms have been proposed to deal with the problem of nonlinear unmixing and determining the endmember signatures simultaneously. Almost all current nonlinear unmixing algorithms require endmembers as algorithm inputs. One potential solution to this problem consists of using a large spectral library rather than directly extracting endmember signatures from scenes. Some sparsity constraints over fractional abundances are clearly necessary, as presented in [Iordache 2012] for the linear case. We intend to solve the problem in the form of

$$\begin{aligned} \psi^* = \arg\min_{\psi} \frac{1}{2}\left(\|\boldsymbol{\alpha}\|^2 + \|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 + \eta\|\boldsymbol{\alpha}\|_1 \\ \text{subject to} \quad e_\ell = r_\ell - \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} + \psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}) \\ \boldsymbol{\alpha} \succeq \mathbf{0} \end{aligned} \tag{6.1}$$

with the endmember matrix $\boldsymbol{M}$ is a large enough spectral library. This strategy can be considered as semi-supervised unmixing. NMF-type unsupervised strategy for our algorithm would be an alternate optimization scheme over abundances and endmember spectra. If this strategy undoubtedly deserves being exploited, note that the corresponding optimization problem is no longer convex, and the risk of overfitting dramatically increases.

6.2.4 Nonlinear unmixing with spatial regularization

It is clear that hyperspectral image unmixing can benefit from both spectral information and spatial information. In Chapter 5, the nonlinear unmixing with spatial information improved the unmixing performance at the cost of an increase of computational burden, as the new algorithm consists of several iterations of the

K-Hype algorithm with additional manipulations. Future works may include more localized or adaptive solution strategies to reduce the computational complexity of unmixing algorithms that jointly consider spatial and spectral information, to accelerate the computational speed without much loss of performance. Regularization with more general information, such as manifold related terms, can also be taken into consideration.

Bibliography

[Acito 2011] N. Acito, M. Diani and G. Corsini. *Signal-dependent noise modeling and model parameter estimation in hyperspectral images.* IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 8, pages 2957–2971, 2011. (Cited on page 111.)

[Aiazzi 2006] B. Aiazzi, L. Alparone, S. Baronti A. Barducci, P. Marcoionni, I. Pippi and M. Selva. *Noise modeling and estimation of hyperspectral data from airborne imaging spectrometer.* Annals of Geophysics, vol. 49, no. 1, 2006. (Cited on page 111.)

[Alligood 1997] K. Alligood, T. Sauer and J. A. Yorke. Chaos : An introduction to dynamical systems. New York : Springer-Verlag, 1997. (Cited on page 22.)

[Almeida 2005] S. J. M Almeida, J. C. M. Bermudez and N. J. Bershad. *A statistical analysis of the affine projection algorithm for unity step size and autoregressive inputs.* IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications, vol. 52, no. 7, pages 1394–1405, 2005. (Cited on page 35.)

[Alparone 2009] L. Alparone, M. Selva, B. Aiazzi, S. Baronti, F. Butera and L. Chiarantini. *Signal-dependent noise modeling and estimation of new-generation imaging spectrometers.* In Proc. IEEE WHISPERS, 2009. (Cited on page 111.)

[Altmann 2011] Y Altmann, A Halimi, N. Dobigeon and J.-Y. Tourneret. *Supervised nonlinear spectral unmixing using a polynomial post nonlinear model for hyperspectral imagery.* In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2011. (Cited on page 79.)

[Aronszajn 1950] N. Aronszajn. *Theory of reproducing kernels.* Transactions of the American Mathematical Society, vol. 68, 1950. (Cited on page 81.)

[Bachmann 2005] C. M. Bachmann, T. L. Ainsworth and R. A. Fusina. *Exploiting manifold geometry in hyperspectral imagery.* IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pages 441–454, 2005. (Cited on page 78.)

[Baldridge 2009] A. M. Baldridge, S. J. Hook, C. I. Grove and G. Rivera. *The ASTER Spectral Library Version 2.0.* Remote Sensing of Environment, vol. 113, no. 711-715, 2009. (Cited on page 108.)

[Bardsley 2008] J. M. Bardsley, J.K. Merikoski and R. Vio. *The stabilizing properties of nonnegativity constraints in least-squares image reconstruction.* International Journal of Pure and Applied Mathematics, vol. 43, no. 1, pages 95–109, 2008. (Cited on page 4.)

[Barzilai 1988] J. Barzilai and J. M. Borwein. *Two-point step size gradient methods.* IMA Journal of Numerical Analysis, vol. 8, no. 1, pages 141–148, 1988. (Cited on page 16.)

[Benvenuto 2010] F. Benvenuto, R. Zanella, L. Zanni and M. Bertero. *Nonnegative least-squares image deblurring : improved gradient projection approaches.* Inverse Problems, vol. 26, no. 1, 2010. (Cited on page 16.)

[Berger 2010] C.R. Berger, Z. Wang, J. Huang and S. Zhou. *Application of compressive sensing to sparse channel estimation.* IEEE on Communications Magazine, vol. 48, no. 11, pages 164–174, 2010. (Cited on page 5.)

[Berry 2007] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca and R. J. Plemmons. *Algorithms and applications for approximate nonnegative matrix factorization.* Computational Statistics and Data Analysis, vol. 52, no. 1, pages 155–173, 2007. (Cited on page 16.)

[Bertsekas 1999] D. Bertsekas. Nonlinear programming. Athena Scientific, second edition édition, 1999. (Cited on pages 84 and 89.)

[Bioucas-Dias 2010] J. M. Bioucas-Dias and A. Plaza. *Hyperspectral unmixing : Geometrical, statistical, and sparse regression-based approaches.* In Proc. of SPIE Image and Signal Processing for Remote Sensing XVI, volume 7830, pages 78300A1–78300A15, 2010. (Cited on page 78.)

[Bioucas-Dias 2012] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader and J. Chanussot. *Hyperspectral unmixing overview : geometrical, statistical, and sparse regression-based approaches.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 2, pages 354– 379, 2012. (Cited on pages 75 and 78.)

[Boardman 1993] J. Boardman. *Atomatic spectral unmixing of AVIRIS data using convex geometry concepts.* In Proc. of JPL Airborne Geoscience Workshop, 1993. (Cited on page 75.)

[Bonnans 1998] J. F. Bonnans and A. Shapiro. *Optimization problems with perturbations : A guided tour.* SIAM review, vol. 40, no. 2, pages 207–227, 1998. (Cited on page 89.)

[Boyd 2004] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004. (Cited on pages 3, 17, 87 and 88.)

[Boyd 2011] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein. *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Foundations and Trends in Machine Learning, vol. 3, no. 1, pages 1–122, 2011. (Cited on page 109.)

[Bro 1997] R. Bro and S. De Jong. *A fast non-negativity-constrained least squares algorithm.* Journal of Chemometrics, vol. 11, no. 5, pages 393–401, 1997. (Cited on page 16.)

[Broadwater 2007] J. Broadwater, R. Chellappa, A. Banerjee and P. Burlina. *Kernel fully constrained least squares abundance estimates.* In Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARRS), 2007. (Cited on pages 79, 91 and 92.)

[Broadwater 2009] J. Broadwater and A. Banerjee. *A comparison of kernel functions for intimate mixture models.* In Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARRS), 2009. (Cited on pages 79 and 91.)

[Bruckstein 2009] A. M. Bruckstein, D. L. Donoho and M. Elad. *From sparse solutions of systems of equations to sparse modeling of signals and images.* SIAM review, vol. 51, no. 1, pages 34–81, 2009. (Cited on page 5.)

[Cai 2009] J. F. Cai, S. Osher and Z. Shen. *Linearized bregman iterations for compressed sensing.* Mathematical of Computation, vol. 78, no. 267, pages 1515–1536, 2009. (Cited on page 60.)

[Cai 2011] D. Cai, X. He, J. Han and T.S. Huang. *Graph regularized nonnegative matrix factorization for data representation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pages 1548–1560, 2011. (Cited on page 79.)

[Calamai 1987] P. H. Calamai and J. J. Moré. *Projected gradient methods for linearly constrained problems.* Mathematical Programming, vol. 39, no. 1, pages 93–116, 1987. (Cited on page 16.)

[Camps-Valls 2005] G. Camps-Valls and L. Bruzzone. *Kernel-based methods for hyperspectral image classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 6, pages 1351–1362, 2005. (Cited on page 79.)

[Candès 2008] E. J. Candès and M.B. Wakin. *An introduction to compressive sampling.* IEEE Signal Processing Magazine, vol. 25, no. 2, pages 21–30, 2008. (Cited on pages 5 and 53.)

[Chan 2009] T.-H. Chan, C.-Y. Chi, Y.-M. Huang and W.-K. Ma. *A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing.* IEEE Transactions on Signal Processing, vol. 57, no. 11, pages 4418–4432, 2009. (Cited on page 75.)

[Chen 2009a] D. Chen and R. Plemmons. *Nonnegativity constraints in numerical analysis.* In Proc. of Symposium on the Birth of Numerical Analysis, 2009. (Cited on pages 3 and 4.)

[Chen 2009b] Y. Chen, Y. Gu and A. O. Hero. *Sparse LMS for system identification.* In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2009. (Cited on page 60.)

[Chen 2010a] Jie Chen, Cédric Richard, Paul Honeine and Jose. C. M. Bermudez. *Non-negative distributed regression for data inference in wireless sensor networks.* In The 44th Asilomar Conference on Signals, Systems, and Computers (ASILOMAR), Pacific Grove (CA), USA, Nov. 2010. (Cited on page 4.)

[Chen 2010b] Y. Chen, Y. Gu and A. O. Hero. *Regularized least-mean-square algorithms. Avaialbe at arxiv.org : http ://arxiv.org/pdf/1012.5066*, 2010. (Cited on page 60.)

[Chen 2011] J. Chen, C. Richard, H. Lanteri, C. Theys and P. Honeine. *Online system identification under non-negativity and ℓ_1-norm Constraints - Algorithm*

and weight behavior analysis. In The 19th European Conference on Signal Processing (EUSIPCO), 2011. (Cited on pages 57, 58 and 119.)

[Chen 2013] J. Chen, C. Richard and P. Honeine. *Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model.* IEEE Transactions on Signal Processing, vol. 61, no. 2, pages 480–492, 2013. (Cited on pages 79, 93, 103, 104 and 109.)

[Cichocki 2009] A. Cichocki, R. Zdunek and A.H. Phan. Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation. Wiley, 2009. (Cited on page 16.)

[Cotter 2002] S.F. Cotter and B.D. Rao. *Sparse channel estimation via matching pursuit with application to equalization.* IEEE Transactions on Communications, vol. 50, no. 3, pages 374–377, 2002. (Cited on page 5.)

[Dasgupta 1990] S. Dasgupta, Jr. Johnson C. R. and A. M. Baksho. *Sign-sign LMS convergence with independent stochastic inputs.* EEE Transactions on Information Theory, vol. 36, no. 1, pages 197–201, 1990. (Cited on page 37.)

[Daubechies 2004] I. Daubechies, M. Defrise and C. De Mol. *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.* Communications on Pure and Applied Mathematics, vol. 57, no. 11, pages 1413–1457, 2004. (Cited on page 60.)

[Dobigeon 2009] N. Dobigeon, S. Moussaoui, M. Coulon, J.-Y. Tourneret and A. O. Hero. *Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery.* IEEE Transactions on Signal Processing, vol. 57, no. 11, pages 4355–4368, 2009. (Cited on page 78.)

[Dòpido 2011] I. Dòpido, M. Zortea, A. Villa, A. Plaza and P. Gamba. *Unmixing Prior to Supervised Classification of Remotely Sensed Hyperspectral Images.* IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 4, pages 760–764, 2011. (Cited on page 114.)

[Du 2009] Q. Du, L. Zhang and N. Raksuntorn. *Improving the quality of extracted endmembers.* In Proc. of WHISPERS, 2009. (Cited on page 103.)

[Duchi 2008] J. Duchi, S. Shalev-Shwartz, Y. Singer and T. Chandra. *Efficient projections onto the ℓ_1-ball for learning in high dimensions.* In Proc. of the 25th international conference on Machine learning, 2008. (Cited on pages 54 and 57.)

[Eches 2010] O. Eches, N. Dobigeon, C. Mailhes and J.-Y. Tourneret. *Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery.* IEEE Transactions on Image Processing, vol. 19, no. 6, pages 1403–1413, 2010. (Cited on page 78.)

[Eches 2011] O. Eches, N. Dobigeon and J.-Y. Tourneret. *Enhancing hyperspectral image unmixing with spatial correlations.* IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 11, pages 4239–4247, 2011. (Cited on pages 102 and 103.)

[Eckstein 1992] J. Eckstein and D. P. Bertsekas. *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators.* Mathematical Programming, vol. 55, no. 1, pages 293–318, 1992. (Cited on page 108.)

[Elad 2010] M. Elad. Sparse and redundant representations : from theory to applications in signal and image processing. Springer, 2010. (Cited on page 5.)

[Eweda 1999] E. Eweda. *Transient and tracking performance bounds of the sign-sign algorithm.* IEEE Transactions on Signal Processing, vol. 47, no. 8, pages 2200–2210, 1999. (Cited on page 37.)

[Fauvel 2012] M. Fauvel, Y. Tarabalka, J. A Benediktsson, J. Chanussot and J Tilton. *Advances in spectral-spation classification of Hyperspectral Images.* Proceedings of IEEE, 2012. (Cited on page 102.)

[Figueiredo 2005] M. Figueiredo and R. D. Nowak. *A bound optimization approach to wavelet-based image deconvolution.* In Proc. of IEEE International Conference on Image Processing (ICIP), 2005. (Cited on page 60.)

[Figueiredo 2007] M. Figueiredo, R. D. Nowak and S. J. Wright. *Gradient projection for sparse reconstruction : Application to compressed sensing and other inverse problems.* IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pages 586–597, 2007. (Cited on page 60.)

[Friedman 2007] J. Friedman, T. Hastie, H. Höfling and R. Tibshirani. *Pathwise coordinate optimization.* The Annals of Applied Statistics, vol. 2, no. 1, pages 302–332, 2007. (Cited on page 59.)

[G72 1994] *ITU-T Recommendation G.726 (former CClTT Recommendation G.721)*, 1994. (Cited on page 33.)

[Gaines 1977] B. R. Gaines. *System Identification, approximation and complexity.* International Journal of General Systems, vol. 3, no. 3, pages 145–174, 1977. (Cited on page 1.)

[Gevers 2006] M. Gevers. *A personal view of the development of system identification : A 30-year journey through an exciting field.* IEEE Control Systems Magazine, vol. 26, no. 6, pages 93–105, 2006. (Cited on page 2.)

[Goldstein 2009] T. Goldstein and S. Osher. *The split Bregman method for L1 regularized problems.* SIAM Journal on Imaging Sciences, vol. 2, no. 2, pages 323–343, 2009. (Cited on page 105.)

[Gönen 2011] M. Gönen and E. Alpaydın. *Multiple kernel learning algorithms.* Journal of Machine Learning Research, vol. 12, pages 2211–2268, 2011. (Cited on pages 5 and 54.)

[Gu 2009] Y. Gu, J. Jin and S. Mei. *ℓ_0-norm constraint LMS algorithm for sparse system identification.* IEEE Signal Processing Letters, vol. 16, no. 7, pages 774–777, 2009. (Cited on page 60.)

[Guan 2011] N. Guan, D. Tao, Z. Luo and B. Yuan. *Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent.* IEEE

Transactions on Image Processing, vol. 20, no. 7, pages 2030–2048, 2011. (Cited on page 79.)

[Guilfoyle 2001] K. J. Guilfoyle, M. L. Althouse and C.-I. Chang. *A quantitative and comparative analysis of linear and nonlinear spectral mixture models using radial basis function neural networks.* IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 10, pages 2314–2318, 2001. (Cited on page 79.)

[Guo 2009] Z. Guo, T. Wittman and S. Osher. *L1 unmixing and its application to hyperspectral image enhancement.* In Proc. SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, volume 7334, pages 73341M–73341M, 2009. (Cited on page 78.)

[Hale 2007] E. T. Hale, W. Yin and Y. Zhang. *A fixed-point continuation method for ℓ_l1-regularized minimization with applications to compressed sensing.* Rapport technique, CAAM TR07-07, Rice University, 2007. (Cited on page 59.)

[Halimi 2011] A. Halimi, Y. Altman, N. Dobigeon and J.-Y. Tourneret. *Nonlinear unmixing of hyperspectral images using a generalized bilinear model.* IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 11, pages 4153–4162, 2011. (Cited on pages 76, 79, 85 and 92.)

[Hapke 1981] B. Hapke. *Bidirectional reflectance spectroscopy, 1, Theory.* Journal of Geophysical Research, vol. 86, no. B4, pages 3039–3054, 1981. (Cited on pages 76 and 78.)

[Haussler 1999] D. Haussler. *Convolution kernels on discrete structures.* Rapport technique, Computer Science Department, University of California at Santa Cruz, 1999. (Cited on page 83.)

[Heesung 2005] K. Heesung and N. M. Nasrabadi. *Kernel orthogonal subspace projection for hyperspectral signal classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 12, pages 2952–2962, 2005. (Cited on page 79.)

[Heinz 2001] D. C. Heinz and C.-I. Chang. *Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery.* IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 3, pages 529–545, 2001. (Cited on pages 4, 77, 92 and 109.)

[Heylen 2011] R. Heylen, D. Burazerovic and P. Scheunders. *Nonlinear spectral unmixing by geodesic simplex volume maximization.* IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 3, pages 534–542, 2011. (Cited on page 78.)

[Honeine 2012] P. Honeine and C. Richard. *Geometric unmixing of large hyperspectral images : a barycentric coordinate approach.* IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 6, pages 2185–2195, 2012. (Cited on page 75.)

[Hubscher 2003] P. I. Hubscher and J. C. M. Bermudez. *An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm.* IEEE Transac-

tions on Signal Processing, vol. 51, no. 3, pages 664–671, 2003. (Cited on pages 27 and 28.)

[Iordache 2011] M. D. Iordache, J. M. Bioucas-Dias and A. Plaza. *Sparse unmixing of hyperspectral data*. IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 6, pages 2014–2039, 2011. (Cited on pages 5 and 78.)

[Iordache 2012] M. D. Iordache, J. M. Bioucas-Dias and A. Plaza. *Total variation spatial regularization for sparse hyperspectral unmixing*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (to appear), vol. 50, no. 11, pages 4484–4502, 2012. (Cited on pages 102, 103, 104, 108, 109 and 119.)

[Jia 2007] S. Jia and Y. Qian. *Spectral and spatial complexity-based hyperspectral unmixing*. IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 12, pages 3867–3879, 2007. (Cited on page 102.)

[Johansen 1998] T. A. Johansen. *Constrained and regularized system identification*. Modeling Identification and Control, vol. 19, pages 109–116, 1998. (Cited on page 2.)

[Jutten 2003] C. Jutten and J. Karhunen. *Advances in nonlinear blind source separation*. In Proc. International Symposium on Independent Component Analysis and Blind Signal Separation (ICA), pages 245–256, 2003. (Cited on page 92.)

[Keshava 2002] N. Keshava and J. F. Mustard. *Spectral unmixing*. IEEE Signal Processing Magazine, vol. 19, no. 1, pages 44–57, 2002. (Cited on pages 4, 5, 54 and 75.)

[Khajehnejad 2011] M. A. Khajehnejad, A. G. Dimakis, W. Xu and B. Hassibi. *Sparse Recovery of Nonnegative Signals With Minimal Expansion*. IEEE Transactions on Signal Processing, vol. 59, no. 1, pages 196–208, 2011. (Cited on page 5.)

[Koike 1998] S. Koike. *Analysis of the sign-sign algorithm based on Gaussian distributed tap weights*. In Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1998. (Cited on pages 33 and 37.)

[Lanckriet 2004] R. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui and M. Jordan. *Learning the kernel matrix with semidefinite programming*. Journal of Machine Learning Research, vol. 5, pages 27–72, 2004. (Cited on page 86.)

[Lantéri 2001] H. Lantéri, M. Roche, O. Cuevas and C. Aime. *A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints*. Signal Processing, vol. 81, no. 5, pages 945–974, 2001. (Cited on page 17.)

[Lawson 1995] C. L. Lawson and R. J. Hanson. Solving least squares problems. Society for Industrial and Applied Mathematics, 1995. (Cited on page 16.)

[Lee 1999] D. D. Lee and H. S. Seung. *Learning the parts of objects by non-negative matrix factorization.* Nature, vol. 401, no. 6755, pages 788–791, 1999. (Cited on page 16.)

[Lee 2001] D. D. Lee and H. S. Seung. *Algorithms for non-negative matrix factorization.* Advances in neural information processing systems, NIPS, pages 556–562, 2001. (Cited on pages 16 and 17.)

[Li 2008] J. Li and J. M. Bioucas-Dias. *Minimum volume simplex analysis : a fast algorithm to unmix hyperspectral data.* In Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARRS), 2008. (Cited on page 75.)

[Li 2011] J. Li, J. M. Bioucas-Dias and A. Plaza. *Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields".* IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 3, pages 809–823, 2011. (Cited on pages 102 and 103.)

[Lin 2006] Y Lin and D. D. Lee. *Bayesian Regularization and Nonnegative Deconvolution for Room Impulse Response Estimation.* IEEE Transactions on Signal Processing, vol. 54, no. 3, pages 839–847, Mar. 2006. (Cited on page 16.)

[Lin 2007a] C. J. Lin. *On the convergence of multiplicative update algorithms for nonnegative matrix factorization.* IEEE Transactions on Neural Networks, vol. 18, no. 6, pages 1589–1596, 2007. (Cited on page 17.)

[Lin 2007b] C. J. Lin. *Projected gradient methods for nonnegative matrix factorization.* Neural Computation, vol. 19, no. 10, pages 2756–2779, 2007. (Cited on page 16.)

[Ljung 1999] L. Ljung. System identification : theory for the user. Prentice-Hall PTR, Upper Saddle River, New Jersey, 1999. (Cited on page 2.)

[Ljung 2010] L. Ljung. *Perspectives on system identification.* Annual Reviews in Control, vol. 34, no. 1, pages 1–12, 2010. (Cited on page 1.)

[Luenberger 2008] D. G. Luenberger and Y. Ye. Linear and nonlinear programming. Springer, 2008. (Cited on pages 3 and 84.)

[Ma 2010] L. Ma, M.M. Crawford and J. Tian. *Local manifold learning-based k-nearest-neighbor for hyperspectral image classification.* IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 11, pages 4099–4109, 2010. (Cited on page 79.)

[Martin 2011] G. Martin and A. Plaza. *Region-Based spatial preprocessing for endmember extraction and spectral unmixing.* IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 4, pages 745–749, 2011. (Cited on page 102.)

[May 1976] R. M. May. *Simple mathematical models with very complicated dynamics.* Nature, vol. 261, no. 10, pages 459–467, 1976. (Cited on page 22.)

[Meola 2011] J. Meola, M.-T. Eismann, R.-L. Moses and J.-N. Ash. *Modeling and estimation of signal-dependent noise in hyperspectral imagery.* Applied Optics, vol. 50, no. 21, pages 3829–3846, 2011. (Cited on page 111.)

[Mercer 1909] J. Mercer. *Functions of positive and negative type and their connection with the theory of integral equations.* Philos. Trans. Roy. Soc. London Ser. A, vol. 209, pages 415–446, 1909. (Cited on page 81.)

[Miao 2007] L. Miao and H. Qi. *Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization.* IEEE Transactions on Geoscience and Remote Sensing, vol. 45, no. 3, pages 765–777, 2007. (Cited on page 75.)

[Miller 1964] K. S. Miller. Multidimensional gaussian distributions. Wiley, New York, 1964. (Cited on page 44.)

[Minkoff 2001] J. Minkoff. *Comment : On the unnecessary assumption of statistical independence between reference signal and fillter weights in feedforward adaptive systems.* IEEE Transactions on Signal Processing, vol. 49, no. 5, page 1109, 2001. (Cited on page 26.)

[Moussaoui 2006] S. Moussaoui, D. Brie, A. Mohammad-Djafari and C. Carteret. *Separation of non-negative mixture of non-negative sources using a bayesian approach and MCMC sampling.* IEEE Transactions on Signal Processing, vol. 54, no. 11, pages 4133–4145, 2006. (Cited on page 16.)

[Moussaoui 2008] S. Moussaoui, H. Hauksdottir, F. Schmidt, C. Jutten, J. Chanussot, D. Brie, S. Douté and J. A. Benediktsson. *On the decomposition of Mars hyperspectral data by ICA and bayesian positive source separation.* Neurocomputing, vol. 71, no. 10, pages 2194–2208, 2008. (Cited on page 78.)

[Narkiss 2005] G. Narkiss and M. Zibulevsky. *Sequential subspace optimization method for large-scale unconstrained problems.* Rapport technique, The Technion, Haifa, Israel, Tech. Rep. CCIT No.559, 2005. (Cited on page 59.)

[Nascimento 2005] J. M. P. Nascimento and J. M. Bioucas-Dias. *Vertex component analysis : a fast algorithm to unmix hyperspectral data.* IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 4, pages 898–910, April 2005. (Cited on pages 75 and 96.)

[Nascimento 2009] J. M. P. Nascimento and J. M. Bioucas-Dias. *Nonlinear mixture model for hyperspectral unmixing.* In Proc. of SPIE : Image and Signal Processing for Remote Sensing XV, 2009. (Cited on pages 78 and 86.)

[Nascimento 2010] J. M. P. Nascimento and J. M. Bioucas-Dias. *Unmixing hyperspectral intimate mixtures.* In Proc of SPIE : Image and Signal Processing for Remote Sensing XVI, 2010. (Cited on page 78.)

[Nesterov 1994] Y. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming.* In Studies in Applied Mathematics, volume 13. Philadelphia, PA : SIAM, 1994. (Cited on page 59.)

[Nesterov 2007] Y. Nesterov. *Gradient methods for minimizing composite objective function.* Rapport technique, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007. (Cited on page 60.)

[O'Grady 2008] P. D. O'Grady and Rickard S. T. *Compressive sampling of non-negative signals.* In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2008. (Cited on page 5.)

[Pauca 2006] V. P. Pauca, J. Piper and Robert J. Plemmons. *Nonnegative matrix factorization for spectral data analysis.* Linear Algebra and its Applications, vol. 416, no. 1, pages 29–47, 2006. (Cited on page 4.)

[Perrin 2008] D. Perrin. *La suite logistique et le chaos.* Rapport technique, Dép. Math. d'Orsay., Univ. de Paris-Sud, France, 2008. (Cited on page 22.)

[Plaza 2004] J. Plaza, P. Martínez, R. Pérez and A. Plaza. *Nonlinear neural network mixture models for fractional abundance estimation in AVIRIS hyperspectral images.* In Proc. of XIII JPL Airborne Earth Science Workshop, 2004. (Cited on page 80.)

[Plaza 2011a] A. Plaza, Q. Du, J. M. Bioucas-Dias, X. Jia and F. Kruse. *Foreword to the special issue on spectral unmixing of remotely sensed data.* IEEE Transactions on Geoscience and Remote Sensing, vol. 22, no. 9, pages 1419–1434, 2011. (Cited on page 101.)

[Plaza 2011b] A. Plaza, G. Martín, J. Plaza, M. Zortea and S. Sánchez. *Recent developments in endmember extraction and spectral unmixing.* In Optical Remote Sensing : Advances in Signal Processing and Exploitation Techniques, pages 235–267. Springer, 2011. (Cited on page 75.)

[Plumbley 2003] M. D. Plumbley. *Algorithms for nonnegative independent component analysis.* IEEE Transactions on Neural Networks, vol. 14, no. 3, pages 534–543, 2003. (Cited on page 16.)

[Price 1958] R. Price. *A useful theorem for nonlinear devices having Gaussian inputs.* IRE Transactions on Information Theory, vol. 4, no. 2, pages 69–72, 1958. (Cited on page 37.)

[Puetter 2005] R. C. Puetter, T. R. Gosnell and A. Yahil. *Digital image reconstruction : Deblurring and denoising.* Annual Review of Astronomy and Astrophysics, vol. 43, pages 139–194, 2005. (Cited on page 3.)

[Rakotomamonjy 2008] A. Rakotomamonjy, F. Bach, S. Canu and Y. Granvalet. *SimpleMKL.* Journal of Machine Learning Research, vol. 9, pages 2491–2521, 2008. (Cited on pages 86, 87, 89 and 90.)

[Raksuntorn 2010] N. Raksuntorn and Q. Du. *Nonlinear spectral mixture analysis for hyperspectral imagery in an unknown environment.* IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 4, pages 836–840, 2010. (Cited on pages 78, 86 and 92.)

[Rao 2005] Y. N. Rao, S. P. Kim, J. C. Sanchez, D. Erdogmus, J. C. Principe, J. M. Carmena, M. A. Lebedev and M. A. Nicolelis. *Learning mappings in brain machine interfaces with echo state networks.* In Proc. of IEEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2005. (Cited on page 54.)

[Ray 1996] T. W. Ray and B. C. Murray. *Nonlinear spectral mixing in desert vegetation*. Remote Sensing of Environment, vol. 55, no. 1, pages 59–64, 1996. (Cited on page 75.)

[Richard 2009] C. Richard, J.-C. M. Bermudez and P. Honeine. *Online prediction of time series data with kernels*. IEEE Transactions on Signal Processing, vol. 57, no. 3, pages 1058–1067, 2009. (Cited on page 95.)

[Rogge 2007] D. M. Rogge, B. Rivard, J. Zhang, A. Sanchez, J. Harris and J. Feng. *Integration of spatial-spectral information for the improved extraction of endmembers*. Remote Sensing of Environment, vol. 110, no. 3, pages 287–303, 2007. (Cited on page 102.)

[Rosen 1960] J. B. Rosen. *The Gradient Projection Method for Nonlinear Programming. Part 1 : Linear Constraints*. Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 1, pages 181–217, 1960. (Cited on page 16.)

[Samson 1983] C. Samson and V. U. Reddy. *Fixed Point Error Analysis of the Normalized Ladder Algorithms*. IEEE Transactions on Acoustics, Speech, Signal Processing, vol. 31, no. 10, pages 1177–1191, 1983. (Cited on page 35.)

[Saul 2003] L. K. Saul, F. Sha and D. D. Lee. *Statistical signal processing with nonnegativity constraints*. In Proc. of EuroSpeech, 2003. (Cited on page 4.)

[Sayed 2008] A. Sayed. Adaptive filters. New York : Wiley-Interscience, 2008. (Cited on pages 14, 20, 24, 50 and 118.)

[Schölkopf 1999] B. Schölkopf, J. C. Burges and A. J. Smola. Advances in kernel methods. MIT Press, Cambridge, MA, 1999. (Cited on pages 81 and 104.)

[Suykens 2002] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle. Least squares support vector machines. World Scientific, Singapore, 2002. (Cited on page 82.)

[Themelis 2012] K. E. Themelis, A. A. Rontogiannis and K. D. Koutroumbas. *A Novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing*. IEEE Transactions on Signal Processing, vol. 60, no. 2, pages 585–599, 2012. (Cited on page 78.)

[Theodoridis 2011] S. Theodoridis, K. Slavakis and I. Yamada. *Adaptive Learning in a World of Projections*. IEEE Signal Processing Magazine, vol. 28, no. 1, pages 97 –123, Jan. 2011. (Cited on page 50.)

[Tibshirani 1996] R. Tibshirani. *Regression shrinkage and selection via the lasso*. Journal of the Royal Statistical Society (Serie B), vol. 58, no. 1, pages 267–288, 1996. (Cited on page 108.)

[Van Benthem 2004] M. H. Van Benthem and M. R. Keenan. *Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems*. Journal of Chemometrics, vol. 18, pages 441–450, 2004. (Cited on page 16.)

[Vapnik 1995] V. N. Vapnik. The nature of statistical learning theory. Springer, New York, NY, 1995. (Cited on pages 81 and 104.)

[Vapnik 2000] V. N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York Incorporated, 2000. (Cited on page 3.)

[Warner 1997] T. A. Warner and Shank M. C. *Spatial autocorrelation analysis of hyperspec- tral imagery for feature selection.* Remote Sensing of Environment, vol. 60, no. 1, pages 58–70, 1997. (Cited on page 102.)

[Waterschoot 2011] T.-V. Waterschoot and G. Leus. *Static field estimation using a wireless sensor network based on the finite element method.* In Proc. of 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011. (Cited on page 4.)

[Winter 1999] M. E. Winter. *N-FINDR : an algorithm for fast autonomous spectral end-member determination in hyperspectral data.* In Proc. of SPIE Spectrometry V, volume 3753, pages 266–277, 1999. (Cited on page 75.)

[Wright 2009] S. J. Wright, R. D. Nowak and M. Figueiredo. *Sparse reconstruction by separable approximation.* IEEE Transactions on Signal Processing, vol. 57, no. 7, pages 2479–2493, 2009. (Cited on page 60.)

[Yeredor 2006] A. Yeredor. *On the role of constraints in system identification.* In Proc. of the 4th International Workshop on Total Least Squares and Errors-in-variables Modeling, 2006. (Cited on page 2.)

[Yin 2008] W. Yin, S. Osher, D. Goldfarb and J. Darbon. *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing.* SIAM Journal on Imaging Sciences, vol. 1, no. 1, pages 143–168, 2008. (Cited on page 60.)

[Zadeh 1962] L. A. Zadeh. *From circuit theory to system theory.* Proc. of Institution of Radio Engineers, vol. 50, no. 5, pages 856–865, 1962. (Cited on page 1.)

[Zare 2011] A. Zare. *Spatial-spectral unmixing using fuzzy local information.* In Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARRS), 2011. (Cited on pages 102 and 103.)

[Zortea 2009] M. Zortea and A. Plaza. *Spatial preprocessing for endmember extraction.* IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 8, pages 2679–2693, 2009. (Cited on page 102.)

[Zymnis 2007] A. Zymnis, S. J. Kim, J. Skaf, M. Parente and S. Boyd. *Hyperspectral image unmixing via alternating projected subgradients.* In Proc. of 41th Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2007. (Cited on pages 102, 103 and 104.)

Annexe A

Résumé en Français

A.1 Introduction

A.1.1 Identification de système et rôles des contraintes

L'identification de système consiste à appliquer ou observer des signaux en entrée d'un système et en analyser la sortie dans le but d'obtenir un modèle mathématique du système. Il s'agit d'un sujet vaste par les nombreuses techniques envisageables, selon les propriétés des systèmes étudiés. Dans de nombreuses applications, l'identification de système consiste à estimer un ensemble de coefficients ou paramètres qui caractérisent un système inconnu décrit par une fonction mathématique sous-jacente liant les entrées aux sorties. Cette catégorie de méthodes d'identification, dite paramétrique, est fréquemment formulée sous la forme d'un problème d'optimisation d'un critère prédéfini. Ce critère est minimisé par rapport aux paramètres à estimer, étant donné les informations disponibles sur les entrées et sorties.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \tag{A.1}$$

avec $\boldsymbol{\theta}$ le vecteur des paramètres à estimer et Ψ le critère. Un critère couramment utilisé est le coût quadratique, optimum au sens du maximum de vraisemblance sous hypothèses gaussiennes.

Au lieu de laisser les paramètres totalement libres et de s'appuyer uniquement sur les données, il peut être intéressant en pratique d'introduire des contraintes dans l'espace des paramètres, à la manière d'informations a priori. Dans ces conditions, l'idée générale du problème d'identification paramétrique est de déterminer un ensemble de paramètres optimums au sens du critère considéré soumis à des contraintes. En introduisant des contraintes sur l'espace des paramètres, le problème d'identification se réécrit en restreignant l'espace des paramètres à un sous-espace de solutions admissibles Θ

$$\begin{aligned} \boldsymbol{\theta}^* = &\arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \\ &\text{sous contrainte } \boldsymbol{\theta} \in \Theta \end{aligned} \tag{A.2}$$

Comme évoqué précédemment, l'introduction de contraintes peut être notamment motivée par les raisons suivantes

– Intégrer des informations a priori sur le système dans le but d'améliorer la précision de l'estimation et l'interprétabilité des résultats, afin de réduire la

taille de l'ensemble des solutions admissibles. Ceci comprend des intervalles présumés pour des variables, les positions de zéros ou pôles de réponse du système, la régularité des signaux, etc. Il s'agit généralement d'une motivation fondamentale pour l'introduction de contraintes.
- Éviter des résultats absurdes et non interprétables physiquement. Par exemple, on peut citer les contraintes de non-négativité qui seraient imposées à des solutions correspondant à des grandeurs physiques, associées à des calculs de fréquence, d'intensités de pixels, ou encore de concentrations chimiques. Ceci est une autre motivation fréquente pour l'ajout de contraintes.
- Éviter une solution triviale dans la minimisation du critère de performance considéré, où le problème d'optimisation ne peut pas conduire à une solution utile sans avoir exclu au préalable les solutions triviales. Par exemple, la contrainte de norme unité est souvent imposée sur les équations homogènes car on ne s'intéresse évidemment pas à la solution triviale 0.
- Imposer certaines propriétés structurelles liées aux signaux considérés, comme la structure de Toeplitz imposée sur une matrice, ou la parcimonie imposée sur des coefficients de décomposition des signaux sur un dictionnaire.
- D'autres éléments comme garantir la stabilité du système estimé, réduire le biais, etc.

Considérons le cas où la fonction Ψ et l'ensemble Θ sont convexes, le problème d'optimisation est alors convexe et possède une solution unique. Le problème ci-dessous, formulé en utilisant des contraintes d'inégalité et de l'égalité est souvent étudié

$$\begin{aligned} \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}) \\ \text{sous contrainte } g_i(\boldsymbol{\theta}) \leq 0 \\ h_i(\boldsymbol{\theta}) = 0 \end{aligned} \tag{A.3}$$

avec des fonctions Ψ, g_i convexes et h_i affines. Cette formulation standard a été largement étudiée dans la littérature sur l'optimisation convexe. De nombreux problèmes pratiques, y compris ceux qui seront discutés dans notre travail, peuvent être formulés sous cette forme.

Dans cette thèse, nous nous sommes concentrés sur les problèmes d'identification sous contraintes de non-négativité et de norme ℓ_1.

A.1.2 Contraintes de non-négativité en identification

En raison de caractéristiques physiques inhérentes à certains systèmes étudiés, la non-négativité des paramètres à estimer est une information a priori parfois naturelle qu'il convient d'exploiter afin de se prémunir contre d'éventuels résultats non-interprétables. Il en est par exemple ainsi dans l'analyse de rayonnements ou l'estimation de la concentration de composés chimiques. La contrainte de non-négativité a récemment connu un regain d'intérêt en traitement du signal et des images. L'en-

semble des solutions admissibles s'écrit

$$\Theta_+ = \{\boldsymbol{\theta} \,|\, \theta_i \geq 0, \quad \forall i\} \tag{A.4}$$

Cette contrainte a reçu une attention considérable dans la communauté du traitement du signal au cours de la dernière décennie. Elle trouve des applications dans de nombreux problèmes. Quelques-uns sont cités ci-après.

Dans les problèmes de machines à vecteurs de support, on s'attèle à minimiser l'erreur de prédiction et à maintenir en-même temps la capacité de généralisation, ce qui revient pour ce problème à minimiser la norme de régresseurs sous les contraintes définies par les règles de classification ou de régression. Souvent, ce problème est résolu dans son espace dual, ce qui conduit à des problèmes d'optimisation quadratiques sous les contraintes de non-négativité imposées aux variables duales. Ainsi a-t-on rencontré ce type de problème dans la formulation duale de nos algorithmes de démélange non-linéaire pour des données hyperspectrales.

Les images numériques sont représentées par des matrices dont les composantes sont non-négatives, en raison de la nature des intensités des pixels. La non-négativité peut donc être une contrainte souhaitable pour le traitement d'image. Dans les problèmes de déconvolution d'image, si la fonction d'étalement de point, ou opérateur de convolution, est connu, l'image originale estimée est souvent reconstituée en résolvant un problème de moindres-carrés avec contraintes de non-négativité. La littérature a montré que l'application de cette contrainte peut conduire à une solution plus satisfaisante.

Dans le contexte de la télédétection, en raison de la réflexion des photons et de la diversité des matériaux, le spectre observé en chaque pixel est un mélange de signatures spectrales des matériaux présents. La démélange spectral vise à décomposer chaque pixel en des spectres de composés purs, et à estimer les abondances associées à chaque matériau. Pour l'interprétabilité physique, les signatures spectrales estimées et les fractions d'abondance doivent satisfaire les contraintes de non-négativité, à laquelle s'ajoute la contrainte de somme unité fréquemment imposée de façon simultanée sur les abondances. Considérant le modèle de mélange linéaire, une façon de déterminer les signatures spectrales des matériaux, et conjointement les abondances, est de résoudre le problème de factorisation en matrice non-négative. Dans le cas où les signatures ont été déterminées préalablement, le démélange se réduit à un problème des moindres carrés non-négatifs. Dans les problèmes de démélange non-linéaires, les contraintes de non-négativité sont également à considérer.

Dans certaines applications de réseaux de capteurs sans fil, comme la surveillance de la concentration d'un composé chimique ou de la diffusion thermique d'une source de chaleur, les champs sont décrits par des valeurs non-négatives. Pour faciliter la reconstruction du champ à partir des mesures obtenues avec les capteurs, il est généralement supposé que le champ s'exprime par la somme pondérée de fonctions de base. Pour assurer que la solution inférée est non-négative en n'importe quel point, les coefficients de poids doivent se soumettre à la contrainte de non-négativité si les fonctions de base utilisées sont non-négatives.

Au delà de ces quelques exemples, il en existe de nombreuses autres impliquant la contrainte de non-négativité, telles que la reconnaissance vocale, la reconnaissance de texte, etc.

A.1.3 Contrainte de norme ℓ_1 en identification

Une autre contrainte largement discutée au sein de la communauté du traitement du signal est celle concernant la norme ℓ_1, due à ses propriétés favorables pour structurer la solution. Dans cette thèse nous aborderons deux types de contraintes avec celle-ci.

La première est la contrainte de norme ℓ_1 constante, qui restreint l'ensemble des solutions à la sphère ℓ_1, c'est-à-dire

$$\Theta_{\ell_1} = \{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\|_1 = \delta\} \tag{A.5}$$

avec $\|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i|$. Si cette contrainte est combinée avec la contrainte de non-négativité, la solution est alors restreinte à l'orthant positif sur cette sphère, c'est-à-dire que la contrainte devient une exigence de somme

$$\begin{aligned} \Theta_{\ell_1^+} &= \Theta_{\ell_1} \cap \Theta_+ \\ &= \{\boldsymbol{\theta} \mid \sum_i \theta_i = \delta \text{ et } \theta_i \geq 0 \quad \forall i\} \end{aligned} \tag{A.6}$$

Cette contrainte joue un rôle important dans certains problèmes, en particulier le cas $\delta = 1$ qui signifie que chaque valeur de θ_i représente la proportion du mode associé. Par exemple, dans le problème de l'apprentissage multi-noyau, la combinaison de noyaux candidats est utilisée en lieu et place d'un seul noyau afin d'améliorer la précision de la classification / régression. Pour garantir le caractère défini-positif du noyau construit et éviter les solutions triviales, il faut que les coefficients de pondération à estimer soient un élément de $\Theta_{\ell_1^+}$. Un autre exemple est le problème de démélange des données hyperspectrales, où on a pour but d'estimer les abondances associés aux matériels. Pour l'interprétabilité physique, il est souvent nécessaire d'imposer deux contraintes : la non-négativité, et somme-à- un des abondances. Ceci implique que le vecteur des abondances est un élément dans l'ensemble $\Theta_{\ell_1^+}$.

Une autre contrainte importante consiste à limiter la norme ℓ_1 du vecteur de paramètres estimés, via la contrainte d'inégalité

$$\Theta_{\ell_1'} = \{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\|_1 \leq \delta\} \tag{A.7}$$

Cette contrainte a connu un intérêt général de la communauté, avec la théorie de «compressed sensing» (CS). La mesure exacte de la parcimonie par la norme ℓ_0 est démontré un problème NP-difficile. La contrainte $\Theta_{\ell_1'}$ permet une reconstruction exacte des signaux avec une forte probabilité dans certains scénarios. Elle est en outre beaucoup plus aisée à manipuler que la norme ℓ_0. Les algorithmes basés sur la norme ℓ_1 possèdent une large gamme d'applications en raison de propriété favorisant la parcimonie. Dans le problème de l'estimation de canal, de nombreux

systèmes réels admettent des représentations parcimonieuses. Les exemples incluent les canaux multiples de communication sans fil, où les réflexions atteignant le récepteur avec de longs retards. Beaucoup de ces applications nécessitent des techniques d'estimation adaptative compte tenu de la complexité de calcul minimale recherchée, de la dynamique temporelle des variables, et du grand nombre de paramètres potentiels à estimer. Les canaux de communication sans fil constituent un exemple typique de cette situation. Dans le problème de démélange spectrale, le nombre de matériaux à déterminer dans la scène ne représente qu'un petit sous-ensemble d'une bibliothèque comportant de nombreux candidats. La régression parcimonieuse est une direction en cours d'exploration pour la démélange spectral. Le problème est formulé de sorte à reconstruire les spectres de réflectance observés à l'aide d'un mélange comportant un nombre limité de signatures spectrales issues de la bibliothèque contenant l'ensemble des candidats potentiels. La combinaison de la contrainte de non-négativité Θ_+ et la contrainte de norme $\ell_1 - \Theta_{\ell_1'}$ a également été étudiée dans le contexte de l'apprentissage automatique.

Il convient de noter enfin que cette contrainte est aussi étroitement liée à la contrainte de non-négativité car elle peut être reformulée avec deux vecteurs non-négatifs tels que

$$\Theta_{\ell_1'} = \{\boldsymbol{\theta} = \boldsymbol{\theta}^+ - \boldsymbol{\theta}^- \,\big|\, \mathbf{1}^\top\boldsymbol{\theta}^+ + \mathbf{1}^\top\boldsymbol{\theta}^- \leq \delta \text{ and } \boldsymbol{\theta}^+ \in \Theta_+, \boldsymbol{\theta}^- \in \Theta_+\} \tag{A.8}$$

Nous utiliserons ce principe afin d'élaborer des méthodes d'identification en-ligne favorisant la parcimonie de la solution.

A.2 Motivations

L'objectif de cette thèse est d'étudier les théories et algorithmes pour l'identification de systèmes sous contraintes, en particulier, les contraintes de non-négativité et de norme ℓ_1. La motivation de cette thèse est double en ce qui concerne le cas linéaire et le cas non-linéaire. Dans le cas linéaire, la thèse se concentre sur l'élaboration d'algorithmes en-ligne, où les méthodes de filtrage adaptatif occupent une situation centrale. Dans le cas non-linéaire, la thèse s'intéresse à un problème particulier d'identification, le démélange non-linéaire des images hyperspectrales.

Le problème d'identification d'un système linéaire sous contraintes de non-négativité peut être formulé comme un problème d'optimisation quadratique. Plusieurs stratégies ont été proposées afin de résoudre le problème de minimisation du coût quadratique sous contraintes de non-négativité, incluant les méthodes de contraintes actives, les méthodes de points intérieurs, les méthodes de gradient-projeté, etc. Récemment, les méthodes multiplicatives ont connu un nouveau regain d'intérêt grâce au problème de factorisation en matrices non-négatives. De nombreuses approches ont également proposées pour la contrainte de norme ℓ_1, y compris les méthodes de points intérieurs, la méthode d'homothopie et ses variantes, et les algorithmes gloutons. Ces algorithmes nécessitent cependant un traitement hors-ligne, ce qui n'est pas approprié pour des problèmes d'identification de système

en-ligne. Par conséquent, cette thèse étudie d'abord des algorithmes en-ligne pour la contrainte de non-négativité par l'utilisation de la structure de filtre adaptatif, qui traite les entrées de manière séquentielle. Après cela, nous étendons cet algorithme afin de résoudre le problème avec la contrainte de norme ℓ_1, également en-ligne.

Dans le cas non-linéaire, au lieu de résoudre un problème quadratique spécifique, nous nous concentrons sur la façon de modéliser un système non-linéaire sous de telles contraintes. Comme présenté brièvement précédemment, en raison de l'interprétation physique des abondances dans les images hyperspectrales, les contraintes de non-négativité et de somme-à-un sont souvent imposées. Dans le cadre d'un modèle de démélange linéaire, ces contraintes nous amènent à un problème de programmation quadratique sous contraintes. Cependant, il est clair que dans une scène observée, les photons peuvent interagir entre plusieurs matériaux en raison des réflexions multiples et du mélange intime des substances. Un modèle non-linéaire est ainsi plus raisonnable et plus général pour décrire ces effets. Il est important dans l'analyse des données hyperspectrales. L'incorporation des deux contraintes mentionnées dans le modèle non-linéaire est également incontournable pour les problèmes de démélange non-linéaire. Pour cela, nous proposons un modèle non-linéaire général et des méthodes de démélange associées.

A.3 Contributions principales de la thèse

Les contributions principales de cette thèse comprennent les aspects suivants :

- Etudier le problème d'identification de système sous contrainte de non-negativité. Proposer l'algorithme "non-negative LMS" (NNLMS) qui opère à la façon d'une méthode de points intérieurs. Le modèle de convergence est aussi étudié pour comprendre le comportement de l'algorithme aux premier et second ordre dans un environnement stationnaire.
- Proposer trois variants utiles de l'algorithme NNLMS (Normalized NNLMS, Exponential NNLMS, Sign-Sign NNLMS) afin d'améliorer certaines de ses propriétés. Les études de convergence sont par ailleurs étendues aux environnements non-stationnaires.
- Proposer un algorithme pour résoudre des problèmes d'identification de système sous contraintes concernant la norme ℓ_1 d'une manière en-ligne, sur la base de l'algorithme NNLMS. Les propriétés de convergence sont aussi étudiées.
- Proposer un modèle général de mélange non-linéaire, consistant en une combinaison d'une loi de mélange linéaire et d'un terme de fluctuations non-linéaire issu d'un espace fonctionnel de Hilbert à noyau reproduisant.
- Généraliser la méthode de démélange évoquée ci-dessus afin qu'elle soit à même d'ajuster automatiquement la proportion entre les composantes linéaire et non-linéaire du mélange .
- Introduire la corrélation spatiale dans le problème de démélange non-linéaire et proposer l'algorithme associé pour améliorer la performance.

A.4 Organisation de thèse

Le corps principal de cette thèse est divisé en deux parties. La première partie, qui se compose des Chapitres 2 et 3, concerne des algorithmes adaptatifs sous contraintes pour résoudre les problèmes d'identification linéaires en-ligne. Dans le Chapitre 2, nous présentons la méthode fondamentale de cette partie, l'algorithme "Non-negative LMS". Dans le chapitre 3, nous nous employons à généraliser cet algorithme pour traiter les problèmes d'identification sous contraintes de norme ℓ_1. La deuxième partie, qui se compose des Chapitres 4 et 5, étudie les algorithmes non-linéaires pour le démélange des données hyperspectrales. Dans le chapitre 4, nous introduisons un modèle non-linéaire pour caractériser le mélange des spectres et proposons une méthode de noyau pour estimer les abondances. Dans le Chapitre 5, nous intégrons la corrélation spatiale dans la méthode de démélange non-linéaire en utilisant un mode de régularisation faisant intervenir la norme ℓ_1, qui mesure les différences entre pixels voisins.

A.5 Filtrage adaptatif avec contrainte de non-négativité

Plusieurs stratégies ont été proposées afin de résoudre le problème de minimisation du coût quadratique sous contraintes de non-négativité. Les méthodes de contraintes actives reposent par exemble sur le fait que, sous réserve de connaitre les variables activant les contraintes, la solution du problème de moindres carrés peut être obtenue par résolution d'un problème non-contraint n'impliquant que les variables encore libres. L'algorithme dit de Lawson et Hanson est une approche hors-ligne de référence exploitant ce principe. Une autre classe de méthodes, dite de type gradient projeté, repose sur des projections successives des solutions intermédiaires sur l'espace admissible. Par leur simplicité algorithmique, sous réserve que l'opération de projection s'exprime aisément, ces approches s'avèrent particulièrement séduisantes. On déplore toutefois le fait que les solutions intermédiaires, avant projection, puissent ne pas respecter les contraintes et se trouver, le cas échéant, à l'extérieur du domaine de définition du critère. On compte enfin les méthodes multiplicatives, rendues populaires par le problème de factorisation en matrices non-négatives, mais largement utilisées auparavant en restauration d'images astronomiques. Elles reposent sur une mise-à-jour de la solution à l'aide d'un facteur vectoriel ou matriciel à composantes positives, assurant la non-négativité de la solution. La vitesse de convergence des méthodes multiplicatives est toutefois réputée lente, en raison d'un pas non-modifiable puisqu'il garantit en l'état la forme multiplicative de la mise-à-jour, et à une direction de descente non-colinéaire au gradient.

Dans cette partie, nous présentons une méthodes aux caractéristiques comparables à celles de l'algorithme LMS, qui garantit la non-négativité des coefficients du filtre. On en étudie alors analytiquement les propriétés de convergence, ce qui permet d'exhiber un critère de convergence portant sur l'initialisation de l'algorithme et le choix du pas.

A.5.1 Principe de la méthode

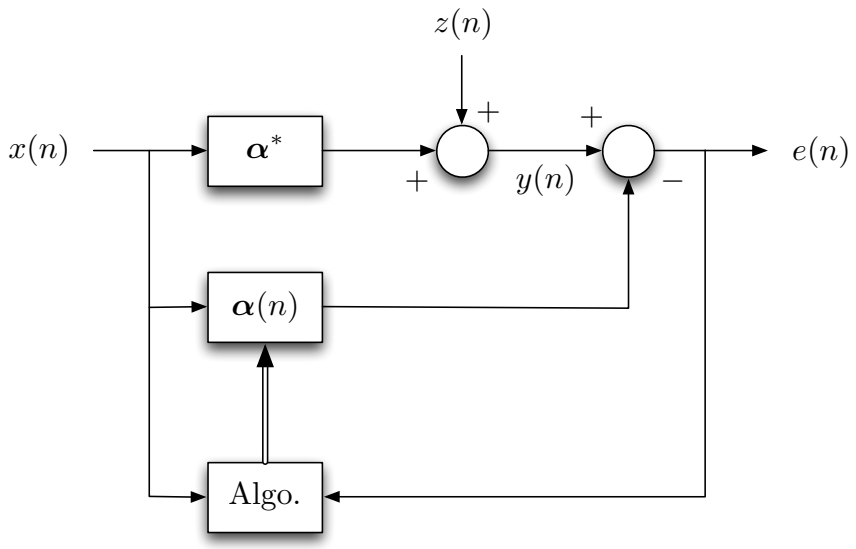


FIGURE A.1 – Identification de système avec filtre adaptatif.

Considérons un système inconnu, uniquement caractérisé par un ensemble de réponses en temps discret à des sollicitations stationnaires connues. Le problème traité est celui de la conception d'un filtre transverse

$$y(n) = \boldsymbol{\alpha}^\top \boldsymbol{x}(n) + z(n), \tag{A.9}$$

avec $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^\top$ le vecteur de coefficients à estimer, et

$$\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^\top$$

le vecteur d'observations. Le signal d'entrée $x(n)$ et le signal de sortie $y(n)$ désiré sont supposés stationnaires et de moyenne nulle. La séquence $z_1(n)$ traduit les bruits de mesure et autres erreurs de modélisation (Figure A.1). On s'intéresse à la résolution du problème suivant

$$\begin{aligned} \boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} & J(\boldsymbol{\alpha}) \\ & \text{sous contrainte } \alpha_i \geq 0, \quad \forall i, \end{aligned} \tag{A.10}$$

où $J(\boldsymbol{\alpha})$ est un coût convexe, et $\boldsymbol{\alpha}^o$ la solution du problème d'optimisation contraint. Afin de résoudre le problème (A.10), on considère le Lagrangien $Q(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ défini par

$$Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}^\top \boldsymbol{\alpha}, \tag{A.11}$$

avec $\boldsymbol{\lambda}$ le vecteur des multiplicateurs de Lagrange, tous non-négatifs. Les conditions dites de Karush-Kuhn-Tucker doivent être vérifiées à l'optimum, caractérisé par $\boldsymbol{\alpha}^o$

et $\boldsymbol{\lambda}^o$, c'est-à-dire

$$\begin{aligned} \nabla_\alpha Q(\boldsymbol{\alpha}^o, \boldsymbol{\lambda}^o) &= 0 \\ \alpha_i^o \, [\boldsymbol{\lambda}^o]_i &= 0, \ \forall i \end{aligned}$$

où ∇_α désigne le gradient par rapport à $\boldsymbol{\alpha}$. En utilisant $\nabla_\alpha Q(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \nabla_\alpha J(\boldsymbol{\alpha}) - \boldsymbol{\lambda}$, ces deux équations peuvent être combinées en une seule expression

$$\alpha_i^o \, [-\nabla_\alpha J(\boldsymbol{\alpha}^o)]_i = 0. \tag{A.12}$$

En constatant que les équations de la forme $\varphi(u) = 0$ peuvent être résolues par la mise en œuvre d'une méthode de point fixe en considérant le problème $u = u + \varphi(u)$, on aboutit à la méthode de descente de gradient exprimée par composante

$$\alpha_i(n+1) = \alpha_i(n) + \eta \, \alpha_i(n) [-\nabla_\alpha J(\boldsymbol{\alpha}(n))]_i \tag{A.13}$$

avec η un pas strictement positif. Il convient de noter que celui-ci doit être choisi afin d'assurer le caractère contractant du schéma de point fixe adopté, ce qui constitue l'objet de l'étude de convergence ci-après. En considérant le coût quadratique $J(\boldsymbol{\alpha}) = E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\}$, et en approchant les moments du second ordre par des grandeurs instantanées, soit $\boldsymbol{R}_x \approx \boldsymbol{x}(n) \, \boldsymbol{x}^\top(n)$ et $\boldsymbol{r}_{xy} \approx y(n) \, \boldsymbol{x}(n)$, on aboutit à l'algorithme *non-negative LMS* (NNLMS) proposé :

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta \, e(n) \, \boldsymbol{D}_\alpha(n) \, \boldsymbol{x}(n), \quad \eta > 0 \tag{A.14}$$

où $\boldsymbol{D}_\alpha(n)$ désigne la matrice diagonale de termes diagonaux donnés par $\boldsymbol{\alpha}(n)$, et $e(n)$ l'erreur d'estimation $y(n) - \boldsymbol{\alpha}^\top(n) \, \boldsymbol{x}(n)$. Il est intéressant de remarquer le rôle de $\boldsymbol{D}_\alpha(n)$ dans cette expression. Celui-ci inverse le sens de progression de l'algorithme pour les composantes $\alpha_i(n)$ qui seraient négatives, afin de les ramener dans le domaine admissible, et réduit le pas à l'approche de la borne 0.

A.5.2 Comportement de l'algorithme

On s'intéresse à présent au comportement de la méthode, que l'on souhaite décrire à l'aide d'un modèle analytique. Ce dernier est ici destiné à définir les conditions de convergence de l'algorithme, et pourra être exploité ultérieurement afin d'anticiper les performances dans un contexte applicatif donné. On note $\boldsymbol{\alpha}^*$ la solution du problème sans contrainte

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} E\{[y(n) - \boldsymbol{\alpha}^\top \boldsymbol{x}(n)]^2\}. \tag{A.15}$$

En définissant le vecteur d'erreur comme suit

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^* = [v_1(n), v_2(n), \ldots, v_N(n)]^\top, \tag{A.16}$$

l'équation de mise-à-jour (A.14) peut s'écrire

$$\boldsymbol{v}(n+1) = \boldsymbol{v}(n) + \eta \, e(n) \, \boldsymbol{D}_x(n) \, \big(\boldsymbol{v}(n) + \boldsymbol{\alpha}^*\big). \tag{A.17}$$

En utilisant $e(n) = y(n) - \boldsymbol{\alpha}^\top(n)\,\boldsymbol{x}(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$, on aboutit à l'expression suivante

$$\begin{aligned}\boldsymbol{v}(n+1) = \boldsymbol{v}(n) &+ \eta\, z(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{v}(n) + \eta\, z(n)\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}^* \\ &- \eta\, \boldsymbol{D}_x(n)\, \boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n) - \eta\, \boldsymbol{D}_x(n)\, \boldsymbol{\alpha}^*\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n).\end{aligned} \tag{A.18}$$

En considérant l'espérance de l'expression ci-dessus, en négligeant la dépendance statistique de $\boldsymbol{x}(n)$ et $\boldsymbol{v}(n)$ et en utilisant $E\{z(n)\,\boldsymbol{D}_x(n)\} = 0$, on aboutit à

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} \approx \big(\boldsymbol{I} - \eta E\{\boldsymbol{D}_x(n)\boldsymbol{\alpha}^*\boldsymbol{x}^\top(n)\}\big) E\{\boldsymbol{v}(n)\} \\ - \eta E\{\boldsymbol{D}_x(n)\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\boldsymbol{x}(n)\}.\end{aligned} \tag{A.19}$$

La première espérance dans le terme de droite de l'équation (A.19) est donnée par

$$E\{\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}^*\,\boldsymbol{x}^\top(n)\} = E\{\boldsymbol{D}_{\alpha^*}\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\} = \boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x. \tag{A.20}$$

La seconde espérance est évaluée par

$$E\{\boldsymbol{D}_x(n)\,\boldsymbol{v}(n)\,\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\} \approx \mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\} \tag{A.21}$$

où $\mathrm{Diag}\{\boldsymbol{A}\}$ représente le vecteur dont la i-ème composante est définie par $[\boldsymbol{A}]_{ii}$. En utilisant ces résultats dans l'expression (A.19), on aboutit à la récurrence suivante décrivant le comportement moyen de l'erreur sur les poids

$$E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} - \eta\,\mathrm{Diag}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}. \tag{A.22}$$

Cette équation nécessite le calcul de moments d'ordre 2 par le biais de $\boldsymbol{K}(n)$ afin d'évaluer $E\{\boldsymbol{v}(n)\}$. Un modèle récursif pourrait être défini pour $\boldsymbol{K}(n)$. Nous avons toutefois abouti à un modèle suffisamment précis et plus aisément exploitable peut être obtenu à partir de l'hypothèse de séparation suivante

$$\boldsymbol{K}(n) \approx E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}. \tag{A.23}$$

En utilisant (A.23) dans (A.22), on obtient le résultat suivant

$$\begin{aligned}E\{\boldsymbol{v}(n+1)\} = \big(\boldsymbol{I} - \eta\,\boldsymbol{D}_{\alpha^*}\,\boldsymbol{R}_x\big)\,E\{\boldsymbol{v}(n)\} \\ - \eta\,\mathrm{Diag}\{\boldsymbol{R}_x\,E\{\boldsymbol{v}(n)\}\,E\{\boldsymbol{v}^\top(n)\}\}.\end{aligned} \tag{A.24}$$

L'approximation (A.23) suppose que

$$\mathrm{Cov}\{v_i(n), v_j(n)\} \ll E\{v_i(n)\}E\{v_j(n)\} \tag{A.25}$$

En général, l'approximation (A.25) est d'autant plus raisonnable que les poids sont loin de leur valeur de convergence, dans la mesure où $E\{v_i(n)\}$ tend alors à être plus grand que les fluctuations quantifiées par $\mathrm{std}\{v_i(n)\}$. De nombreuses simulations nous ont permis de confirmer que le modèle (A.24) offre une qualité de prédiction de l'évolution des poids suffisante pour anticiper les performances du filtre dans un contexte applicatif donné.

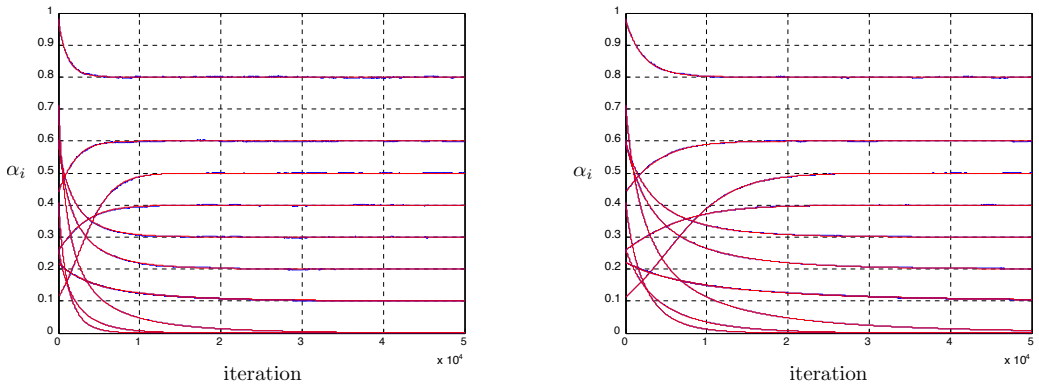


FIGURE A.2 – Convergence des coefficients $\alpha_i(n)$ pour NNLMS dans le cas d'une entrée blanche avec le pas $\eta = 10^{-3}$ (gause) et $\eta = 5 \cdot 10^{-4}$ (droit). Les courbes théorique (rouge) sont parfaitement superposées aux courbes obtenues par moyennage 100 simulations de Monte Carlo (bleu).

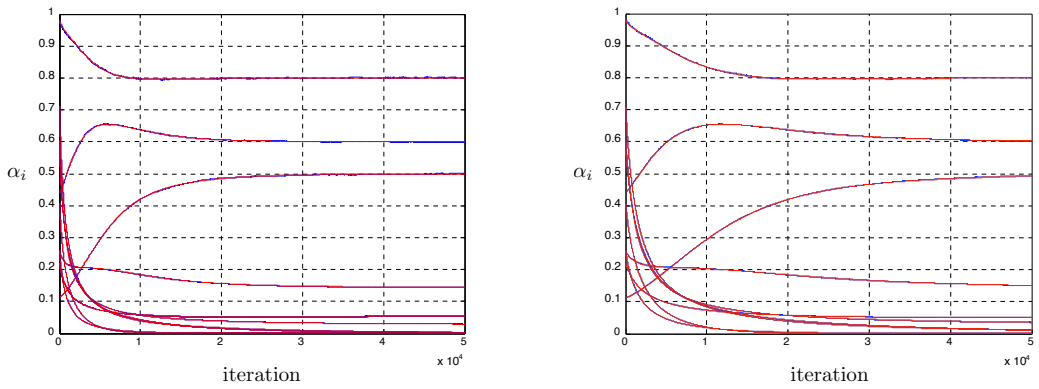


FIGURE A.3 – Convergence des coefficients $\alpha_i(n)$ pour NNLMS dans le cas d'une entrée corrélée avec le pas $\eta = 10^{-3}$ (gause) et $\eta = 5 \cdot 10^{-4}$ (droit). Les courbes théorique (rouge) sont parfaitement superposées aux courbes obtenues par moyennage 100 simulations de Monte Carlo (bleu).

Après cette analyse au premier ordre, nous présentons une brève analyse au second ordre. En utilisant $e(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$, en négligeant la corrélation entre $\boldsymbol{x}(n)$ et $\boldsymbol{v}(n)$, et en considérant les propriétés de $z(n)$, l'erreur quadratique moyenne (MSE) est exprimée par

$$\begin{aligned} E\{e^2(n)\} &= E\{(z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n))(z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n))\} \\ &= \sigma_z^2 + E\{\boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)\,\boldsymbol{x}^\top(n)\,\boldsymbol{v}(n)\} \\ &= \sigma_z^2 + \mathrm{trace}\{\boldsymbol{R}_x\,\boldsymbol{K}(n)\}. \end{aligned} \tag{A.26}$$

avec $\boldsymbol{K}(n) = E\left\{\boldsymbol{v}(n)\boldsymbol{v}^\top(n)\right\}$. Ainsi nous avons déterminons dans la thèse une expression analytique récursive de $\boldsymbol{K}(n)$ pour évaluer la MSE définie par (A.26).

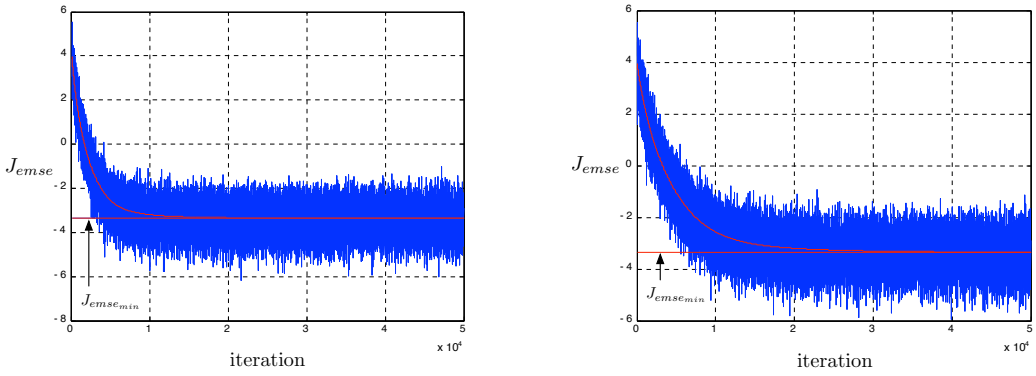


FIGURE A.4 – Convergence de EMSE pour NNLMS dans le cas d'une entrée blanche avec le pas $\eta = 10^{-3}$ (gause) et $\eta = 5 \cdot 10^{-4}$ (droit). Les courbes théorique (rouge) sont parfaitement superposées aux courbes obtenues par moyennage 100 simulations de Monte Carlo (bleu).

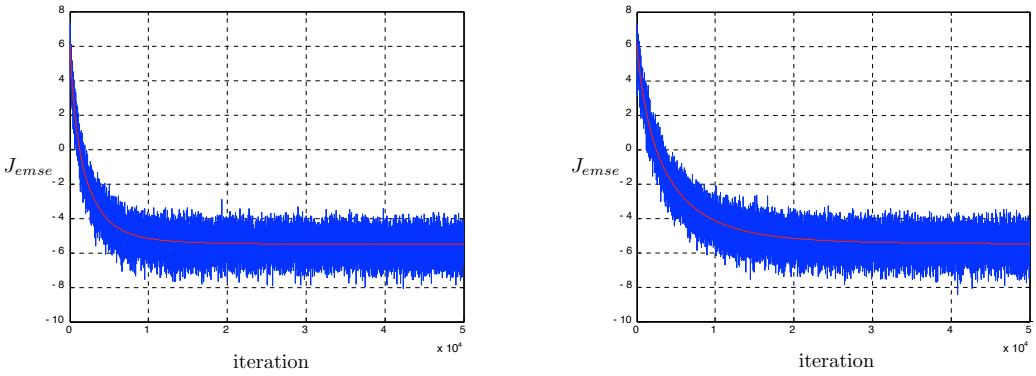


FIGURE A.5 – Convergence de EMSE pour NNLMS dans le cas d'une entrée corrélée avec le pas $\eta = 10^{-3}$ (gause) et $\eta = 5 \cdot 10^{-4}$ (droit). Les courbes théorique (rouge) sont parfaitement superposées aux courbes obtenues par moyennage 100 simulations de Monte Carlo (bleu).

A.5.3 Variantes de l'algorithme NNLMS

Dans les sections précédentes, le problème d'identification en-ligne sous contrainte de non-négativité sur les paramètres a été étudiée. Ci-après, nous étendons ce travail et proposons des variantes de l'algorithme NNLMS. Chacune de ces variantes est proposée afin d'améliorer les performances de l'algorithme NNLMS selon un certain objectif. Un algorithme normalisé est proposé pour réduire la sensibilité des comportements de NNLMS à la puissance d'entrée. Un algorithme exponentiel est proposé pour équilibrer des vitesses de convergence des poids. Enfin, un algorithme de type signe est proposé pour réduire les coûts de mise-en-œuvre dans des applications où la contrainte de temps-réel est critique.

A.5.3.1 Normalized NNLMS

Une extension directe de l'algorithme original est le Normalized-NNLMS. Conditionné par $\boldsymbol{\alpha}(n)$, le produit $e(n)\boldsymbol{D}_x(n)$ dans (A.14) a pour dimension la puissance du signal d'entrée. Ainsi, η est inversement proportionnelle à la puissance du signal. Définir une valeur constante pour η conduit à différentes vitesses de convergence selon la puissance du signal d'entrée. Une façon fréquente de remédier à cette sensibilité à normaliser le pas de mise-à-jour par la norme ℓ_2 des entrées.

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n)}\, e(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{N}}(n) \tag{A.27}$$

Comme pour l'algorithme NLMS, une petite valeur positive ε est souvent ajoutée au dénominateur $\boldsymbol{x}^{\top}(n)\boldsymbol{x}(n)$ pour éviter les difficultés numériques lorsque $\boldsymbol{x}^{\top}(n)\boldsymbol{x}(n)$ devient faible. On aboutit à l'algorithme ε-Normalized NNLMS

$$\boldsymbol{\alpha}_{\mathrm{N}}(n+1) = \boldsymbol{\alpha}_{\mathrm{N}}(n) + \frac{\eta}{\boldsymbol{x}^{\top}(n)\,\boldsymbol{x}(n) + \varepsilon}\, e(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{N}}(n) \tag{A.28}$$

A.5.3.2 Exponential NNLMS

Chaque composante $\alpha_i(n)$ dans l'équation de mise-à-jour (A.14) peut être considérée comme un ajustement d'amplitude différente selon l'indice i. Ainsi, chaque élément de $\boldsymbol{\alpha}(n)$ a une vitesse convergence différente en général. En particulier pour les coefficients de l'ensemble actif, la vitesse de convergence diminue progressivement à l'approche de l'état d'équilibre. Pour remédier à cette disparité de vitesses de convergence, nous introduisons l'algorithme Exponentiel NNLMS. En choisissant un paramètre $\gamma = p/q$ avec p, q impairs et $0 < p < q$, on aboutit à l'algorithme suivant, dénommé Exponential NNLMS

$$\boldsymbol{\alpha}_{\mathrm{E}}(n+1) = \boldsymbol{\alpha}_{\mathrm{E}}(n) + \eta\, e(n)\,\boldsymbol{D}_x(n)\,\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n) \tag{A.29a}$$

où la i-ème composante de $\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)$ est definie par

$$[\boldsymbol{\alpha}_{\mathrm{E}}^{(\gamma)}(n)]_i = \mathrm{sign}\{\alpha_{\mathrm{E}_i}(n)\}\,|\alpha_{\mathrm{E}_i}(n)|^{\gamma}. \tag{A.29b}$$

Un exposant compris entre $0< \gamma < 1$ permet de compenser la décroissance de la vitesse de convergence de $\alpha_{\mathrm{E}_i}(n)$ à l'approche du point d'équilibre 0.

A.5.3.3 Sign-Sign NNLMS

Motivé par la même raison que l'algorithme Sign-Sign LMS, qui a été inclus dans la norme CCITT pour la modulation différentielle adaptative de codes impulsifs, l'introduction de l'algorithme Sign-Sign NNLMS a pour but la simplicité de calcul et la robustesse aux perturbations. Remplacer le vecteur d'entrée et l'erreur d'estimation dans l'équation de mise-à-jour par leur signe réduit le temps de calcul. L'algorithme Sign-Sign NNLMS est donné par

$$\boldsymbol{\alpha}_{\mathrm{S}}(n+1) = \boldsymbol{\alpha}_{\mathrm{S}}(n) + \eta\,\mathrm{sgn}\{e(n)\}\,\mathrm{sgn}\{\boldsymbol{D}_x(n)\}\,\boldsymbol{\alpha}_{\mathrm{S}}(n) \tag{A.30}$$

Il peut être observé que la non-negativité de $\boldsymbol{\alpha}_{\mathrm{S}}(n)$ est toujours garantie avec un pas $0 < \eta < 1$.

A.5.3.4 Modèles de convergence pour les variantes de NNLMS

Les propriétés de convergence de NNLMS ont été étudiée dans un environnement stationnaire. Pour généraliser cette analyse, nous étudions les comportements des variantes de NNLMS dans des environnements non-stationnaires. Supposons que la solution du système sans contrainte évolue au cours de temps selon

$$\boldsymbol{\alpha}^*(n) = \boldsymbol{\alpha}_{\mathrm{o}}^*(n) + \boldsymbol{\xi}(n) \tag{A.31}$$

où $\boldsymbol{\alpha}_{\mathrm{o}}^*(n)$ est un terme de moyenne déterministe mais pouvant varier au cours du temps, $\boldsymbol{\xi}(n)$ un signal indépendant des autres et à moyenne nulle avec pour matrice de covariance $\boldsymbol{\Xi} = \sigma_\xi^2 \boldsymbol{I}$. Ce modèle nous permet d'étudier les performances des algorithmes dans un environnement variant au cours du temps.

Définissons le vecteur d'erreur comme suit

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}_{\mathrm{o}}^*(n) \tag{A.32}$$

Pour poursuivre l'analyse de l'algorithme NNLMS que nous avons présentée, le modèle au premier ordre de Normalized NNLMS est donné par

$$E\{\boldsymbol{v}_{\mathrm{N}}(n+1)\} = -\frac{\eta}{N\sigma_x^2 + \varepsilon}\, \mathrm{Diag}\Big\{\boldsymbol{R}_x\, E\{\boldsymbol{v}_{\mathrm{N}}(n)\}\, E\{\boldsymbol{v}_{\mathrm{N}}^\top(n)\}\Big\} + \boldsymbol{\Delta}(n). \tag{A.33}$$

avec $\boldsymbol{\Delta}(n) = \boldsymbol{\alpha}_{\mathrm{o}}^*(n) - \boldsymbol{\alpha}_{\mathrm{o}}^*(n+1)$. Le modèle au premier ordre de Exponential NNLMS est donné par

$$E\{\boldsymbol{v}_{\mathrm{E}}(n+1)\} = \Big(\boldsymbol{I}_N - \eta\, \boldsymbol{D}_r(n)\boldsymbol{R}_x\Big) E\{\boldsymbol{v}_{\mathrm{E}}(n)\} + \boldsymbol{\Delta}(n)$$

Le modèle au premier ordre de Sign-Sign NNLMS est donné par

$$E\Big\{\mathrm{sgn}\{z(n) - \boldsymbol{v}_{\mathrm{S}}^\top(n)\, \boldsymbol{x}(n) + \boldsymbol{\xi}^\top(n)\boldsymbol{x}(n)\}\, \mathrm{sgn}\{x_i(n)\}\Big\} \approx \frac{2}{\pi} \sin^{-1}\left(-\frac{\boldsymbol{R}_i^\top\, E\{\boldsymbol{v}_{\mathrm{S}}(n)\}}{\sigma_x \sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}}}\right)$$

avec

$$\sigma_{e|E\{\boldsymbol{v}_{\mathrm{S}}(n)\},\boldsymbol{\Xi}} = \sqrt{\sigma_z^2 + \mathrm{tr}\Big\{\boldsymbol{R}_x\, E\{\boldsymbol{v}_{\mathrm{S}}(n)\} E\{\boldsymbol{v}_{\mathrm{S}}^\top(n)\}\Big\} + \mathrm{trace}\{\boldsymbol{R}_x \boldsymbol{\Xi}\}}$$

Pour les modèles au seconde ordre des algorithmes, l'erreur quadratique moyenne en excès (EMSE) est donnée par

$$\begin{aligned} \zeta(n) &= E\Big\{(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))^\top \boldsymbol{x}(n)\boldsymbol{x}^\top(n)(\boldsymbol{v}(n) - \boldsymbol{\xi}(n))\Big\} \\ &= \mathrm{trace}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\} + \mathrm{trace}\{\boldsymbol{R}_x\, \boldsymbol{\Xi}\} \end{aligned} \tag{A.34}$$

Ainsi nous avons déterminons les expressions analytiques récursives de $\boldsymbol{K}(n)$ pour les trois variantes d'algorithme au cours de la thèse. L'estimation de la EMSE fournie par ces modèles s'est avérée en parfait accord avec les simulations effectuées.

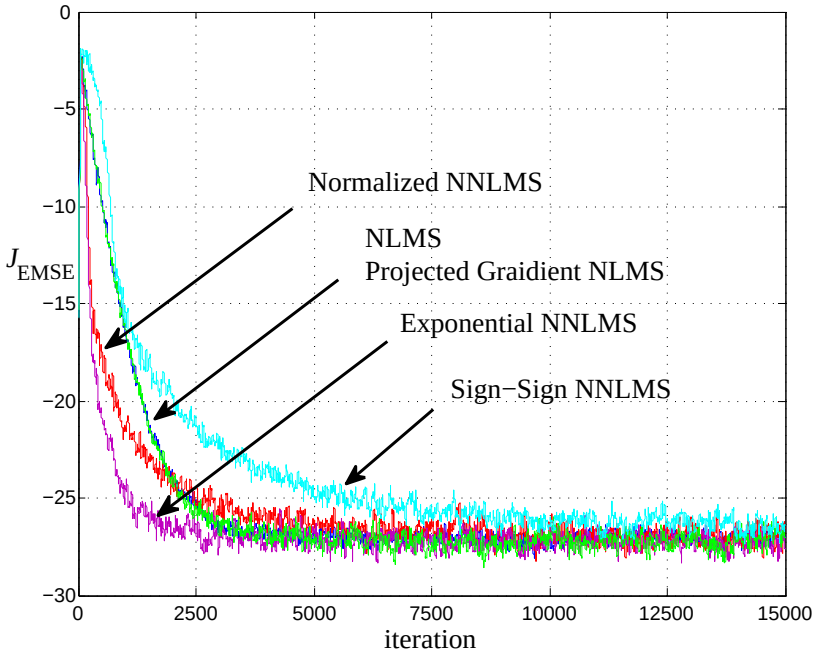


FIGURE A.6 – Une comparaison des algorithms de classes NNLMS.

A.6 Filtrage adaptatif avec contrainte de norme ℓ_1

Ces dernières années ont vu un réel engouement de la communauté pour les techniques de régularisation reposant sur des critères de type ℓ_1. Des termes comme parcimonie, échantillonnage compressif ou variation totale font régulièrement l'objet d'articles dans des conférences et revues spécialisées. De nombreux modèles et algorithmes d'optimisation ont été proposés puis appliqués à la masse croissante des problèmes inverses rencontrés en ingénierie du signal.

Cette section s'inscrit dans la continuité de notre travail présenté précédemment. Nous y envisageons d'étendre l'algorithme NNLMS pour l'identification de système en-ligne avec une contrainte/régularisation de norme ℓ_1. Tout d'abord, nous allons brièvement aborder une extension directe de NNLMS pour traiter le problème sous les contraintes de non-négativité et de norme ℓ_1, c'est à dire l'ensemble des contraintes $\Theta^{+}_{\ell_1}$. Après cela, nous allons aborder en détail un algorithme en-ligne pour identifier des systèmes avec la contrainte de norme ℓ_1 pour promouvoir la parcimonie, grâce à une extension de l'algorithme NNLMS.

A.6.1 Identification de système sous contraintes de non-négativité et norme ℓ_1 constante

Le problème d'identification sous contraintes de non-négativité et norme ℓ_1 constante est formulé par

$$\begin{aligned} \boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} & J(\boldsymbol{\alpha}) \\ \text{sous contrainte} \quad & \boldsymbol{\alpha} \geq \mathbf{0} \\ & \|\boldsymbol{\alpha}\|_1 = \varepsilon_0 \end{aligned} \tag{A.35}$$

avec $J(\alpha)$ un critère convexe, $\|\cdot\|_1$ la norme ℓ_1, et ε_0 une constante pré-définie. Dans le cas où la non-négativité des coefficients est satisfaite, la contrainte de norme ℓ_1 constante devient une contrainte de somme constante $\sum_i \alpha_i = \varepsilon_0$. En introduisant des variables intermédiaire w_i tel que $\alpha_j = \frac{w_j}{\sum_{\ell=1}^{N} w_\ell}\varepsilon_0$, et en appliquant l'algorithme NNLMS sur w_i, on aboutit à

$$\alpha_i(k+1) = \alpha_i(k) + \eta\,\alpha_i(k)\big(-\varepsilon_0[\nabla_{\boldsymbol{\alpha}} J]_i - \sum_{j=1}^{N} \alpha_j \nabla_{\boldsymbol{\alpha}} J]_j\big) \tag{A.36}$$

La contrainte de somme est toujours garantie lorsque l'initialisation $\boldsymbol{\alpha}(0)$ satisfait les deux contraintes. En appliquant cette relation sur le critère d'erreur quadratique moyenne et utilisant le gradient instantané, un algorithme de type LMS est obtenu

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \eta \boldsymbol{D}_{\boldsymbol{\alpha}}(n)\big(\varepsilon_0 \boldsymbol{x}(n)e(n) - \mathbf{1}\,\boldsymbol{\alpha}^\top(n)x(n)e(n)\big) \tag{A.37}$$

Cet algorithme peut traiter le problème (A.35) d'une manière efficace et en-ligne.

A.6.2 Identification de système par régularisation ℓ_1

La contrainte de norme ℓ_1 encourage la propriété de parcimonie des solutions. Elle permet une reformulation équivalente du problème par le biais d'un terme de régularisation. Dans cette section, nous allons résoudre le problème d'identification en-ligne en minimisant l'erreur quadratique moyenne régularisée par la norme ℓ_1 induisant la parcimonie

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} E\{[\boldsymbol{\alpha}^\top \boldsymbol{x}(n) - y(n)]^2\} + \lambda \|\boldsymbol{\alpha}\|_1 \tag{A.38}$$

où le paramètre λ contôle le compromis entre la fidélité aux données et le niveau de parcimonie. Comme nous l'avons évoqué précédemment, cette contrainte/régularisation peut être abordée en introduisant au préalable deux vecteurs non-négatifs $\boldsymbol{\alpha}^+$ et $\boldsymbol{\alpha}^-$ de taille $N \times 1$ tels que

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \tag{A.39}$$

$$\boldsymbol{\alpha}^+ \succeq 0 \tag{A.40}$$

$$\boldsymbol{\alpha}^- \succeq 0 \tag{A.41}$$

Par simplicité, on définit un vecteur $\tilde{\boldsymbol{\alpha}}$ de dimension $2N \times 1$ en superposant les deux sous-vecteurs $\boldsymbol{\alpha}^+$ et $\boldsymbol{\alpha}^-$

$$\tilde{\boldsymbol{\alpha}} = \begin{bmatrix} \boldsymbol{\alpha}^+ \\ \boldsymbol{\alpha}^- \end{bmatrix} \tag{A.42}$$

Le problème (A.38) se réécrit donc ainsi

$$\begin{aligned} \tilde{\boldsymbol{\alpha}}^o = \arg\min_{\tilde{\boldsymbol{\alpha}}} \frac{1}{2} E\left\{ \left(\begin{bmatrix} \boldsymbol{x}(n) \\ -\boldsymbol{x}(n) \end{bmatrix}^\top \tilde{\boldsymbol{\alpha}} - y(n) \right)^2 \right\} + \lambda \mathbf{1}_{2N}^\top \tilde{\boldsymbol{\alpha}} \\ \text{sous contrainte } \tilde{\boldsymbol{\alpha}} \succeq 0 \end{aligned} \tag{A.43}$$

avec $\mathbf{1}_{2N}$ un vecteur de taille $2N$ composé de 1. Ce problème est reformulé comme un problème d'optimisation sous la contrainte de non-négativité, nous sommes donc à même de le résoudre en-ligne par l'algorithme NNLMS

$$\tilde{\boldsymbol{\alpha}}(n+1) = (1 - \eta\,\lambda)\tilde{\boldsymbol{\alpha}}(n) + \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}}(n)\, e(n)\, \tilde{\boldsymbol{x}}(n) \tag{A.44}$$

A.6.3 Modèle de convergence de l'algorithme

Définissons le vecteur d'erreur comme suit

$$\begin{aligned} \boldsymbol{v}(n) &= \tilde{\boldsymbol{\alpha}}(n) - \tilde{\boldsymbol{\alpha}}^* \\ &= \begin{bmatrix} \boldsymbol{\alpha}^+(n) \\ \boldsymbol{\alpha}^-(n) \end{bmatrix} - \begin{bmatrix} \boldsymbol{\alpha}^{+*} \\ \boldsymbol{\alpha}^{-*} \end{bmatrix} \end{aligned} \tag{A.45}$$

Profitant de l'analyse faite pour NNLMS, nous pouvons obtenir un modèle pour caractériser les comportements au premier ordre de cet algorithme

$$\begin{aligned} E\left\{\boldsymbol{v}(n+1)\right\} = \Big((1 - \eta\,\lambda)\boldsymbol{I} - \eta\, \boldsymbol{D}_{\tilde{\boldsymbol{\alpha}}^*} \tilde{\boldsymbol{R}}_x\Big) E\left\{\boldsymbol{v}(n)\right\} - \eta\,\lambda\,\tilde{\boldsymbol{\alpha}}^* \\ - \eta\, \mathrm{diag}\{\tilde{\boldsymbol{R}}_x\, E\left\{\boldsymbol{v}(n)\right\} E\{\boldsymbol{v}^\top(n)\}\} \end{aligned} \tag{A.46}$$

avec $\tilde{\boldsymbol{R}}_x = \begin{pmatrix} \boldsymbol{R}_x - \boldsymbol{R}_x & \\ -\boldsymbol{R}_x & \boldsymbol{R}_x \end{pmatrix}$. Voir figure A.7.

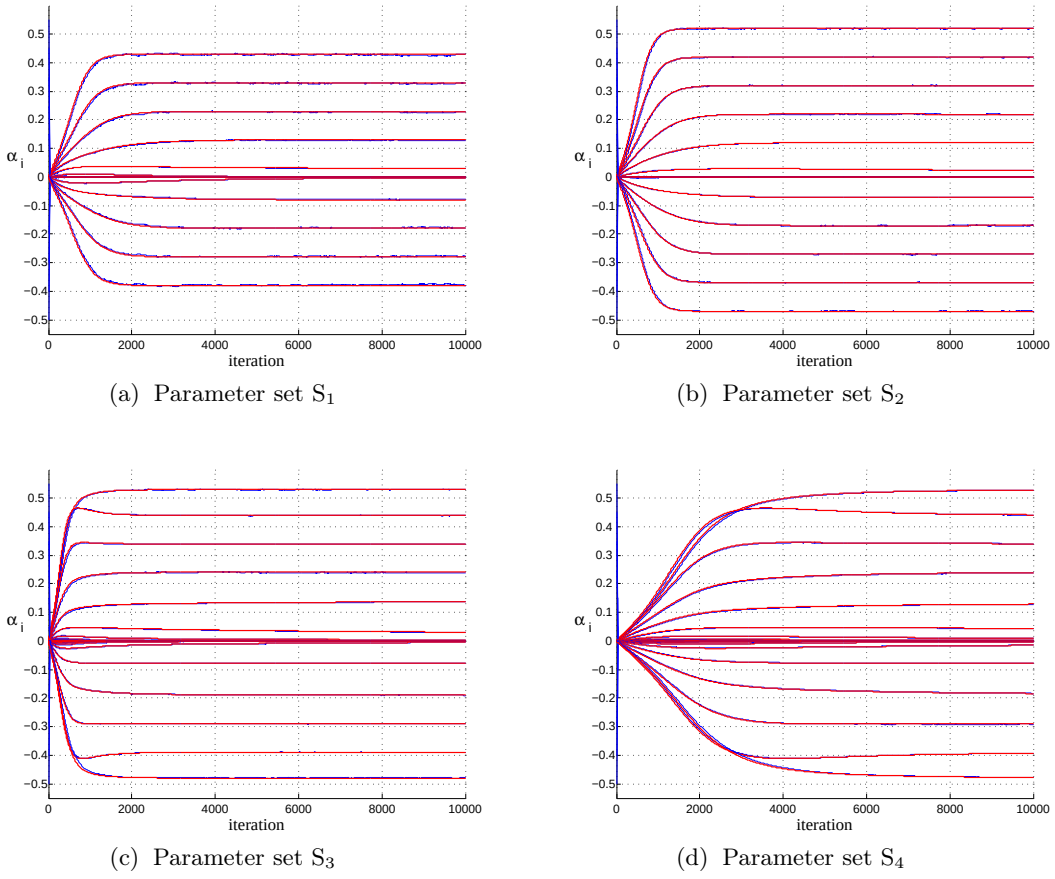
Par ailleurs nous avons déterminé l'expression analytique récursive pour la matrice de covariance de $\boldsymbol{v}(n)$, à savoir $\boldsymbol{K}(n)$, afin d'évaluer la EMSE de l'algorithme (voir Figure A.8)

Ses comportements dans l'environnement non-stationaire est illustrées dans figure A.9.

A.7 Démélange non-linéaire des images hyperspectrales

A.7.1 Contexte

L'imagerie hyperspectrale consiste à acquérir des images dans des centaines de bandes spectrales contigües et identiquement géo-référencées, avec une résolution



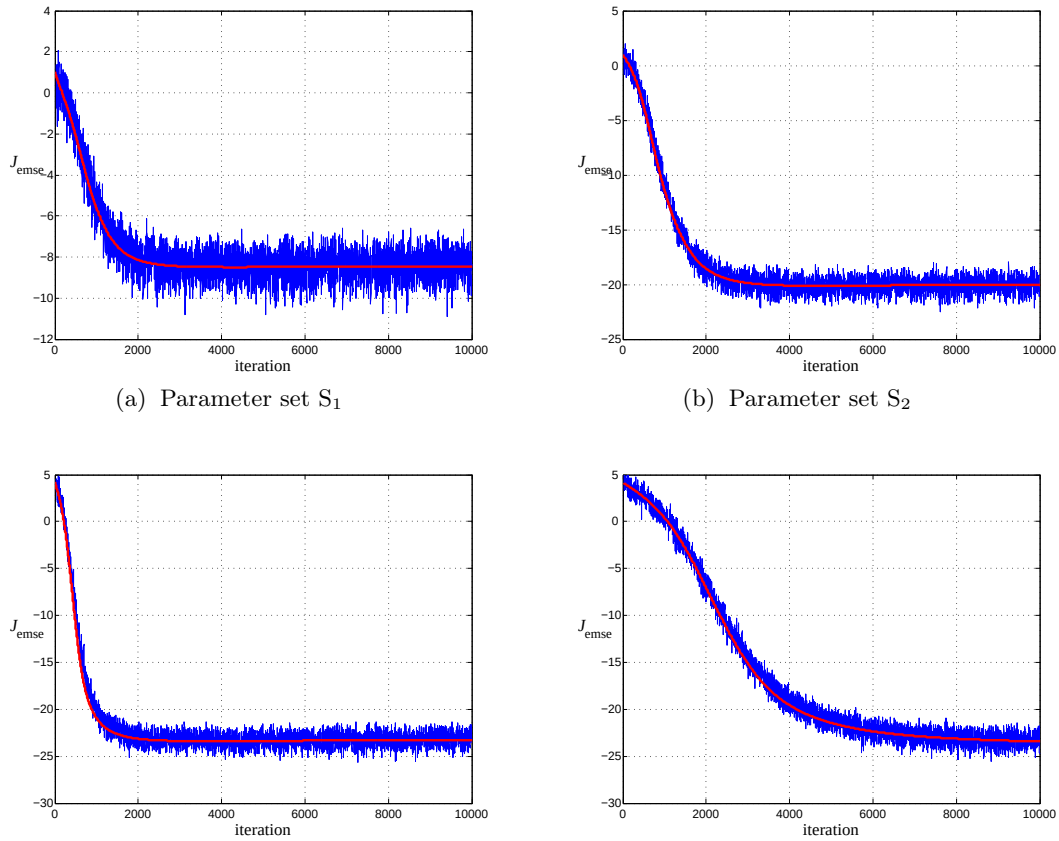
(a) Parameter set S_1 (b) Parameter set S_2

(c) Parameter set S_3 (d) Parameter set S_4

FIGURE A.7 – Convergence des coefficients pour NNLMS traitant le problème régularisé avec la norm-ℓ_1

suffisante pour résoudre la variabilité naturelle d'une scène. Le démélange spectral constitue l'un des problèmes importants dans ce domaine. Il s'agit de décomposer un ensemble de vecteurs spectraux sur une collection de signatures spectrales de composants purs, supposés connus dans un cadre supervisé, et d'estimer la fraction d'abondance de ces derniers dans le mélange. Selon l'échelle d'observation adoptée, les modèles de mélange considérés peuvent être soit linéaire, soit non-linéaire. Les premiers sont propre à des considérations macroscopiques au sens où ils négligent les interactions entre composants purs. Plus réaliste, les seconds visent à traduire la complexité des phénomènes physiques mis en jeu. On y distingue essentiellement le modèle bilinéaire, qui complète le modèle linéaire avec des termes d'interaction éponymes, et le modèle *intimate* reposant sur des considérations physiques avancées.

La littérature rapporte de nombreux travaux ayant trait aux modèles de mélange non-linéaires en imagerie hyperspectrale. Un réseaux de neurones à fonction radiale de base a été employé pour le traitement de mélanges. Un perceptron multi-couches couplé à un réseau de Hopfield a aussi été construit. Ces deux approches souffrent



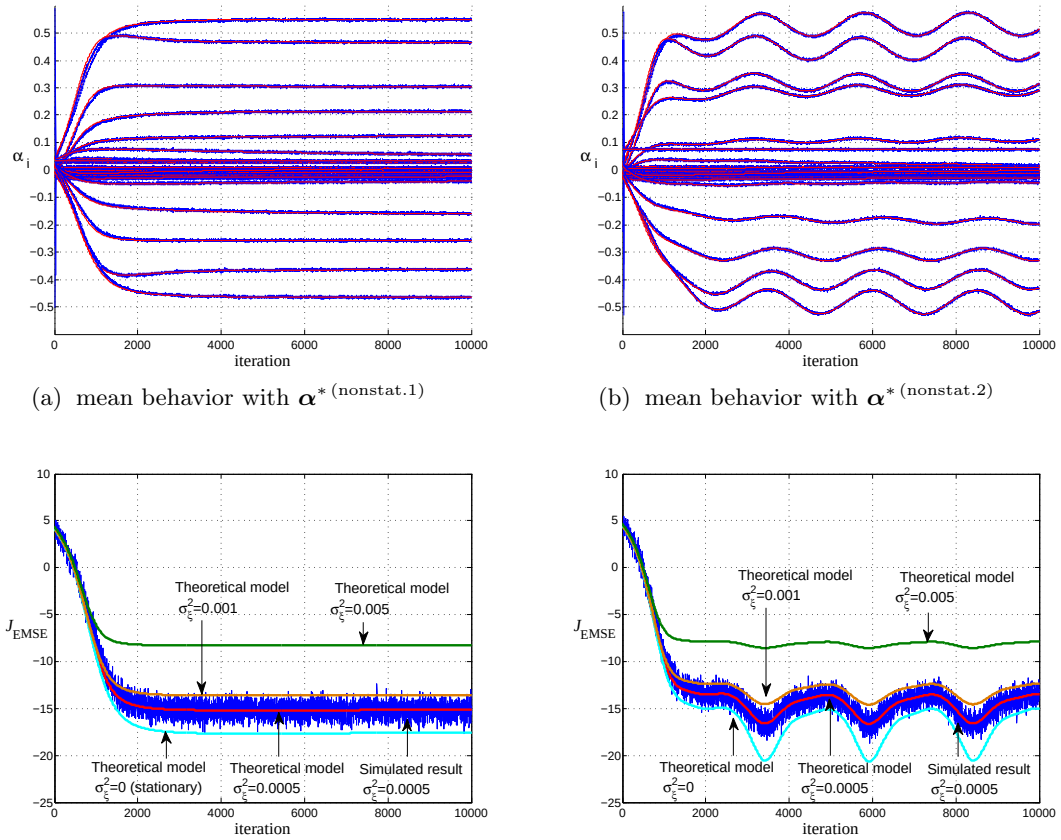
(a) Parameter set S_1 (b) Parameter set S_2

(c) Parameter set S_3 (d) Parameter set S_4

FIGURE A.8 – Convergence de EMSE pour NNLMS traitant le problème régularisé avec la norm-ℓ_1.

malheureusement des inconvénients inhérents aux approches connexionnistes, leur caractère boite noire et une optimisation rendue délicate par une fonction coût non-convexe. Des approches bayésiennes ont également été proposées. Elles s'avèrent performantes mais souffrent d'une charge calculatoire conséquente. Elles sont également uniquement dédiées au modèle bilinéaire. Enfin, certains travaux consistent à compléter la collection des signatures spectrales des éléments purs avec un ensemble de spectres hybrides supposés reproduire les interactions entre matériaux. Le nombre et la composition de ces éléments demeurent toutefois des questions délicates, aux conséquences importantes sur les résultats.

Des méthodes à noyau, dont les performances font référence en reconnaissance des formes, ont été mises en œuvre pour la classification de données hyperspectrales. Récemment, on y a également eu recours pour des questions de démélange non-linéaire. Les noyaux sont toutefois appliqués à la signature spectrale de chacun des composants purs, dans leur globalité et indépendamment des interactions entre matériaux, opérant ainsi à la façon d'une fonction de distorsion non-linéaire. Il



(a) mean behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.1})}$

(b) mean behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.2})}$

(c) 2nd-order behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.1})}$

(d) 2nd-order behavior with $\boldsymbol{\alpha}^{*\,(\text{nonstat.2})}$

FIGURE A.9 – Comportements d'algorithme NNLMS traitant le problème régularisé avec la norm-ℓ_1(environnement non-stationaire).

n'existe par ailleurs pas de formule de reconstruction de l'image pour cette approche, l'ensemble des calculs étant effectué dans l'espace de Hilbert à noyau reproduisant.

Dans cette section, nous formulons un nouveau paradigme où chaque noyau est appliqué selon une longueur d'onde qui lui est propre, y combinant ainsi les contributions de chaque composant selon une règle de mélange non-linéaire. Un choix approprié de noyau permet alors d'exhiber les abondances. Le problème correspondant peut être résolu par régression à noyau sous contrainte.

Soit $\boldsymbol{r} = [r_1, r_2, ..., r_L]^\top$ un vecteur-pixel hyperspectral, avec p le nombre de longueurs d'onde considérées. On suppose que $\boldsymbol{r}$ résulte du mélange de R composants purs, corrompu par un bruit. On note $\boldsymbol{M}$ la matrice de taille $L \times R$ regroupant l'ensemble des signatures spectrales des composants purs, qui constituent donc chacune une colonne $\boldsymbol{m}_i$ de la matrice $\boldsymbol{M}$. Aussi chaque ligne $\boldsymbol{m}_{\lambda_\ell}$ de $\boldsymbol{M}$ est-elle associée à une longueur d'onde donnée. Enfin, on note $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_R]^\top$ le vecteur des abondances associé au pixel $\boldsymbol{r}$.

Il existe schématiquement deux scénarios justifiant l'usage de modèles de mé-

lange non-linéaires pour décrire la complexité des interactions matériaux/lumière. Le premier implique des réflexions successives de la lumière sur différents composants purs. Le modèle de mélange bilinéaire tente de rendre compte de ce phénomène en introduisant des termes d'interaction dans le modèle linéaire sous la forme suivante

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{\alpha} + \sum_{i=1}^{\ell-1} \sum_{j=i+1}^{\ell} \gamma_{ij}\, \alpha_i\, \alpha_j\, \boldsymbol{m}_i \odot \boldsymbol{m}_j + \boldsymbol{n} \tag{A.47}$$

où $\boldsymbol{n}$ désigne un bruit additif que l'on pourra supposé Gaussien, et $\odot$ le produit d'Hadamard, avec les contraintes suivantes sur les paramètres

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{\ell} \alpha_i = 1 \qquad 0 \leq \gamma_{ij} \leq 1. \tag{A.48}$$

Pour le second scénario, le matériau réfléchissant est un modèle *intimate* de composants purs à la manière de grains de sable de compositions différentes.

A.7.2 Algorithme de démélange non-linéaire - KHYPE

Afin d'exhiber les abondances $\boldsymbol{\alpha}$, nous proposons de caractériser le mécanisme de mélange par un modèle qui consiste en une loi de mélange linéaire paramétrée par les abondances $\boldsymbol{\alpha}$, combinée à une fonction de fluctuation non-linéaire

$$\begin{aligned} \psi(\boldsymbol{m}_{\lambda_\ell}) &= \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} + \psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}) \\ &\text{sous contrainte} \quad \boldsymbol{\alpha} \succeq \mathbf{0} \quad \text{and} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1 \end{aligned} \tag{A.49}$$

avec ψ_{nlin} un élément d'un espace fonctionnel $\mathcal{H}_{\text{nlin}}$ à définir. Pour des facilités de mise en œuvre, soit $\mathcal{H}_{\text{nlin}}$ un espace de Hilbert à noyau reproduisant (RKHS) de fonctions à valeurs réelles sur un compact $\mathcal{M}$, et $\langle \cdot, \cdot \rangle$ son produit scalaire. On désigne par $\kappa_{\text{nlin}} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ le noyau de cet espace, vérifiant ainsi la propriété reproduisante $\psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell}) = \langle \psi_{\text{nlin}}(\cdot), \kappa_{\text{nlin}}(\cdot, \boldsymbol{m}_{\lambda_\ell}) \rangle$ pour toute fonction ψ_{nlin} de $\mathcal{H}_{\text{nlin}}$et pour tout $\boldsymbol{m}_{\lambda_\ell}$ de $\mathcal{M}$.

Nous proposons d'effectuer l'opération de démélange en résolvant le problème d'optimisation convexe

$$\begin{aligned} \psi^* = \arg\min_{\psi} \frac{1}{2} \left(\|\psi_{\text{lin}}\|^2_{\mathcal{H}_{\text{lin}}} + \|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}} \right) + \frac{1}{2\mu} \sum_{\ell=1}^{L} e_\ell^2 \\ \text{avec} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \quad \text{with} \quad \psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{\alpha}^\top \boldsymbol{m}_{\lambda_\ell} \\ \text{sous contrainte} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \\ \boldsymbol{\alpha} \succeq \mathbf{0} \quad \text{et} \quad \mathbf{1}^\top \boldsymbol{\alpha} = 1 \end{aligned} \tag{A.50}$$

Selon la propriété de dualité forte, nous pouvons obtenir un problème dual qui a la même solution que le problème primal ci-dessus. En introduisant les multiplicateurs de Lagrange β_ℓ, γ_r et λ, la fonction de Lagrange associée au problème (A.50) peut s'écrire comme

$$\begin{aligned} G =& \frac{1}{2}\left(\|\boldsymbol{\alpha}\|^2 + \|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 - \sum_{\ell=1}^{L}\beta_\ell\left(e_\ell - r_\ell + \psi(\boldsymbol{m}_{\lambda_\ell})\right) \\ & - \sum_{r=1}^{R}\gamma_r\,\alpha_r + \lambda(\mathbf{1}^\top\boldsymbol{\alpha} - 1) \end{aligned} \tag{A.51}$$

Les conditions d'optimalité de G par rapport aux variables primales sont données par

$$\begin{cases} \boldsymbol{\alpha}^* = \sum_{\ell=1}^{L}\beta_\ell^*\,\boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}^* - \lambda^*\mathbf{1} \\ \psi^*_{\text{nlin}} = \sum_{\ell=1}^{L}\beta_\ell^*\,\kappa_{\text{nlin}}(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_\ell^* = \mu\,\beta_\ell^* \end{cases} \tag{A.52}$$

En remplaçant ces conditions dans (A.51), on aboutit au problème dual

$$\begin{aligned} \max_{\boldsymbol{\beta},\boldsymbol{\gamma},\lambda} G'(\boldsymbol{\beta},\boldsymbol{\gamma},\lambda) = -\frac{1}{2}\left(\begin{array}{c}\boldsymbol{\beta}\\\hline\boldsymbol{\gamma}\\\hline\lambda\end{array}\right)^\top \left(\begin{array}{c|c|c} \boldsymbol{K}+\mu\boldsymbol{I} & \boldsymbol{M} & -\boldsymbol{M}\mathbf{1} \\\hline \boldsymbol{M}^\top & \boldsymbol{I} & -\mathbf{1} \\\hline -\mathbf{1}^\top\boldsymbol{M}^\top & -\mathbf{1}^\top & R \end{array}\right) \left(\begin{array}{c}\boldsymbol{\beta}\\\hline\boldsymbol{\gamma}\\\hline\lambda\end{array}\right) + \left(\begin{array}{c}\boldsymbol{r}\\\hline\mathbf{0}\\\hline -1\end{array}\right)^\top \left(\begin{array}{c}\boldsymbol{\beta}\\\hline\boldsymbol{\gamma}\\\hline\lambda\end{array}\right) \\ \text{sous contrainte} \quad \boldsymbol{\gamma} \succeq \mathbf{0} \end{aligned} \tag{A.53}$$

avec $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top + \boldsymbol{K}_{\text{nlin}}$. Une fois les variables duales déterminées par résolution de ce problème quadratique, le vecteur des abondances $\boldsymbol{\alpha}^*$ peut être estimé par

$$\boldsymbol{\alpha}^* = \boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^* - \lambda^*\mathbf{1} \tag{A.54}$$

et le pixel est reconstruit par

$$\begin{aligned} \boldsymbol{r}^* &= [\psi^*(\boldsymbol{m}_{\lambda_1}), \ldots, \psi^*(\boldsymbol{m}_{\lambda_L})]^\top \\ &= \boldsymbol{M}(\boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^* - \lambda^*\mathbf{1}) + \boldsymbol{K}_{\text{nlin}}\,\boldsymbol{\beta}^* \end{aligned} \tag{A.55}$$

A.7.3 Algorithme généralisé - SKHYPE

Le modèle proposé est basé sur l'hypothèse que le mécanisme de mélange peut être décrit par un mélange linéaire de composantes des spectres, avec une fluctuation additive non-linéairs ψ_{nlin} définie dans un RKHS. Ceci justifie l'utilisation de la matrice de Gram sous la forme $\boldsymbol{K} = \boldsymbol{M}\boldsymbol{M}^\top + \boldsymbol{K}_{\text{nlin}}$ dans l'algorithme présenté précédemment. Le modèle (A.49) a cependant certaines limites lié à l'équilibre entre la composante linéaire $\boldsymbol{\alpha}^\top\boldsymbol{m}_{\lambda_\ell}$ et la composante non-linéaire $\psi_{\text{nlin}}(\boldsymbol{m}_{\lambda_\ell})$, qui ne peut pas être réglé. Il nous est donc nécessaire de relaxer cette absence de souplesse entre les deux composantes.

Afin de régler l'équilibre entre ψ_{lin} et ψ_{nlin}, nous proposons de démélanger les

données hyperspectrales en résolvant le problème primal suivant

$$\begin{aligned} \psi^*, u^* = \arg\min_{\psi,u} \frac{1}{2}\left(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}'_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}'_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\ \text{sous contrainte} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \quad \text{and} \quad 0 \leq u \leq 1 \\ \text{avec} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \end{aligned} \tag{A.56}$$

où u permet de contrôler la balance entre ψ_{lin} et ψ_{nlin} via leurs normes. Les espaces $\mathcal{H}'_{\text{lin}}$ and $\mathcal{H}'_{\text{nlin}}$ sont des RKHS de forme générale telle que

$$\mathcal{H}'_{\text{lin/nlin}} = \left\{ \psi \in \mathcal{H}_{\text{lin/nlin}} : \frac{\|\psi\|_{\mathcal{H}_{\text{lin/nlin}}}}{u} < \infty \right\} \tag{A.57}$$

avec la convention $\frac{x}{0} = 0$ if $x = 0$, et ∞ sinon. La convexité de ce problème (par rapport à ψ et u) nous permet de formuler une procédure d'optimisation en deux étapes, sur ψ et u successivement

$$\min_u J(u) \qquad \text{sous contrainte} \qquad 0 \leq u \leq 1 \tag{A.58}$$

avec

$$J(u) = \left\{ \begin{aligned} \min_{\psi} F(u,\psi) = \frac{1}{2}\left(\frac{1}{u}\|\psi_{\text{lin}}\|^2_{\mathcal{H}'_{\text{lin}}} + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}'_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 \\ \text{sous contrainte} \quad e_\ell = r_\ell - \psi(\boldsymbol{m}_{\lambda_\ell}) \quad \text{with} \quad \psi = \psi_{\text{lin}} + \psi_{\text{nlin}} \\ \text{avec} \quad \psi_{\text{lin}}(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{h}^\top \boldsymbol{m}_{\lambda_\ell} \quad \text{et} \quad \boldsymbol{h} \succeq \boldsymbol{0} \end{aligned} \right. \tag{A.59}$$

Par la propriété de dualité forte, nous pouvons obtenir un problème dual qui a la même solution que le problème primal ci-dessus. En introduisant les multiplicateurs de Lagrange β_ℓ, γ_r et λ, la fonction de Lagrange associée au problème (A.59) peut s'écrire comme

$$G = \frac{1}{2}\left(\frac{1}{u}\|\boldsymbol{h}\|^2 + \frac{1}{1-u}\|\psi_{\text{nlin}}\|^2_{\mathcal{H}_{\text{nlin}}}\right) + \frac{1}{2\mu}\sum_{\ell=1}^{L} e_\ell^2 - \sum_{\ell=1}^{L}\beta_\ell\left(e_\ell - r_\ell + \psi(\boldsymbol{m}_{\lambda_\ell})\right) - \sum_{r=1}^{R}\gamma_r\, h_r \tag{A.60}$$

avec $\gamma_r \geq 0$. Les conditions d'optimalité par rapport aux variables primales sont données par

$$\left\{ \begin{aligned} \boldsymbol{h}^* &= u\left(\textstyle\sum_{\ell=1}^{L}\beta_\ell^*\,\boldsymbol{m}_{\lambda_\ell} + \boldsymbol{\gamma}^*\right) \\ \psi^*_{\text{nlin}} &= (1-u)\textstyle\sum_{\ell=1}^{L}\beta_\ell^*\,\kappa_{\text{nlin}}(\cdot, \boldsymbol{m}_{\lambda_\ell}) \\ e_\ell^* &= \mu\,\beta_\ell^* \end{aligned} \right. \tag{A.61}$$

En remplaçant ces conditions dans (A.60), ceci conduit au problème dual

$$J(u) = \left\{ \begin{aligned} \max_{\boldsymbol{\beta},\boldsymbol{\gamma}} G'(u,\boldsymbol{\beta},\boldsymbol{\gamma}) = -\frac{1}{2}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix}^\top \left(\begin{array}{c|c} \boldsymbol{K}_u + \mu\boldsymbol{I} & u\boldsymbol{M} \\ \hline u\boldsymbol{M}^\top & u\boldsymbol{I}\end{array}\right)\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix} + \begin{pmatrix}\boldsymbol{r}\\ \boldsymbol{0}\end{pmatrix}^\top\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix} \\ \text{sous conrainte} \quad \boldsymbol{\gamma} \succeq \boldsymbol{0} \end{aligned} \right. \tag{A.62}$$

avec $\boldsymbol{K}_u = u\boldsymbol{M}\boldsymbol{M}^\top + (1-u)\boldsymbol{K}_{\text{nlin}}$. En considérons la convexité du problème par rapport à la variable u, on note que cette dernière peut être mise à jour, soit par une itération d'un pas de gradient

$$\left.\frac{dJ(u)}{du}\right|_{u=u_0} = -\frac{1}{2}\left(\|\boldsymbol{M}^\top\boldsymbol{\beta}_0^* + \boldsymbol{\gamma}_0^*\|^2 - \boldsymbol{\beta}_0^{*\top}\boldsymbol{K}_{\text{nlin}}\boldsymbol{\beta}_0^*\right) \tag{A.63}$$

soit par une recherche directe de la valeur optimum donnée par

$$u^* = \frac{1}{1 + \|\psi_{\text{nlin}}\|_{\mathcal{H}_{\text{nlin}}}/\|\psi_{\text{nlin}}\|_{\mathcal{H}_{\text{lin}}}} \tag{A.64}$$

Finalement, le vecteur des abondances est donné par

$$\boldsymbol{\alpha}^* = \frac{\boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^*}{\mathbf{1}^\top(\boldsymbol{M}^\top\boldsymbol{\beta}^* + \boldsymbol{\gamma}^*)} \tag{A.65}$$

et la reconstruction du pixel est obtenue en utilisant $\boldsymbol{r}^* = [\psi^*(\boldsymbol{m}_{\lambda_1}), \ldots, \psi^*(\boldsymbol{m}_{\lambda_L})]^\top$ avec $\psi^*(\boldsymbol{m}_{\lambda_\ell}) = \boldsymbol{m}_{\lambda_\ell}^\top \boldsymbol{h}^* + \psi_{\text{nlin}}^*(\boldsymbol{m}_{\lambda_\ell})$ défini par (A.61).

Les tableaux (A.1) à (A.3) les résultats des algorithmes de démélange avec les différentes configurations.

A.8 Démélange non-linéaire avec corrélation spatiale

A.8.1 Contexte

Les algorithmes d'inversion développés dans les sections précédentes ont montré des estimations des coefficients d'abondance très satisfaisantes. Cependant, ces algorithmes comme la plupart des méthodes d'inversion, analysent les pixels de l'image indépendamment les uns des autres. Or une image réelle présente dans bien des cas des zones homogènes (zones de lac, de terres agricoles, etc). L'hypothèse d'indépendance des pixels appartenant à de telles zones paraît donc très éloignée de la réalité. C'est pourquoi nous pensons que l'introduction de corrélations entre les pixels voisins d'une image pourrait sérieusement améliorer les performances d'estimation.

Autant que l'on sache, la corrélation spatiale n'a jamais été intégrée dans un processus de démélange non-linéaire. La principale raison de ce manque est que le démélange non-linéaire lui-même est constitue une tâche importante mais difficile. Il semble difficile de répondre à ces deux problèmes en même temps. Dans la section précédente, un nouveau modèle non-linéaire a été proposé, où nous avons supposé qu'un mélange pouvait être décomposé en une tendance linéaire, et un terme de fluctuation additif non-linéaire dans un espace de Hilbert à noyau reproduisant pour modéliser les effets non-linéaires. A partir de cette avancée, nous prenons à présent en compte l'information spatiale dans le processus de démélange en utilisant la régularisation spatiale avec la norme ℓ_1. Une méthode d'optimisation basée sur des itérations de type Split-Bregman est proposée pour traiter ce problème qui souffre de la non-linéarité du modèle et de la non-régularité du terme de régularisation.

TABLE A.1 – Scène 1 (Trois matériaux) : Comparaison de RMSE

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	$0.0037 \pm 2 \times 10^{-5}$	0.0758±0.0019	0.0604±0.0017
ExtM	0.0079±0.0001	0.0312±0.0013	0.0601±0.0016
KFCLS	$0.0054 \pm 3 \times 10^{-5}$	0.2711±0.0516	0.2371±0.0197
BilBay	0.0384±0.0013	0.0285±0.0006	0.1158±0.0058
K-Hype (G)	0.0208±0.0004	0.0349±0.0013	0.0446±0.0020
K-Hype (P)	0.0346±0.0011	0.0281±0.0011	0.0569±0.0031
SK-Hype (G)	0.0104±0.0001	0.0315±0.0012	0.0230±0.0007
SK-Hype (P)	0.0106±0.0002	0.0310±0.0011	0.0245±0.0007
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0212±0.0005	0.0960±0.0060	0.0886±0.0063
ExtM	0.0404±0.0031	0.0991±0.096	0.0869±0.0066
KFCLS	0.0296±0.0009	0.2694±0.0498	0.2372±0.0235
BilBay	0.1135±0.0098	0.1059±0.0085	0.1191±0.0091
K-Hype (G)	0.0562±0.0041	0.0611±0.0048	0.0786±0.0067
K-Hype (P)	0.0589±0.0041	0.0628±0.0053	0.0794±0.0066
SK-Hype (G)	0.0562±0.0044	0.0598±0.0048	0.0757±0.0073
SK-Hype (P)	0.0561±0.0043	0.0602±0.0048	0.0742±0.0075

A.8.2 Formulation et solution de problème

En prenant compte de la corrélation spatiale parmi les pixels, le problème de démélange peut être formulé par minimisation d'une fonction de coût par rapport à la matrice des abondances $\boldsymbol{A}$

$$J(\boldsymbol{A}) = J_{\text{err}}(\boldsymbol{A}) + \eta\, J_{\text{sp}}(\boldsymbol{A}) \tag{A.66}$$

sous les contraintes de non-négativité imposées sur chaque élément de $\boldsymbol{A}$ et les contraintes de somme unité imposées à chaque colonne de $\boldsymbol{A}$, soit $\boldsymbol{\alpha}_n$. Pour simplifier les notations, ces deux contraintes sont notées par

$$\begin{aligned} &\boldsymbol{A} \succeq \mathbf{0} \\ &\boldsymbol{A}^\top \mathbf{1}_R = \mathbf{1}_N \end{aligned} \tag{A.67}$$

Dans l'expression générale (A.66), la fonction J_{err} représente l'erreur de modélisation, et J_{sp} est un terme de régularisation pour promouvoir les similarités entre des abondances voisines. Le paramètre non-négatif η contrôle le compromis entre la fidélité aux données et la similarité de pixels.

TABLE A.2 – Scène 2 (Cinq matériaux) : Comparaison de RMSE

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	0.0134±0.0002	0.1137±0.0032	0.1428±0.0039
ExtM	0.0157±0.0003	0.0575±0.0024	0.1427±0.0040
KFCLS	0.0200±0.0004	0.2051±0.0148	0.1955±0.0115
BilBay	0.0585 ±0.0017	0.0441±0.0010	0.1741±0.0082
K-Hype (G)	0.0231±0.0004	0.0307±0.0008	0.0398±0.0012
K-Hype (P)	0.0218±0.0004	0.0465±0.0012	0.0386±0.0011
SK-Hype (G)	0.0196±0.0004	0.0288±0.0007	0.0346±0.0010
SK-Hype (P)	0.0195±0.0004	0.0349±0.0008	0.0346±0.0010
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0657±0.0047	0.1444±0.0116	0.1611±0.0134
ExtM	0.0761±0.0060	0.1207±0.0160	0.1678±0.0139
KFCLS	0.0890±0.0080	0.1884±0.0113	0.1572±0.0114
BilBay	0.1465±0.0109	0.1007±0.0063	0.1609±0.0124
K-Hype (G)	0.1076±0.0093	0.0748±0.0046	0.0823±0.0053
K-Hype (P)	0.0738±0.0043	0.0847±0.0052	0.0828±0.0054
SK-Hype (G)	0.0675±0.0040	0.0778±0.0043	0.0942±0.0065
SK-Hype (P)	0.0673±0.0040	0.0830±0.0046	0.0965±0.0071

En utilisant la fonction de coût que nous avons développée pour KHYPE dans J_{err}, et la norme ℓ_1 de la différence des pixels voisins pour la régularisation spatiale, nous obtenons le problème

$$\begin{aligned} \boldsymbol{A}^*, \boldsymbol{\psi}^* = \underset{\boldsymbol{A},\boldsymbol{\psi}}{\arg\min} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 \right) + \eta \, \|\boldsymbol{A}\boldsymbol{H}\|_{1,1} \\ \text{sous contrainte} \quad \boldsymbol{A} \succeq 0 \quad \text{et} \quad \boldsymbol{A}^\top \mathbf{1}_R = \mathbf{1}_N \end{aligned} \tag{A.68}$$

où la matrice $\boldsymbol{H}$ définit l'opération de différence entre les abondances. La méthode split-Bregman est utilisée à résoudre ce problème d'optimisation convexe mais non-régularité. En introduisant deux nouvelle variables $\boldsymbol{U}$ et $\boldsymbol{V}$, et en notant les contraintes sur $\boldsymbol{A}$ par $\boldsymbol{A} \in \mathcal{S}_A$, on aboutit au problème

$$\begin{aligned} \min_{\boldsymbol{A}\in\mathcal{S}_A,\boldsymbol{\psi}} \sum_{n=1}^{N} \frac{1}{2} \left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2 \right) + \eta \, \|\boldsymbol{U}\|_{1,1} \\ \text{sous contrainte} \quad \boldsymbol{V} = \boldsymbol{A} \quad \text{and} \quad \boldsymbol{U} = \boldsymbol{V}\boldsymbol{H} \end{aligned} \tag{A.69}$$

TABLE A.3 – Scène 3 (Huit matériaux) : Comparaison de RMSE

	SNR = 30 dB		
	linear	bilinear	PNMM
FCLS	0.0148±0.0002	0.0930±0.0024	0.1079±0.0018
ExtM	0.0173±0.0003	0.0560±0.0017	0.1126±0.0019
KFCLS	0.0216±0.0004	0.1431±0.0059	0.1274±0.0039
BilBay	0.0448±0.0007	0.0369±0.0004	0.1159±0.0029
K-Hype (G)	0.0203±0.0003	0.0202±0.0003	0.0300±0.0006
K-Hype (P)	0.0195±0.0003	0.0330±0.0006	0.0297±0.0006
SK-Hype (G)	0.0185±0.0003	0.0221±0.0003	0.0291±0.0006
SK-Hype (P)	0.0184±0.0002	0.0247±0.0004	0.0313±0.0007
	SNR = 15 dB		
	linear	bilinear	PNMM
FCLS	0.0652±0.0031	0.1177±0.0068	0.1252±0.0065
ExtM	0.0743±0.0038	0.1066±0.0062	0.1322±0.0063
KFCLS	0.0647±0.0032	0.1270±0.0038	0.2250±0.0220
BilBay	0.0745±0.0020	0.0792±0.0026	0.1040±0.0430
K-Hype (G)	0.0562±0.0020	0.0548±0.0018	0.0642±0.0024
K-Hype (P)	0.0585±0.0021	0.0646±0.0024	0.0657±0.0026
SK-Hype (G)	0.0561±0.0019	0.0573±0.0020	0.0696±0.0027
SK-Hype (P)	0.0571±0.0021	0.0620±0.0021	0.0736±0.0031

dont la solution est obtenue par les itérations

$$\begin{aligned}\boldsymbol{A}^{(k+1)},\boldsymbol{\psi}^{(k+1)},\boldsymbol{V}^{(k+1)},\boldsymbol{U}^{(k+1)} = \underset{\boldsymbol{A}\in\mathcal{S}_A,\boldsymbol{\psi},\boldsymbol{V},\boldsymbol{U}}{\arg\min} \sum_{n=1}^{N} \frac{1}{2}\left(\|\boldsymbol{\alpha}_n\|^2 + \|\psi_n\|_{\mathcal{H}}^2 + \frac{1}{\mu}\|\boldsymbol{e}_n\|^2\right) \\ + \eta\,\|\boldsymbol{U}\|_{1,1} + \frac{\zeta}{2}\|\boldsymbol{A}-\boldsymbol{V}-\boldsymbol{D}_1^{(k)}\|_F^2 + \frac{\zeta}{2}\|\boldsymbol{U}-\boldsymbol{V}\boldsymbol{H}-\boldsymbol{D}_2^{(k)}\|_F^2\end{aligned} \tag{A.70}$$

avec

$$\begin{aligned}\boldsymbol{D}_1^{(k+1)} &= \boldsymbol{D}_1^{(k)} + \left(\boldsymbol{V}^{(k+1)} - \boldsymbol{A}^{(k+1)}\right) \\ \boldsymbol{D}_2^{(k+1)} &= \boldsymbol{D}_2^{(k)} + \left(\boldsymbol{V}^{(k+1)}\boldsymbol{H} - \boldsymbol{U}^{(k+1)}\right)\end{aligned} \tag{A.71}$$

La minimisation par rapport à $\boldsymbol{A}$ et $\boldsymbol{\psi}$ peut être conduite de la même manière que dans KHYPE, ce qui donne

$$\boldsymbol{\alpha}_n^* = \frac{1}{\zeta+1}\left(\boldsymbol{M}^\top\boldsymbol{\beta}_n^* + \boldsymbol{\gamma}_n^* - \lambda_n^*\mathbf{1} + \zeta\boldsymbol{\xi}_n^{(k)}\right) \tag{A.72}$$

TABLE A.4 – Performance de classification avec les abondances estimées (FCLS, K-Hype, proposé)

	5%	10%	15%
FCLS	56.41	61.36	62.32
K-Hype	67.67	71.39	74.68
Proposed	93.82	96.80	97.02

où $\boldsymbol{\xi}_n^{(k)} = \boldsymbol{V}_n^{(k)} + \boldsymbol{D}_{1,n}^{(k)}$. Les minimisations par rapport à $\boldsymbol{V}$ et $\boldsymbol{U}$ sont assez simples avec les solutions

$$\boldsymbol{V}^{(k+1)} = \left(\boldsymbol{A}^{(k+1)} - \boldsymbol{D}_1^{(k)} + (\boldsymbol{U}^{(k)} - \boldsymbol{D}_2^{(k)})\boldsymbol{H}^\top\right)(\boldsymbol{I} + \boldsymbol{H}\boldsymbol{H}^\top)^{-1} \tag{A.73}$$

$$\boldsymbol{U}^{(k+1)} = \text{Thresh}\left(\boldsymbol{V}^{(k+1)}\boldsymbol{H} + \boldsymbol{D}_2^{(k)}, \frac{\eta}{\zeta}\right) \tag{A.74}$$

où $\text{Thresh}(\cdot, \tau)$ est la fonction de seuil

$$\text{Thresh}(x, \tau) = \text{sign}(x)\ \max(|x| - \tau, 0) \tag{A.75}$$

La figure A.10 illustrate une exemple de démélange avec régularisation spatiale. Les performances des algorithmes avec donnée réelles sont testées sur l'observation d'Indiana Pine. Pour éviter la difficulté de manque de référence de démélange, on utilise les algorithme de démélange à générer les abondances, et les utilise comme les entrées d'algorithmes de classification. Voir le tableau A.4 et la figure A.11 pour les résultats.

A.9 Conclusion et perspectives

Dans cette thèse, nous avons étudié des problèmes d'optimisation sous des contraintes de non-négativité, et des contraintes relatives à la norme ℓ_1. Ces deux contraintes ont attiré une attention considérable au cours de la dernière décennie, car elles constituent une information a priori importante et relativement générique par rapport aux applications rencontrées. Le travail de cette thèse a consisté à explorer des problèmes importants liés à ces deux contraintes dans le contexte linéaire et non-linéaire, avec des applications à l'identification de systèmes en-ligne par des méthode de filtrage adaptatif, et de démélange non-linéaire de donnée hyperspectrales. Nous avons mis un point d'honneur à adopter un point de vue théorique pour décrites ces algorithmes et leurs performances, tant à convergence qu'en phase transitoire. Cependant, de nombreux problèmes intéressants subsistent, dans la continuité des travaux présentés. Ils comprennent des études des expressions analytiques de l'erreur en régime permanent pour l'algorithme NNLMS, ou encore l'intégration la détermination des signatures spectrales des composés purs dans les modèles de mélange non-linéaires, pour n'en citer que quelques-uns.

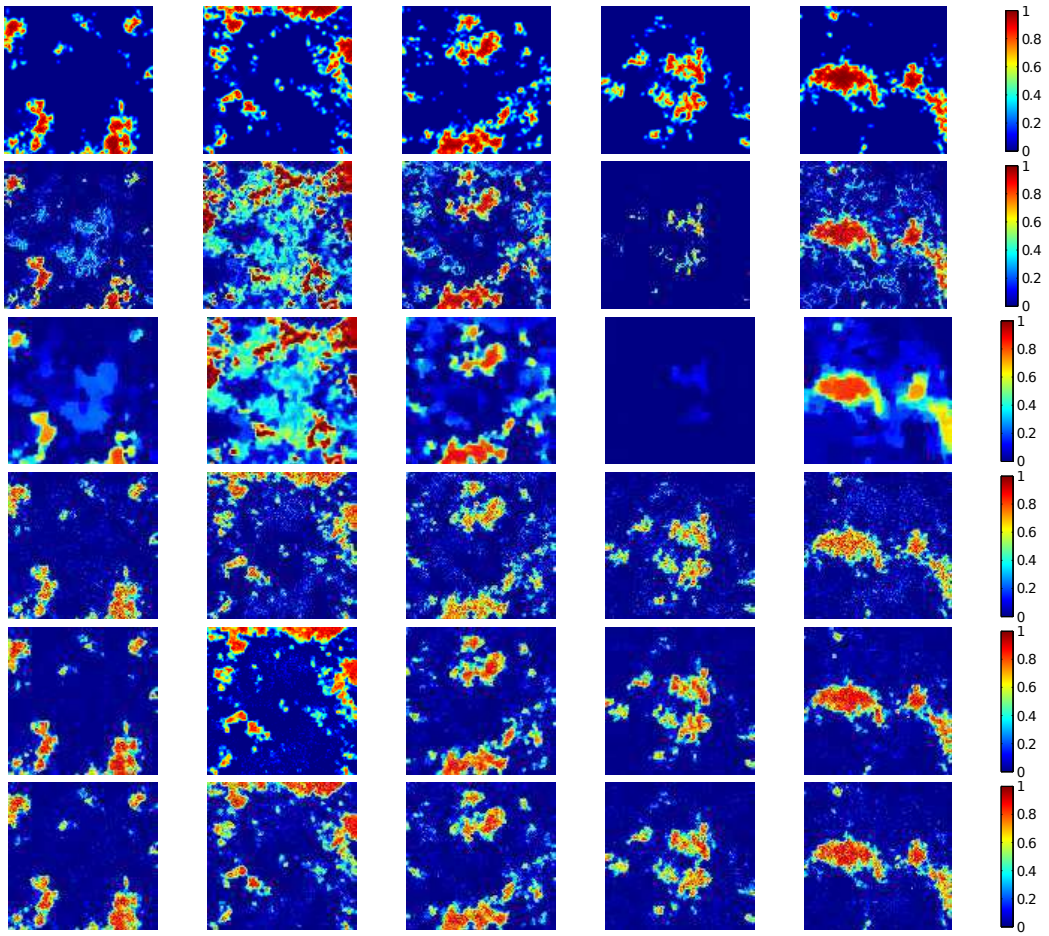


FIGURE A.10 – Carte d'abondances estimées. Chaque range, d'haute en bas : abondances réelles, résultat de FCLS, FCLS avec régularisation spatiale, K-Hype, algorithme proposé avec 4 voisinages, algorithme proposé avec 8 voisinages.

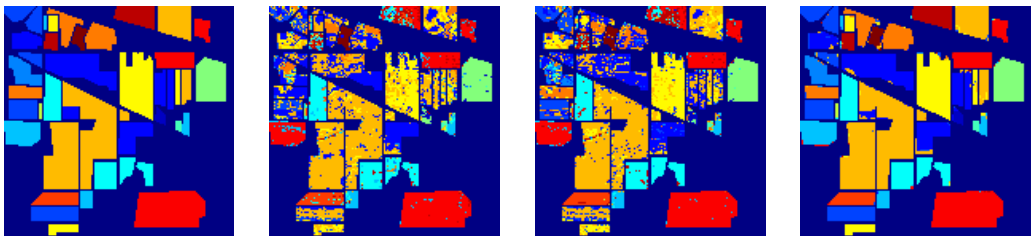


FIGURE A.11 – Résultats de classification avec les abondances : FCLS (61.36%), K-Hype (71.39%), Proposé (96.80%)