COPPE
UFRJ
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia

ENERGY-EFFICIENT ADAPTIVE FILTERS

Markus Vinícius Santos Lima

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Paulo Sergio Ramirez Diniz

Rio de Janeiro
Dezembro de 2013

ENERGY-EFFICIENT ADAPTIVE FILTERS

Markus Vinícius Santos Lima

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Paulo Sergio Ramirez Diniz, Ph.D.

Prof. Marcello Luiz Rodrigues de Campos, Ph.D.

Prof. José Antonio Apolinário Jr., D.Sc.

Prof. Renato da Rocha Lopes, Ph.D.

Prof. Cássio Guimarães Lopes, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2013

Lima, Markus Vinícius Santos

Energy-Efficient Adaptive Filters/Markus Vinícius Santos Lima. – Rio de Janeiro: UFRJ/COPPE, 2013.

XXI, 127 p.: il.; $29,7$cm.

Orientador: Paulo Sergio Ramirez Diniz

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2013.

Referências Bibliográficas: p. 101 – 109.

1. adaptive signal processing. 2. data-selective algorithms. 3. set-membership filtering. 4. sparsity. 5. energy saving. I. Diniz, Paulo Sergio Ramirez. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

To my parents and Bruna for
their love and support.

Acknowledgments

I would like to thank my advisor, Professor Paulo S. R. Diniz, for his support, guidance, and generosity. I have learned a lot from him, not only in technical and professional aspects, but also in personal issues. Hence, I thank him for being not only an advisor, but also a mentor and friend. I am particularly amazed that throughout all these years I have never seen Professor Diniz in a bad mood.

I would like to thank Professors Marcello Campos, José Apolinário Jr., Renato Lopes, and Cássio Lopes for their invaluable comments and suggestions. I have learned a lot from them and this thesis benefited from their comments.

I would like to thank the professors of the Programa de Engenharia Elétrica (PEE) who have contributed to my education. I am particularly grateful to Professors Amit Bhaya and Luiz Wagner for the inspiring courses they taught.

My special thanks go to all the friends which make the SMT lab a special place to work: Adriana Schulz, Alan Tygel, Dr. Alessandro Dutra, Alexandre Leizor, Amanda Loiola, Prof. Amaro Lima, Ana Fernanda, Anderson Oliveira, André Targino, Dr. Andreas Ellmauthaler, Dr. Bernardo da Costa, Breno Espíndola, Camila Gussen, Carlos Carlim, Carlos Jr., Catia Valdman, Claudio Verdun, Diego Felix, Prof. Diego Haddad, Prof. Fabiano Castoldi, Dr. Fábio Freeland, Felipe Gonçalves, Felipe Grael, Felipe Ribeiro (Curicica), Dr. Filipe Diniz, Prof. Flávio Ávila, Frederico Wegelin, Prof. Gabriel Araujo (Baiano), Guilherme Pinto, Hugo Carvalho, Dr. Iker Sobron, Isabela Apolinário, Prof. João Dias, Jonathan Gois, Dr. José Fernando (Zé), Leonardo Nunes (Lonnes), Prof. Lisandro Lovisolo, Luiz Gustavo (Maestro), Maurício da Costa, Prof. Michel Tcheou, Michelle Nogueira, Rafael Amado, Rafael de Jesus, Dr. Rafael Paiva, Prof. Rodrigo Peres, Rodrigo Prates, Dr. Rodrigo Torres, Prof. Tadeu Ferreira, Prof. Thiago Prego, and Prof. Wallace Martins.

I would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) for the financial support.

I would like to thank my lovely girlfriend (and future wife) Bruna Cesario for all her love, comprehension, and patience. Without her support and encouragement, this doctorate would be much more difficult to pursue.

I would like to express my deep gratitude to my parents for raising me and teaching me invaluable lessons. Both my dad Luiz Álvaro Ferreira Lima and my mother Aracy Moema Santos Lima had to make many sacrifices so that I could have a good education. I thank them for all they did for me. I would also like to share this happy moment with my brother Álvaro Luiz.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

FILTROS ADAPTATIVOS COM CONSUMO EFICIENTE DE ENERGIA

Markus Vinícius Santos Lima

Dezembro/2013

Orientador: Paulo Sergio Ramirez Diniz

Programa: Engenharia Elétrica

Filtros adaptativos são usados em muitas aplicações, tais como cancelamento de eco, identificação de sistema, redução de ruído, predição de sinal e equalização de canal. Assim, encontramos filtros adaptativos em vários dispositivos, que podem ser tão comuns quanto fones de ouvido e *smartphones*, ou tão complexos quanto estações rádio-base e sistemas de teleconferência. Portanto, a redução do consumo de energia dos filtros adaptativos é de suma importância.

Neste trabalho, a teoria de estimação de conjunto, em particular o conceito de *set-membership filtering* (SMF), é empregado a fim de gerar filtros adaptativos com a propriedade de seleção de dados, o que possibilita uma redução significativa no consumo de energia. Esta tese proporciona uma melhor compreensão dos algoritmos baseados em SMF, explicando como configurar corretamente os parâmetros de tais algoritmos a fim de explorar plenamente as suas capacidades. Em seguida, através da combinação do conhecimento adquirido sobre algoritmos com seleção de dados e de uma forma alternativa de modelar a esparsidade, dois algoritmos são propostos. De fato, esses dois algoritmos promovem esparsidade a cada iteração através de uma aproximação da norma l^0, ao invés de empregar a norma l^1 comumente utilizada. Os algoritmos propostos superam, tanto em acurácia quanto em redução da carga computational, os algoritmos estado-da-arte que exploram a esparsidade dos sinais. Além disso, o conceito de SMF é estendido para o domínio da frequência e dois novos algoritmos com seleção de dados são propostos: o primeiro é um algoritmo para equalização semi-cega para sistemas de comunicações baseados em OFDM que permite uma grande redução na quantidade de pilotos transmitidos, enquanto o segundo é um algoritmo para cancelamento de eco acústico que usa critérios psicoacústicos a fim de atualizar os coeficientes do filtro adaptativo apenas quando o eco residual for audível.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

ENERGY-EFFICIENT ADAPTIVE FILTERS

Markus Vinícius Santos Lima

December/2013

Advisor: Paulo Sergio Ramirez Diniz

Department: Electrical Engineering

Adaptive filters are used in many applications, such as echo cancellation, system identification, noise reduction, signal prediction, and channel equalization. Thus, we find adaptive filters in several devices, which can be as common as headphones and smartphones, or as complex as antenna base stations and teleconference systems. Therefore, reducing the energy consumption of adaptive filters is of paramount importance.

In this work, set estimation theory, in particular the set-membership filtering (SMF) concept, is employed in order to generate adaptive filters featuring the *data selection* property, which enables a significant reduction in energy consumption. This thesis provides a better understanding of SMF-based algorithms, explaining how to properly set the parameters of such algorithms in order to fully exploit their capabilities. Then, by combining the acquired knowledge about data-selective algorithms with an alternative form of modeling sparsity, two novel sparsity-aware data-selective algorithms are proposed. Indeed, these two algorithms promote sparsity at each iteration through an approximation to the l^0 norm, instead of employing the commonly used l^1 norm. The proposed algorithms overcome the state-of-the-art algorithms designed to exploit sparsity both in accuracy and in reduction of computational burden. In addition, the SMF concept is extended to the frequency domain and two new data-selective algorithms are proposed: the first is a semi-blind equalization algorithm for OFDM-based communications systems that allows a large reduction in the amount of pilots transmitted, whereas the second is an acoustic echo cancellation algorithm that uses psychoacoustics criteria in order to update the adaptive filter coefficients only when the residual echo becomes audible.

Contents

List of Figures

List of Tables

List of Symbols

$(\cdot)^H$ Hermitian transposition of $(\cdot)$, i.e., $(\cdot)$ is transposed and its entries are conjugated, p. 75

$(\cdot)^T$ Transposition of $(\cdot)$, p. 5

F_β Continuous and almost everywhere (a.e.) differentiable function that approximates the l^0 norm; β is a parameter that determines the quality of the approximation, p. 49

L Data reuse factor, p. 12

N Order of the FIR adaptive filter, p. 7

P Probability operator, p. 5

$P_{\rm up}(k)$ Probability of updating the filter coefficients at iteration k, p. 32

E Expected value operator, p. 5

$\mathbf{F}$ DFT matrix, p. 75

$\mathbf{F}^H$ IDFT matrix, p. 75

$\mathbf{I}$ Identity matrix, p. 5

Ψ_0^k Exact membership set, i.e., intersection of the constraint sets $\mathcal{H}(0), \mathcal{H}(1), \ldots, \mathcal{H}(k)$, p. 15

$\Psi_{k_1}^{k_2}$ Intersection of the constraint sets $\mathcal{H}(k_1), \mathcal{H}(k_1+1), \ldots, \mathcal{H}(k_2)$, p. 15

$\mathbf{R}(k)$ Auxiliary matrix $\mathbf{R}(k) \triangleq \mathbf{X}^T(k)\mathbf{X}(k)$, p. 21

$\mathbf{S}(k)$ Auxiliary matrix $\mathbf{S}(k) \triangleq [\mathbf{R}(k)+\delta\mathbf{I}]^{-1} \triangleq \left[\mathbf{X}^T(k)\mathbf{X}(k)+\delta\mathbf{I}\right]^{-1}$, p. 21

$\mathrm{tr}\{\cdot\}$ Trace of matrix $\{\cdot\}$, p. 5

Θ	Feasibility set (set of acceptable solutions), p. 15
$\mathbf{X}(k)$	Input Matrix, p. 13
$\mathcal{S}$	Set comprised of all possible pairs $(\mathbf{x}, d)$, p. 14
$\circ$	Element-wise multiplication of two vectors, p. 75
$\mathbf{d}(k)$	Desired vector, p. 13
δ	Regularization factor, p. 11
$\div$	Element-wise division of two vectors, i.e., each entry of the vector on the left is divided by its corresponding entry of the vector on the right, p. 75
$\mathbf{e}(k)$	Error vector or *a priori* error vector, p. 13
$\mathbf{f}_\beta(\mathbf{z})$	Gradient of $F_\beta(\mathbf{z})$ with respect to $\mathbf{z}$, p. 51
$\overline{\gamma}$	Upper bound for the magnitude of the error signal, i.e., it defines how much error is acceptable, p. 14
$\boldsymbol{\gamma}(k)$	Constraint vector (CV), p. 20
$\mathbb{C}$	Set of complex numbers, p. 75
$\mathbb{N}$	Set of natural numbers, p. 5
$\mathbb{R}$	Set of real numbers, p. 5
$\mathbb{R}_+$	Set of nonnegative real numbers, p. 5
$\mathbb{Z}$	Set of integer numbers, p. 5
$\mathbf{0}$	Vector or matrix with all entries equal to 0, p. 5
μ	Step size or convergence factor, p. 10
$\mathbf{n}(k)$	Noise vector, p. 20
$\mathbf{p}(k)$	Perturbation/step added to $\mathbf{w}(k)$ to generate $\mathbf{w}(k+1)$, p. 25
σ_n^2	Variance of the noise signal, p. 14
EMSE	Excess MSE: $\lim_{k\rightarrow\infty} \mathrm{E}[\tilde{e}^2(k)]$, p. 32
$\tilde{e}(k)$	noiseless *a priori* error signal, p. 31
$\triangleq$	Definition, p. 5

$\boldsymbol{\varepsilon}(k)$	*A posteriori* error vector, p. 20
$\mathbf{w}(k)$	Coefficient vector (coefficients of the adaptive filter), p. 7
$\mathbf{w}_*$	Impulse response of the unknown system, p. 62
$\mathbf{w}_o$	Wiener filter, p. 11
$\mathbf{x}(k)$	Input vector or regressor, p. 7
$\mathbf{y}(k)$	Output vector, p. 13
$d(k)$	Desired or reference signal, p. 7
$e(k)$	Error signal, p. 8
$f_\beta(z_n)$	Derivative of $F_\beta(\mathbf{z})$ with respect to z_n, p. 50
k	Iteration counter, p. 7
$n(k)$	Noise signal, p. 8
$y(k)$	Output signal (output of the adaptive filter), p. 7
$\mathcal{H}(k)$	Constraint set using the kth data-pair $(\mathbf{x}(k), d(k))$, p. 15
$\text{diag}\{\mathbf{x}\}$	Diagonal matrix having $\mathbf{x}$ on its main diagonal, p. 5

List of Abbreviations

APA-SSI	AP Algorithm for Sparse System Identification, p. 54
AP	Affine Projection, p. 3, 7, 17, 46
BER	Bit-Error Rate, p. 83
BNLMS	BiNormalized LMS, p. 8
CFD	Constrained Frequency-Domain, p. 87
CV	Constraint Vector, p. 3, 18
DFT	Discrete Fourier Transform, p. 75
ED-CV	Exponential Decay CV, p. 26
EMSE	Excess MSE, p. 31
F-SM-NLMS	Frequency-domain SM-NLMS, p. 75
F-SMF	Frequency-domain Set-Membership Filtering, p. 75
FFT	Fast Fourier Transform, p. 74
FIR	Finite-duration Impulse Response, p. 7
FMEB-CV	Fixed Modulus Error-Based CV, p. 24
GMF	Geman-McClure Function, p. 50
IDFT	Inverse DFT, p. 75
IPAPA	Improved PAPA, p. 47
IPNLMS	Improved PNLMS, p. 46
LEM	Loudspeaker-Enclosure-Microphone, p. 86
LF	Laplace Function, p. 50

LMS	Least-Mean-Square, p. 8, 17, 46
LS	Least Squares, p. 9
MMSE	Minimum MSE, p. 9
MSE	Mean-Squared Error, p. 3, 9, 17, 46
NLMS	Normalized LMS, p. 8, 46
OFDM	Orthogonal Frequency-Division Multiplexing, p. 4, 74
PAPA	Proportionate AP Algorithm, p. 47
PNLMS	Proportionate NLMS, p. 46
QAPA-SSI	Quasi APA-SSI, p. 54
QPSK	Quadrature Phase-Shift Keying, p. 79
QSSM-AP	Quasi SSM-AP, p. 59
RLS	Recursive Least-Squares, p. 8, 17, 46
RV	Random Variable or Random Vector, p. 31
RZA-APA	Reweighted ZA-APA, p. 47, 54
SBE	Semi-Blind Equalization, p. 80
SC-CV	Simple Choice CV, p. 24
SM-AP	Set-Membership Affine Projection, p. 3, 8, 17
SM-BNLMS	Set-Membership BiNormalized LMS, p. 8, 17
SM-CFD	Set-Membership Constrained Frequency-Domain, p. 87
SM-NLMS	Set-Membership Normalized LMS, p. 8, 17
SM-PAPA	Set-Membership PAPA, p. 47
SM-PNLMS	Set-Membership PNLMS, p. 46
SMF	Set-Membership Filtering, p. 1, 6, 17
SM	Set-Membership, p. 1, 17
SNR	Signal-to-Noise Ratio, p. 83

SSM-AP	Sparsity-aware SM-AP, p. 59
WLS	Weighted Least Squares, p. 9
WSS	Wide-Sense Stationary, p. 10
ZA-APA	Zero-Attracting AP Algorithm, p. 47, 54

Chapter 1

Introduction

In recent years, mainly due to lower prices of sensors and storage devices, we have seen a huge increase in the amount of data to be processed and/or stored. For example, multiple antennas have been used to increase the capacity of wireless communications systems, multiple microphones have been employed in multimedia applications such as sound source localization and speech/sound enhancement, multiple sensors have also been deployed in networks to monitor faults in nodes and to control data traffic, as well as the databases are constantly growing.

In a world where data are so abundant, the capability of evaluating data automatically is of paramount importance. Such a capability, herein called *data selection*, gives rise to *data-selective filters*, which can be regarded as automatic tools for evaluation and selection of data. These filters are responsible for discarding unwanted data, such as redundancy and outliers, providing the end user with the portion of the data that contains information of his interest. An example of data-selective filter is the widely used Google Search.

If we add the ability of self-tuning to the data-selective filters, then we obtain the so-called *data-selective adaptive filters* and, in the context of *adaptive signal processing*, such a self-tuning is known as *learning* process. Thus, data-selective adaptive filters are filters that learn as data are evaluated and processed. Since these filters discard irrelevant data, only the relevant portion of the data are used in their learning process, a key feature of data-selective adaptive filters. Indeed, such a feature yields adaptive filters that are superior to the classical adaptive filters, both in performance and in reducing the total number of arithmetic operations performed, which enables the *reduction in energy consumption*.

The energy-efficient adaptive filters addressed in this thesis are data-selective adaptive filters that employ the *set-membership filtering* (SMF) concept [1]. This concept is fairly recent since it was proposed in 1998, and has given rise to adaptive filters that exhibit lower computational burden and estimation error, as compared to the classical adaptive filters [2]. The literature on set-membership (SM) algorithms still lacks contributions, both in applications and analyses, and this thesis aims at reducing this gap.

This chapter is organized as follows. In Section 1.1, we present the main motivations and possible impact of this thesis. The goals are presented in Section 1.2. Section 1.3 presents an outline of this thesis and the notation is introduced in Section 1.4.

1.1 Motivations and Impact

Adaptive filters are employed in many devices, such as smartphones, modern headphones, and teleconference systems, and appear in applications like acoustic echo cancellation, noise reduction, equalization, and signal prediction.

An adaptive filter adjusts itself according to an algorithm, consisting of a set of instructions that determines how the filter coefficients shall be updated. The algorithms used in adaptive filtering are closely related to optimization techniques, since these algorithms are iterative solutions to optimization problems. However, unlike optimization techniques [3], adaptive filtering algorithms do not possess a stopping criterion, i.e., they keep running until the application or device is shut down.[1] Therefore, the development of energy-efficient adaptive filtering algorithms is an important issue, especially in the case of algorithms that run on battery-operated devices.

In addition, the classical algorithms used in adaptive filtering perform coefficient updates at every iteration, consuming a fixed amount of energy per iteration. This is very counterintuitive! Indeed, as the number of iterations increases and the adaptive filter learns from the observed data, it is expected that the adaptive filter could reduce its energy consumption since there is less to be learned. This is exactly what data-selective adaptive filters do: they evaluate the input data to check how much innovation is conveyed and, based on this evaluation, the algorithm decides whether an update is required or not.

As an example, one data-selective algorithm that we propose was able to achieve a better estimation performance and also reduce the number of arithmetic operations in 85%, as compared to a classical algorithm (the details are shown in Chapter 4).

In summary, the widespread use of adaptive filters together with the absence of a stopping criterion in adaptive filtering algorithms make us believe that data-selective algorithms have great potential to reduce energy consumption. Therefore, this topic has relevant practical appeal, especially in a moment when there is a growing concern in developing technologies that have a reduced environmental impact, the so-called

[1] The absence of a stopping criterion in adaptive filtering algorithms is justified by the fact that the filter specifications, in addition to being unknown, can also be time-varying.

green technologies.

1.2 Goal

The goal of this thesis is twofold:

- To develop data-selective adaptive filtering algorithms that exhibit competitive to better performance and lower computational burden, as compared to the classical adaptive filtering techniques;
- To obtain a deeper understanding of the data-selective algorithms through analytic tools and also considering the geometric viewpoint. Some of the aspects discussed are: stability criteria, mean-squared error (MSE), and updating scheme.

In summary, in this thesis we develop and analyze data-selective adaptive filtering algorithms.

1.3 Thesis Outline

In this thesis, we address data-selective adaptive filters for: (i) general applications involving filtering, see Chapters 3 and 4; and (ii) specific applications, such as echo cancellation and equalization, see Chapter 5. In addition, background material is condensed in Chapter 2, whereas Chapters 3 to 5 include all contributions of this work. This thesis is organized as follows.

Chapter 2 addresses set estimation theory in adaptive signal processing aiming at presenting the set-membership filtering (SMF) concept and its advantages over the classical algorithms used in adaptive filtering. It also provides a brief review of adaptive signal processing and presents some of the classical algorithms, among which we focus on the affine projection (AP) algorithm due to its importance to the subsequent chapters. Indeed, in Chapters 3 and 4 we present some SM-versions of the AP algorithm. Chapter 2 basically constitutes background material to the following chapters.

Chapter 3 presents the set-membership affine projection (SM-AP) algorithm, which is one of the most general SMF-based algorithms. This chapter begins with an explanation of how the SM-AP algorithm manages to estimate a member of the exact membership set using a finite number of constraint sets, i.e., using only a portion of the observed data. Such an explanation motivates the optimization problem related to the SM-AP algorithm and also explains how to set properly one of its parameters, which is known as constraint vector (CV), in order to improve accuracy.

In addition, two novel CVs are described and analyzed from both geometric and analytic viewpoints. Then, an expression for the steady-state MSE of the SM-AP algorithm using a specific CV is derived. This theoretic expression matches the simulation results very well. Moreover, using such an expression we prove that the steady-state MSE of the SM-AP algorithm can always be lower than the MSE of the AP algorithm. In other words, by properly setting the SM-AP algorithm, it exhibits better accuracy with a reduced number of updates (i.e., saving energy), as compared to its non-SM counterpart. Most of the mathematical derivations and proofs are left to Appendix A.

Chapter 4 presents sparsity-aware data-selective adaptive filters, i.e., we propose data-selective algorithms capable of exploiting the sparse nature of signals and systems. Based on the literature on classical adaptive filtering algorithms, it is widely known that their convergence speed degrades as the impulse response of the involved systems/filters become longer, even when most of their energy is concentrated in few coefficients. The goal of sparsity-aware algorithms is to overcome this issue by exploiting the sparsity of such filters. In order to do that, these algorithms rely on sparsity-promoting functions, usually the l^0 and l^1 norms. In Chapter 4, we follow a different approach by using an approximation to the l^0 norm as the sparsity-promoting function, and we explain the advantages of such a function over the l^0 and l^1 norms. We devise two sparsity-aware data-selective adaptive algorithms and study some of their properties, such as their updating schemes, stability, and number of operations per iteration. Simulation results considering sparse and compressible signals show that the proposed algorithms outperform the state-of-the-art algorithms designed to exploit sparsity. Some mathematical derivations and proofs are left to Appendix B.

Chapter 5 addresses set-membership filtering in the frequency domain. Frequency-domain algorithms are very appealing to applications involving filters with very-high orders, such as the ones required in acoustic echo cancellation, since convolutions in time domain are replaced by scalar multiplications in the frequency domain [2]. On the other hand, such algorithms require block-processing and, therefore, they introduce latency, which might be unacceptable for some applications. In Chapter 5, we develop two new algorithms. The first is a semi-blind equalization algorithm for communications systems based on orthogonal frequency-division multiplexing (OFDM). Many technologies and standards employ OFDM as their core modulation scheme [4] and the proposed algorithm allows a significant reduction on the amount of pilots transmitted. The second algorithm is designed for acoustic echo cancellation and uses psychoacoustics criteria in order to update the adaptive filter coefficients only when the residual echo becomes audible.

The conclusions and future works are addressed in Chapter 6.

1.4 Notation

In this section, we define most of the notation used throughout this thesis. We prefer not to include here definitions that are not extensively used. Such definitions are introduced only at the moment they are necessary so that the reader does not have to return to this section every time he/she finds a new symbol.

Equalities are represented by $=$, but when they represent a definition we use $\triangleq$. We use $\mathbb{R}$ to denote the real field, $\mathbb{R}_+$ to denote the set of nonnegative real numbers, and the set of integer and natural numbers are denoted by $\mathbb{Z}$ and $\mathbb{N}$, respectively. The natural numbers are defined as $\mathbb{N} \triangleq \{0, 1, 2, \ldots\}$, i.e., 0 is included.

In addition, scalars are denoted by lowercase letters (e.g., x), vectors by lowercase boldface letters (e.g., $\mathbf{x}$), and matrices by uppercase boldface letters (e.g., $\mathbf{X}$). The symbol $(\cdot)^T$ denotes transposition and all vectors are column vectors so that the inner product between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is $\mathbf{x}^T\mathbf{y} \in \mathbb{R}$.

We define $\mathrm{E}[\cdot]$, $\mathrm{tr}\{\cdot\}$, and $P[\cdot]$ as the expected value, the trace, and the probability operators, respectively, and define $\mathbf{I}$ as the identity matrix and $\mathbf{0}$ as a vector (or matrix) whose elements are all zeros. In addition, $\mathrm{diag}\{\mathbf{x}\}$ is a diagonal matrix having $\mathbf{x}$ on its main diagonal.

Chapter 2

Set Estimation Theory in Adaptive Signal Processing

For a long time, machine learning mechanisms have been based on the classical theory of point estimation [5, 6]. Nevertheless, the importance of set estimation theory is growing as the advantages of such a paradigm become clearer. Set theoretic optimization is the proper approach to tackle problems in which uncertainty is unavoidable since, instead of searching for a unique point within the feasible region that minimizes or maximizes some objective function, it searches for a set of points which are acceptable solutions given the inherent uncertainty of the problem [7].

Hence, set estimation theory has a great appeal to practical applications [7]. Indeed, since most applications present some sources of uncertainty, like imprecision due to measurements and modeling, noise, and interference effects, it makes more sense to search for acceptable solutions rather than a single solution that one would find using the traditional point estimation theory.

From the practitioner viewpoint, it is quite common to have some knowledge about the uncertainties of the phenomenon being studied. For instance, we usually know the precision of the instruments used for measuring, we know how much a signal is distorted due to quantization, as well as we can estimate or measure noise and interference. The set estimation theory provides means to incorporate such knowledge into the estimation problem by properly defining the set of acceptable solutions.

The inclusion of a priori information into the optimization problem results in additional degrees of freedom that one can exploit in order to obtain an estimation technique with interesting properties, such as better accuracy, robustness against noise, and *data selection*. The set-membership filtering (SMF) concept is a representative of set estimation theory that yields data-selective adaptive filters.

This chapter presents a brief review of adaptive signal processing aiming at presenting the SMF concept. We present just what we consider essential to follow the remaining chapters. The reader should refer to the books [2, 6, 8, 9] for a thorough discussion of the principles of adaptive filtering. Section 2.1 presents the essential elements in adaptive filtering and discusses the differences between point estimation theory and set estimation theory. In Section 2.2 we present some classical adaptive filtering algorithms focusing on the affine projection (AP) algorithm. Such a focus is justified by the generality of the AP algorithm and also by the fact that the next two chapters present set-theoretic versions of the AP algorithm. Then, in Section 2.3 we address the set-theoretic adaptive filtering and the SMF.

2.1 Adaptive Signal Processing

In this section we present the basics of adaptive signal processing. We describe the operation of an adaptive filter in a general setup in Subsection 2.1.1. Then, in Subsection 2.1.2, we address the differences between point estimation theory and set estimation theory.

2.1.1 Basics

Figure 2.1 depicts a general adaptive filter configuration. At a given iteration k, the coefficients of the adaptive filter, herein called *coefficient vector*, are represented by $\mathbf{w}(k) \in \mathbb{R}^{N+1}$, where

$$\mathbf{w}(k) \triangleq [w_0(k) \;\; w_1(k) \;\; \ldots \;\; w_N(k)]^T \tag{2.1}$$

and $N \in \mathbb{N}$ is the *order of the adaptive filter*, which is assumed to be a finite-duration impulse response (FIR) adaptive filter [2, 10]. The adaptive filtering algorithm requires two signals: $d(k) \in \mathbb{R}$, which is the *desired signal* (also known as reference signal), and $\mathbf{x}(k) \in \mathbb{R}^{N+1}$, which is the *input vector* (also called regressor [8]). The input vector has the following structure:

$$\mathbf{x}(k) \triangleq [x(k) \;\; x(k-1) \;\; \ldots \;\; x(k-N)]^T, \tag{2.2}$$

i.e., the adaptive filter has a transversal structure (also known as tapped-delay line [2, 6]).

The output of the adaptive filter $y(k) \in \mathbb{R}$ due to a given input vector $\mathbf{x}(k)$ is defined as

$$y(k) \triangleq \mathbf{w}^T(k)\mathbf{x}(k). \tag{2.3}$$

It is expected that $y(k)$ becomes closer to $d(k)$ as k grows. In order to measure the

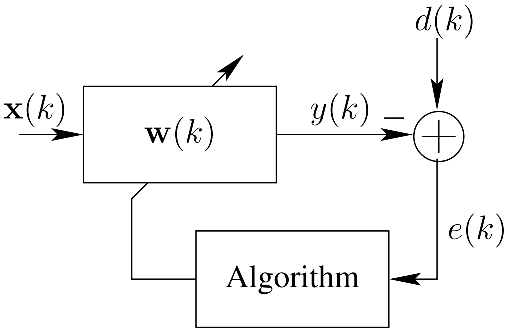


Figure 2.1: *General adaptive filter configuration.*

discrepancy between these two variables, we define the error signal $e(k) \in \mathbb{R}$ as

$$e(k) \triangleq d(k) - y(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k). \tag{2.4}$$

The error signal is used by the algorithm during the update of the coefficient vector. It is worth mentioning that, in many practical applications, $d(k)$ is usually corrupted by some kind of noise, which we denote as $n(k) \in \mathbb{R}$.

The variables depicted in Figure 2.1 and defined in this section are extensively used in Chapters 3 and 4, which correspond to the core material of this thesis. In Chapter 5 we deal with complex-valued variables, and a slight modification on the definition of the output signal $y(k)$ will be necessary. In addition, in that chapter we also deal with the case in which the desired signal $d(k)$ is not always available, which is denominated as semi-supervised or *semi-blind* setup. Such particularities are introduced only when necessary.

2.1.2 Adaptive Filtering: Point *vs.* Set Estimation Theories

In the adaptive filtering context, the algorithms can be classified according to their estimation procedure into two groups: the *point estimation* and the *set estimation* techniques.

The point estimation group contains the majority of the adaptive filtering algorithms, including the classical ones, such as the least-mean-square (LMS), the normalized LMS (NLMS), the binormalized LMS (BNLMS), the affine projection (AP), and the recursive least-squares (RLS) [2, 6, 8]. These algorithms try to find a single solution that is optimal with respect to a given objective function, usually a quadratic function of the parameters to be estimated, i.e., the coefficient vector.

The group of set estimation techniques, on the other hand, has fewer representatives. Some examples are the set-membership NLMS (SM-NLMS) [1], the set-membership binormalized LMS (SM-BNLMS) [11], and the set-membership AP (SM-AP) [12] algorithms. Unlike the point estimation techniques, these algorithms search for any acceptable solution. In addition, these algorithms employ the SMF concept, which confers them the data selection property. Thus, these algorithms evaluate data before performing an update and, as a result, they are able to operate in an economic mode, in which most of the time they are not updating the filter coefficients. In addition to this reduction of energy consumption due to fewer updates, these algorithms have proven to be superior to their classical counterparts [2, 13].

2.2 Classical Adaptive Filtering

In the classical adaptive filtering, the algorithms are devised so that a function of the error signal $e(k)$, defined in Eq. (2.4), is minimized. Two widely used objective functions are [2, 6]:

- Weighted Least Squares (WLS): $\sum_{i=0}^{k} \lambda^i e^2(k-i)$, where $\lambda \in \mathbb{R}_+$ is a constant that should be chosen as $0 < \lambda \leq 1$;
- Mean-squared error (MSE): $\mathrm{E}\left[e^2(k)\right]$;

The WLS can be interpreted as a generalization of the standard least squares (LS). Indeed, by choosing $\lambda = 1$, the WLS becomes the LS. Adaptive filtering algorithms that minimize the WLS belong to the family of LS algorithms, which has the RLS algorithm as its most popular member.

The family of algorithms that minimize the MSE is known as minimum MSE (MMSE). Among the MMSE adaptive filtering techniques, we must highlight the importance of one of its subgroups of algorithms which is known as *stochastic gradient* algorithms. The stochastic gradient algorithms are characterized by updating the coefficient vector $\mathbf{w}(k)$ following the opposite direction of an approximation to the gradient of the MSE function. These algorithms are very important due to their low computational complexity and the LMS algorithm is certainly the most famous of them. Some modifications of the LMS algorithm have also been proposed and, therefore, the term LMS-based algorithms is also used. Examples of such algorithms are the NLMS, the BNLMS, and the AP [2, 6].

In the following subsections we present a brief review of the LMS, NLMS, RLS, and AP algorithms.

2.2.1 Least-Mean-Square (LMS) Algorithm

The field known as adaptive signal processing (or simply adaptive filtering) emerged in 1960, when Bernard Widrow and his Ph.D. student Marcian E. Hoff proposed the LMS algorithm [14]. In the LMS algorithm, the coefficient vector is updated according to the following recursion:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k), \tag{2.5}$$

where $\mu \in \mathbb{R}_+$ is the step size, also known as convergence factor, and should be chosen as [2]

$$0 < \mu < \frac{1}{\mathrm{tr}\{\mathrm{E}\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]\}}, \tag{2.6}$$

where $\mathrm{E}\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]$ is the autocorrelation matrix of the input signal.

Observe that (2.6) does not provide a practical way of computing the upper bound for μ. Indeed, even if we assume that the input signal is wide-sense stationary (WSS), it might be unfeasible to have an accurate estimate for such a matrix (or its trace). In practice, however, (2.6) states that we can choose a small μ and, if it is not enough to guarantee stability, then we reduce it even further.

2.2.2 Normalized LMS (NLMS) Algorithm

The NLMS algorithm was independently proposed in [15] and [16], both in 1967, and is probably the most widely used algorithm nowadays. This algorithm overcomes two major problems in the LMS algorithm [6]:

1. Choice of μ;

2. Gradient noise amplification.

The problem related to the choice of μ was already explained in the previous subsection. The gradient noise amplification occurs because the gradient (and thus, the updating rule) of the LMS algorithm depends on the input vector $\mathbf{x}(k)$, which may have a high norm. Thus, when the observed data is corrupted by some kind of noise, a large gradient will amplify this noise leading to a higher MSE.

The NLMS algorithm solves these problems by normalizing the energy of the input vector, which leads to the following updating rule:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{\|\mathbf{x}(k)\|^2} e(k)\mathbf{x}(k), \tag{2.7}$$

where $\|\cdot\|$ stands for the 2-norm of a vector, also known as Euclidean norm. Such a modification eliminates the problems due to input vectors with high $\|\mathbf{x}(k)\|^2$, and also allows the step size to be chosen as:

$$0 < \mu < 2. \tag{2.8}$$

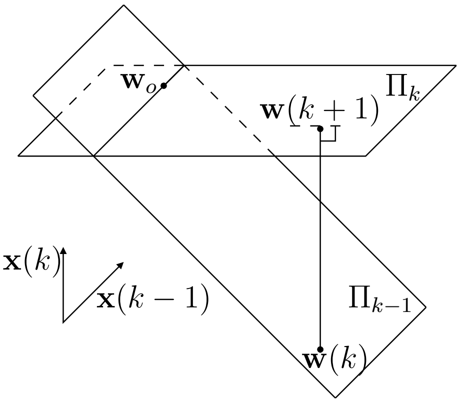


Figure 2.2: *Updating scheme of the NLMS algorithm with* $\mu = 1$.

The range of μ in (2.8) guarantees the convergence of the algorithm in noiseless scenarios. In practice, it is recommended to set μ as $0 < \mu \leq 1$.

In addition, the NLMS algorithm is usually implemented as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu}{\|\mathbf{x}(k)\|^2 + \delta} e(k)\mathbf{x}(k), \tag{2.9}$$

where $\delta \in \mathbb{R}_+$ is the regularization factor, which is used to avoid numerical problems that may occur when $\|\mathbf{x}(k)\|^2$ is close to 0.

Let us define the hyperplane Π_k as

$$\Pi_k \triangleq \left\{ \mathbf{w} \in \mathbb{R}^{N+1} : d(k) - \mathbf{w}^T \mathbf{x}(k) = 0 \right\}, \tag{2.10}$$

i.e., Π_k is the hyperplane where the a posteriori error (i.e., the error computed using the coefficient vector after the update) is equal to 0. Then, Figure 2.2 provides an interesting interpretation of the updating process of the NLMS algorithm. In this figure, we consider that $\mathbf{w}(k) \in \mathbb{R}^3$, thus Π_{k-1} and Π_k are actually planes, and we also consider $\mu = 1$. In such a setup, the NLMS algorithm generates $\mathbf{w}(k+1)$ as a projection of $\mathbf{w}(k)$ onto Π_k.[1] Vector $\mathbf{w}_o$ represents the optimal coefficients of the adaptive filter given by the Wiener solution, which belongs to the intersection of Π_k and Π_{k-1} considering the noiseless case.

2.2.3 Recursive Least-Squares (RLS) Algorithm

The RLS algorithm for transversal adaptive filters was first proposed in [18]. Later, alternative implementations have been proposed. These different versions of the RLS algorithm were proposed to reduce the complexity of the algorithm, or due to stability issues. Here, we follow the approach of [2, 18].

The RLS algorithm aims at minimizing the sum of the weighted squared-errors,

[1] A projection of a point $\mathbf{z}$ on a set Z is any point $\mathbf{z}_p \in Z$ which is closest to $\mathbf{z}$ [17].

i.e., its objective function is the WLS. It uses an approximation to the correlation matrix in order to “whiten” the input signal and, consequently, increase the convergence speed. Therefore, in stationary scenarios, the RLS algorithm is much faster than the LMS and NLMS algorithms. However, its computational complexity is also much higher due to several matrix-vector multiplications.[2]

The RLS algorithm is characterized by the following recursion:

$$\mathbf{w}(k) = \mathbf{S}_D(k)\mathbf{p}_D(k), \tag{2.11}$$

where $\mathbf{S}_D(k) \in \mathbb{R}^{(N+1)\times(N+1)}$ and $\mathbf{p}_D(k) \in \mathbb{R}^{N+1}$ are called the inverse of the deterministic autocorrelation matrix of the input signal and the deterministic cross-correlation vector between input and desired signals, respectively, and they are defined as:

$$\mathbf{S}_D(k) \triangleq \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}\right], \tag{2.12}$$

$$\mathbf{p}_D(k) \triangleq \lambda\mathbf{p}_D(k-1) + d(k)\mathbf{x}(k), \tag{2.13}$$

where $\lambda \in \mathbb{R}_+$ is the forgetting factor and should be chosen as $0 < \lambda \leq 1$. In practice, we usually choose $\lambda \in [0.9, 1.0)$.

2.2.4 Affine Projection (AP) Algorithm

The AP algorithm was developed in [19], in 1984, aiming at increasing the convergence speed of stochastic gradient algorithms. In particular, the target of the AP algorithm was to solve an issue related to the NLMS algorithm: the degradation of its convergence speed when the input signal is colored.

Motivated by Figure 2.2, Ozeki and Umeda noticed that the convergence speed of the NLMS algorithm could be increased if the update were a projection of $\mathbf{w}(k)$ onto $\Pi_{k-1}\bigcap\Pi_k$. Since $\mathbf{x}(k)$ is orthogonal to Π_k, for each k, they figured out that the perturbation applied to $\mathbf{w}(k)$ that yields $\mathbf{w}(k+1) \in \Pi_{k-1}\bigcap\Pi_k$ had to be a linear combination of the current input vector $\mathbf{x}(k)$ and the previous input vector $\mathbf{x}(k-1)$, as shown in Figure 2.3.

Figure 2.3 depicts the updating process of $\mathbf{w}(k) \in \mathbb{R}^3$ considering an AP algorithm with $\mu = 1$ and reusing one previous input vector. Observe that due to the data reuse, the AP algorithm was able to get close to $\mathbf{w}_o$ faster than the NLMS algorithm did. Such an idea can be generalized to $(N+1)$-dimensional spaces in which L previous input vectors are used simultaneously with the current input vector, where

[2] This comment considers the standard transversal RLS algorithm. As already mentioned, there exist alternative versions of the RLS algorithm that are less complex, but these implementations are usually affected by stability issues [2].

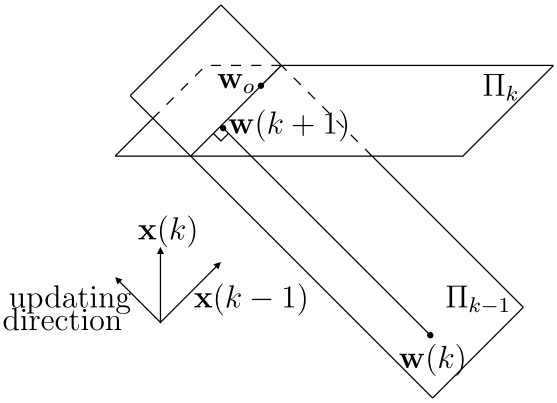


Figure 2.3: *Updating process of the AP algorithm considering $L = 1$ and $\mu = 1$.*

L is known as data reuse factor.

The updating equation of the AP algorithm is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{X}(k) \left[\mathbf{X}^T(k)\mathbf{X}(k)\right]^{-1} \mathbf{e}(k), \tag{2.14}$$

but like the NLMS algorithm, we usually include a regularization factor δ to avoid numerical issues due to matrix inversion, and we actually implement the following recursion:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{X}(k) \left[\mathbf{X}^T(k)\mathbf{X}(k) + \delta \mathbf{I}\right]^{-1} \mathbf{e}(k), \tag{2.15}$$

where $\mathbf{X}(k) \in \mathbb{R}^{(N+1)\times(L+1)}$ is the input matrix defined as

$$\mathbf{X}(k) \triangleq \left[\mathbf{x}(k) \quad \mathbf{x}(k-1) \quad \ldots \quad \mathbf{x}(k-L)\right], \tag{2.16}$$

and the error vector $\mathbf{e}(k) \in \mathbb{R}^{L+1}$ is defined as

$$\mathbf{e}(k) \triangleq \mathbf{d}(k) - \mathbf{y}(k). \tag{2.17}$$

The desired vector $\mathbf{d}(k) \in \mathbb{R}^{L+1}$ and the output vector $\mathbf{y}(k) \in \mathbb{R}^{L+1}$ are, respectively, given by

$$\mathbf{d}(k) = \left[d(k) \quad d(k-1) \quad \ldots \quad d(k-L)\right]^T, \tag{2.18}$$

$$\mathbf{y}(k) = \mathbf{X}^T(k)\mathbf{w}(k). \tag{2.19}$$

The geometric interpretation of Figure 2.3 also reveals the objective function that is actually minimized by the AP algorithm with $\mu = 1$. Indeed, the optimization problem is:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = 0, \end{aligned} \tag{2.20}$$

i.e., we want to find $\mathbf{w}(k+1)$ which is as close as possible to $\mathbf{w}(k)$ such that $\mathbf{w}(k+1) \in \Pi_k \bigcap \Pi_{k-1} \bigcap \ldots \bigcap \Pi_{k-L}$.

Similarly to the NLMS algorithm, in order to guarantee the convergence of the AP algorithm we should choose the step size as $0 < \mu < 2$, considering noiseless scenarios. Once again, in practice we should choose $0 < \mu \leq 1$.

In this section, we focused on the AP algorithm because it is not as complex as the RLS algorithm and because it generalizes the NLMS algorithm. The AP algorithm has the data reuse factor L, which controls the trade-off between complexity and convergence speed. The AP algorithm with $L = 0$ and $L = 1$ becomes the NLMS and the BNLMS algorithms, respectively. It is already known that as L increases, the convergence speed increases as well, but the steady-state MSE of the AP algorithm also increases [2, 20].

2.3 Set-theoretic Adaptive Filtering and the Set-Membership Filtering

As previously mentioned, set estimation theory is very appealing due to its additional degrees of freedom that enable one to take into consideration the uncertainties inherent to the problem.

Despite its advantages, set estimation theory is rarely found in the adaptive filtering literature. An important example of set estimation theory in adaptive signal processing is the *set-membership filtering* (SMF). Indeed, the SMF is remarkable because it connects set-theoretic estimation with *data selection*, enabling a reduction of the required computational burden and, consequently, *saving energy*. In addition to the reduction of the computational burden, SMF-based algorithms are also more robust against noise [13].

The SMF concept appeared in [1] and is suitable to adaptive filtering problems that are linear-in-parameters, i.e., given the input vector $\mathbf{x} \in \mathbb{R}^{N+1}$ and the filter coefficients $\mathbf{w} \in \mathbb{R}^{N+1}$, the output of the filter is given by $y = \mathbf{w}^T\mathbf{x} \in \mathbb{R}$.

The SMF criterion aims at estimating the parameter $\mathbf{w}$ that leads to an error signal $e \triangleq d - y \in \mathbb{R}$ whose magnitude is upper bounded by a constant $\overline{\gamma} \in \mathbb{R}_+$, for all possible pairs $(\mathbf{x}, d)$, where $d \in \mathbb{R}$ is the desired (or reference) signal [1]. The variable $\overline{\gamma}$ determines how much error is acceptable and is usually chosen based on a priori information about the sources of uncertainty. Most of the time, we assume that such uncertainty is caused by a noise signal n whose variance is σ_n^2, and $\overline{\gamma}$ is chosen as a function of σ_n^2.

Alternatively, denoting by $\mathcal{S}$ the set comprised of all possible pairs $(\mathbf{x}, d)$, we can state the SMF criterion as finding $\mathbf{w}$ that satisfies $|e| = |d - \mathbf{w}^T\mathbf{x}| \leq \overline{\gamma}$, $\forall(\mathbf{x}, d) \in \mathcal{S}$. That is, by defining the *feasibility set* (the set of acceptable solutions) Θ as

$$\Theta \triangleq \bigcap_{(\mathbf{x},d)\in\mathcal{S}} \left\{\mathbf{w} \in \mathbb{R}^{N+1} : |d - \mathbf{w}^T\mathbf{x}| \leq \overline{\gamma}\right\}, \tag{2.21}$$

then the SMF criterion can be summarized as finding a $\mathbf{w} \in \Theta$.

Considering online applications, Eq. (2.21) does not provide a practical way of determining Θ or a point in it since we do not have $\mathcal{S}$. So, it is important to develop iterative techniques, which are referred to as *set-membership adaptive recursive techniques* (SMARTs) [1].

Clearly, the best we can do to iteratively estimate Θ is to use all data available up to the current iteration k, i.e., use the pairs $(\mathbf{x}(0), d(0))$, $(\mathbf{x}(1), d(1))$, …, $(\mathbf{x}(k), d(k))$. Indeed, defining the set

$$\Psi_{k_1}^{k_2} \triangleq \bigcap_{k=k_1}^{k_2} \mathcal{H}(k), \tag{2.22}$$

where $k_1, k_2 \in \mathbb{N}$ and $\mathcal{H}(k)$ denotes the kth *constraint set* given by

$$\mathcal{H}(k) \triangleq \left\{\mathbf{w} \in \mathbb{R}^{N+1} : |d(k) - \mathbf{w}^T\mathbf{x}(k)| \leq \overline{\gamma}\right\}, \tag{2.23}$$

then Θ can be iteratively estimated via the *exact membership set* Ψ_0^k since $\lim_{k\longrightarrow\infty} \Psi_0^k = \Theta$.

Fig. 2.4 depicts the geometrical interpretation of the SMF criterion. The constraint sets correspond to regions between parallel hyperplanes in the parameter space, whereas the exact membership set is a convex polytope. The hypervolume of Ψ_0^k decreases for each k in which the pairs $(\mathbf{x}(k), d(k))$ bring some innovation. Clearly $\Theta \subset \Psi_0^k, \forall k$.

2.4 Conclusion

In this chapter we introduced the SMF concept. Our goal was to define some of the involved sets, especially the constraint set $\mathcal{H}(k)$. Other important sets are the feasibility set Θ, the intersection of the constraint sets from iteration k_1 to k_2 which is given by $\Psi_{k_1}^{k_2}$, and the exact membership set Ψ_0^k. These sets are extensively used

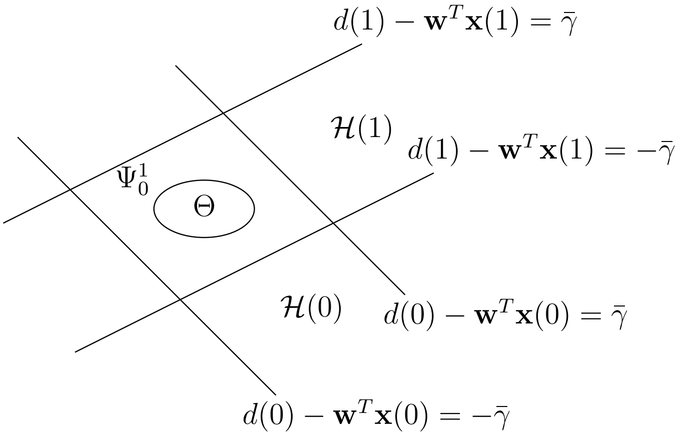

$d(1) - \mathbf{w}^T\mathbf{x}(1) = \bar{\gamma}$
$\mathcal{H}(1)$
$d(1) - \mathbf{w}^T\mathbf{x}(1) = -\bar{\gamma}$
Ψ_0^1
Θ
$\mathcal{H}(0)$
$d(0) - \mathbf{w}^T\mathbf{x}(0) = \bar{\gamma}$
$d(0) - \mathbf{w}^T\mathbf{x}(0) = -\bar{\gamma}$

Figure 2.4: *SMF geometrical interpretation in the parameter space:* $\Psi_0^1 = \mathcal{H}(0) \bigcap \mathcal{H}(1)$.

in the beginning of the next chapter.

In addition, the next two chapters present SMF-based algorithms and, as it will be shown, these algorithms are superior to their classical counterparts, having two very attractive features: *robustness against noise* and *reduced computational burden* [2].

Chapter 3

The Set-Membership Affine Projection Algorithm

The affine projection (AP) algorithm [19] is a widely used and studied adaptive filtering algorithm due to its fast convergence, when compared to the popular least-mean-square (LMS) algorithm, and its low computational complexity as compared to the recursive least-squares (RLS) algorithm [2, 19–22]. The fast convergence of the AP algorithm, especially for highly correlated input signals, originates from the *data reuse*, as explained in Chapter 2. In other words, the convergence speed of the AP algorithm can be increased by reusing more data. The price to be paid is an increase in both the computational complexity and steady-state mean-squared error (MSE) [2].

The set-membership (SM) algorithms [1, 11, 12, 23–29] rely on the concept of set-membership filtering (SMF), which allows the reduction of computational burden by updating the filter coefficients only in the cases where the error is greater than a prescribed threshold, i.e., the innovation in the observed data is checked before the data are used in the learning process. This SMF property, known as *data selection*, is responsible not only for the reduction of computational burden, thus *saving energy*, but also for the robustness against noise of the SM algorithms.

Therefore, the *set-membership affine projection* (SM-AP) algorithm is an interesting alternative to the AP algorithm because it combines the *data reuse*, which increases the convergence speed, with the *data selection*, which makes the algorithm less sensitive to noise and also reduces the computational burden. This combination results in a computationally efficient algorithm with low steady-state MSE and high convergence speed.

Another reason that makes the SM-AP algorithm interesting is the fact that it generalizes many algorithms. Indeed, the set-membership normalized LMS (SM-NLMS), the set-membership binormalized LMS (SM-BNLMS), and their non-SM counterparts (i.e., the NLMS, BNLMS, and AP algorithms) with step size $\mu = 1$ are all particular cases of the SM-AP algorithm. Hence, the results presented in this chapter can also be applied to the aforementioned algorithms.

Almost the entire chapter constitutes contributions of this thesis. The only exception is Section 3.2. The contributions of this chapter are listed below:

1. We connect the theory of the SM-AP algorithm with the SMF concept, i.e., we explain how a finite number of constraint sets $\mathcal{H}(k)$ shall be used in order to obtain an estimate of the exact membership set Ψ_0^k.

2. We describe the role played by the constraint vector (CV) in the updating process, which explains why a general choice of the constraint vector does not lead to accurate estimates.

3. Based on what the SM-AP algorithm should do, i.e., based on its original optimization problem, we propose a guideline for setting the CV.

4. We propose two new CVs: one presents very high convergence speed, whereas the other represents a trade-off between convergence speed and steady-state MSE.

5. By analyzing different CVs from both geometrical and analytical viewpoints, some properties of these CVs are revealed and then confirmed via simulation.

6. We present an analysis of the steady-state MSE of the SM-AP algorithm using one of the CVs, which results in a closed-form expression that matches the experimental MSE results and also agrees with the analysis of the standard AP algorithm.

7. We show, mathematically and via simulation, that the SM-AP algorithm can always achieve lower steady-state MSE than the AP algorithm with $\mu = 1$, provided the error-bound parameter $\overline{\gamma}$ is properly set.

The content of this chapter was published mostly in [13, 30] and also in [31, 32].

This chapter is organized as follows. Section 3.1 provides an interpretation of the SM-AP algorithm based on the sets involved in the SMF concept, which culminates in the original optimization problem corresponding to the SM-AP algorithm. Section 3.2 describes the SM-AP algorithm and presents one of its key components: the CV. In Section 3.3, we provide a comprehensive study of the CV, which encompasses: a guideline for setting the CV, proposition of two new CVs, properties of the updating process, and simulation results that corroborate our expectations. Section 3.4 presents an analysis of the steady-state MSE of the SM-AP algorithm employing a specific CV. Appendix A presents material that is related to Section 3.4. Indeed, it contains most of the mathematical derivations, the model used for some variables, as well as the assumptions required throughout the steady-state MSE analysis.

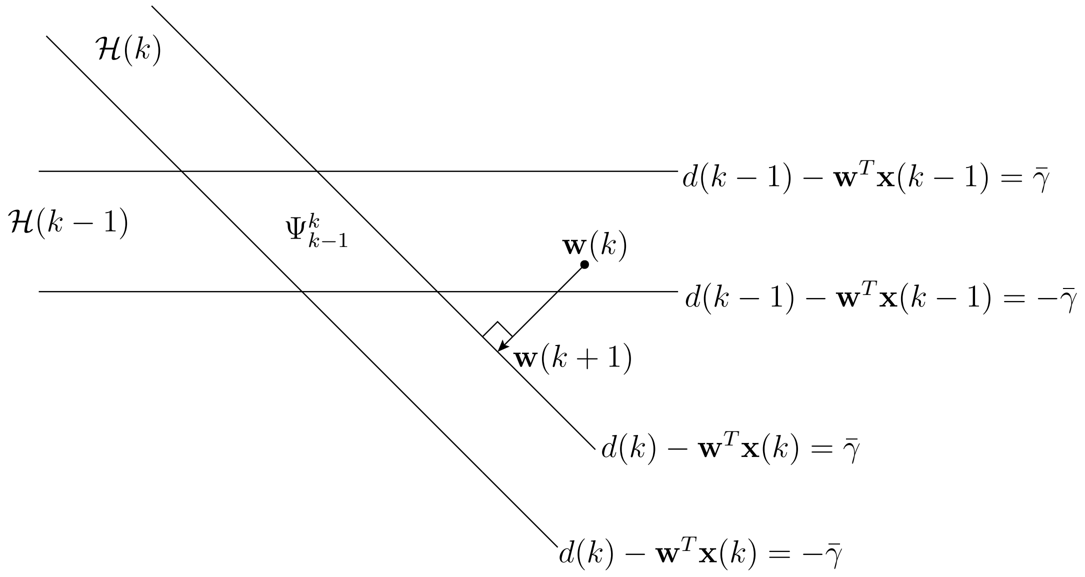

$\mathcal{H}(k)$
$d(k-1)-\mathbf{w}^T\mathbf{x}(k-1)=\bar{\gamma}$
$\mathcal{H}(k-1)$
Ψ_{k-1}^k
$\mathbf{w}(k)$
$d(k-1)-\mathbf{w}^T\mathbf{x}(k-1)=-\bar{\gamma}$
$\mathbf{w}(k+1)$
$d(k)-\mathbf{w}^T\mathbf{x}(k)=\bar{\gamma}$
$d(k)-\mathbf{w}^T\mathbf{x}(k)=-\bar{\gamma}$

Figure 3.1: *Geometrical interpretation of the updating scheme of the SM-NLMS algorithm in the parameter space. Observe that the updating process generated* $\mathbf{w}(k+1) \notin \mathcal{H}(k-1)$.

3.1 Relation with the SMF Concept

The different SMF-based online algorithms in the literature choose distinct forms of working with a finite number of constraint sets $\mathcal{H}(k)$ in order to approximate the exact membership set Ψ_0^k, which converges to the feasibility set Θ as $k \longrightarrow \infty$, as explained in Section 2.3. For instance, at a given iteration k, the SM-NLMS algorithm works only with the current constraint set $\mathcal{H}(k)$, whereas the SM-BNLMS algorithm considers $\Psi_{k-1}^k = \mathcal{H}(k) \bigcap \mathcal{H}(k-1)$.

The set-membership affine projection (SM-AP) algorithm generalizes both the SM-NLMS and SM-BNLMS algorithms. Indeed, the SM-AP algorithm works with the last $L+1$ constraint sets Ψ_{k-L}^k, where L is known as data reuse factor. This basically means that, at a given iteration k, the SM-AP algorithm finds a new parameter vector $\mathbf{w}(k+1)$ that belongs to Ψ_{k-L}^k.

As L increases, more previous data are reused at every iteration resulting in an increase of convergence speed at the cost of increasing the computational complexity. In addition, by progressing from $\mathbf{w}(k) \in \Psi_{k-L-1}^{k-1}$ to $\mathbf{w}(k+1) \in \Psi_{k-L}^k$, it may happen that $\mathbf{w}(k+1) \notin \Psi_{k-L-1}^{k-1}$, or equivalently $\mathbf{w}(k+1) \notin \mathcal{H}(k-L-1)$. Figure 3.1 depicts an example of a valid updating scheme for the SM-NLMS algorithm (i.e., $L=0$) in which the updating process generates $\mathbf{w}(k+1) \in \Psi_k^k = \mathcal{H}(k)$, but $\mathbf{w}(k+1) \notin \Psi_{k-1}^{k-1} = \mathcal{H}(k-1)$. Therefore, in order to mitigate this issue it is important to employ the *minimum disturbance principle* so that the updating of the parameter vector is performed to minimize $\|\mathbf{w}(k+1)-\mathbf{w}(k)\|^2$, where $\|\cdot\|$ stands for the 2-norm of its argument. Intuitively, by constraining $\mathbf{w}(k+1)$ to be close to $\mathbf{w}(k)$, there is a lower probability of $\mathbf{w}(k+1) \notin \Psi_{k-L-1}^{k-1}$ and, consequently, the probability of $\mathbf{w}(k+1) \notin \Psi_0^k$ decreases as well. In other words, the SM-AP algorithm alleviates the fact that it uses only $L+1$ constraint sets at every iteration by constraining the length of the perturbation applied to $\mathbf{w}(k)$ that generates $\mathbf{w}(k+1)$ to be small.

Therefore, the SM-AP algorithm aims at finding a $\mathbf{w}(k+1)$ that solves the following optimization problem:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ &\text{subject to } \mathbf{w}(k+1) \in \Psi_{k-L}^k. \end{aligned} \tag{3.1}$$

This means that if $\mathbf{w}(k) \in \Psi_{k-L}^k$, i.e., the data does not bring enough innovation, then no update is performed and $\mathbf{w}(k+1) = \mathbf{w}(k)$. On the other hand, an update is required when $\mathbf{w}(k) \notin \Psi_{k-L}^k$. Note that in order to check if $\mathbf{w}(k) \in \Psi_{k-L}^k$, it suffices to check if $\mathbf{w}(k) \in \mathcal{H}(k)$, because the iterative process already ensures that $\mathbf{w}(k) \in \Psi_{k-L-1}^{k-1} \subset \Psi_{k-L}^{k-1}$.

We call the optimization problem in (3.1) as the *original optimization problem* of the SM-AP algorithm because it is based on the principles of the SMF theory. We show in the next section that the SM-AP algorithm was derived/proposed as the solution to a different optimization problem.

3.2 The Algorithm

The SM-AP algorithm was proposed in [12] and encompasses many other algorithms such as the SM-NLMS algorithm [1], the SM-BNLMS algorithm [11], and the standard AP algorithm with step size $\mu = 1$. The SM-AP algorithm combines the *data selection* of the SMF with the *data reuse* of the AP algorithm. These two features make the SM-AP a powerful algorithm that converges fast, like an AP algorithm, but requires fewer computations since it does not update very often, especially after convergence [2].

3.2.1 Summary of the Main Variables

Assuming we have available the last $L+1$ data-pairs of input vectors and desired signals, we define, for a given iteration k, the main variables of the SM-AP algorithm:

$$\begin{array}{rclcccl}
\mathbf{x}(k) & = & [x(k) & x(k-1) & \dots & x(k-N)]^T & \text{(input vector)} \\
\mathbf{X}(k) & = & [\mathbf{x}(k) & \mathbf{x}(k-1) & \dots & \mathbf{x}(k-L)] & \text{(input matrix)} \\
\mathbf{w}(k) & = & [w_0(k) & w_1(k) & \dots & w_N(k)]^T & \text{(coefficient vector)} \\
\mathbf{d}(k) & = & [d(k) & d(k-1) & \dots & d(k-L)]^T & \text{(desired vector)} \\
\boldsymbol{\gamma}(k) & = & [\gamma_0(k) & \gamma_1(k) & \dots & \gamma_L(k)]^T & \text{(constraint vector)} \\
\mathbf{e}(k) & = & [e_0(k) & e_1(k) & \dots & e_L(k)]^T & (\textit{a priori} \text{ error vector}) \\
\boldsymbol{\varepsilon}(k) & = & [\varepsilon_0(k) & \varepsilon_1(k) & \dots & \varepsilon_L(k)]^T & (\textit{a posteriori} \text{ error vector}).
\end{array} \tag{3.2}$$

The input matrix $\mathbf{X}(k) \in \mathbb{R}^{(N+1)\times(L+1)}$ contains the last $L+1$ input vectors $\mathbf{x}(k-l) \in \mathbb{R}^{N+1}$ for $l = 0, \dots, L$. The adaptive filter coefficient vector is given by $\mathbf{w}(k) \in \mathbb{R}^{N+1}$, the desired output vector $\mathbf{d}(k) \in \mathbb{R}^{L+1}$ contains the last $L+1$ desired signals $d(k-l) \in \mathbb{R}$ for $l = 0, \dots, L$, the constraint vector (CV) is given by $\boldsymbol{\gamma}(k) \in \mathbb{R}^{L+1}$, and $\mathbf{e}(k), \boldsymbol{\varepsilon}(k) \in \mathbb{R}^{L+1}$ defined as

$$\mathbf{e}(k) \triangleq \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k) \tag{3.3}$$

$$\boldsymbol{\varepsilon}(k) \triangleq \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) \tag{3.4}$$

are the *a priori* and the *a posteriori* error vectors, respectively. Observe that the *a priori* error vector $\mathbf{e}(k)$ is sometimes called error vector.

The noise vector is $\mathbf{n}(k) = [n(k), \dots, n(k-L)]^T \in \mathbb{R}^{L+1}$. The parameter $\overline{\gamma} \in \mathbb{R}_+$ defines an upper bound for the entries of the CV, i.e., $|\gamma_l(k)| \leq \overline{\gamma}$ for $l = 0, \dots, L$, and is commonly chosen as $\overline{\gamma} = \sqrt{5\sigma_n^2}$ [1, 2, 11, 12], where σ_n^2 is the variance of the noise. Later on, in this chapter, we address the problem of choosing $\overline{\gamma}$, which depends on the choice of the CV $\boldsymbol{\gamma}(k)$.

3.2.2 The SM-AP Algorithm

The SM-AP algorithm is characterized by the following recursion [2, 12]:

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] & \text{if } |e_0(k)| > \overline{\gamma}, \\ \mathbf{w}(k) & \text{otherwise,} \end{cases} \tag{3.5}$$

where in order to keep a compact notation we define

$$\mathbf{R}(k) \triangleq \mathbf{X}^T(k)\mathbf{X}(k), \tag{3.6}$$

$$\mathbf{S}(k) \triangleq [\mathbf{R}(k) + \delta\mathbf{I}]^{-1} = \left[\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}\right]^{-1}, \tag{3.7}$$

where $\mathbf{R}(k), \mathbf{S}(k) \in \mathbb{R}^{(L+1)\times(L+1)}$ and the parameter δ is the regularization factor, a small constant used to avoid numerical instability due to the inversion of an ill-conditioned matrix. It is important to keep in mind that, in addition to being very close to 0, δ is also an artificial solution to a numerical problem.[1] Therefore, for the purpose of theoretical analysis we can consider $\delta = 0$. For instance, the SM-AP algorithm, given in (3.5), with $\delta = 0$ was derived in [12] as the solution to the following optimization problem:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k). \end{aligned} \tag{3.8}$$

Observe that the optimization problem in (3.8) is very similar to the original optimization problem given in (3.1), but they are not equivalent. This aspect is explored in Section 3.3.

3.2.3 Geometric Interpretation

The equality constraint in (3.8) defines, in the space of the coefficient vectors, $L+1$ hyperplanes. Therefore, the SM-AP algorithm maps $\mathbf{w}(k)$ to the $\mathbf{w}(k+1) \in \mathbb{R}^{N+1}$ that belongs to the intersection of these $L+1$ hyperplanes given by $\varepsilon_l(k) = \gamma_l(k)$, i.e.,

$$d(k-l) - \mathbf{x}^T(k-l)\mathbf{w}(k+1) = \gamma_l(k), \quad \text{for } l = 0, 1, \ldots, L, \tag{3.9}$$

and is as close as possible to $\mathbf{w}(k)$.

Figure 3.2 illustrates the updating process of the SM-AP algorithm for $L = 1$ considering the most general form of the constraint vector [2, 12], i.e., the only constraint applied to each of its element $\gamma_l(k)$ is $|\gamma_l(k)| \leq \overline{\gamma}$.

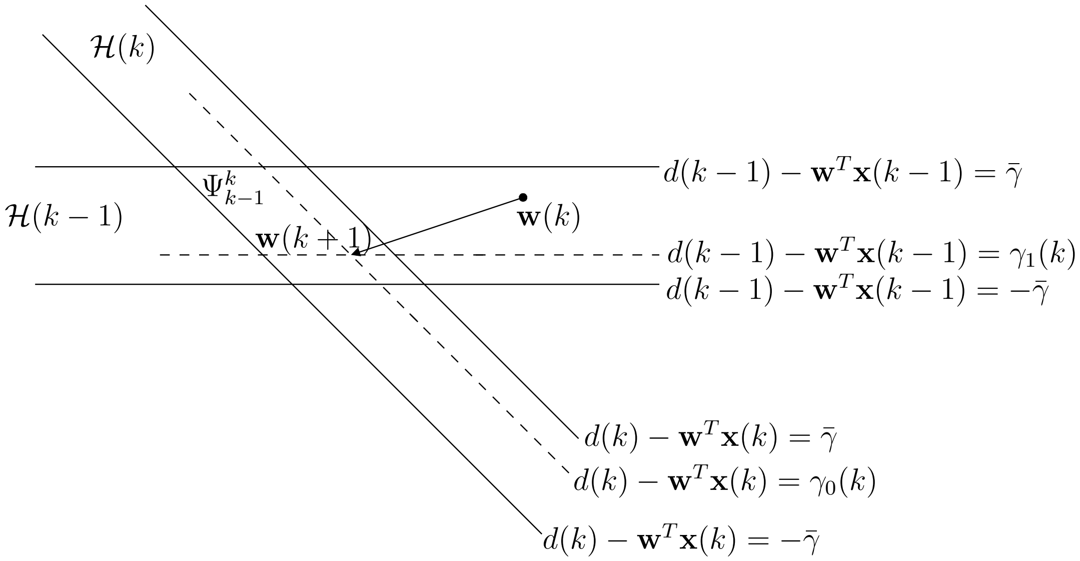
3.3 The Role of the Constraint Vector

Now, we have all the tools to fully understand the role of the CV $\boldsymbol{\gamma}(k)$ in the SM-AP algorithm.

3.3.1 The Optimization Problems

In the SM-AP algorithm, the entries of $\boldsymbol{\gamma}(k)$ are employed to define a subset of Ψ_{k-L}^k where $\mathbf{w}(k+1)$ will lie. Indeed, as observed in Figure 3.2, the equality

[1] Indeed, regularization corresponds to a process in which additional information/constraints are introduced in order to solve an ill-posed problem. The regularization factor δ used in (3.7) is related to the Tikhonov regularization.



$\mathcal{H}(k)$
$\mathcal{H}(k-1)$
Ψ_{k-1}^k
$\mathbf{w}(k+1)$
$\mathbf{w}(k)$
$d(k-1) - \mathbf{w}^T\mathbf{x}(k-1) = \bar{\gamma}$
$d(k-1) - \mathbf{w}^T\mathbf{x}(k-1) = \gamma_1(k)$
$d(k-1) - \mathbf{w}^T\mathbf{x}(k-1) = -\bar{\gamma}$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = \bar{\gamma}$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = \gamma_0(k)$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = -\bar{\gamma}$

Figure 3.2: *Geometrical interpretation of the updating scheme of the SM-AP algorithm in the parameter space for $L = 1$.*

constraints in the optimization problem given in (3.8) represent an intersection of $L + 1$ hyperplanes, which is guaranteed to be a subset of Ψ_{k-L}^k by making $|\gamma_l(k)| \leq \overline{\gamma}$ for $l \in \mathcal{L} \triangleq \{0, 1, \ldots, L\}$, and $\mathbf{w}(k+1)$ is the projection of $\mathbf{w}(k)$ onto such an intersection/subset.

However, in the original optimization problem, which is described in (3.1), it is clear that the the SM-AP algorithm should generate $\mathbf{w}(k+1)$ as a projection[2] of $\mathbf{w}(k)$ onto Ψ_{k-L}^k. This means that $\mathbf{w}(k+1)$ should be the closest point onto Ψ_{k-L}^k, and that is not what the SM-AP algorithm with an arbitrary/random choice of the CV generally does, as depicted in Figure 3.2.

In addition, if we consider the SM-NLMS and SM-BNLMS algorithms, which are particular cases of the SM-AP algorithm, then we observe that their updating rules satisfy the original optimization problem. Indeed, at a given iteration k, the SM-NLMS algorithm generates $\mathbf{w}(k+1)$ as a projection of $\mathbf{w}(k)$ onto $\Psi_k^k = \mathcal{H}(k)$, see [1] or [2]. For the SM-BNLMS algorithm, there are two commonly used forms to generate $\mathbf{w}(k+1)$: (i) as a projection of $\mathbf{w}(k)$ on the set $\Psi_{k-1}^k = \mathcal{H}(k) \bigcap \mathcal{H}(k-1)$ and (ii) as a projection of $\mathbf{w}(k)$ on the set Ψ_{k-1}^k constrained to keep the *a posteriori* error at iteration $k-1$ constant, see [11] or [2]. Regarding the SM-BNLMS algorithm, observe that its updating rule explained in (i) satisfies (3.1), but the other rule explained in (ii) does not.[3] However, for all possible forms of update performed by the SM-NLMS and SM-BNLMS, one must notice that they generate a new estimate $\mathbf{w}(k+1)$ that lies on the closest boundary of their corresponding sets.

[2] A projection of a point $\mathbf{z}$ on a set Z is any point $\mathbf{z}_p \in Z$ which is closest to $\mathbf{z}$ [17].

[3] In fact, (ii) was probably the motivation behind the simple choice CV (SC-CV), which is explained in the next subsection.

3.3.2 Choosing the CV

So far, we have seen that the SM-AP algorithm solves the optimization problem in (3.8). On the other hand, in Section 3.1 we explained the importance of solving the original optimization problem given in (3.1). Indeed, in that section we explained that small steps are important to increase the probability of $\mathbf{w}(k) \in \Psi_0^k$, which is the exact membership set, which in turn converges to the feasibility set Θ as k grows. Here, we explain how to set the CV in order to reduce the discrepancy between these two optimization problems. The term "reduce" was used because sometimes we may prefer to set the CV so that $\mathbf{w}(k+1)$ lies on the closest boundary of Ψ_{k-L}^k, i.e., we might be interested in alleviating the original optimization problem by replacing "the closest point" with "a point belonging to the closest boundary".[4] Observe that the solution to the original optimization problem is also a solution to this "alleviated problem", whose motivation is found in the already explained updating schemes of the SM-NLMS and SM-BNLMS algorithms. Indeed, by extending the observations valid for the SM-NLMS and SM-BNLMS algorithm, two algorithms that yield accurate estimates, we can determine a guideline for setting the CV.

Thus, in the SM-AP algorithm, we are interested in generating $\mathbf{w}(k+1)$ on the border of Ψ_{k-L}^k that is closest to $\mathbf{w}(k)$ whenever an update occurs. Such an updating scheme, in addition to being closer to the original optimization problem, also has the advantage of keeping numerical errors (more specifically, the noise enhancement effect[5]) under control [30].

In order to update $\mathbf{w}(k)$ towards the closest border of Ψ_{k-L}^k, one should notice that the CV must use some information about the error signal. In fact, just the sign of the error is necessary to discover which hyperplanes are closer to $\mathbf{w}(k)$, as can be verified in Figure 3.2 and also in Figure 3.3. Therefore, the main *guideline* for setting the CVs is: $\boldsymbol{\gamma}(k)$ must be a function that takes into account the sign of the components of the error vector $\mathbf{e}(k)$.

The first CV that was proposed following this guideline was the simple choice CV given in Definition 1.

Definition 1 (Simple choice — SC)**.** *The SC constraint vector (SC-CV) is defined as*

$$\gamma_l(k) \triangleq \begin{cases} \overline{\gamma}\,\mathrm{sign}\,[e_l(k)] & \textit{if } l = 0, \\ e_l(k) & \textit{for } l \in \mathcal{L} \setminus \{0\}. \end{cases} \tag{3.10}$$

[4] One possible reason for that is the complexity of defining the CV. Indeed, we will present some CVs with nice properties that demand low computational burden in their calculation.

[5] Noise enhancement occurs when the observed data is corrupted by some kind of noise. Since the step applied to $\mathbf{w}(k)$ to generate $\mathbf{w}(k+1)$ uses the observed data, if we update using large steps, then the noise is amplified accordingly.

The SC-CV was proposed in [12] and its MSE performance was studied in [33]. Similarly to the updating scheme (ii) of the SM-BNLMS algorithm, when an update occurs the SM-AP with SC-CV generates $\mathbf{w}(k+1)$ such that only the 0th error component is changed (i.e., it reduces $|e_0(k)|$ to $\overline{\gamma}$), keeping the other L error components unaltered.

After some experiments with the SM-AP algorithm employing the SC-CV, we observed that its convergence speed exhibited only a slight increase as L grows. For example, the convergence speed of the AP algorithm increases much faster with L than the convergence speed of the SM-AP algorithm with SC-CV does. This observation led us to propose another CV, which is given in Definition 2.

Definition 2 (Fixed modulus error-based — FMEB)**.** *The FMEB constraint vector (FMEB-CV) is defined as*

$$\gamma_l(k) \triangleq \overline{\gamma}\,\mathrm{sign}\left[e_l(k)\right], \text{ for } l \in \mathcal{L}. \tag{3.11}$$

The FMEB-CV was proposed in [32] and its steady-state MSE was analyzed in [13]. When an update occurs, the FMEB-CV yields $\mathbf{w}(k+1)$ such that all errors due to the last $L+1$ data pairs are modified so that their absolute values become $\overline{\gamma}$.

Figure 3.3 illustrates the updating process of the SM-AP algorithm employing the SC-CV and the FMEB-CV for $L = 1$. In the SC-CV, $\mathbf{w}(k)$ is mapped to a $\mathbf{w}(k+1)$ that lies on the closest border of $\mathcal{H}(k)$ constrained not to modify the previous error components (i.e., it follows a line that is parallel to the hyperplanes $d(k-1) - \mathbf{w}^T\mathbf{x}(k-1) = \pm\overline{\gamma}$), whereas the FMEB-CV maps $\mathbf{w}(k)$ to $\mathbf{w}(k+1)$ that lies on the intersection of the two closest hyperplanes. Figure 3.3 also reveals some important characteristics of the updating process. For example, the cyan arrow is usually longer than the blue arrow, especially during the early iterations, which suggests that the FMEB-CV converges faster than the SC-CV. On the other hand, as previously mentioned, since the SM-AP works with a finite number of constraint sets, by using small steps the SC-CV reduces the chance of generating $\mathbf{w}(k+1)$ disrespecting the constraints prior to iteration $k-L$. This indicates that, although the SC-CV is slower than the FMEB-CV, it is capable of achieving both lower MSE and lower probability of update.

Figure 3.3 also illustrates that there are infinitely many solutions between the FMEB-CV and SC-CV. In what follows, we show some analytical interpretations in order to obtain a novel CV that enjoys the good characteristics of both SC-CV and FMEB-CV.

Assuming that $|e_0(k)| > \overline{\gamma}$, then the updating formula of the SM-AP algorithm

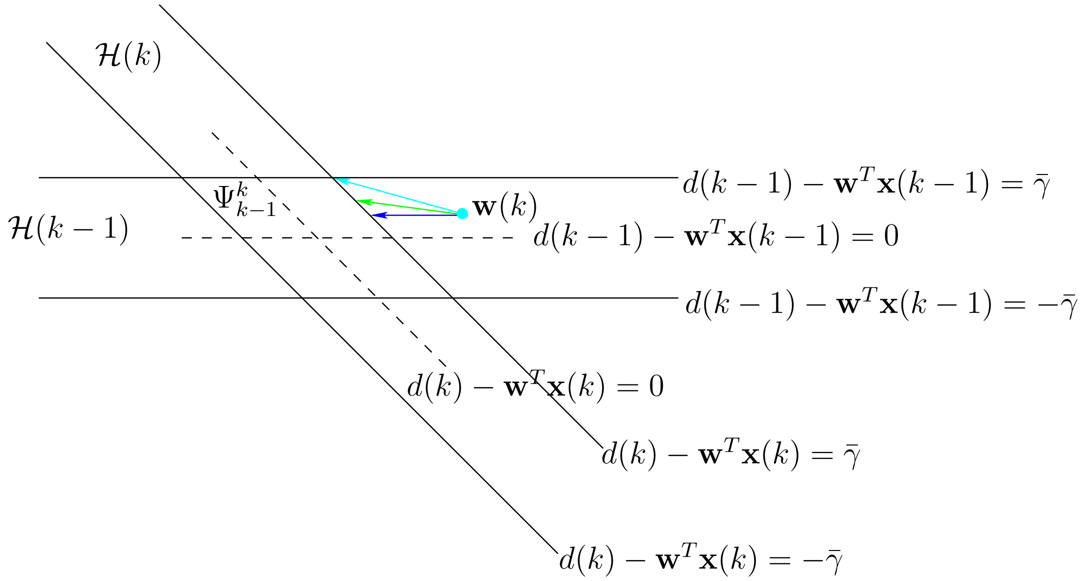

$\mathcal{H}(k)$
$\mathcal{H}(k-1)$
Ψ_{k-1}^{k}
$\mathbf{w}(k)$
$d(k-1)-\mathbf{w}^T\mathbf{x}(k-1)=\bar{\gamma}$
$d(k-1)-\mathbf{w}^T\mathbf{x}(k-1)=0$
$d(k-1)-\mathbf{w}^T\mathbf{x}(k-1)=-\bar{\gamma}$
$d(k)-\mathbf{w}^T\mathbf{x}(k)=0$
$d(k)-\mathbf{w}^T\mathbf{x}(k)=\bar{\gamma}$
$d(k)-\mathbf{w}^T\mathbf{x}(k)=-\bar{\gamma}$

Figure 3.3: *SM-AP updating scheme in the parameter space for $L=1$. The blue, cyan, and green arrows correspond to the step/perturbation $\mathbf{p}(k)$ applied to $\mathbf{w}(k)$ during the update considering the SC-CV, FMEB-CV, and ED-CV, respectively. Thus, the head of these arrows depict $\mathbf{w}(k+1)$.*

given in (3.5) can be rewritten as

$$\mathbf{w}(k+1)=\mathbf{w}(k)+\mathbf{X}(k)\mathbf{S}(k)[\mathbf{e}(k)-\boldsymbol{\gamma}(k)]. \tag{3.12}$$

Now, by using the particular forms of the aforementioned CVs, one can write

$$\mathbf{w}(k+1)=\mathbf{w}(k)+\underbrace{\mathbf{X}(k)\mathbf{S}(k)\mathbf{D}(k)\mathbf{e}(k)}_{\triangleq\mathbf{p}(k)}, \tag{3.13}$$

where $\mathbf{D}(k)\in\mathbb{R}^{(L+1)\times(L+1)}$ is a diagonal matrix containing weights applied to the columns of matrix $\mathbf{X}(k)\mathbf{S}(k)$, and $\mathbf{p}(k)\in\mathbb{R}^{N+1}$ is the step/perturbation added to $\mathbf{w}(k)$ to implement the update. The analytic definitions of $\mathbf{D}(k)$ and $\mathbf{p}(k)$ for the SC-CV and FMEB-CV are, respectively,

$$\begin{aligned}\mathbf{D}_{\rm SC}(k)&=\text{diag}\left\{\left(1-\frac{\bar{\gamma}}{|e_0(k)|},0,\ldots,0\right)\right\},\\ \mathbf{p}_{\rm SC}(k)&=\left(1-\frac{\bar{\gamma}}{|e_0(k)|}\right)e_0(k)\left[\mathbf{X}(k)\mathbf{S}(k)\right]_0, \end{aligned}\tag{3.14}$$

$$\begin{aligned}\mathbf{D}_{\rm FMEB}(k)&=\text{diag}\left\{\left(1-\frac{\bar{\gamma}}{|e_0(k)|},1-\frac{\bar{\gamma}}{|e_1(k)|},\ldots,1-\frac{\bar{\gamma}}{|e_L(k)|}\right)\right\},\\ \mathbf{p}_{\rm FMEB}(k)&=\sum_{l\in\mathcal{L}}\left(1-\frac{\bar{\gamma}}{|e_l(k)|}\right)e_l(k)\left[\mathbf{X}(k)\mathbf{S}(k)\right]_l, \end{aligned}\tag{3.15}$$

where $[\mathbf{X}(k)\mathbf{S}(k)]_l$ denotes the lth column of matrix $\mathbf{X}(k)\mathbf{S}(k)$ and diag$\{\mathbf{z}\}$ is a diagonal matrix with $\mathbf{z}$ on its main diagonal.

According to (3.13), the SM-AP algorithm with SC-CV or FMEB-CV can be thought as a standard AP algorithm featuring *data selection* through the innovation check that decides whether an update is necessary or not, and employing a *variable step-size matrix* $\mathbf{D}(k)$, whose diagonal entries represent the step-sizes applied to each column of $\mathbf{X}(k)\mathbf{S}(k)$. Observe that, for the SC-CV, only the first column of $\mathbf{X}(k)\mathbf{S}(k)$ is employed in the update, whereas the FMEB-CV yields a perturbation vector that is a linear combination of all columns of $\mathbf{X}(k)\mathbf{S}(k)$, thus featuring more degrees of freedom. Therefore, it is expected that the SM-AP using the FMEB-CV has higher convergence speed.

Both the SC-CV and the FMEB-CV could be thought as two extreme cases regarding the level of importance given to past data pairs. Indeed, the SC-CV considers only the current error $e_0(k)$ to define the perturbation vector employed in its related update equation. On the other hand, the FMEB-CV gives similar weights to all the last $L+1$ errors $e_l(k)$ to define its corresponding perturbation vector. However, since $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{p}(k)$ and $\mathbf{w}(k)$ has already taken into account past error signals, then giving similar weights to all error components when defining $\mathbf{p}(k)$ corresponds to giving more importance to previous error signals (i.e., $e_l(k)$ with l relatively large) in the definition of $\mathbf{w}(k+1)$, since those signals were used in $\mathbf{w}(k)$. Therefore, it makes sense to look for a solution that gives less importance to past error signals in the definition of $\mathbf{p}(k)$. A simple way of doing that is inspired by the RLS algorithm, which applies an exponential decaying to the error signal. We propose the novel CV of Definition 3 [30, 31], which is represented in Figure 3.3 as the green arrow.

Definition 3 (Exponential decay — ED)**.** *The ED constraint vector (ED-CV) requires that $\overline{\gamma} \leq 1$ and is given by*

$$\gamma_l(k) \triangleq \overline{\gamma}^{\,l+1} \operatorname{sign}\left[e_l(k)\right], \text{ for } l \in \mathcal{L}. \tag{3.16}$$

By utilizing the constraint vector definition above in (3.13), one has

$$\begin{aligned}
\mathbf{D}_{\mathrm{ED}}(k) &= \operatorname{diag}\left\{\left(1 - \frac{\overline{\gamma}}{|e_0(k)|}, 1 - \frac{\overline{\gamma}^2}{|e_1(k)|}, \ldots, 1 - \frac{\overline{\gamma}^{L+1}}{|e_L(k)|}\right)\right\}, \\
\mathbf{p}_{\mathrm{ED}}(k) &= \sum_{l \in \mathcal{L}} \left(1 - \frac{\overline{\gamma}^{\,l+1}}{|e_l(k)|}\right) e_l(k) \left[\mathbf{X}(k)\mathbf{S}(k)\right]_l .
\end{aligned} \tag{3.17}$$

3.3.3 Simulation: Comparing the CVs

In this subsection, some aspects of the SM-AP algorithm using different choices for the CV are studied. These aspects are:

1. Steady-state MSE level in stationary and nonstationary environments.

2. Convergence speed.

3. Influence of $\overline{\gamma}$ in the steady-state MSE.

The CVs considered are the SC-CV, FMEB-CV, ED-CV (see Definitions 1, 2, and 3), and the general-choice CV (GC-CV), which is defined as $\gamma_l(k) \triangleq \overline{\gamma}$ for $l \in \mathcal{L}$, i.e., it satisfies $|\gamma_l(k)| \leq \overline{\gamma}$ for every l, but it does not use any information regarding the error vector (i.e., it does not follow the proposed guideline for setting the CV).[6]

Scenario

We consider the problem of identifying an unknown system whose impulse response is $\mathbf{h}(k) = \mathbf{w}_o$ for all k, where $\mathbf{w}_o \triangleq [0.1 \;\; 0.3 \;\; 0 \;\; -0.2 \;\; -0.4 \;\; -0.7 \;\; -0.4 \;\; -0.2]^T$. When evaluating the nonstationary behavior, which happens only in Figure 3.5, the impulse response of the unknown system is given by $\mathbf{h}(k+1) = \lambda_{\mathbf{h}}\mathbf{h}(k) + \mathbf{n}_{\mathbf{h}}(k)$, where $\mathbf{h}(0) = \mathbf{w}_o$, $\lambda_{\mathbf{h}}$=0.99, and $\mathbf{n}_{\mathbf{h}}(k)$ is white and Gaussian with variance 0.0015.

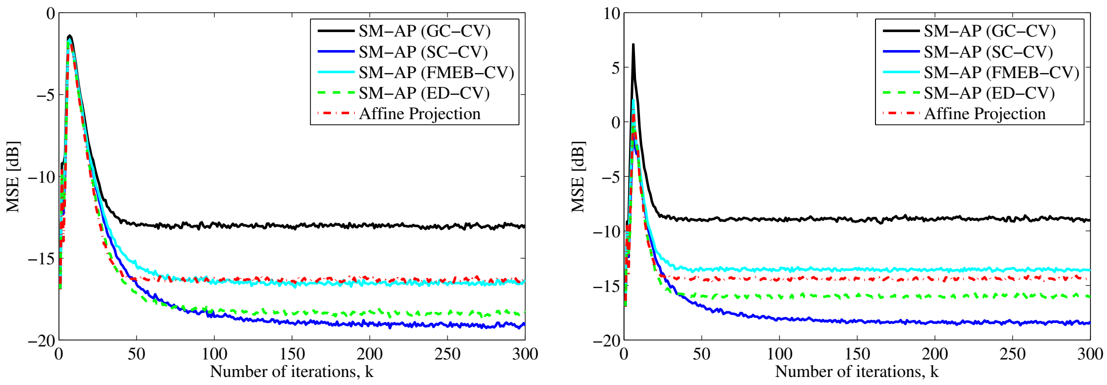
The input signal is drawn from a standard normal distribution and the noise variance is $\sigma_n^2 = 10^{-2}$. Most of the results were obtained by repeating the experiment 5×10^3 times except for the results in Figure 3.7, in which we took an average of the last 10^4 samples from each of the 100 experiments, and then averaged over the experiments, as done in [13]. In addition, the adaptive filter order is $N = 7$, which is the same order of the unknown system, and is initialized with $\mathbf{w}(0) = \mathbf{0}$. The regularization factor is $\delta = 10^{-12}$.

Results

Figures 3.4 to 3.6 depict the MSE learning curves for the SM-AP algorithm with different CVs. The standard AP algorithm with $\mu = 1$ is used as benchmark. Figure 3.7 depicts steady-state excess MSE (EMSE) as a function of $\overline{\gamma}$.

In Figures 3.4 and 3.5, the steady-state MSE in stationary and nonstationary environments is evaluated, respectively. In order to allow a fair comparison, the algorithms were set so that they have similar convergence speeds in the early iterations. In addition, the parameter $\overline{\gamma}$ is chosen as $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$ with $\tau = 3$, which is a recommended value for both the SC-CV [33] and FMEB-CV [13] in order to achieve a balance between low steady-state MSE and low probability of update. Figures 3.4(a) and 3.5(a) consider $L = 1$ (binormalized version of the algorithms), whereas $L = 4$ in Figures 3.4(b) and 3.5(b).

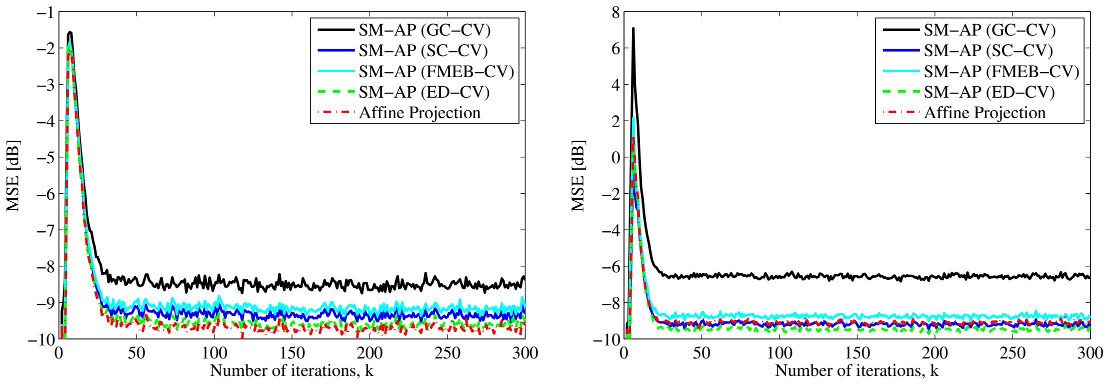
[6] We have also tested $\gamma_l(k)$ as a random number drawn from a uniform distribution in the range of $[0, \overline{\gamma}]$ and obtained similar results.



(a) $L = 1$

(b) $L = 4$

Figure 3.4: *MSE learning curves considering $\tau = 3$ in $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$. Algorithms were set so that they have similar transient responses, i.e., SM algorithms use the same $\overline{\gamma}$, whereas $\mu = 1$ for the AP algorithm.*



(a) $L = 1$

(b) $L = 4$

Figure 3.5: *MSE learning curves in nonstationary environment using $\tau = 3$ in $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$. Algorithms were set so that they have similar transient responses, i.e., SM algorithms use the same $\overline{\gamma}$, whereas $\mu = 1$ for the AP algorithm.*

Comparing Figures 3.4(a) and 3.4(b), despite the higher convergence speed and also a bit higher steady-state MSE exhibited by the algorithms using $L = 4$, these figures follow the same pattern. Indeed, the GC-CV led to the worst steady-state MSE, while the FMEB-CV led to a steady-state MSE level similar to the one achieved by the AP algorithm. The SC-CV led to the lowest steady-state MSE level, while the ED-CV reached an intermediate (i.e., between the SC-CV and the FMEB-CV) MSE level. In addition, using Figure 3.4(a) as example, the SM-AP employing the GC-CV, SC-CV, FMEB-CV, and ED-CV updated only about $45\%, 20\%, 30\%$, and 21% of the iterations, respectively. Figure 3.4 illustrates the importance of a proper choice for the CV and also shows that the SM-AP can achieve similar to better results compared to the AP algorithm besides saving computational power.

In Figure 3.5 the nonstationary behavior of the SM-AP algorithm is assessed. Once again, the GC-CV led to the worst results, while the results obtained using

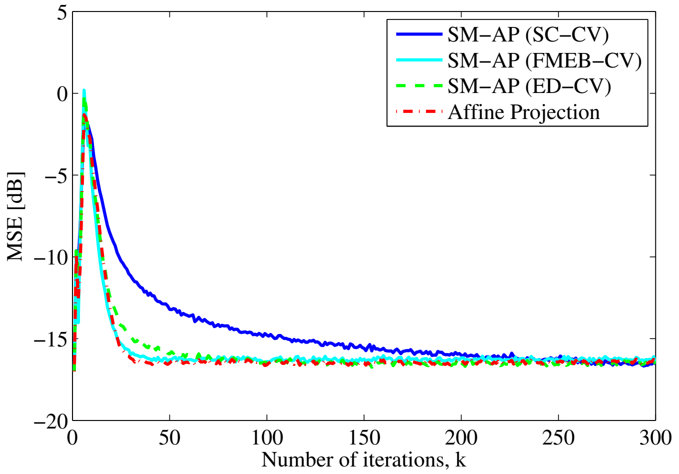


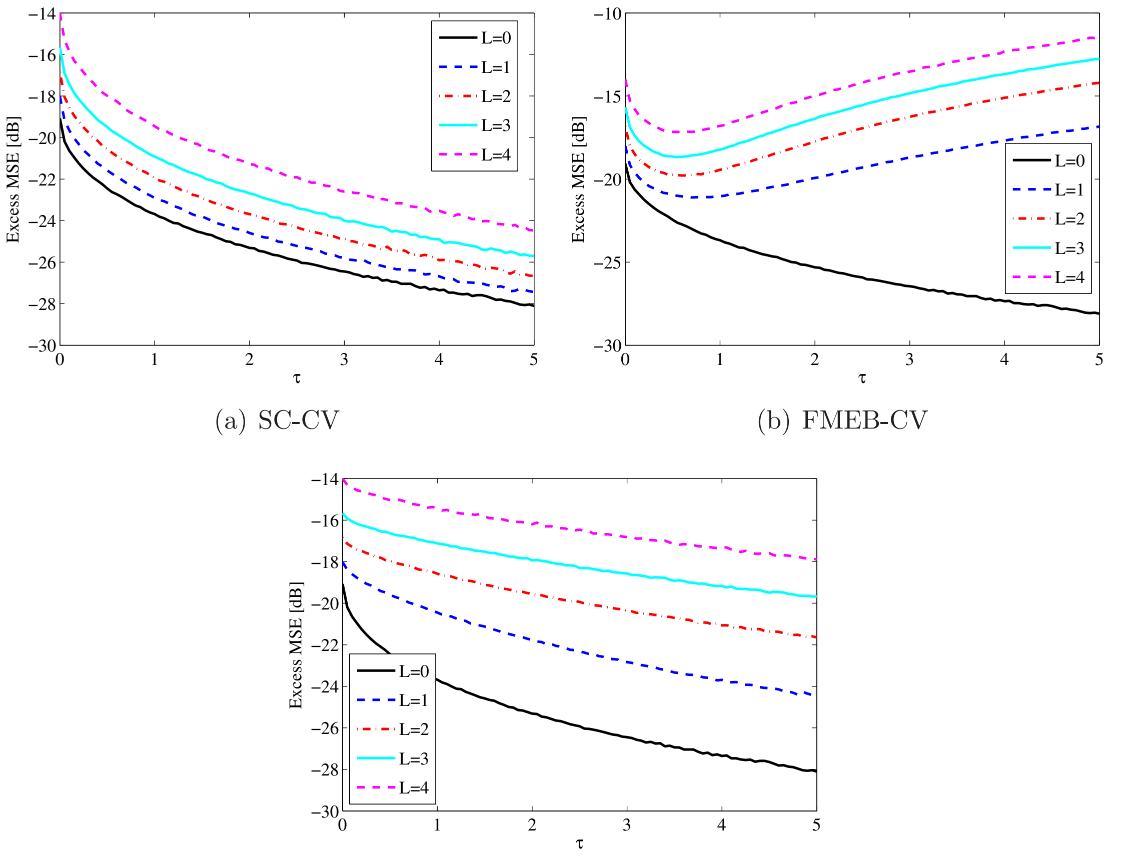
Figure 3.6: *MSE learning curves considering $L = 4$. Algorithms were set so that they reach a similar steady-state MSE, i.e., different values of $\overline{\gamma}$ were used.*

the SC-CV, FMEB-CV, and ED-CV are similar to the one of the AP algorithm. In fact, we observed that the ED-CV was a bit better than the others since it achieved the lowest steady-state MSE and also had the lowest probability of update. In Figure 3.5(b), e.g., the SM-AP employing the GC-CV, SC-CV, FMEB-CV, and ED-CV updated about $70\%, 62\%, 63\%$, and 60% of the iterations, respectively.

In Figure 3.6 the convergence speed is studied. Hence, the algorithms were set so that they reach a similar steady-state MSE level. Figure 3.6 shows that, for an arbitrarily given steady-state MSE level, the SM-AP with SC-CV was the slowest algorithm but also had the lowest probability of update. The highest convergence speeds were achieved by the SM-AP with FMEB-CV and the AP algorithms, but the former has the advantage of not updating at every iteration. Interestingly, the convergence speed provided by the ED-CV was very close to the one of the FMEB-CV, but the ED-CV requires fewer updates. In this particular setup, the SM-AP employing the SC-CV, FMEB-CV, and ED-CV updated about $8\%, 65\%$, and 11% of the iterations, respectively.[7]

Figure 3.7 depicts the steady-state excess MSE (EMSE) as a function of τ, a parameter that determines $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, for different values of L. Observe that when $\tau = 0$, i.e., $\overline{\gamma} = 0$, the SM-AP employing the FMEB-CV and the ED-CV become the *standard AP algorithm* (i.e., AP with step size equal to 1). In Proposition 2 introduced in the next section, we show that by judiciously choosing τ the SM-AP with FMEB-CV can always achieve a steady-state MSE lower than the one of the standard AP algorithm in stationary environments, a fact that is corroborated by Figure 3.7(b) (because the EMSE for τ close to 0 is lower than the EMSE for $\tau = 0$).

[7] In Figure 3.6, we fixed the steady-state MSE level obtained by the algorithms. However, each of the SM-AP algorithms could be made faster at the cost of increasing their percentage of updates.



(a) SC-CV

(b) FMEB-CV

(c) ED-CV

Figure 3.7: *Steady-state Excess MSE vs. τ, where $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$. For the SM-AP employing the FMEB-CV or ED-CV, $\tau = 0$ corresponds to the standard AP algorithm.*

Figure 3.7(c) indicates that the same result should be valid for the SM-AP with ED-CV (because it exhibits many values of τ that yield lower EMSE, as compared to the EMSE for $\tau = 0$), but the ED-CV has an advantage: a wider range of values of τ leading to low EMSE, which means that it is easier to set the SM-AP with ED-CV. Finally, Figures 3.7(a) and 3.7(c) show that the SM-AP algorithm using the SC-CV and the ED-CV, respectively, can use high values of τ and still achieve low steady-state MSE. Recall that higher τ implies lower probability of update, i.e., more energy/computational saving.

3.4 Steady-State MSE Analysis

Although the SM-AP algorithm has been successfully employed in many applications, see [29] and references in Chapter 6 of [2], analytical results for this algorithm are so far lacking in the open literature. Simpler SM algorithms were analyzed in [11, 33, 34]. In this section, the SM-AP algorithm with FMEB-CV is analyzed leading to useful closed-form expressions for its EMSE and misadjustment. The analysis result encompasses a number of algorithms such as the SM-NLMS and SM-BNLMS, and can also be seen as a generalization of the steady-state analysis of the AP algorithm.

The proposed analysis has the attractive feature of not assuming a specific model for the input signal and relies on energy conservation concepts previously employed to derive analytical results related to a number of adaptive filtering algorithms [35, 36]. These analytical results explain some observed experimental results and provide tools to properly choose the algorithm parameters for a given application. Particularly important is the role of the probability of coefficient update controlled by the deterministic threshold parameter inherent to the SM algorithms.

3.4.1 Preliminaries

In this section we use the energy conservation method to derive expressions for two common performance measures, namely the excess mean-squared error (EMSE) and the misadjustment (M), in order to describe the steady-state mean-squared performance of SM-AP algorithms in stationary environments. Assumptions required throughout the analysis are labeled as *As-i*, $i \in \mathbb{N}$, whereas the label *St-i* denotes statements/facts. Both assumptions and statements are discussed in Section A.6 of Appendix A.

We start by considering the signal model described in Definition 4.

Definition 4. *The sequences/random processes* $\{d(k)\}, \{\mathbf{x}(k)\}, \{n(k)\}$ *satisfy the following conditions:*
(a) $\exists \mathbf{w}_o \in \mathbb{R}^{N+1} : d(k) = \mathbf{w}_o^T \mathbf{x}(k) + n(k)$*;*
(b) The random variables (RVs) $d(k), \mathbf{x}(k)$ *have zero mean,* $\forall k$*;*
(c) $n(k)$ *is a zero-mean white Gaussian noise with variance* $\sigma_n^2 = \mathrm{E}[n^2(k)]$*;*
(d) $n(k_1)$ *is independent of* $\mathbf{x}(k_2)$, $\forall k_1, k_2$*;*
(e) The initial condition $\mathbf{w}(0)$ *is independent of the RVs* $d(k), \mathbf{x}(k), n(k)$*.*

Result 1. *As a consequence of Definition 4, the noise* $n(k)$ *is independent of the noiseless a priori error signal* $\tilde{e}(k)$*, which is defined as* $\tilde{e}(k) = (\mathbf{w}_o - \mathbf{w}(k))^T \mathbf{x}(k)$*.*

Considering Definition 4, we can write the *a priori* error signal as

$$\begin{aligned} e(k) = d(k) - y(k) &= \mathbf{w}_o^T \mathbf{x}(k) + n(k) - \mathbf{w}^T(k)\mathbf{x}(k) \\ &= n(k) + \tilde{e}(k). \end{aligned} \tag{3.18}$$

Squaring both sides of the equation above, then taking the expected value, and using

Definition 4 and Result 1 we have

$$\mathrm{E}[e^2(k)] = \mathrm{E}[n^2(k)] + \mathrm{E}[\tilde{e}^2(k)] = \sigma_n^2 + \mathrm{E}[\tilde{e}^2(k)]. \tag{3.19}$$

Assuming the algorithm converges, we can study its steady-state MSE performance by analyzing $\lim_{k\to\infty} \mathrm{E}[e^2(k)] = \sigma_n^2 + \mathrm{EMSE}$ where

$$\mathrm{EMSE} = \lim_{k\to\infty} \mathrm{E}[\tilde{e}^2(k)] \tag{3.20}$$

and the misadjustment is given by

$$\mathrm{M} = \frac{\mathrm{EMSE}}{\sigma_n^2}. \tag{3.21}$$

Now that we have introduced the quantities we aim to calculate, we can start the analysis of the SM-AP algorithm by employing a unified framework based on energy conservation arguments [20, 35].

Here we will consider a slight generalization of the SM-AP algorithm which adopts a step size factor μ.[8] We will refer to this algorithm as GenSM-AP algorithm to distinguish it from the SM-AP algorithm introduced in [12].

The updating equation of the GenSM-AP algorithm is given by

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] & \text{if } |e_0(k)| > \overline{\gamma}, \\ \mathbf{w}(k) & \text{otherwise.} \end{cases} \tag{3.22}$$

The nonlinearity presented in (3.22) due to the innovation check (also called information evaluation) step [11] turns the analysis more difficult. Therefore we will use a simple model, introduced in Section IV of [11], to overcome this problem by representing (3.22) as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + P_{\mathrm{up}}(k)\mu\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right], \tag{3.23}$$

where $P_{\mathrm{up}}(k) \in [0,1]$ is a function that represents the probability of updating the filter coefficients at a given iteration k, i.e., $P_{\mathrm{up}}(k) \triangleq P\left[|e_0(k)| > \overline{\gamma}\right]$. Note that Eq. (3.23) is still much more complex than the AP updating rule, since $\boldsymbol{\gamma}(k)$ is any RV satisfying $|\gamma_l(k)| \leq \overline{\gamma}$.

3.4.2 The Analysis Model

The goal of this section is to determine under which conditions the proposed analysis model is accurate.

[8] The motivation is to highlight the similarities between the SM-AP and AP algorithms.

Let $\mathbf{p}(k) \in \mathbb{R}^{N+1}$ be a RV representing the update applied to $\mathbf{w}(k)$ when $|e_0(k)| > \overline{\gamma}$, i.e,

$$\mathbf{p}(k) = \mu \mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]. \tag{3.24}$$

We can write (3.22) as

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{p}(k) & \text{if } |e_0(k)| > \overline{\gamma}, \\ \mathbf{w}(k) & \text{otherwise.} \end{cases} \tag{3.25}$$

Now, let $\bar{\mathbf{p}}(k)$ be a random vector defined as

$$\bar{\mathbf{p}}(k) = \begin{cases} \mathbf{p}(k) & \text{if } |e_0(k)| > \overline{\gamma}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{3.26}$$

With this definition, (3.25) can be rewritten as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \bar{\mathbf{p}}(k). \tag{3.27}$$

Then, by applying the expected value operator to (3.27), we obtain

$$\mathrm{E}[\mathbf{w}(k+1)] = \mathrm{E}[\mathbf{w}(k)] + \mathrm{E}[\bar{\mathbf{p}}(k)]. \tag{3.28}$$

Computing $\mathrm{E}[\bar{\mathbf{p}}(k)]$ we get

$$\begin{aligned} \mathrm{E}\left[\bar{\mathbf{p}}(k)\right] &= \mathrm{E}\Big[\bar{\mathbf{p}}(k)\Big|\ \{|e_0(k)| \leq \overline{\gamma}\}\Big](1 - P_{\mathrm{up}}(k)) + \mathrm{E}\Big[\bar{\mathbf{p}}(k)\Big|\ \{|e_0(k)| > \overline{\gamma}\}\Big]P_{\mathrm{up}}(k) \\ &= \mathbf{0} + \mathrm{E}\Big[\mathbf{p}(k)\Big|\ \{|e_0(k)| > \overline{\gamma}\}\Big]P_{\mathrm{up}}(k), \end{aligned} \tag{3.29}$$

where $P_{\mathrm{up}}(k) = P\left[|e_0(k)| > \overline{\gamma}\right]$. Assuming that the RV $\mathbf{p}(k)$ is independent of the event $\{|e_0(k)| > \overline{\gamma}\}$, see assumption *As-1*, we can write

$$\mathrm{E}\left[\bar{\mathbf{p}}(k)\right] = \mathrm{E}[\mathbf{p}(k)]P_{\mathrm{up}}(k). \tag{3.30}$$

As a result, it is straightforward to verify that the expected value of (3.22), see (3.28) and (3.30), leads to the same result as the expected value of (3.23). Consequently, provided $\mathbf{p}(k)$ and $\{|e_0(k)| > \overline{\gamma}\}$ are independent, (3.23) approximates (3.22) on average. Since we are interested in the average behavior of the algorithm, our main concern is to maintain the averages correct (in the sense of expected values).

3.4.3 Energy Conservation Approach

Subtracting $\mathbf{w}_o$, the optimal filter coefficients, from both sides of (3.23) we have

$$\Delta\mathbf{w}(k+1) = \Delta\mathbf{w}(k) + P_{\rm up}(k)\mu\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right], \tag{3.31}$$

where $\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o$ is the coefficient error vector.

Premultiplying Eq. (3.31) by $\mathbf{X}^T(k)$ and using (3.6) we have

$$-\tilde{\boldsymbol{\varepsilon}}(k) = -\tilde{\mathbf{e}}(k) + P_{\rm up}(k)\mu\mathbf{R}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right], \tag{3.32}$$

where

$$\tilde{\boldsymbol{\varepsilon}}(k) = -\mathbf{X}^T(k)\Delta\mathbf{w}(k+1) = \boldsymbol{\varepsilon}(k) - \mathbf{n}(k) \tag{3.33}$$

$$\tilde{\mathbf{e}}(k) = -\mathbf{X}^T(k)\Delta\mathbf{w}(k) = \mathbf{e}(k) - \mathbf{n}(k), \tag{3.34}$$

i.e., $\tilde{\boldsymbol{\varepsilon}}(k) = [\tilde{\varepsilon}_0(k) \;\; \tilde{\varepsilon}_1(k) \;\; \ldots \;\; \tilde{\varepsilon}_L(k)]^T$, $\tilde{\mathbf{e}}(k) = [\tilde{e}_0(k) \;\; \tilde{e}_1(k) \;\; \ldots \;\; \tilde{e}_L(k)]^T \in \mathbb{R}^{L+1}$ are the noiseless *a posteriori* error vector and the noiseless *a priori* error vector, respectively. Note that $\tilde{e}_0(k) = \tilde{e}(k)$, and the subscript 0 is used to emphasize that $\tilde{e}_0(k)$ is the first component of vector $\tilde{\mathbf{e}}(k)$.

Assuming $\mathbf{X}(k)$ has full column rank, see statement *St-1*, then $\mathbf{R}(k)$ is invertible and we can write Eq. (3.32) as

$$\mathbf{R}^{-1}(k)\left[\tilde{\mathbf{e}}(k) - \tilde{\boldsymbol{\varepsilon}}(k)\right] = P_{\rm up}(k)\mu\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]. \tag{3.35}$$

Using Eq. (3.35) in Eq. (3.31) follows

$$\Delta\mathbf{w}(k+1) - \mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k) = \Delta\mathbf{w}(k) - \mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k). \tag{3.36}$$

Proposition 1. *By evaluating the energies at both sides of Eq.* (3.36) *one can prove the following relation*

$$\|\Delta\mathbf{w}(k+1)\|^2 + \left[\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)\right] = \|\Delta\mathbf{w}(k)\|^2 + \left[\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)\right], \tag{3.37}$$

which involves the energies of the coefficient-error (also called weight-error) vectors, and the a priori and a posteriori error vectors.

Proposition 1 is an energy conservation relation for GenSM-AP algorithms. Its proof is left to Section A.1 of Appendix A.

Applying the expected value operator to (3.37), assuming the algorithm is properly set, and considering a sufficiently large k, we can assume the algorithm converges. In this case, $\mathrm{E}\left[\|\Delta\mathbf{w}(k+1)\|^2\right] = \mathrm{E}\left[\|\Delta\mathbf{w}(k)\|^2\right]$, so that

$$\mathrm{E}\left[\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)\right] = \mathrm{E}\left[\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)\right] \tag{3.38}$$

holds in the steady-state.

According to the mathematical derivations given in Section A.2 of Appendix A, the following relation involving correlation matrices is valid

$$\begin{aligned}
&(2-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&+2(1-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&-P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&+P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&-(2-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\boldsymbol{\gamma}(k)\tilde{\mathbf{e}}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&-2(1-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\\
&=P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}-2\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{n}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}.
\end{aligned} \tag{3.39}$$

In order to derive closed-form expressions for the EMSE of the GenSM-AP algorithm, in the next subsection we will assume that the correlation matrices presented in the equation above are diagonally dominant, assumption *As-3*. This assumption is important to maintain the mathematical tractability, as it will become clear in the next subsection, and it also allows the approximations $\mathrm{E}\left[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)\right] \approx \mathrm{E}\left[\boldsymbol{\gamma}(k)\tilde{\mathbf{e}}^T(k)\right]$ and $\mathrm{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right] \approx \mathrm{E}\left[\mathbf{n}(k)\boldsymbol{\gamma}^T(k)\right]$ so that (3.39) is simplified to

$$\begin{aligned}
(2-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&\\
+2(1-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&\\
-P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&\\
-2(1-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&\\
+2P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&\\
=P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}&.
\end{aligned} \tag{3.40}$$

The energy relation discussed so far is valid for the GenSM-AP algorithm with any possible choice of $\boldsymbol{\gamma}(k)$. In the next subsections we will address some specific choices for $\boldsymbol{\gamma}(k)$.

3.4.4 EMSE for the SM-AP Algorithm with FMEB-CV

In this subsection, we use $\boldsymbol{\gamma}(k)$ as the FMEB-CV, see Definition 2. The next task is to derive expressions for the correlation matrices in Eq. (3.40) in order to compute the EMSE. Due to the definition of the FMEB-CV, note that these correlations involve a nonlinear function of the error. Therefore, Result 2, in the following, plays a very important role in the analysis since it enables us to eliminate this nonlinearity.

Result 2 (Price's Theorem)**.** *Consider two RVs a and $\mathbf{b}$. If we assume that these variables are jointly Gaussian, the following approximation holds [37, 38]*

$$\mathrm{E}\left[\mathrm{sign}[a]\mathbf{b}\right] \approx \sqrt{\frac{2}{\pi\sigma_a^2}}\mathrm{E}\left[\mathbf{b}a\right], \tag{3.41}$$

where σ_a^2 is the variance of the random variable a.

Expression for $\mathrm{E}[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)]$**:**

As explained in Section A.3 of Appendix A, $\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]$ can be written as

$$\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right] = \mathbf{A}_1\mathrm{E}\left[\tilde{e}_0^2(k)\right] + \mathbf{A}_2 b\left(P_{\mathrm{up}}\mu\right)^2 \tag{3.42}$$

where $\mathbf{A}_1 = \mathrm{diag}\left\{1, a, a^2, \cdots, a^L\right\}$, $\mathbf{A}_2 = \mathrm{diag}\left\{0, 1, 1+a, \cdots, \sum_{l=0}^{L-1} a^l\right\}$, $b = [\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\rho_0(k)\sigma_n^2]$, $a = [1 - P_{\mathrm{up}}\mu + 2P_{\mathrm{up}}\mu\overline{\gamma}\rho_0(k)]\left(1 - P_{\mathrm{up}}\mu\right)$, and $\rho_0(k) = \sqrt{\frac{2}{\pi\mathrm{E}[e_0^2(k)]}}$.

Expression for $\mathrm{E}[\mathbf{n}(k)\mathbf{n}^T(k)]$**:**

According to Definition 4, we can write $\mathrm{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right]$ as

$$\mathrm{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right] = \sigma_n^2\mathbf{I}_{L+1}. \tag{3.43}$$

Expression for $\mathrm{E}[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)]$**:**

Considering that $\mathrm{E}[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)]$ is a diagonally dominant matrix, assumption *As-3*, its diagonal entries can be neglected using statement *St-3* leading to

$$\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right] \approx \mathbf{0}. \tag{3.44}$$

Expression for $\mathrm{E}[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)]$**:**

Using the diagonally dominant assumption, see assumption *As-3*, for the matrix $\mathrm{E}[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)]$, Definition 2 and Result 2, the relation $\mathbf{e}(k) = \tilde{\mathbf{e}}(k) + \mathbf{n}(k)$, and the approximation given in (3.44), we have by direct computation of the diagonal terms that

$$\mathrm{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right] \approx \overline{\gamma}\sigma_n^2\mathbf{C} + \overline{\gamma}\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right]\mathbf{C} \approx \overline{\gamma}\sigma_n^2\mathbf{C}, \tag{3.45}$$

where $\mathbf{C} = \mathrm{diag}\left\{\rho_0(k), \rho_1(k), \cdots, \rho_L(k)\right\}$, and $\rho_l(k) = \sqrt{\frac{2}{\pi\mathrm{E}[e_l^2(k)]}}$, as given in

Eq. (A.17). Using assumption *As-5*, we can simplify the expression above as follows

$$\mathrm{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right] \approx \overline{\gamma}\sigma_n^2\rho_0(k)\mathbf{I}_{L+1}. \tag{3.46}$$

Expression for $\mathrm{E}[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)]$**:**

Using the diagonally dominant assumption, see assumption *As-3*, for the matrix $\mathrm{E}[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)]$, and since $\mathrm{E}\left[\gamma_l^2(k)\right] = \overline{\gamma}^2$, for $l = 0, 1, \ldots, L$, then

$$\mathrm{E}\left[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)\right] \approx \overline{\gamma}^2\mathbf{I}_{L+1}. \tag{3.47}$$

Expression for $\mathrm{E}[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)]$**:**

Invoking the diagonally dominant assumption, see assumption *As-3*, for the matrix $\mathrm{E}[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)]$, utilizing Definition 2, Result 2, the relation $\mathbf{e}(k) = \tilde{\mathbf{e}}(k) + \mathbf{n}(k)$, and assumption *As-5*, we have

$$\begin{aligned}\mathrm{E}\left[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)\right] &\approx \overline{\gamma}\mathbf{C}\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right] + \overline{\gamma}\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right]\mathbf{C} \\ &\approx \overline{\gamma}\rho_0(k)\left\{\mathbf{A}_1\mathrm{E}\left[\tilde{e}_0^2(k)\right] + \mathbf{A}_2 b\left(P_{\mathrm{up}}\mu\right)^2\right\}. \end{aligned} \tag{3.48}$$

EMSE of the SM-AP Algorithm:

Considering that the algorithm has converged, in order to replace $P_{\mathrm{up}}(k)$ by P_{up}, and substituting (3.42), (3.43), (3.44), (3.46), (3.47), and (3.48) in Eq. (3.40), we have

$$\begin{aligned}&(2 - P_{\mathrm{up}}\mu)\mathrm{E}\left[\tilde{e}_0^2(k)\right]\mathrm{tr}\left\{\mathbf{A}_1\mathrm{E}\left[\mathbf{S}(k)\right]\right\} + (2 - P_{\mathrm{up}}\mu)b\left(P_{\mathrm{up}}\mu\right)^2\mathrm{tr}\left\{\mathbf{A}_2\mathrm{E}\left[\mathbf{S}(k)\right]\right\} \\ &- P_{\mathrm{up}}\mu\overline{\gamma}^2\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{S}(k)\right]\right\} - 2(1 - P_{\mathrm{up}}\mu)\overline{\gamma}\rho_0(k)\mathrm{E}\left[\tilde{e}_0^2(k)\right]\mathrm{tr}\left\{\mathbf{A}_1\mathrm{E}\left[\mathbf{S}(k)\right]\right\} \\ &- 2(1 - P_{\mathrm{up}}\mu)\overline{\gamma}\rho_0(k)b\left(P_{\mathrm{up}}\mu\right)^2\mathrm{tr}\left\{\mathbf{A}_2\mathrm{E}\left[\mathbf{S}(k)\right]\right\} + 2P_{\mathrm{up}}\mu\overline{\gamma}\sigma_n^2\rho_0(k)\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{S}(k)\right]\right\} \\ &\qquad = P_{\mathrm{up}}\mu\sigma_n^2\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{S}(k)\right]\right\}. \end{aligned} \tag{3.49}$$

Considering that $P_{\mathrm{up}} \ll 1$, or $\mu \ll 1$, or the product $P_{\mathrm{up}}\mu \ll 1$, see assumption *As-6*, so that the terms depending on $(P_{\mathrm{up}}\mu)^2$ are much smaller than the others and can be neglected, then

$$\begin{aligned}&\left[(2 - P_{\mathrm{up}}\mu) - 2(1 - P_{\mathrm{up}}\mu)\overline{\gamma}\rho_0(k)\right]\mathrm{tr}\left\{\mathbf{A}_1\mathrm{E}\left[\mathbf{S}(k)\right]\right\}\mathrm{E}\left[\tilde{e}_0^2(k)\right] \\ &\qquad = P_{\mathrm{up}}\mu\left[\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\sigma_n^2\rho_0(k)\right]\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{S}(k)\right]\right\}. \end{aligned} \tag{3.50}$$

Rearranging the equation above and considering that the elements on the main diagonal of $\mathrm{E}\left[\mathbf{S}(k)\right]$ are equal, see assumption *As-7*, we get an expression for the

EMSE of the GenSM-AP algorithm

$$\begin{aligned}\text{EMSE}_{\text{GenSM-AP}} = \text{E}\left[\tilde{e}_0^2(k)\right] &= \frac{\left[\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\sigma_n^2\rho_0(k)\right] P_{\text{up}}\mu \text{tr}\left\{\text{E}\left[\mathbf{S}(k)\right]\right\}}{\left[(2 - P_{\text{up}}\mu) - 2(1 - P_{\text{up}}\mu)\overline{\gamma}\rho_0(k)\right] \text{tr}\left\{\mathbf{A}_1 \text{E}\left[\mathbf{S}(k)\right]\right\}} \\ &= \frac{(L+1)\left[\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\sigma_n^2\rho_0(k)\right] P_{\text{up}}\mu}{\left[(2 - P_{\text{up}}\mu) - 2(1 - P_{\text{up}}\mu)\overline{\gamma}\rho_0(k)\right]} \left(\frac{1-a}{1-a^{L+1}}\right),\end{aligned} \tag{3.51}$$

where $a = \left[1 - P_{\text{up}}\mu + 2P_{\text{up}}\mu\overline{\gamma}\rho_0(k)\right](1 - P_{\text{up}}\mu)$ as defined in Eq. (A.20). The misadjustment of the GenSM-AP algorithm is given by

$$\text{M}_{\text{GenSM-AP}} = \frac{\text{EMSE}_{\text{GenSM-AP}}}{\sigma_n^2} = \frac{(L+1)\left[1 + \frac{\overline{\gamma}^2}{\sigma_n^2} - 2\overline{\gamma}\rho_0(k)\right] P_{\text{up}}\mu}{\left[(2 - P_{\text{up}}\mu) - 2(1 - P_{\text{up}}\mu)\overline{\gamma}\rho_0(k)\right]} \left(\frac{1-a}{1-a^{L+1}}\right). \tag{3.52}$$

By choosing $\mu = 1$, (3.51) becomes the EMSE of the SM-AP algorithm with the fixed modulus error-based constraint vector

$$\text{EMSE}_{\text{SM-AP}} = \text{E}\left[\tilde{e}_0^2(k)\right]_{\mu=1} = \frac{(L+1)\left[\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\sigma_n^2\rho_0(k)\right] P_{\text{up}}}{\left[(2 - P_{\text{up}}) - 2(1 - P_{\text{up}})\overline{\gamma}\rho_0(k)\right]} \left(\frac{1-\overline{a}}{1-\overline{a}^{L+1}}\right), \tag{3.53}$$

where $\overline{a} = \left[1 - P_{\text{up}} + 2P_{\text{up}}\overline{\gamma}\rho_0(k)\right](1 - P_{\text{up}})$.

The misadjustment of the SM-AP algorithm is given by

$$\text{M}_{\text{SM-AP}} = \frac{(L+1)\left[1 + \frac{\overline{\gamma}^2}{\sigma_n^2} - 2\overline{\gamma}\rho_0(k)\right] P_{\text{up}}}{\left[(2 - P_{\text{up}}) - 2(1 - P_{\text{up}})\overline{\gamma}\rho_0(k)\right]} \left(\frac{1-\overline{a}}{1-\overline{a}^{L+1}}\right) \tag{3.54}$$

and, since we considered $\mu = 1$, the accuracy of the approximations given by (3.53) and (3.54) will improve as P_{up} approaches zero, see assumption *As-6*.

In Sections A.4 and A.5 of Appendix A, we address the problem of modeling the variables $\rho_0(k)$ and P_{up}, respectively, in such a way that one can regard (3.53) and (3.54) as closed-form expressions.

3.4.5 EMSE for the Affine Projection Algorithm

Definition 5. *The trivial constraint vector is defined as*

$$\boldsymbol{\gamma}(k) = \mathbf{0}. \tag{3.55}$$

When $\overline{\gamma} = 0$, $\boldsymbol{\gamma}(k)$ must be a trivial constraint vector, since the condition $|\gamma_l(k)| \leq \overline{\gamma}$ is met, for $l = 0, 1, \ldots, L$. In this case, the GenSM-AP algorithm given by (3.22) becomes the AP algorithm with step-size μ [2].

Note that the steady-state results derived in the last subsection can be directly applied to the AP algorithm by making $\overline{\gamma} = 0$, see Definition 2. Note also that when $\overline{\gamma} = 0$ we have $P_{\text{up}} = 1$, see Section A.5 of Appendix A, and (3.51) becomes

$$\text{EMSE}_{\text{AP}} = \text{E}\left[\tilde{e}_0^2(k)\right]_{\overline{\gamma}=0, P_{\text{up}}=1} = \frac{(L+1)\mu}{(2-\mu)}\left(\frac{1-(1-\mu)^2}{1-(1-\mu)^{2(L+1)}}\right)\sigma_n^2 \tag{3.56}$$

which is the EMSE of the AP algorithm for small values of μ, see Eq. (4.123) of [2]. The misadjustment of the AP algorithm for small μ is given by

$$\text{M}_{\text{AP}} = \frac{(L+1)\mu}{(2-\mu)}\left(\frac{1-(1-\mu)^2}{1-(1-\mu)^{2(L+1)}}\right) \tag{3.57}$$

which also agrees with Eq. (4.124) of [2].

3.4.6 SM-AP vs. AP

We now discuss how $\text{M}_{\text{SM-AP}}$ relates to M_{AP} with $\mu = 1$. Recall that for $\overline{\gamma} = 0$, the SM-AP algorithm becomes the AP algorithm with $\mu = 1$.

Proposition 2. *By choosing τ satisfying* (3.59) *we generate $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$ in such a way that we can guarantee that the steady-state MSE of the SM-AP algorithm is lower than the one of the AP algorithm with $\mu = 1$.*

Proof. First, rewriting $\text{M}_{\text{SM-AP}}$ considering $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, $\tau \in \mathbb{R}_+$, and $\rho_0(k) = \sqrt{\frac{2}{\pi\text{E}[e_0^2(k)]}}$ as defined in Eq. (A.17), Eq. (3.54) becomes

$$\text{M}_{\text{SM-AP}} = (L+1)\frac{\overbrace{\left[1+\tau-\frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{\frac{\sigma_n^2}{\text{E}[e_0^2(k)]}}\sqrt{\tau}\right]P_{\text{up}}}^{t_1}}{\underbrace{\left[2-P_{\text{up}}-(1-P_{\text{up}})\frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{\frac{\sigma_n^2}{\text{E}[e_0^2(k)]}}\sqrt{\tau}\right]}_{t_2}}\left(\frac{\overbrace{1-\overline{a}}^{t_3}}{\underbrace{1-\overline{a}^{L+1}}_{t_4}}\right).$$

If $\tau \leq \frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{\frac{\sigma_n^2}{\text{E}[e_0^2(k)]}}\sqrt{\tau}$, i.e., if $\tau \leq \frac{8}{\pi}\frac{\sigma_n^2}{\text{E}[e_0^2(k)]}$, then $t_1 \leq P_{\text{up}}$. Since $\tau > 0$, then $P_{\text{up}} < 1$, thus $t_1 < 1$. In addition, in order to have $t_2 \geq 1$ we must satisfy

$$(1-P_{\text{up}}) \geq (1-P_{\text{up}})\frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{\frac{\sigma_n^2}{\text{E}[e_0^2(k)]}}\sqrt{\tau} \tag{3.58}$$

which leads to the condition $\tau \leq \frac{\pi}{8}\frac{\text{E}[e_0^2(k)]}{\sigma_n^2}$.

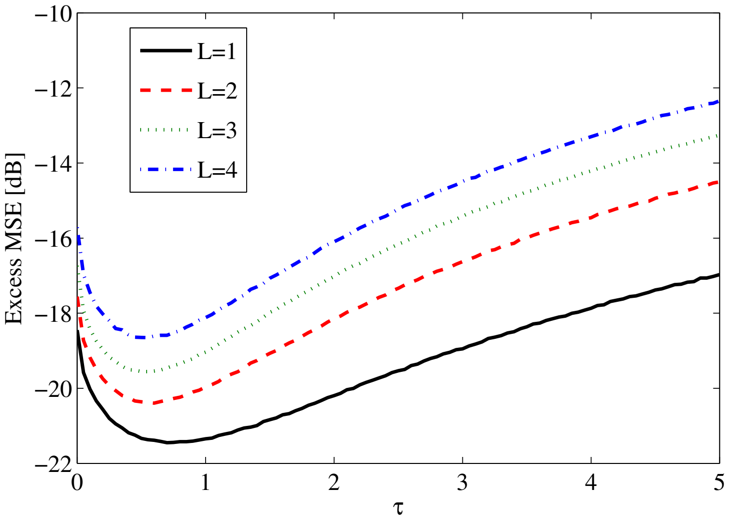


Figure 3.8: *Experimental EMSE vs. τ, where $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, for $L \in \{1,2,3,4\}$ — Basic Scenario.*

So, if we choose τ satisfying

$$0 < \tau \leq \min\left\{\frac{8}{\pi}\frac{\sigma_n^2}{\mathrm{E}[e_0^2(k)]} \ , \ \frac{\pi}{8}\frac{\mathrm{E}[e_0^2(k)]}{\sigma_n^2}\right\}, \tag{3.59}$$

we have $t_1 < 1$ and $t_2 \geq 1$. Therefore, $t_1/t_2 < 1$.
Expanding $\overline{a}$ we obtain

$$\begin{aligned}\overline{a} &= (1-P_{\mathrm{up}})^2 + P_{\mathrm{up}}(1-P_{\mathrm{up}})\frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{\frac{\sigma_n^2}{\mathrm{E}[e_0^2(k)]}}\sqrt{\tau} \\ &\leq (1-P_{\mathrm{up}})^2 + P_{\mathrm{up}}(1-P_{\mathrm{up}}) = (1-P_{\mathrm{up}}),\end{aligned}$$

where the inequality follows due to (3.58). For τ satisfying (3.59), we have $0 < 1 - P_{\mathrm{up}} < 1$, thus $\overline{a} \in (0,1)$. For these values of $\overline{a}$ we have $t_3 \leq t_4$ (the equality occurs when $L = 0$).
Since $t_1/t_2 < 1$ and $t_3/t_4 \leq 1$, we have proven that for τ satisfying (3.59) we guarantee $\mathrm{M}_{\text{SM-AP}} < (L+1) = \mathrm{M}_{\mathrm{AP}}\big|_{\mu=1}$. Additionally, since $0 < \sigma_n^2 \leq \mathrm{E}[e_0^2(k)] < \infty$, then the upper bound given in (3.59) is well defined.

□

Figure 3.8 shows the experimental EMSE for different values of L considering the Basic Scenario, to be defined in Subsection 3.4.7. This figure illustrates that τ_o, the value of τ that minimizes the steady-state MSE, is shifted (slightly) to the left as L increases. In this figure, the values of τ_o for $L = 1$, 2, 3, and 4 are 0.7, 0.6, 0.55, and 0.5, respectively. Another important observation is that, regardless the value of L, the EMSE, and consequently the misadjustment, first decreases and then increases, as τ increases.[9] This corroborates Proposition 2.

3.4.7 Simulation Results

In this section, we present simulation results in order to confirm the validity of the theoretical steady-state MSE expressions derived for the SM-AP algorithm. We consider a system identification configuration where the unknown system $\mathbf{h}$ is modeled as an FIR filter of order N.

In what follows, we present simulation results considering different scenarios. We start with a *Basic Scenario* (BS) and then we apply changes to this scenario in order to test the robustness of the proposed analysis.

Figures 3.9 to 3.11 depict the excess MSE (EMSE) versus τ, where τ is an auxiliary variable such that $\overline{\gamma} = \sqrt{\tau \times \sigma_n^2}$, for each scenario. Values of L greater than 4 are not addressed here since they lead to higher EMSE and usually require updates more often, as it was experimentally observed.

The experimental EMSE results were computed in the following way. For each realization of the stochastic process we let the algorithm run for 20×10^3 iterations. For the proposed scenarios the SM-AP algorithm converged in fewer than 2×10^3 iterations, for all tested values of L. The experimental steady-state MSE results were obtained by computing a time-average of the squared-error over the last 10^4 iterations, and then computing an ensemble-average over 100 independent runs. Subtracting the corresponding noise variances σ_n^2 from the experimental steady-state MSE yields the experimental EMSE. In addition, the theoretical EMSE was computed via Eq. (3.53).

Scenarios

There are four simulation scenarios. They are:

1. *Basic Scenario* (BS): The unknown system impulse response $\mathbf{h}$ was generated using the MATLAB command `randn(N+1,1)` with $N = 9$, and then normalizing it to obtain the following unitary-energy impulse response

$$\begin{aligned} \mathbf{h} = \; & [-0.0520 \quad -0.1228 \quad 0.1624 \quad 0.1592 \quad -0.4400 \\ & -0.0153 \quad -0.0839 \quad 0.3193 \quad 0.5561 \quad 0.5643]^T. \end{aligned}$$

   The input signal is a white noise drawn from a standard Gaussian distribution. The adaptive filter order is also N and the initial coefficient vector is $\mathbf{w}(0) = \mathbf{0}$.

[9] The same behavior is also valid for $L = 0$. The only reason we did not plot a curve for $L = 0$ is because the point τ_o in which such a curve changes its inclination is out of the range of τ shown in Figure 3.8. In addition, if we were to increase the range of τ to accommodate the case $L = 0$, it would become harder to distinguish the other values of τ_o for $1 \leq L \leq 4$.

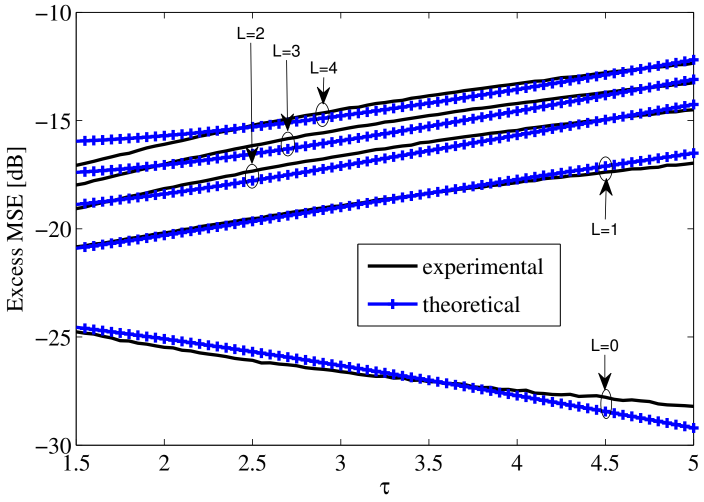


Figure 3.9: *EMSE vs.* τ*, where* $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$*, for* $L \in \{0, 1, 2, 3, 4\}$ *— Basic Scenario (BS).*

In addition, the variance of the additive white Gaussian noise is $\sigma_n^2 = 10^{-2}$.

2. Scenario 2: This scenario is similar to the BS, but with a correlated input signal. This input signal has eigenvalue spread equal to 20 and is obtained as the output of a first-order autoregressive process to a white noise signal drawn from a standard Gaussian distribution [2].

3. Scenario 3: This scenario corresponds to the BS, but with a lower noise variance σ_n^2, i.e., $\sigma_n^2 = 10^{-3}$.

4. Scenario 4: This scenario corresponds to the BS, but with a different unknown system given by:

$$\begin{aligned}\mathbf{h}_2 = \quad & [0.0809 \quad 0.2760 \quad -0.3399 \quad 0.1297 \quad 0.0480 \\ & -0.1968 \quad -0.0652 \quad 0.0516 \quad 0.5385 \quad 0.4167 \\ & -0.2031 \quad 0.4567 \quad 0.1092 \quad -0.0095 \quad 0.1076]^T.\end{aligned}$$

Once again the adaptive filter order is the same of the unknown system, i.e., $N = 14$.

Results

Figures 3.9, 3.10, and 3.11 depict the excess MSE (EMSE) versus τ considering different values of L for the BS, Scenario 2, and Scenario 3, respectively. The experimental and theoretical curves were generated as explained in the beginning of Subsection 3.4.7. The range of the variable τ is justified by assumption *As-6*. As can be observed in these figures, the proposed theoretical expression provides accu-

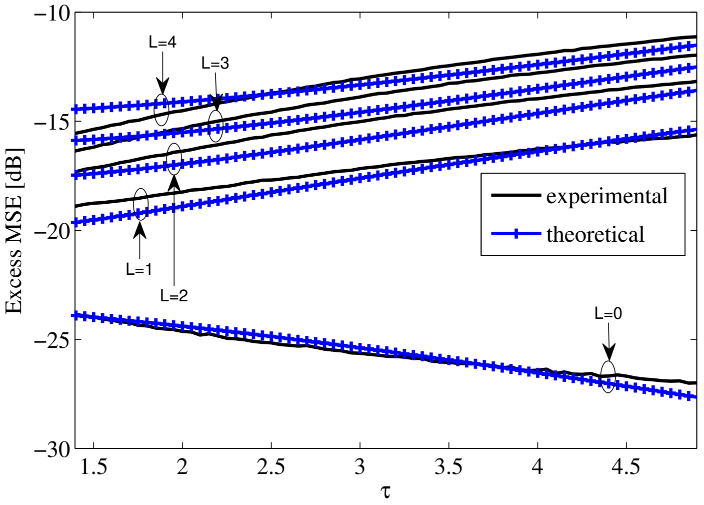


Figure 3.10: *EMSE vs. τ, where $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, for $L \in \{0, 1, 2, 3, 4\}$ — Scenario 2.*

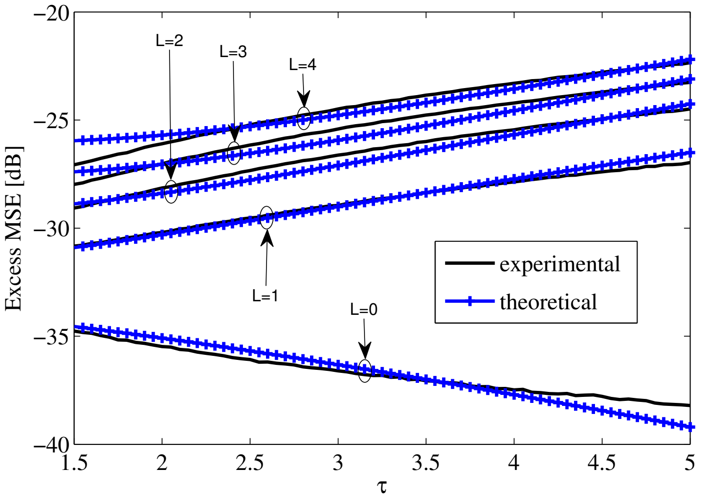


Figure 3.11: *EMSE vs. τ, where $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, for $L \in \{0, 1, 2, 3, 4\}$ — Scenario 3.*

rate estimates of the experimental EMSE for different scenarios, corroborating the proposed analysis. The results for the scenarios where the input signal is white (BS and Scenario 3) are slightly more accurate than the results considering a correlated input signal (Scenario 2).

In addition, the results show that the popular choice $\overline{\gamma} = \sqrt{5\sigma_n^2}$ [1, 2, 11, 12], i.e. $\tau = 5$, leads to low steady-state MSE only for $L = 0$. In practice, for $L = 0$, low EMSE is achieved by choosing $\tau \in [4, 5]$. For $L \neq 0$, however, the results show that $\tau = 5$ does not yield low EMSE. In such cases, one should choose $\tau \in [0.5, 1]$ in order to achieve low EMSE, see Figure 3.8. The drawback of choosing a small value for τ, and thus $\overline{\gamma}$, is that it reduces the probability of the constraint set $\mathcal{H}(k)$ contain the optimum filter coefficients $\mathbf{w}_o$, which increases the computational burden. In practice, for $L \neq 0$, choosing $\tau \in [1.5, 4]$ provides a fine balance between achieving

Table 3.1: *Experimental and theoretical EMSE — Scenario 4.*

L	$\tau = 3$		$\tau = 4$	
	experimental	theoretical	experimental	theoretical
0	0.00209	0.002232	0.001722	0.001694
1	0.01214	0.01257	0.01563	0.01688
2	0.01981	0.0195	0.02624	0.02705
3	0.025	0.255	0.03342	0.03496
4	0.02916	0.03325	0.0395	0.04404

low steady-state EMSE and low computational burden (see also Figure A.1).

Table 3.1 presents EMSE results for Scenario 4 considering two values of τ that are particularly interesting because they represent a compromise between low EMSE and high probability that $\mathbf{w}_o = \mathbf{h}_2 \in \mathcal{H}(k)$, which is related to the probability of update. Specifically, higher probability that $\mathbf{w}_o \in \mathcal{H}(k)$ implies lower probability of update, which in turn leads to lower computational burden. Observe that the theoretical values approximate quite well the experimental EMSE.

3.5 Conclusion

In this chapter we performed a thorough study of the SM-AP algorithm, which can be seen as an iterative method based on the intersection of constraint sets whose aim is to estimate a member of the feasibility set.

We started explaining the relation between the SM-AP algorithm and the SMF concept in order to motivate the original optimization problem, i.e., the optimization problem that expresses what the SM-AP algorithm should do. Then, we described the SM-AP algorithm following the article in which it was proposed. We observed that the SM-AP algorithm was proposed as the solution to an optimization problem which, in general, is different from the original optimization problem. Indeed, the constraint vector (CV) provides additional degrees of freedom that, if not properly set, they can deteriorate the accuracy of the estimates.

We explained the role played by the CV in the updating process. It is interesting to notice that although any point in the set Ψ_{k-L}^k is an acceptable solution, due to the presence of noise and also because of the finite number of constraint sets used per iteration, some points of this set are better than the others for a given $\mathbf{w}(k)$. More specifically, we show that the CV should take the error signal (at least the sign of the error signal) into account so that $\mathbf{w}(k+1)$ lies on the border of Ψ_{k-L}^k which is closest to $\mathbf{w}(k)$. In this part, it becomes clear why the general choice for the CV exhibits inaccurate results.

Next, we present a guideline explaining how to set the CV and also showed three types of CV: the simple choice CV (SC-CV), the fixed-modulus error-based CV

(FMEB-CV), and the exponential decay CV (ED-CV). The properties of such CVs were discussed and we focused on the ED-CV because it makes the SM-AP algorithm almost as fast as the FMEB-CV, but with steady-state MSE and probability of update as low as the SC-CV. In addition, the ED-CV as well as the SC-CV enable one to use high values of $\overline{\gamma}$, which means that these CVs achieve accurate results and require few updates.

Then, we presented a steady-state MSE analysis for the SM-AP algorithm employing the FMEB-CV. The proposed analysis relies on energy conservation arguments and is robust to changes in the input-signal model. The results encompass a number of algorithms such as the SM-NLMS, the SM-BNLMS, and the AP algorithms. The mathematical derivations presented here are general enough to be applied to the analysis of the SM-AP algorithm with other choices for the constraint vector. The theoretical expressions for the excess MSE and misadjustment predict well the MSE performance of the SM-AP algorithm in stationary environments. Moreover, it has been shown that the SM-AP algorithm can always have lower steady-state MSE than the AP algorithm. Simulation results corroborate the accuracy of the proposed analysis and validate Proposition 2.

Chapter 4

Sparsity-Aware Data-Selective Adaptive Filters

Many signals of different nature might have an interesting common feature: they admit a representation in which most of their components are null (sparse signal) or have a negligible magnitude (compressible signal). Indeed, such signals are usually found in a redundant representation so that they can be transformed to a domain where most of their energy is concentrated in few samples. This is the case of image and audio signals after sampling, for example.

Sparse signals and systems are found in many scenarios, such as echo cancellation, channel equalization, and system identification. The practical appeal of such applications has driven the development of many adaptive filtering algorithms aiming at exploiting the sparse nature of the involved signals. However, traditional algorithms such as the least-mean-square (LMS), normalized LMS (NLMS), affine projection (AP), and recursive least-squares (RLS) do not take advantage of sparsity in the signal models, thus disregarding the inherent structure of the problem that could be employed to improve convergence speed and steady-state error.[1]

In the adaptive filtering context, the most widely used approach to exploit sparsity is by performing coefficient updates that are proportional to the magnitude of the related coefficient, leading to the so-called *proportionate* family of algorithms. This family includes the proportionate NLMS (PNLMS) [41], the PNLMS++ [42], improved PNLMS (IPNLMS) [43], IPNLMS-l_0 [44], improved $\mu-$law PNLMS (IMPNLMS) [45], among others [46] [47]. In addition, the set-membership PNLMS (SM-PNLMS) [48] [49] can be interpreted as a data-selective version of the IPNLMS algorithm. In comparison to the original PNLMS algorithm, the SM-PNLMS has higher convergence speed, lower steady-state mean-squared error (MSE), and reduced computational burden due to data selection, which leads to

[1] For instance, it is widely known that the convergence speed of the classical algorithms degrades as the impulse response of the involved system becomes longer [39, 40].

sparse updates.

In addition to the proportionate algorithms based on NLMS, different adaptations of the affine projection algorithm employing the proportionate idea have been proposed, leading to the *proportionate AP algorithms* (PAPAs). Among the motivations for such algorithms we may mention that: (i) they include their PNLMS counterparts as a particular case when there is no data reuse, i.e., when $L = 0$; (ii) they can accelerate convergence by reusing previous data. The PAPA and improved PAPA (IPAPA) algorithms were proposed in [50]. In [49], the proposed set-membership PAPA (SM-PAPA) was shown to have faster convergence than the PAPA.

Recently, a different approach to deal with sparsity has been exploited, where a penalty function accounting for the sparsity is added to the original objective function and a gradient-based algorithm is derived. Examples of resulting algorithms are the *zero-attracting AP algorithm* (ZA-APA) and reweighted ZA-APA (RZA-APA) [40], whose penalty functions are related to the l^1 norm of the coefficient vector.

In this chapter, we present sparsity-aware data-selective solutions that bring together some advantageous properties of the aforementioned algorithms, while yielding low computational burden. Unlike most of the methods that tackle sparsity by minimizing the l^1 norm of the coefficient vector, the proposed algorithms use a penalty function based on an approximation to the l^0 norm. In summary, the contributions of this chapter are:

1. We present a comprehensive material explaining how to approximate the l^0 norm and the advantages of such an approach (Section 4.1). Although the approximations used are widely known functions, we establish for the first time connections with the l^1 norm and the commonly used reweighted technique [51]. In addition, the content of this section is not restricted to the adaptive filtering context and, therefore, it can find applications in other areas involving sparse/compressible signals and systems.

2. We verify the advantages of the approximation to the l^0 norm over the widely used approach that consists of minimizing the l^1 norm. These advantages are confirmed through simulation of simple algorithms that allow us to focus only on the effect of the sparsity-promoting scheme (Section 4.2).

3. We propose two data-selective algorithms that exploit the sparsity of the signals (Section 4.3), provide geometric interpretations to their updating schemes and prove two theorems regarding the stability of the proposed algorithms (Section 4.4).

4. We calculate the number of operations required by the proposed and competing algorithms (Section 4.6).
5. We quantify the reduction of computational burden as compared to traditional algorithms and show simulation results proving that the proposed algorithms outperform the other adaptive filtering algorithms designed to exploit sparsity (Section 4.7).

The content of this chapter was published mostly in [31], but also in [52, 53].

This chapter is organized as follows. Section 4.1 addresses the problem of modeling sparsity. In this section, we explain and motivate the approach we used by showing its advantages over the l^0 and l^1 norms. In Section 4.2, we provide some simulation results that corroborate these advantages. Next, in Section 4.3, two novel data-selective algorithms tailored to exploit the sparsity of the involved signals are proposed, and some of their properties are addressed in Section 4.4. A literature review regarding adaptive filtering algorithms designed to exploit sparsity is presented in Section 4.5. Section 4.6 discusses the computational burden of the aforementioned algorithms. Section 4.7 presents simulation results considering an extensive set of scenarios and the conclusions are drawn in Section 4.8. Appendix B provides some proofs related to the properties of the proposed algorithms.

4.1 Modeling Sparsity

In this chapter, *sparse signals* are vectors of a finite-dimensional vector space which can be represented as a linear combination of a small amount of basis vectors of the related space [54]. Usually, algorithms that are originally developed to deal with sparse signals are also employed in the context of *compressible signals*, which are not—strictly speaking—sparse signals, but can be well approximated as such [54]. We shall first develop the proposed algorithms for sparse signals and then describe some implications of using them with compressible signals.

It is well known that the sparsity of a parameter vector can be promoted by minimizing its l^0 norm. However, working directly with such a norm is a very difficult task since it leads to an NP-hard problem, which turns its use prohibitive in online applications [54]. In this section, we show how to approximate the l^0 norm by using *almost everywhere* (a.e.) differentiable functions, which allows the related (non-convex) optimization problem to be solved by using stochastic gradient methods, a key feature of our proposal. In addition, we explain the advantages of minimizing such functions over minimizing the l^0 and l^1 norms. The material presented in this section is key to address some properties of the proposed algorithms, which are discussed in Section 4.4.

4.1.1 Approximating the l^0 norm

Let us define the set of indexes $\mathcal{N} \triangleq \{0, 1, \ldots, N\} \subset \mathbb{N}$. The l^0 norm of a vector $\mathbf{z} = [z_0 \ z_1 \ \ldots \ z_N]^T \in \mathbb{R}^{N+1}$ is defined as the number of nonzero elements of $\mathbf{z}$, i.e., $\|\mathbf{z}\|_0 \triangleq \#\{n \in \mathcal{N} : z_n \neq 0\} \in \mathbb{N}$, in which # denotes the cardinality of a finite set. Therefore, by recalling the definition of sparse signals adopted in this chapter, one can observe that the sparsity of a vector is directly revealed by its l^0 norm.[2]

In many practical applications it is desirable to find the sparsest approximation of corresponding compressible signals, which is related to minimizing their l^0 norm. In addition to difficulties inherent to combinatorial searches for minimum l^0-norm solutions, the resulting optimization problems are also ill-conditioned, since small perturbations on $\mathbf{z}$ may yield large changes on $\|\mathbf{z}\|_0$. These facts hinder the attempt to directly minimize the l^0 norm in many practical cases, especially when noise is present.

These difficulties are due to the discontinuity of the l^0 norm. Thus, they could be overcome by approximating the l^0 norm using a continuous function $F_\beta : \mathbb{R}^{N+1} \longrightarrow \mathbb{R}_+$, in which $\beta \in \mathbb{R}_+$ is a parameter responsible for controlling the compromise between quality of the approximation and "smoothness" of F_β [55]. A common practice is to analytically define a continuous function F_β so that

$$\lim_{\beta \longrightarrow \infty} F_\beta(\mathbf{z}) = \|\mathbf{z}\|_0. \tag{4.1}$$

In order for F_β to satisfy this property for all $\mathbf{z}$, one must have, in particular, $F_\beta(z_n \mathbf{e}_n) \longrightarrow 1$ as long as $\beta \longrightarrow \infty$ and the real number $z_n \neq 0$, where $\mathbf{e}_n$ is the nth vector[3] of the canonical basis of $\mathbb{R}^{N+1}$. If $z_n = 0$, then $F_\beta(\mathbf{0}) \longrightarrow 0$, as long as $\beta \longrightarrow \infty$. This means that, $\forall n \in \mathcal{N}$, F_β must satisfy

$$\lim_{\beta \longrightarrow \infty} [1 - F_\beta(z_n \mathbf{e}_n)] = \begin{cases} 0 & \text{if } z_n \neq 0, \\ 1 & \text{if } z_n = 0. \end{cases} \tag{4.2}$$

As a rule of thumb, we can analytically define F_β so that

$$1 - F_\beta(z_n \mathbf{e}_n) \begin{cases} \approx 0 & \text{if } z_n \neq 0, \\ = 1 & \text{if } z_n = 0, \end{cases} \tag{4.3}$$

where the approximation above becomes more accurate as β increases. Besides continuity, we want to work with functions F_β that are also differentiable (at least, a.e. differentiable), since this will allow us to employ gradient-based optimization

[2] Observe that the l^0 norm is not truly a norm, because in general we have $\|c\mathbf{z}\|_0 \neq |c|\|\mathbf{z}\|_0$, for $c \in \mathbb{R}$. However, the term "l^0 norm" is widely accepted/used in the literature.

[3] That is, it has only 0 elements, except for a 1 in its nth coordinate, with $n \in \mathcal{N}$.

methods. Next subsection presents some examples of F_β.

4.1.2 Standard Approximations

There is a variety of functions F_β that can be used to approximate the l^0 norm of a vector. Four examples of such functions are [44, 52, 56, 57]:

$$F_\beta(\mathbf{z}) = \sum_{n\in\mathcal{N}} \left(1 - \mathrm{e}^{-\beta|z_n|}\right), \tag{4.4a}$$

$$F_\beta(\mathbf{z}) = \sum_{n\in\mathcal{N}} \left(1 - \mathrm{e}^{-\frac{1}{2}\beta^2 z_n^2}\right), \tag{4.4b}$$

$$F_\beta(\mathbf{z}) = \sum_{n\in\mathcal{N}} \left(1 - \frac{1}{1+\beta|z_n|}\right), \text{ and} \tag{4.4c}$$

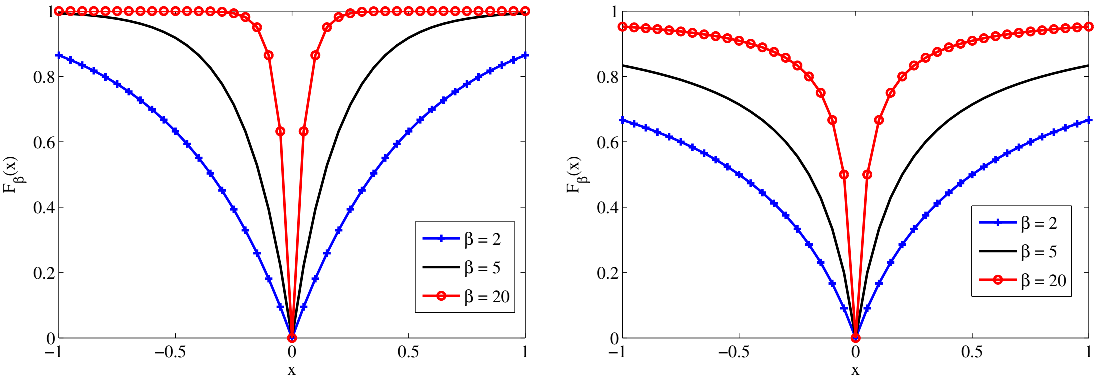
$$F_\beta(\mathbf{z}) = \sum_{n\in\mathcal{N}} \left(1 - \frac{1}{1+\beta^2 z_n^2}\right). \tag{4.4d}$$

By carefully observing Eqs. (4.4a)-(4.4d), one can verify that all of them satisfy the expressions in (4.2) and (4.3). For instance, by considering $\mathbf{z} = z_n\mathbf{e}_n$ (i.e., only the nth component of $\mathbf{z}$ can be nonzero), then Eq. (4.4a) tells us that $1 - F_\beta(z_n\mathbf{e}_n) = \mathrm{e}^{-\beta|z_n|}$, which is approximately equal to 0, when $z_n \neq 0$ and β is large, whereas it is equal to 1, when $z_n = 0$. Note also that $\mathrm{e}^{-\beta|z_n|} \longrightarrow 0$, when $\beta \longrightarrow \infty$ and $z_n \neq 0$. Besides, $\mathrm{e}^{-\beta|z_n|}$ is differentiable with respect to z_n, for any $z_n \neq 0$ (a.e. differentiable). The same properties hold for the other approximations.

Eq. (4.4a), denominated as multivariate Laplace function (LF) [56, 58], is probably the most widely used approximation to the l^0 norm. In addition, Eq. (4.4c) describes the multivariate Geman-McClure function (GMF) [56, 59]. Eqs. (4.4b) and (4.4d) are modifications of the LF and GMF, respectively, so that their derivatives are also continuous functions (see Eq. (4.5)).

Figure 4.1 depicts the univariate LF and GMF for different values of β. Notice that these functions are not convex and that β trades off smoothness for quality of approximation as it increases. Figure 4.2 illustrates the bivariate GMF for $\beta = 5$. Observe that such a function has only one global minimum at $\mathbf{z} = \mathbf{0}$ and that an optimization method following the opposite direction of the gradient will converge to this minimum regardless the initial point due to the smoothness of the depicted F_β. Clearly, the same observations are valid for the other functions F_β.

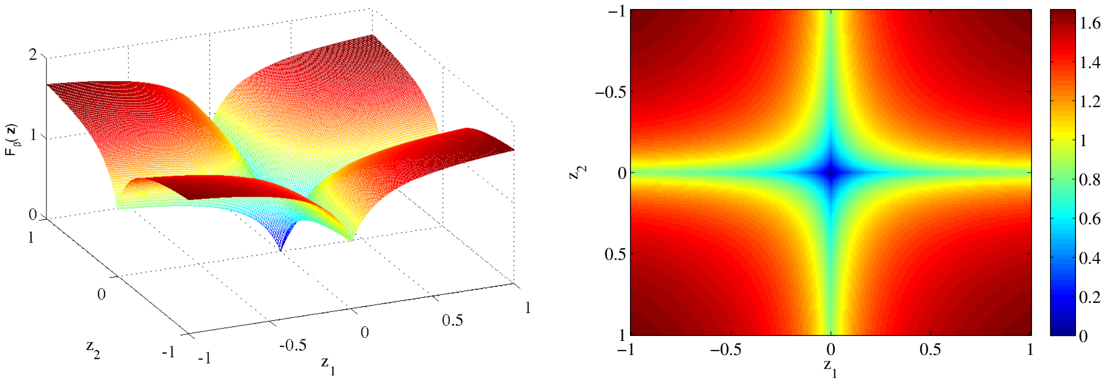
Defining $f_\beta(z_n) \triangleq \frac{\partial F_\beta(\mathbf{z})}{\partial z_n}$, the derivatives corresponding to Eqs. (4.4a)-(4.4d) are,



(a) LF.

(b) GMF.

Figure 4.1: *Univariate functions $F_\beta(x)$, with $x \in [-1, 1] \subset \mathbb{R}$, for different values of β.*



(a) GMF.

(b) GMF contour.

Figure 4.2: *Bivariate GMF for $\beta = 5$ and $\mathbf{z} \in [-1, 1] \times [-1, 1] \subset \mathbb{R}^2$.*

respectively,

$$f_\beta(z_n) = \beta \mathrm{sign}(z_n) \mathrm{e}^{-\beta |z_n|}, \tag{4.5a}$$

$$f_\beta(z_n) = \beta^2 z_n \mathrm{e}^{-\frac{1}{2}\beta^2 z_n^2}, \tag{4.5b}$$

$$f_\beta(z_n) = \frac{\beta \mathrm{sign}(z_n)}{(1 + \beta |z_n|)^2}, \text{ and} \tag{4.5c}$$

$$f_\beta(z_n) = \frac{2\beta^2 z_n}{(1 + \beta^2 z_n^2)^2}, \tag{4.5d}$$

where the function sign : $\mathbb{R} \longrightarrow \{-1, 0, 1\}$ maps negative real numbers into -1, positive real numbers into 1, and 0 into 0.

Finally, we can define the gradient of $F_\beta(\mathbf{z})$ with respect to $\mathbf{z}$ as

$$\nabla F_\beta(\mathbf{z}) \triangleq \mathbf{f}_\beta(\mathbf{z}) \triangleq [f_\beta(z_0) \ f_\beta(z_1) \ \ldots \ f_\beta(z_N)]^T. \tag{4.6}$$

4.1.3 Choosing β

We have already seen that the parameter β represents a trade-off between smoothness and quality of approximation. Here we show that this characteristic is very useful to incorporate *a priori* knowledge about the nature of the involved signals.

High values of β lead to steep declines in $F_\beta(\mathbf{z})$ for $\mathbf{z}$ very close to $\mathbf{0}$, whereas $F_\beta(\mathbf{z})$ is almost constant for the remaining values of $\mathbf{z}$. Taking the univariate case depicted in Figure 4.1 as example, high values of β, such as $\beta = 20$, imply that $F_\beta(x)$ presents a small zero-attraction interval (around 0) in which any x belonging to this interval is strongly pushed to 0. As β decreases toward 0, $F_\beta(\mathbf{z})$ becomes less steep (see $\beta = 2$ in Figure 4.1) and the zero-attraction interval expands, but the strength with which $\mathbf{z}$ is attracted to $\mathbf{0}$ decreases.

Therefore, when dealing with sparse signals we could use high values of β because we only need to attract the values which are really close to $\mathbf{0}$. In practice, it is preferable to use moderate values of β because smoothness is important to guarantee the effectiveness of gradient-based optimization methods, as previously explained. On the other hand, we must reduce β even further when dealing with compressible signals. Indeed, we must enlarge the zero-attraction interval so that small components of the compressible signal lie on such an interval. Section 4.7 shows some choices for β considering both sparse and compressible signals.

4.1.4 Comparing F_β to the l^1 norm

In Subsection 4.1.1, the difficulties of working directly with the l^0 norm were explained and the function F_β, a continuous and a.e. differentiable function that approximates the l^0 norm, was introduced to circumvent such issues. Here, we discuss the most widely used approach to promote sparsity, which consists of minimizing the l^1 norm rather than the l^0 norm, and then we connect the minimization of the l^1 norm with the proposed approach based on F_β.

In addition to continuity and a.e. differentiability, the l^1 norm is also convex, which turns gradient-based methods very suitable to its minimization.[4] Indeed, the l^1 norm of the vector $\mathbf{z}$ is given by $\|\mathbf{z}\|_1 = \sum_{n=0}^{N} |z_n|$ and its derivative is $g(z_n) \triangleq \frac{\partial \|\mathbf{z}\|_1}{\partial z_n} = \text{sign}(z_n)$ so that its gradient is $\nabla \|\mathbf{z}\|_1 \triangleq [g(z_0)\ g(z_1)\ \cdots\ g(z_N)]^T$. Clearly, the facility of working with such a norm explains why it has been extensively used, as can be seen in [60–63] and references therein. On the other hand, there are two major issues related to minimizing the l^1 norm: (i) the conditions that guarantee equivalence between minimizing the l^0 and l^1 norms may not be satisfied in practical applications [52, 64] and (ii) its derivative $g(z_n)$, and thus the gradient

[4] In fact, interior point methods are also a common choice in other contexts and are out of the scope of this thesis.

$\nabla\|\mathbf{z}\|_1$, does not take into account how close z_n is to 0.

Observe that one expects a good sparsity-promoting scheme to push z_n to 0 for small $|z_n|$, whereas z_n should be less attracted to 0 as $|z_n|$ increases. Intuitively, if a component z_n has large $|z_n|$, then the sparsity-promoting scheme should not waste energy in an attempt to make it equal to 0. Indeed, by giving priority to components which are close to 0, the numerical method is able to reduce $\|\mathbf{z}\|_0$ (the original problem) rapidly and, as a result, we can perform updates using small step sizes, keeping numerical errors under control. Hence, to cope with issue (ii) mentioned above, the sparsity-promoting scheme using gradient-based minimization of the l^1 norm could be improved by employing a technique known as reweighted l^1 minimization proposed in [51] and used in adaptive filtering algorithms such as those in [40, 65]. In this heuristic approach, the functions $g(z_n)$ that compose $\nabla\|\mathbf{z}\|_1$ are replaced by $\hat{g}(z_n)$, which are defined as

$$\hat{g}(z_n) \triangleq \frac{\text{sign}(z_n)}{1+\epsilon|z_n|}, \tag{4.7}$$

where ϵ is a predefined positive real constant. Observe that expressions (4.7) and (4.5c) are very similar in the way they consider the information regarding the proximity of z_n to 0. Nevertheless, besides being formally justified, expression (4.5c) also has the advantage of employing a parameter β which has a clear meaning, making its choice easier.

4.2 Validation

Before we move on to the development of sparsity-aware data-selective adaptive filters employing F_β, it is wise to check if the advantages of using an approximation to the l^0 norm over using the l^1 norm are really verified in practice. That is, since data-selective algorithms introduce a nonlinearity due to the innovation check, then it would be difficult to draw conclusions about the scheme used to promote sparsity. Indeed, due to such nonlinearity we would observe a combined/coupled effect that would prevent us from drawing any conclusions regarding the sparsity-promoting scheme. Therefore, in this section we consider the classical AP algorithm and we add different sparsity-promoting schemes to it. Another reason that justifies this choice is the existence of AP algorithms using the l^1 norm to promote sparsity, i.e., the competing algorithms were already available in the literature.

In Subsection 4.2.1 we present these AP algorithms tailored for applications involving sparse signals, which we call sparsity-aware affine projection algorithms. Then, in Subsection 4.2.2 we compare these algorithms via simulation considering scenarios with different degrees of sparsity.

4.2.1 Sparsity-Aware Affine Projection Algorithms

In this subsection we present four affine projection algorithms that were designed for applications involving sparse signals. First, we present two algorithms that we proposed in [52], viz. the *affine projection algorithm for sparse system identification* (APA-SSI) and the quasi APA-SSI (QAPA-SSI). These algorithms were analyzed regarding their stability and steady-state MSE in [53]. Then, we present two algorithms, which were proposed in [40], viz. the *zero-attracting affine projection algorithm* (ZA-APA) and the reweighted ZA-APA (RZA-APA). These four algorithms have been derived by adding to the objective function of the AP algorithm a penalty function based on a sparsity-promoting function of the coefficient vector. Indeed, the APA-SSI and QAPA-SSI use a penalty function based on an approximation to the l^0 norm, i.e., based on the F_β explained in the previous section, whereas the penalty function of the ZA-APA and RZA-APA are related to the l^1 norm.

The APA-SSI and ZA-APA are dual algorithms, in the sense that they are almost identical, but the penalty function used in the APA-SSI is $F_\beta(\mathbf{w}(k+1))$, whereas the ZA-APA uses $\|\mathbf{w}(k+1)\|_1$ as penalty function. Therefore, we are mainly interested in the comparison between these two algorithms.

It is worth highlighting that although we have proposed the APA-SSI and QAPA-SSI, we opted for a succinct introduction of these two algorithms in this thesis because they are not our focus and also because they can be regarded as particular cases of the data-selective algorithms that we propose in the following section. Thus, we present all the mathematical steps to derive these data-selective algorithms as well as some of their properties, which can be easily mapped to corresponding results for the APA-SSI and QAPA-SSI. Alternatively, the reader may refer to [52, 53] for more details about APA-SSI and QAPA-SSI.

APA-SSI

As previously mentioned, the APA-SSI adds a penalty function based on an approximation to the l^0 norm of $\mathbf{w}(k+1)$ to the AP optimization problem in order to promote sparsity at each iteration. Thus, the optimization problem from which the APA-SSI originates is given by

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1)-\mathbf{w}(k)\|_2^2+\alpha F_\beta(\mathbf{w}(k+1)) \\ &\text{subject to } \mathbf{d}(k)-\mathbf{X}^T(k)\mathbf{w}(k+1)=\mathbf{0}, \end{aligned} \tag{4.8}$$

where $\alpha \in \mathbb{R}_+$ is a nonnegative parameter that determines the weight given to the penalty function.

The updating equation for the APA-SSI is

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) &+ \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) \\ &+ \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{f}_\beta\left(\mathbf{w}(k)\right),\end{aligned} \tag{4.9}$$

in which $\mathbf{S}(k) \triangleq \left(\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}\right)^{-1}$, δ is a regularization factor, and μ is the step size.

QAPA-SSI

The QAPA-SSI is characterized by the following updating rule:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) - \mu\frac{\alpha}{2}\mathbf{f}_\beta\left(\mathbf{w}(k)\right). \tag{4.10}$$

The QAPA-SSI has a reduced computational complexity, in comparison with the APA-SSI. In addition, the QAPA-SSI also generalizes the l_0-NLMS algorithm proposed in [44]. Indeed, the l_0-NLMS algorithm can be achieved by setting the QAPA-SSI in the following way: (i) $L = 0$, (ii) $f_\beta(z_n)$ as a first-order approximation via Taylor series of the exponential function in Eq. (4.5a). On the other hand, the QAPA-SSI does not satisfy the equality constraint present in the optimization problem related to AP algorithm and, therefore, it is not truly an affine projection algorithm.

ZA-APA

The ZA-APA was derived by directly minimization of the AP cost function plus a penalty function based on the l^1 norm of $\mathbf{w}(k+1)$. That is, its optimization problem is given by:

$$\begin{aligned}&\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 + \alpha\|\mathbf{w}(k+1)\|_1 \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \mathbf{0},\end{aligned} \tag{4.11}$$

where $\alpha \in \mathbb{R}_+$ is a nonnegative parameter that determines the weight given to the penalty function.

In order to highlight the similarities between the APA-SSI and the ZA-APA, we write the updating equation of the ZA-APA as[5]

$$\begin{aligned}\mathbf{w}(k+1) = &\ \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) \\ &+ \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{sign}\left(\mathbf{w}(k)\right),\end{aligned} \tag{4.12}$$

where $\mathbf{sign}\left(\mathbf{w}(k)\right) \triangleq [g(w_0(k))\ g(w_1(k))\ \cdots\ g(w_N(k))]^T$ is the element-wise sign function and the definition of g is in Subsection 4.1.4.

RZA-APA

The RZA-APA uses the reweighted technique explained in Subsection 4.1.4 leading to the following recursion:

$$\begin{aligned}\mathbf{w}(k+1) = &\ \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) \\ &+ \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{P}\left(\mathbf{w}(k)\right),\end{aligned} \tag{4.13}$$

where $\mathbf{P}\left(\mathbf{w}(k)\right) \triangleq [\hat{g}(w_0(k))\ \hat{g}(w_1(k))\ \cdots\ \hat{g}(w_N(k))]^T$, see Subsection 4.1.4 for the definition of $\hat{g}$.

4.2.2 Comparing methods

Here, we present simulation results for the aforementioned algorithms considering scenarios with different degrees of sparsity.

We first present the simulation scenarios and evaluate the APA-SSI and QAPA-SSI considering different functions F_β. Then, using the approximation function F_β that leads to the best results, we compare the proposed algorithms based on F_β *versus* the algorithms based on the l^1 norm.

Scenarios

The simulation scenarios that we consider are the same three experiments proposed in [40]. Those scenarios allow us to assess the performance of the proposed algorithms for different degrees of sparsity. The experiments consist of identifying an unknown system composed of 16 coefficients, whose taps are set as follows: (i) Exp. 1: 4th tap equal to 1, others equal to 0; (ii) Exp. 2: odd taps equal to 1, even taps equal to 0; and (iii) Exp. 3: all taps equal to 1.

Regarding the adaptive filter parameters, the number of coefficients is 16 and the following algorithms are tested: the proposed ones (APA-SSI and QAPA-SSI), the

[5] The only difference between Eq. (4.12) and the ZA-APA of [40] is a regularization factor δ which appears in the definition of $\mathbf{S}(k)$. In addition, we incorporated the step size μ in every term that is added to $\mathbf{w}(k)$, which essentially implies that our α is a scaled version of the α in [40].

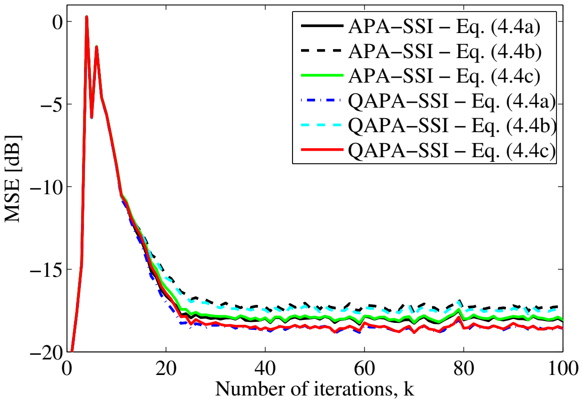


Figure 4.3: *Comparing the performances of the APA-SSI and QAPA-SSI for different functions F_β considering Exp. 1. The curve obtained using* (4.4d) *is omitted, but it coincides with the curve obtained using* (4.4b)*.*

proposals of [40] (ZA-APA and RZA-APA), and the classical ones (AP and NLMS) to serve as benchmarks for comparisons. The algorithms were set so that they have a similar convergence speed.[6] Thus, we use step-size $\mu = 0.9$, regularization factor $\delta = 10^{-12}$, data reuse factor $L = 4$, $\beta = 5$ (following the suggestion of [44]), and, in accordance with the suggested values in [40] we use $\alpha = 5 \times 10^{-3}$ and $\epsilon = 100$. In addition, the reference signal $d(k)$ is assumed to be corrupted by an additive white Gaussian measurement noise with variance $\sigma_n^2 = 0.01$.

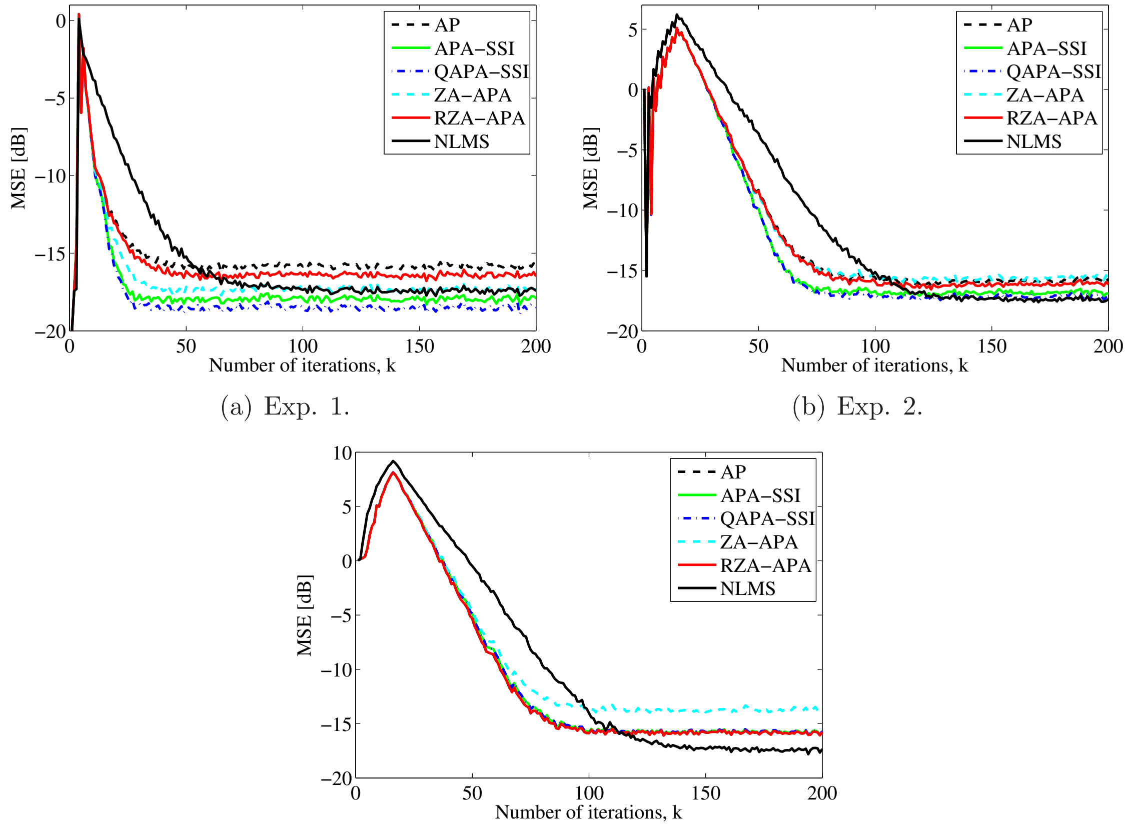
Results for different approximations

The APA-SSI and QAPA-SSI were tested in several scenarios using the functions F_β given in Eqs. (4.4a) to (4.4d). Throughout all scenarios we tested, the following observations always held: (i) convergence speed was indeed similar for all functions; (ii) Eqs. (4.4b) and (4.4d) led to the worst results in terms of steady-state MSE; and (iii) approximations based on Eqs. (4.4a) and (4.4c) exhibited almost identical steady-state MSE performances.

Figure 4.3 depicts an example of such comparison considering Exp. 1. For the sake of clearness, the curve corresponding to F_β given in Eq. (4.4d) was omitted, but such F_β yielded results very similar to the ones obtained when using Eq. (4.4b).

In what follows, we consider that both the APA-SSI and QAPA-SSI employ the GMF given in Eq. (4.4c). We opted for the GMF rather than the LF because the former is cheaper to compute since the latter requires the computation of an exponential, i.e., the computation of (4.5c) requires fewer arithmetic operations than the computation of (4.5a).

[6] This observation is valid for the AP-based algorithms. By using the same step-size μ of the AP algorithm, the NLMS algorithm will be slower.



(a) Exp. 1.

(b) Exp. 2.

(c) Exp. 3.

Figure 4.4: *MSE learning curve for experiments involving different degrees of sparsity.*

Results for different degrees of sparsity

Figure 4.4 depicts the MSE results for Exps. 1, 2, and 3. It can be observed that the convergence speeds are similar for all AP-based methods. Figure 4.4(a) shows that by exploiting the sparsity of the underlying unknown system, all four algorithms (APA-SSI, QAPA-SSI, ZA-APA, and RZA-APA) outperformed the AP algorithm. Actually, the proposed APA-SSI and QAPA-SSI achieved the best results. As depicted in Figures 4.4(b) and 4.4(c), as the unknown system becomes less sparse, the performance of the algorithms which explicitly take sparsity into account in their formulations become worse and converge to the performance of the AP algorithm when there is no sparsity, as shown in Figure 4.4(c). In fact, the result obtained by the ZA-APA in a dispersive environment was worse than the one obtained by the AP algorithm. In addition, one may note that when the sparsity factor is 50% (Exp. 2), the performance of the methods based on l^1 norm is not very different from that of the AP algorithm, whereas the proposed APA-SSI and QAPA-SSI are still able to take advantage of this somewhat low degree of sparsity.

Now that we have confirmed the advantages of using an approximation to the l^0 norm, we can proceed to propose sparsity-aware data-selective algorithms.

4.3 Sparsity-Aware Data-Selective Algorithms

In this section, we propose two algorithms that exploit the sparsity of the involved signals/systems in order to achieve higher convergence speed and lower computational burden than traditional algorithms. In addition, the proposed algorithms combine data reuse and data selection mechanisms, as the SM-AP algorithm does. The first proposal, abbreviated by SSM-AP, is a sparsity-aware version of the SM-AP algorithm, whereas the second, viz. quasi SSM-AP (QSSM-AP), is a simplified version of the former designed to reduce the computational burden even further.

4.3.1 SSM-AP Algorithm

The SSM-AP algorithm updates whenever $|e_0(k)| > \overline{\gamma}$, following an updating recursion that is an *approximation* of the solution to the optimization problem:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 + \alpha\|\mathbf{w}(k+1)\|_0 \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k), \end{aligned} \tag{4.14}$$

where $\alpha \in \mathbb{R}_+$ denotes the weight given to the l^0 norm penalty. We used the word “approximation” since we actually “solve” the following problem:

$$\begin{aligned} &\text{minimize } \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 + \alpha F_\beta\left(\mathbf{w}(k+1)\right) \\ &\text{subject to } \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k), \end{aligned} \tag{4.15}$$

In order to solve this optimization problem, we form the Lagrangian $\mathbb{L}$ as

$$\mathbb{L} = \|\mathbf{w}(k+1) - \mathbf{w}(k)\|_2^2 + \alpha F_\beta\left(\mathbf{w}(k+1)\right) + \boldsymbol{\lambda}^T(k)\left[\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) - \boldsymbol{\gamma}(k)\right], \tag{4.16}$$

differentiate it with respect to $\mathbf{w}(k+1)$ and $\boldsymbol{\lambda}(k)$, and equal the resulting expressions to zero (i.e., $\nabla\mathbb{L} = \mathbf{0}$), thus yielding

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k)\frac{\boldsymbol{\lambda}(k)}{2} - \frac{\alpha}{2}\nabla F_\beta\left(\mathbf{w}(k+1)\right), \tag{4.17}$$

$$\mathbf{X}^T(k)\mathbf{w}(k+1) = \mathbf{d}(k) - \boldsymbol{\gamma}(k), \tag{4.18}$$

respectively. Then, the left-multiplication of Eq. (4.17) by $\mathbf{X}^T(k)$ and the substitution of Eq. (4.18) into the resulting equation generates

$$\frac{\boldsymbol{\lambda}(k)}{2} = \left(\mathbf{X}^T(k)\mathbf{X}(k)\right)^{-1}\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] + \frac{\alpha}{2}\left(\mathbf{X}^T(k)\mathbf{X}(k)\right)^{-1}\mathbf{X}^T(k)\nabla F_\beta\left(\mathbf{w}(k+1)\right), \tag{4.19}$$

where we assumed that $\mathbf{X}^T(k)\mathbf{X}(k)$ is invertible. Substituting Eq. (4.19) into Eq. (4.17) leads to the following updating equation of the SSM-AP algorithm:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] + \frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{f}_\beta(\mathbf{w}(k)), \tag{4.20}$$

where we replace $\mathbf{f}_\beta(\mathbf{w}(k+1))$ with $\mathbf{f}_\beta(\mathbf{w}(k))$, defined in Subsection 4.1.2, in order to form the recursion and the term $\left(\mathbf{X}^T(k)\mathbf{X}(k)\right)^{-1}$ was replaced by $\mathbf{S}(k)$ to incorporate the regularization factor δ.

4.3.2 QSSM-AP Algorithm

In Eq. (4.20), the term $\mathbf{f}_\beta(\mathbf{w}(k))$, which is responsible for promoting sparsity, is left-multiplied by a $(N+1)\times(N+1)$ matrix whose rank is $N-L$. Therefore, such a matrix-vector multiplication restricts the influence of $\mathbf{f}_\beta(\mathbf{w}(k))$ to an $(N-L)$-dimensional space, thus losing $L+1$ degrees of freedom. In order to benefit from these degrees of freedom and also to reduce the computational complexity, the proposed QSSM-AP algorithm is characterized by the following recursion:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] - \frac{\alpha}{2}\mathbf{f}_\beta(\mathbf{w}(k)). \tag{4.21}$$

4.4 Properties of the Proposed Algorithms

In this section, we address some geometric characteristics of the proposed algorithms and prove two theorems regarding their stability, a property that is related to the choice of α.

4.4.1 Interpreting the gradient $\mathbf{f}_\beta$

From the discussion presented in Section 4.1, we know that $\mathbf{f}_\beta(\mathbf{w}(k))$ takes into account the distance from each component $w_n(k)$ to 0. This means that if we have $w_0(k) = w_1(k) = \cdots = w_N(k)$ (i.e., components equidistant to 0), then each $w_n(k)$ is pushed to 0 with equal strength. Hence, if we represent $-\mathbf{f}_\beta(\mathbf{w}(k))$ by an arrow with its tail at $\mathbf{w}(k)$, then such an arrow points to $\mathbf{0}$. On the other hand, if we have one component $w_n(k)$ closer to 0 than the others, then $w_n(k)$ will be forced to 0 with more strength. These aspects are illustrated in Figure 4.5 for the $\mathbb{R}^2$ space and two different $\mathbf{w}(k)$, one given by $[1/2\ 1/2]^T$ and the other by $[1/2\ 1/4]^T$. The red arrow represents the vector $-\alpha\mathbf{f}_5(\mathbf{w}(k))$ and, just to make such an example reproducible, we used the GMF presented in Subsection 4.1.2 and $\alpha = 1/5$. However, note that the

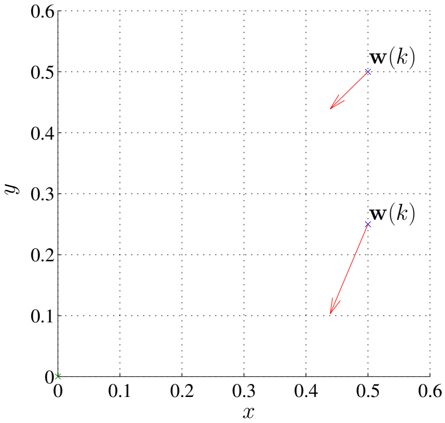


Figure 4.5: *Arrows illustrate vector* $-\alpha\mathbf{f}_\beta(\mathbf{w}(k)) \in \mathbb{R}^2$.

discussion presented is this subsection is not dependent on the choices of parameters and function F_β.

4.4.2 Updating process

Let us consider the SSM-AP algorithm and rewrite its updating process given in Eq. (4.20):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \underbrace{\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]}_{\triangleq \mathbf{p}_1} + \underbrace{\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{f}_\beta(\mathbf{w}(k))}_{\triangleq \mathbf{p}_2}. \tag{4.22}$$

It is easy to show that $\mathbf{p}_1$ is orthogonal to $\mathbf{p}_2$ (see Section B.1 of Appendix B). Hence, in order to generate $\mathbf{w}(k+1)$, the SSM-AP first applies the perturbation $\mathbf{p}_1$, which is exactly the same perturbation used by the SM-AP algorithm, that moves $\mathbf{w}(k)$ to the closest point lying on the intersection of the hyperplanes described by $\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k)$ [2, 52]. Then, $\mathbf{p}_2$ is applied yielding $\mathbf{w}(k+1)$ that still lies on the intersection of hyperplanes,[7] but with components $w_n(k+1)$ closer to zero than the components of $\mathbf{w}(k)$.

Now, let us examine the updating equation of the QSSM-AP algorithm given in Eq. (4.21):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \underbrace{\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]}_{\triangleq \mathbf{p}_1} \underbrace{- \frac{\alpha}{2}\mathbf{f}_\beta(\mathbf{w}(k))}_{\triangleq \mathbf{p}_3}. \tag{4.23}$$

The term $\mathbf{p}_1$ was explained in the previous paragraph, whereas the term $\mathbf{p}_3$, as

[7] Following the same reasoning shown in Section B.1 of Appendix B, one can prove that this $\mathbf{w}(k+1)$ satisfies the constraints $\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) = \boldsymbol{\gamma}(k)$.

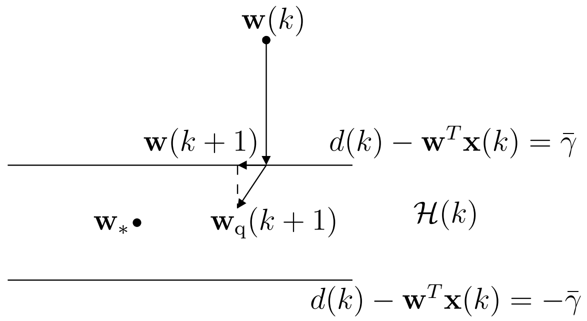

$\mathbf{w}(k)$
$\mathbf{w}(k+1)$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = \bar{\gamma}$
$\mathbf{w}_*$
$\mathbf{w}_\mathrm{q}(k+1)$
$\mathcal{H}(k)$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = -\bar{\gamma}$

Figure 4.6: *Updating process of SSM-AP and QSSM-AP algorithms for $L = 0$ (normalized LMS version) yielding $\mathbf{w}(k+1)$ and $\mathbf{w}_\mathrm{q}(k+1)$, respectively. It is assumed that $\mathbf{w}_*$, the filter modeling the unknown system, is sparse.*

explained in the previous subsection, aims at forcing the components of $\mathbf{w}(k)$ to 0. It is important to notice that, unlike the SSM-AP algorithm, $\mathbf{p}_3$ is not orthogonal to $\mathbf{p}_1$. Hence, the QSSM-AP algorithm generates $\mathbf{w}(k+1)$ that does not belong to the intersection of hyperplanes.

So far, we addressed the direction of $\mathbf{p}_2$ and $\mathbf{p}_3$, which are related to the updating process of the SSM-AP and QSSM-AP algorithms, respectively. Now let us analyze the length of these vectors, i.e., how strong the proposed algorithms push the components of $\mathbf{w}(k)$ to 0. It is not difficult to verify that $\mathbf{p}_2$ is the projection of $\mathbf{p}_3$ onto a space that is orthogonal to the column space of $\mathbf{X}(k)$. Therefore, we have $\|\mathbf{p}_2\|_2 \leq \|\mathbf{p}_3\|_2$.

Figure 4.6 illustrates the updating process of the SSM-AP and QSSM-AP for $L = 0$, in which $\mathbf{w}(k+1)$ and $\mathbf{w}_\mathrm{q}(k+1)$ are the updated parameter vectors of the SSM-AP and QSSM-AP algorithms, respectively, and $\mathbf{w}_*$ denotes the actual coefficients of the sparse unknown system. Observe that $\mathbf{p}_1$ is orthogonal to $\mathbf{p}_2$, which is the projection of $\mathbf{p}_3$ on the corresponding hyperplane.

4.4.3 Stability

Theorems 1 and 2 below guarantee the existence of α that yields estimates $\mathbf{w}(k+1)$ which are always closer to $\mathbf{w}_*$ than $\mathbf{w}(k)$ when the SSM-AP and QSSM-AP algorithms update their coefficient vectors.

Theorem 1. *For the SSM-AP algorithm, there exists α so that $\|\mathbf{w}_* - \mathbf{w}(k)\|_2$ is a monotonically nonincreasing sequence.*

Proof. Proof is left to Section B.2 of Appendix B. □

Theorem 2. *For the QSSM-AP algorithm, there exists α so that $\|\mathbf{w}_* - \mathbf{w}(k)\|_2$ is a monotonically nonincreasing sequence.*

Proof. Proof is left to Section B.3 of Appendix B. □

Sections B.2 and B.3 of Appendix B exhibit such values of α. Indeed, we found that $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ for the SSM-AP algorithm (Section B.2), whereas $\alpha \in [\alpha_{q,\min}, \alpha_{q,\max}]$ for the QSSM-AP algorithm (Section B.3). In addition, $\alpha = 0$ belongs to both intervals, i.e., $\alpha_{\min}, \alpha_{q,\min} < 0$ and $\alpha_{\max}, \alpha_{q,\max} > 0$.

Even though Sections B.2 and B.3 provide closed-form expressions for $\alpha_{\min}$, $\alpha_{\max}$, $\alpha_{q,\min}$, and $\alpha_{q,\max}$, these expressions are not practical for choosing α because they depend on $\mathbf{w}_*$, which is not known.[8] In addition, these bounds for α depend implicitly on k, i.e., they must be calculated at every iteration, which goes in the opposite way of reducing the number of operations required by the proposed algorithms. Hence, in practice, it is preferable to set α as a small nonnegative number that is *constant for all k*. The nonnegativity ensures that the algorithms are following a direction that promotes sparsity in the coefficient vector as shown in Figure 4.5, whereas α should be small enough to guarantee $\alpha < \alpha_{\max}$ (SSM-AP) or $\alpha < \alpha_{q,\max}$ (QSSM-AP) for all k.

4.5 Related Algorithms

In this section, we provide an overview of the subclass of online[9] adaptive filtering algorithms designed to exploit sparsity.

As mentioned before, the so-called *proportionate* family of algorithms finds widespread use in the context of sparse systems/signals. The first algorithm of this family is the PNLMS [41] whose updating rule is:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{G}(k) \frac{e(k)}{\mathbf{x}^T(k)\mathbf{G}(k)\mathbf{x}(k) + \delta} \mathbf{x}(k), \tag{4.24}$$

where $\mu \in \mathbb{R}_+$ is the step-size and $\mathbf{G}(k) \triangleq \mathrm{diag}\{[g_0(k) \ g_1(k) \ \ldots \ g_N(k)]^T\}$ is a diagonal matrix whose entry $g_n(k)$ is proportional to the magnitude of its associated filter coefficient $|w_n(k)|$.

Many algorithms based on Eq. (4.24) have been proposed in order to achieve a better model for sparsity, and they differ from each other just on the choice of $\mathbf{G}(k)$. For example, the choice of $\mathbf{G}(k)$ in the PNLMS algorithm makes it very sensitive to the sparsity degree. Indeed, PNLMS is very efficient when the sparsity degree is high, but its convergence becomes slower as the sparsity degree decreases. Since the sparsity degree usually varies in most applications, other algorithms that mitigate this issue have been proposed. Examples of algorithms include PNLMS++ [42], improved PNLMS (IPNLMS) [43], IPNLMS-l_0 [44], im-

[8] In fact, for the QSSM-AP algorithm, one can use the expressions in Section B.3 to safely set α as $\alpha = -b_q/2a_q$, where a_q and b_q can be computed at every iteration using (B.10) and (B.11).

[9] That is, algorithms that do not use block processing and, therefore, do not introduce latency.

proved $\mu-$law PNLMS (IMPNLMS) [45], among others [46] [47]. In addition, SM-PNLMS [48] [49] can be interpreted as a data-selective version of the IPNLMS algorithm.

Moreover, proportionate data-reuse algorithms, such as PAPA and improved PAPA (IPAPA) [50], were also proposed. In [49], the proposed set-membership PAPA (SM-PAPA) was shown to have faster convergence than the PAPA algorithm. The updating rule of the SM-PAPA generalizes almost all the PNLMS- and PAPA-based algorithms, depending on the proper set-up of the algorithm parameters, as shown in the following,

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{G}(k)\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{G}(k)\mathbf{X}(k) + \delta\mathbf{I}]^{-1}[\mathbf{e}(k) - \boldsymbol{\gamma}(k)] & \text{if } |e_0(k)| > \gamma, \\ \mathbf{w}(k) & \text{otherwise.} \end{cases} \quad (4.25)$$

With respect to those algorithms that include a penalty function accounting for the sparsity in the original objective function, the already explained ZA-APA and RZA-APA [40] (which are based on the l^1 norm), as well as the APA-SSI and QAPA-SSI [52] (which use a penalty function based on F_β) are remarkable examples. Here we recall their updating recursions for convenience. The updating rule employed by the ZA-APA is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) + \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{sign}\left(\mathbf{w}(k)\right), \quad (4.26)$$

where $\mathbf{sign}\left(\mathbf{w}(k)\right) \triangleq [g(w_0(k))\ g(w_1(k))\ \cdots\ g(w_N(k))]^T$ is the element-wise sign function and the definition of g is in Subsection 4.1.4. The RZA-APA uses the reweighted technique explained in Subsection 4.1.4 leading to the following recursion:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) + \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{P}\left(\mathbf{w}(k)\right), \quad (4.27)$$

where $\mathbf{P}\left(\mathbf{w}(k)\right) \triangleq [\hat{g}(w_0(k))\ \hat{g}(w_1(k))\ \cdots\ \hat{g}(w_N(k))]^T$ (see Subsection 4.1.4 for the definition of $\hat{g}$).

The updating equation corresponding to the APA-SSI is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) + \mu\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{f}_\beta\left(\mathbf{w}(k)\right), \quad (4.28)$$

whereas the updating process of QAPA-SSI is described by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{S}(k)\mathbf{e}(k) - \mu\frac{\alpha}{2}\mathbf{f}_\beta\left(\mathbf{w}(k)\right). \quad (4.29)$$

Both APA-SSI and QAPA-SSI with $\mu = 1$ are particular cases of the SSM-AP and

QSSM-AP algorithms, respectively.

4.6 Number of Operations

In this section, we determine the number of (arithmetic and bitwise) operations performed by the following algorithms: SSM-AP, QSSM-AP, and SM-PAPA.

Compared to traditional algorithms, set-membership (SM) algorithms present reduced computational burden due to data-selective updates. Indeed, for SM algorithms there are two types of iteration. The first corresponds to the iterations where no update is performed, which requires $C_{\rm nup} \in \mathbb{N}$ operations, while the second one accounts for the iterations where an update is performed, requiring $C_{\rm up} \in \mathbb{N}$ operations. Thus, denoting by $F_{\rm up} \in [0, 1]$ the fraction of the iterations in which updates occur, the average number of operations $C_{\rm av} \in \mathbb{R}_+$ per iteration is expressed by:

$$C_{\rm av} = F_{\rm up} C_{\rm up} + (1 - F_{\rm up}) C_{\rm nup}. \tag{4.30}$$

In addition, there are two important facts: (i) $C_{\rm nup} \ll C_{\rm up}$ and (ii) $F_{\rm up}$ is a small number, especially in stationary environments. These facts together imply that for most of the iterations the SM algorithms perform only $C_{\rm nup}$ operations, while non-SM algorithms perform $C_{\rm up}$ operations at every iteration.

In addition to being SM algorithms, the SSM-AP, QSSM-AP, and SM-PAPA can accelerate convergence and reduce the average number of operations even further when dealing with sparse signals. For these three algorithms, Table 4.1 presents the number of operations required in both types of iterations. This table discriminates each type of operation, which can be an arithmetic operation (addition, subtraction, multiplication, and division), a comparison operation (the “if” statement), and bitwise operations (sign and absolute value). Thus, one can compute $C_{\rm up}$ and $C_{\rm nup}$ for one of the algorithms by summing the values along a column of Table 4.1.

When calculating the number of operations we had to make some choices so that a fair comparison could be established. Hence, we are not claiming that the results in Table 4.1 represent the lowest number of operations. Some examples of choices we made are: (i) all quantities that do not vary with k (e.g., $\alpha/2$) are precomputed and stored in memory, requiring no operation; (ii) Gauss-Jordan elimination is used for the inversion of the $(L + 1) \times (L + 1)$ matrices; and (iii) multiplications by 0 or ± 1 (so-called trivial multiplications) are not taken into account. In addition, Table 4.1 does not take the computation of $\boldsymbol{\gamma}(k)$ into consideration. Thus, the use of the SC-CV requires 1 extra *sign* operation, whereas the ED-CV requires $L + 1$ extra *sign* operations. In addition to the aforementioned choices that affect all three algorithms, for the SSM-AP and QSSM-AP we consider that $\mathbf{f}_\beta(\mathbf{w}(k))$ follows

Table 4.1: *Number of operations for the SM-PAPA, SSM-AP, and QSSM-AP algorithms.*

	Update (C_{up})			No Update (C_{nup})
Operation \ Algorithm	SSM-AP	QSSM-AP	SM-PAPA	For all three algorithms
Addition	$(L^2+6L+6)N+$ $(2L^3+6L^2+7L+4)$	$(L^2+4L+4)N+$ $(2L^3+5L^2+5L+3)$	$\boldsymbol{N^2}+$ $(L^2+4L+4)N+$ $(2L^3+5L^2+5L+2)$	N
Subtraction	$\boldsymbol{N}+(2L+3)$	$\boldsymbol{N}+(2L+3)$	$\boldsymbol{N}+(2L+3)$	1
Multiplication	$\boldsymbol{(L^2+6L+9)N}+$ $(2L^3+7L^2+12L+11)$	$\boldsymbol{(L^2+4L+6)N}+$ $(2L^3+6L^2+8L+7)$	$\boldsymbol{(L^2+5L+7)N}+$ $(2L^3+6L^2+9L+8)$	$\boldsymbol{N}+$ 1
Division	$\boldsymbol{N}+(2L^2+4L+3)$	$\boldsymbol{N}+(2L^2+4L+3)$	$\boldsymbol{2N}+(2L^2+4L+4)$	0
Comparison	1	1	1	1
Bitwise	$2N+3$	$2N+3$	$\boldsymbol{N^2}+$ $2N+2$	1

Eq. (4.5c). In order to facilitate comparisons, since it is common practice to have $N \gg L$, the number of operations are represented by polynomials in N with the dominant terms per operation being highlighted in boldface.

From Table 4.1, we observe that the number of operations for iterations with *no update* is equal for all three algorithms, since they evaluate new data in the same manner. Therefore, we can focus on the iterations where an *update occurs*. In this case, the QSSM-AP algorithm presents the lowest number of operations for all kinds of operation, thus being the least computationally demanding algorithm among the three. Although the SSM-AP requires more multiplications than SM-PAPA, the N^2 additions and bitwise operations required by the SM-PAPA dominate the computational burden, making the SM-PAPA's update the most computationally demanding one.

4.7 Results

In this section we compare the performance of the proposed algorithms, viz. SSM-AP and QSSM-AP, with an extensive list of algorithms designed to exploit sparsity, mentioned in Section 4.5. The competing algorithms are: the ZA-APA and RZA-APA [40], the APA-SSI and QAPA-SSI [52], and the SM-PAPA [49] representing the proportionate family of algorithms.[10] For each of the SM algorithms we use both the SC-CV and the ED-CV and, whenever we mention an SM algorithm employing a specific constraint vector (CV) we abbreviate as "*algorithm (CV-type)*". Thus, SSM-AP (SC-CV) stands for the SSM-AP algorithm using the simple-choice constraint

[10] The SM-PAPA was chosen because: (i) most proportionate algorithms are special cases of the SM-PAPA, and (ii) it was shown in [49] that, in addition to reducing computational burden, the SM-PAPA also has better MSE performance than their non-SM special cases, such as the PAPA and IPNLMS algorithms.

vector (SC-CV). It is worth highlighting that the SM-PAPA (ED-CV) is actually a contribution of this thesis that has faster convergence speed, in comparison with the SM-PAPA (SC-CV) proposed in [49].

The algorithms are evaluated via three figures of merit: the percentage of updates, the MSE, and the normalized misalignment M which is defined as

$$\mathrm{M} \triangleq \frac{\|\mathbf{w}(k) - \mathbf{w}_*(k)\|_2^2}{\|\mathbf{w}_*(k)\|_2^2}, \tag{4.31}$$

where $\mathbf{w}_*(k)$ is the actual impulse response of the unknown system to be identified.

We consider $\mathbf{w}_*(k)$ representing both a sparse impulse response (Subsection 4.7.1) and a compressible impulse response (Subsection 4.7.2). In each case we perform different experiments to assess the robustness of the algorithms against the sparsity degree of $\mathbf{w}_*(k)$. Indeed, we use impulse responses that are very sparse (one or two "nonzero" coefficients), sparse ("nonzero" coefficients are equispaced), block-sparse ("nonzero" coefficients are adjacent), and dispersive (all coefficients are "nonzero"). The quotation marks in "nonzero" were used to address both the sparse and compressible cases, i.e., the cases where we do have coefficients equal to 0 and the cases where we have coefficients *close* to 0. Therefore, in the compressible case, "nonzero" coefficient actually means a coefficient whose magnitude is much greater than the magnitude of the coefficients which are close to 0. In addition, in order to investigate how fast the algorithms can readapt, we perform an abrupt change in $\mathbf{w}_*(k)$ at $k = 1000$.

4.7.1 Sparse Impulse Response

Scenario

We consider four experiments (Exp.) consisting of the identification of an unknown system $\mathbf{w}_*(k) = [w_{*,0}(k)\ w_{*,1}(k)\ \cdots\ w_{*,15}(k)]^T \in \mathbb{R}^{16}$ whose coefficients are set as:

- Exp. 1 (very sparse):
  - For $k < 10^3$: $w_{*,3}(k) = 1$ and $w_{*,n}(k) = 0$, for the other indices n;
  - For $k \geq 10^3$: $w_{*,3}(k) = w_{*,10}(k) = 1$ and $w_{*,n}(k) = 0$, for the other indices n.
- Exp. 2 (sparse):
  - For $k < 10^3$: $w_{*,n}(k) = 1$, for $n \in \{0, 4, 8, 12\}$, and $w_{*,n}(k) = 0$, for the other indices n;
  - For $k \geq 10^3$: $w_{*,n}(k) = 1$, for $n \in \{1, 5, 9, 13\}$, and $w_{*,n}(k) = 0$, for the other indices n.

- Exp. 3 (block-sparse):
  - For $k < 10^3$: $w_{*,n}(k) = 1$, for $n \in \{5, 6, 7, 8\}$, and $w_{*,n}(k) = 0$, for the other indices n;
  - For $k \geq 10^3$: $w_{*,n}(k) = 1$, for $n \in \{7, 8, 9, 10\}$, and $w_{*,n}(k) = 0$, for the other indices n.
- Exp. 4 (dispersive):
  - For $k < 10^3$: $w_{*,n}(k) = 1$, for $n \in \{0, 1, \ldots, 15\}$;
  - For $k \geq 10^3$: $w_{*,n}(k) = 2$, for $n \in \{0, 1, \ldots, 7\}$, and $w_{*,n}(k) = 1$, for $n \in \{8, 9, \ldots, 15\}$.

In order to identify the unknown system $\mathbf{w}_*(k)$, we use an adaptive filter $\mathbf{w}(k) \in \mathbb{R}^{16}$ that is initialized as $\mathbf{w}(0) = [0 \ \cdots \ 0]^T$. The reference signal $d(k)$ is assumed to be corrupted by an additive white Gaussian measurement noise with variance $\sigma_n^2 = 10^{-2}$.

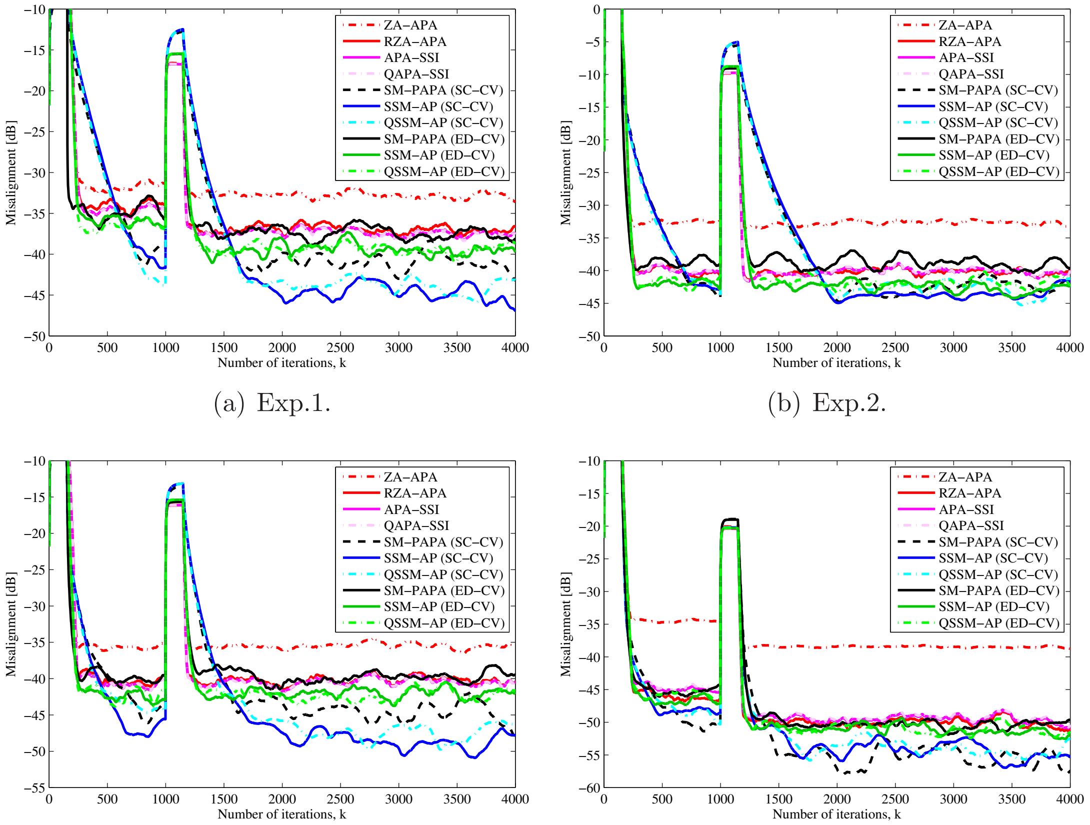
The input signal $x(k)$ is a real-valued random sequence generated as follows. First, a white Gaussian real-valued noise sequence $n_x(k)$, uncorrelated with $n(k)$, is filtered by an IIR filter defined by the following equation $x(k) = 0.95x(k-1) + 0.19x(k-2) + 0.09x(k-3) - 0.5x(k-4) + n_x(k)$, and then the variance of $x(k)$ is normalized to 1. This corresponds to a 4th order autoregressive (AR) process and $x(k)$ can be seen as a colored noise sequence. AR processes are very useful to model some signals found in practice, such as speech signals. The IIR filter used here is exactly the same used in [49].

The simulation results are generated as follows. We let each algorithm run during 4000 iterations and repeat this procedure 2000 times, forming the ensemble. Then, we perform an ensemble average to generate estimates of our figures of merit for each iteration k. The results in steady-state are generated by averaging these figures of merit over the last 1000 iterations.

Algorithm's parameters

The updating process of the adaptive filter is governed by one of the following algorithms: ZA-APA, RZA-APA, APA-SSI, QAPA-SSI, SM-PAPA (SC-CV), SM-PAPA (ED-CV), SSM-AP (SC-CV), SSM-AP (ED-CV), QSSM-AP (SC-CV), and QSSM-AP (ED-CV). Their updating rules can be found in Sections 4.3 and 4.5, whereas the constraint vectors are defined in Chapter 3.

These algorithms are set so that they yield similar convergence speed at the early iterations, except for the algorithms employing the SC-CV, which are naturally slower. The following parameters were used: step-size $\mu = 0.9$, regularization factor



(a) Exp.1.

(b) Exp.2.

(c) Exp.3.

(d) Exp.4.

Figure 4.7: *Misalignment evolution for different experiments involving sparse systems ($\beta = 5$).*

$\delta = 10^{-12}$, data-reuse factor $L = 4$, penalty-function weight $\alpha = 5 \times 10^{-3}$ [40, 52], $\beta = 5$ with F_β chosen as the GMF [52], $\overline{\gamma} = \sqrt{5\sigma_n^2}$ [30], and $\epsilon = 100$ for the RZA-APA [40]. Such choices agree with the recommended values for the competing algorithms.

Results

Figure 4.7 depicts the misalignment evolution for the ten aforementioned algorithms, whereas Table 4.2 summarizes their results (percentage of update, MSE, and misalignment) during steady-state.

Observing Figure 4.7, we can classify the algorithms into two categories according to the speed in which their related misalignments converge: the “slow” algorithms, which require more than 1000 iterations to converge, and the “fast” algorithms, which converge in less than 250 iterations. The category of “slow” algorithms is formed by the SM-PAPA (SC-CV), SSM-AP (SC-CV), and QSSM-AP (SC-CV), whereas the other seven algorithms belong to the group of “fast” algorithms. In what follows, we shall establish comparisons among algorithms within the same group before comparing algorithms of different groups.

Table 4.2: *Results during steady-state. The proposed algorithms are shadowed.*

Algorithms	Exp. 1			Exp. 2			Exp. 3			Exp. 4		
	Updates	MSE [dB]	M [dB]	Updates	MSE [dB]	M [dB]	Updates	MSE [dB]	M [dB]	Updates	MSE [dB]	M [dB]
ZA-APA	100%	-13.76	-32.90	100%	-13.69	-32.69	100%	-13.69	-35.48	100%	-13.31	-38.53
RZA-APA	100%	-13.40	-37.04	100%	-13.39	-40.25	100%	-13.39	-40.08	100%	-13.38	-50.04
APA-SSI	100%	-13.83	-37.53	100%	-13.76	-40.17	100%	-13.74	-40.34	100%	-13.36	-49.54
QAPA-SSI	100%	-13.83	-37.56	100%	-13.75	-40.10	100%	-13.74	-40.32	100%	-13.36	-49.40
SM-PAPA (SC-CV)	15.8%	-15.84	-40.65	16.0%	-15.78	-41.84	16.1%	-15.78	-44.29	17.0%	-15.57	-55.77
SSM-AP (SC-CV)	8.7%	-17.62	-44.89	9.6%	-17.37	-43.32	9.5%	-17.40	-48.87	17.1%	-15.54	-54.71
QSSM-AP (SC-CV)	9.1%	-17.50	-43.96	10.7%	-17.09	-42.71	10.5%	-17.13	-47.14	17.1%	-15.55	-54.56
SM-PAPA (ED-CV)	24.2%	-14.18	-37.73	24.6%	-14.10	-39.06	24.5%	-14.12	-39.58	25.1%	-14.00	-50.31
SSM-AP (ED-CV)	23.4%	-14.33	-39.64	23.7%	-14.27	-42.87	23.6%	-14.29	-41.77	25.1%	-14.01	-51.57
QSSM-AP (ED-CV)	23.2%	-14.35	-39.40	23.6%	-14.30	-41.64	23.5%	-14.30	-41.75	25.1%	-14.00	-51.05

Considering the group of "slow" algorithms, we can observe that the misalignment results of the SSM-AP (SC-CV) and QSSM-AP (SC-CV) algorithms are lower than the misalignment of the SM-PAPA (SC-CV) for the experiments in which the unknown system is sparse, i.e., experiments 1, 2, and 3. In addition, the proposed algorithms also achieved better steady-state results than SM-PAPA (SC-CV) in such experiments, as shown in Table 4.2. Indeed, they yielded a misalignment that is 1 to 4.5 dB lower, MSE around 2 dB lower, and they updated $\mathbf{w}(k)$ only a little more than a half the times that SM-PAPA (SC-CV) updated. In the dispersive case, on the other hand, these three algorithms yielded similar performance, with a small advantage for the SM-PAPA (SC-CV), see Exp. 4 in Table 4.2.

Considering the group of "fast" algorithms, we can see that the SSM-AP (ED-CV) and QSSM-AP (ED-CV) algorithms achieved the lowest misalignments for all the four experiments. In addition, these two proposed algorithms also achieved better steady-state results than the other five algorithms, as shown in Table 4.2. In this group of algorithms, the SM-PAPA (ED-CV) also deserves attention since its MSE and percentage of updates is almost as low as the ones of the two proposals, while its misalignment is about 1 to 4 dB higher.

Comparing the two best algorithms of each of the two categories, we observe that the steady-state results of the "slow" algorithms are better. On the other hand, the SSM-AP (ED-CV) and QSSM-AP (ED-CV) have the advantage of converging almost four times faster.

Finally, observing Figure 4.7 and Table 4.2 we notice that the results of the APA-SSI are superior to the results of the ZA-APA, which shows the practical benefits of using an approach based on approximating the l^0 norm over using the l^1 norm.[11] In addition, observe that the results corresponding to the RZA-APA are almost as good as the results of the APA-SSI, corroborating the connections between our approach based on F_β and the reweighted technique.

[11] Indeed, recall that the only difference between these two algorithms is that the ZA-APA uses l^1 norm, whereas the APA-SSI uses F_β.

4.7.2 Compressible Impulse Response

Scenario

In this subsection we consider almost the same scenario presented in Subsection 4.7.1. The only difference is that now our impulse response $\mathbf{w}_*$ is a compressible version of the $\mathbf{w}_*$ shown in Subsection 4.7.1. Indeed, here we redefine Exp. 1, Exp. 2, Exp. 3, and Exp. 4 so that every coefficient $w_{*,n}(k)$ that was equal to 0 is now replaced by a nonzero value that is drawn from a zero-mean Gaussian distribution with variance σ_c^2.

Algorithm's parameters

Here we consider the same choices of parameters shown in Subsection 4.7.1, but we will make β vary as σ_c^2 increases, as it will be explained.

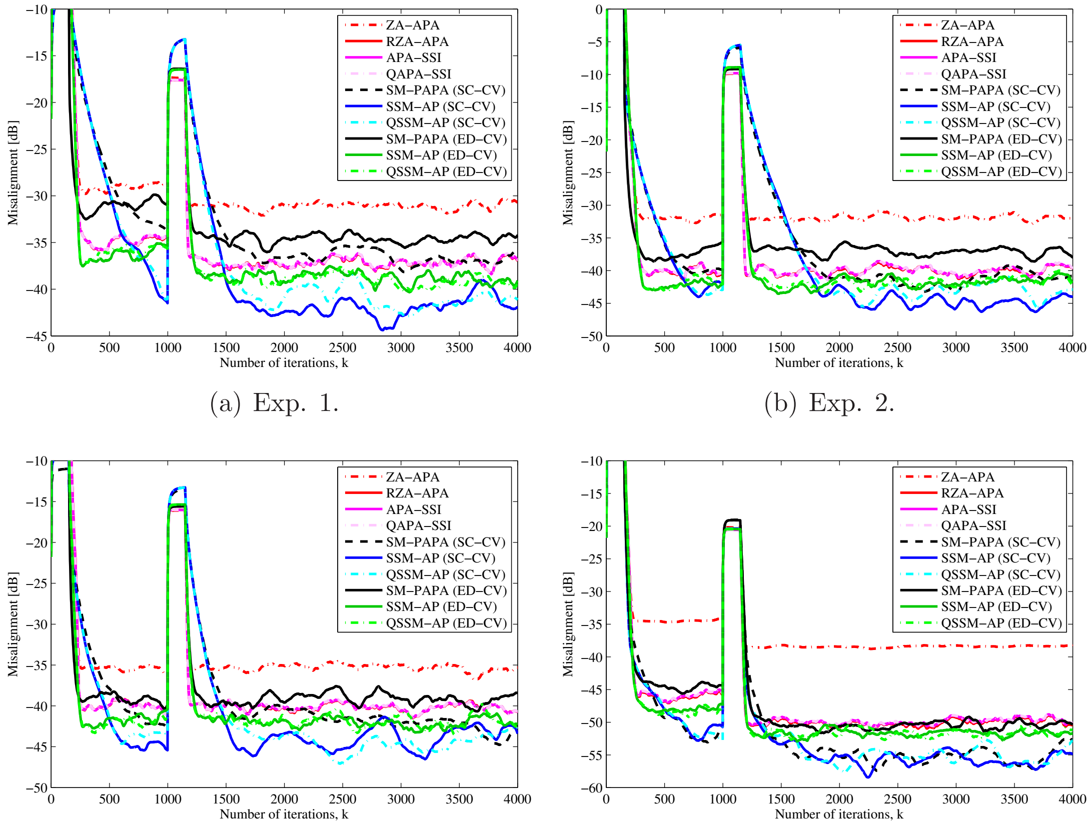
Results

Initially, for small values of σ_c^2 such as $\sigma_c^2 = 10^{-6}$, we used $\beta = 5$ (as in the sparse case) and obtained results similar to the ones presented in Subsection 4.7.1. We also observed that β should decrease as σ_c^2 increases, corroborating the comments in Subsection 4.1.3. For instance, using a high value such as $\sigma_c^2 = 10^{-2}$, we used $\beta = 0.0625$ in order to obtain the results shown in Figure 4.8. Once again we observe that there is a group of "slow" algorithms, comprised by the three algorithms using the SC-CV, while the remaining seven algorithms belong to the group of "fast" algorithms. In summary, within each group, the proposed algorithms achieved lower misalignments than the competing algorithms. The steady-state MSE and percentage of updates are not mentioned here because they are very similar for the SM-PAPA, SSM-AP, and QSSM-AP algorithms employing the same CV. Note that, similarly to the sparse scenario, the APA-SSI also achieved better results than the ZA-APA in the compressible case.

4.7.3 Additional Remarks

We have tested the proposed algorithms in many other scenarios which are not included in this chapter. Thus, in this subsection we summarize some of our observations.

Firstly, we have always observed the two clusters: "fast" and "slow" algorithms. Considering the group of "fast" algorithms, the SSM-AP (ED-CV) and QSSM-AP (ED-CV) have always exhibited very similar performances, which were better than the competing algorithms. In addition, the number of operations required by QSSM-AP is much lower than for the other algorithms. For instance, considering Exp. 1 in



(a) Exp. 1.

(b) Exp. 2.

(c) Exp. 3.

(d) Exp. 4.

Figure 4.8: *Misalignment evolution for different experiments involving compressible systems ($\beta = 0.0625; \sigma_c^2 = 10^{-2}$).*

Subsection 4.7.1, the average number of operations per iteration of the QSSM-AP (ED-CV) is $C_{\text{av}} = 0.232 \times N_{up} + (1 - 0.232) \times N_{nup} \approx 447.1$, which is almost 30% less than for the SM-PAPA (ED-CV) and about 85% less than for the traditional AP algorithm with the same data reuse factor ($L = 4$). Therefore, the QSSM-AP (ED-CV) has the best cost-effectiveness among the group of "fast" algorithms.

The group of "slow" algorithms, which employ the SC-CV, are much more sensitive to the choice of β and to the sparsity degree of the unknown system than the algorithms using the ED-CV. For instance, Table 4.2 shows that the figures of merit of the algorithms using the ED-CV are more robust to the sparsity degree than the algorithms using the SC-CV. This characteristic facilitates the specification of a hardware.

4.8 Conclusion

In this chapter, we proposed two algorithms that can be regarded as generalizations of the SM-AP algorithm to exploit the sparsity of the involved signals in order to increase the accuracy of the estimates or accelerate convergence speed. Thus, such algorithms rely on two independent parts: (i) models to reveal/promote sparsity and (ii) the SM-AP algorithm, which combines the data selection and data reuse properties. In part (i), we compare and establish connections between the approach we used, which is based on approximations to the l^0 norm, and the approaches based on l^0 and l^1 norms in order to motivate our choice. We also show that our approach is more effective than the other based on the l^1 norm considering both sparse and compressible signals. In part (ii), we use the results of the SM-AP algorithm, which were presented in the previous chapter. In addition, we analyze the proposed algorithms in order to properly set their parameters. Indeed, we show some geometric properties of the proposed algorithms as well as a proof of stability for each of them. Finally, the number of operations performed by the proposed algorithms is addressed and simulation results show that the proposed algorithms outperform the state-of-the-art algorithms designed to exploit sparsity.

Chapter 5

Frequency-Domain Data-Selective Algorithms

Frequency-domain adaptive filters are able to reduce the computational complexity of adaptive filters, since convolutions in the time domain are exchanged by scalar multiplications in the frequency domain [10], and to increase convergence speed [2]. In addition, there is a computationally efficient algorithm that converts time-domain representation to frequency-domain representation: the fast Fourier transform (FFT) algorithm.

Therefore, an adaptive filter can dramatically decrease its complexity for high values of M by operating in the frequency domain. Indeed, for high values of M, the complexity of time-domain adaptive filters may turn their use prohibitive. On the other hand, working in the frequency domain has a major drawback: latency due to block processing. This is the reason why frequency-domain adaptive filters are not as widely used as time-domain filters.

In this chapter we address frequency-domain adaptive filters which also have the data-selection property. After introducing the frequency-domain version of the set-membership filtering concept, we tackle two applications: channel equalization and acoustic echo cancellation.

In summary, the contributions of this chapter are:

1. We propose a semi-blind equalization algorithm which can increase the throughput of communications systems employing the orthogonal frequency-division multiplexing (OFDM) as their core modulation scheme (Section 5.2).

2. We propose a data-selective algorithm employing psychoacoustics criteria for acoustic echo cancellation (Section 5.3). The main idea of our proposal is to eliminate the residual signal/echo only in the cases when a person would be capable of perceiving them.

The content of this chapter was partially published in [66, 67].

This chapter is organized as follows. Section 5.1 presents the frequency-domain set-membership filtering (F-SMF) concept and introduces the frequency-domain SM-NLMS (F-SM-NLMS) algorithm. Sections 5.2 and 5.3 address applications of the F-SMF concept where the F-SM-NLMS is used as the basic structure for the algorithms proposed in these two sections. In Section 5.2, we present a semi-blind equalization algorithm for OFDM-based systems, whereas a perception-based acoustic echo cancellation was proposed in Section 5.3. Conclusions are drawn in Section 5.4.

Notation: The complex field is represented by $\mathbb{C}$. The length of the adaptive filter is denoted as M, thus M can be interpreted as the number of frequency bins. In addition to the previously established notation, we represent the discrete Fourier transform (DFT) matrix as $\mathbf{F} \in \mathbb{C}^{M\times M}$, in which its entry on the mth row and nth column is given by

$$[\mathbf{F}]_{m,n} \triangleq \frac{1}{\sqrt{M}} \mathrm{e}^{-\jmath \frac{2\pi}{M} mn}, \text{ for } m, n = 0, \ldots, M-1, \tag{5.1}$$

where $\jmath$ is the imaginary unit so that $\jmath^2 = -1$ and $\mathrm{e} \approx 2.718$ is the Euler's number. The inverse DFT (IDFT) is represented by $\mathbf{F}^H$, where the superscript ^{H} denotes Hermitian transposition. Operators $\circ$ and $\div$ denote element-wise multiplication and division of two vectors, respectively.

5.1 Frequency-domain Set-Membership Filtering (F-SMF)

In this section we present the frequency-domain SMF (F-SMF) proposed in [28]. This concept uses a block-based processing and is applicable to adaptive filtering problems that are linear-in-parameters and which involve circular convolutions.

Figure 5.1 depicts a general frequency-domain filtering. In this figure, we have the following time-domain vectors: $\mathbf{x}, \mathbf{d}, \mathbf{y}, \mathbf{e} \in \mathbb{C}^M$, which represent the input vector, the desired vector, the output vector, and the error vector, respectively. The frequency-domain vectors corresponding to $\mathbf{x}, \mathbf{d}, \mathbf{y}, \mathbf{e}$ are $\bar{\mathbf{x}}, \bar{\mathbf{d}}, \bar{\mathbf{y}}, \bar{\mathbf{e}} \in \mathbb{C}^M$. Thus, superscript $\bar{(\cdot)}$ denotes the frequency-domain representation of vector $(\cdot)$. In addition, the adaptive filter coefficients are represented by vector $\bar{\mathbf{w}} \in \mathbb{C}^M$.

As shown in Figure 5.1, the input vector $\mathbf{x}$ is first mapped to $\bar{\mathbf{x}} \triangleq \mathbf{F}\mathbf{x}$ through an M-point FFT, which is then filtered by $\bar{\mathbf{w}}$ yielding the output $\bar{\mathbf{y}} \triangleq \bar{\mathbf{w}} \circ \bar{\mathbf{x}}$. Next, this output is transformed back to the time-domain leading to $\mathbf{y} \triangleq \mathbf{F}^H\bar{\mathbf{y}}$. Then, the error vector $\mathbf{e} \triangleq \mathbf{d} - \mathbf{y}$ is computed, transformed to the frequency-domain generating $\bar{\mathbf{e}} \triangleq \mathbf{F}\mathbf{e} = \mathbf{F}\mathbf{d} - \mathbf{F}\mathbf{y} = \bar{\mathbf{d}} - \bar{\mathbf{y}}$, and fed back to the adaptive filter.

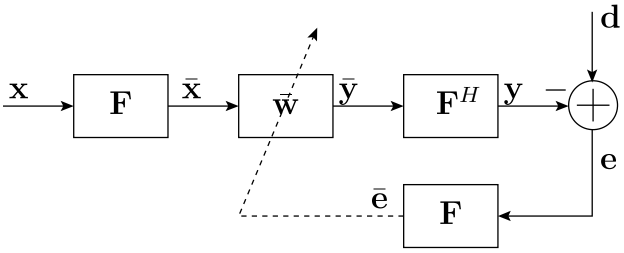


Figure 5.1: *General frequency-domain filtering.*

Denoting by $\mathcal{S}$ the set comprised of all possible pairs $(\mathbf{x}, \mathbf{d})$ and representing its frequency-domain version by $\bar{\mathcal{S}}$, i.e., $\bar{\mathcal{S}}$ is the set containing all possible pairs $(\bar{\mathbf{x}}, \bar{\mathbf{d}})$, the F-SMF criterion aims at finding $\bar{\mathbf{w}}$ such that $\|\mathbf{e}\|^2 \leq \overline{\gamma}^2$, for all $(\mathbf{x}, \mathbf{d}) \in \mathcal{S}$, where $\overline{\gamma} \in \mathbb{R}_+$ is an upper bound representing the amount of error that is acceptable. Since $\mathbf{F}$ is a unitary matrix so that $\|\bar{\mathbf{e}}\|^2 = \|\mathbf{F}\mathbf{e}\|^2 = \mathbf{e}^H\mathbf{F}^H\mathbf{F}\mathbf{e} = \mathbf{e}^H\mathbf{e} = \|\mathbf{e}\|^2$, then we can restate the F-SMF criterion as finding $\bar{\mathbf{w}}$ satisfying:

$$\|\bar{\mathbf{e}}\|^2 \leq \overline{\gamma}^2, \text{ for all } (\bar{\mathbf{x}}, \bar{\mathbf{d}}) \in \bar{\mathcal{S}}. \tag{5.2}$$

Similarly to the feasibility set of the SMF concept, we can define its frequency-domain counterpart as

$$\bar{\Theta} \triangleq \bigcap_{(\bar{\mathbf{x}}, \bar{\mathbf{d}}) \in \bar{\mathcal{S}}} \left\{ \bar{\mathbf{w}} \in \mathbb{C}^M : \|\bar{\mathbf{d}} - \bar{\mathbf{w}} \circ \bar{\mathbf{x}}\|^2 \leq \overline{\gamma}^2 \right\}, \tag{5.3}$$

where $\bar{\Theta}$ is denominated as frequency-domain feasibility set.

In order to iteratively estimate $\bar{\Theta}$ or a point in it, we define the frequency-domain constraint set as

$$\bar{\mathcal{H}}(k) \triangleq \left\{ \bar{\mathbf{w}} \in \mathbb{C}^M : \|\bar{\mathbf{d}}(k) - \bar{\mathbf{w}} \circ \bar{\mathbf{x}}(k)\|^2 \leq \overline{\gamma}^2 \right\}. \tag{5.4}$$

Hence, $\bar{\mathcal{H}}(k)$ represents the set of adaptive filter coefficients $\bar{\mathbf{w}}$ whose squared norm of the error is upper bounded by a constant $\overline{\gamma}^2$ considering the kth data pair $(\bar{\mathbf{x}}(k), \bar{\mathbf{d}}(k))$. In addition, we can also define the frequency-domain exact membership set as

$$\bar{\Psi}_0^k \triangleq \bigcap_{k_1=0}^{k} \bar{\mathcal{H}}(k_1). \tag{5.5}$$

Clearly, $\lim\limits_{k \longrightarrow \infty} \bar{\Psi}_0^k = \bar{\Theta}$.

5.1.1 Adaptive F-SMF Algorithms

One of the main advantages of subband adaptive filtering is the possibility of treating each subband independently. In this chapter we are considering a particular version of subband adaptive filtering, which is known as frequency-domain adaptive filtering, and we would also like to treat each frequency bin independently. However, the definition of $\bar{\mathcal{H}}(k)$ in (5.4) hinders our attempt to decouple the coefficients of $\bar{\mathbf{w}}$ and $\bar{\mathbf{x}}(k)$ due to the existence of a single parameter $\overline{\gamma}$ for all frequency bins, i.e., we have a single prescribed scalar threshold.

Such an issue can be circumvented by replacing the scalar threshold $\overline{\gamma}$ with a vector threshold $\bar{\boldsymbol{\gamma}} \in \mathbb{R}_+^M$ defined as $\bar{\boldsymbol{\gamma}} \triangleq \left[\overline{\gamma}_0 \ \overline{\gamma}_1 \ \dots \ \overline{\gamma}_{M-1}\right]^T$, where the upper bound associated with the mth frequency bin is $\overline{\gamma}_m \in \mathbb{R}_+$, for $m = 0, 1, \dots, M-1$. Then, we can define the *modified constraint set* as

$$\bar{\mathcal{H}}_{\text{mod}}(k) \triangleq \bigcap_{m=0}^{M-1} \bar{\mathcal{H}}_m(k) \tag{5.6}$$

where

$$\bar{\mathcal{H}}_m(k) \triangleq \left\{\bar{\mathbf{w}} \in \mathbb{C}^M : |\bar{d}_m(k) - \bar{w}_m \bar{x}_m(k)|^2 \leq \overline{\gamma}_m^2\right\}. \tag{5.7}$$

That is, $\bar{\mathcal{H}}_m(k)$ is comprised of the coefficients $\bar{\mathbf{w}}$ that lead to an error in the mth frequency bin $e_m(k)$ such that $|e_m(k)|^2 \leq \overline{\gamma}_m^2$, considering the kth data pair/block. In addition, $\bar{\mathcal{H}}_{\text{mod}}(k)$ is the intersection of $\bar{\mathcal{H}}_m(k)$ for all frequency bins, i.e., for $m = 0, 1, \dots, M-1$.

In addition, the entries of $\bar{\boldsymbol{\gamma}}$ should be defined so that $\bar{\boldsymbol{\gamma}}^T\bar{\boldsymbol{\gamma}} = \overline{\gamma}^2$. Observe that with such a definition we have $\bar{\mathcal{H}}_{\text{mod}}(k) \subset \bar{\mathcal{H}}(k)$, which means that if we find $\bar{\mathbf{w}} \in \bar{\mathcal{H}}_{\text{mod}}(k)$, then we have $\bar{\mathbf{w}} \in \bar{\mathcal{H}}(k)$, but the converse is not true in general (except when $\overline{\gamma} = 0$).

5.1.2 F-SM-NLMS Algorithm

Here we define the frequency-domain counterpart of the SM-NLMS algorithm, namely F-SM-NLMS algorithm [28].

The objective function is analogous to that of the SM-NLMS algorithm. Indeed, the goal is to find the new coefficient vector $\bar{\mathbf{w}}(k+1)$ that solves the following optimization problem

$$\begin{aligned} &\text{minimize } \|\bar{\mathbf{w}}(k+1) - \bar{\mathbf{w}}(k)\|^2 \\ &\text{subject to } \bar{\mathbf{w}}(k+1) \in \bar{\mathcal{H}}_{\text{mod}}(k). \end{aligned} \tag{5.8}$$

From the definition of $\bar{\mathcal{H}}_{\text{mod}}(k)$ in (5.6), and observing that the property that defines the set $\bar{\mathcal{H}}_{m_1}(k)$ is independent from the property that defines $\bar{\mathcal{H}}_{m_2}(k)$ for $m_1 \neq m_2$, we can decouple and solve such an optimization problem in the following way. If a component of $\bar{\mathbf{w}}(k)$, say $\bar{w}_m(k)$, already belongs to $\bar{\mathcal{H}}_m(k)$, then no update is performed and $\bar{w}_m(k+1) = \bar{w}_m(k)$. Otherwise, $\bar{w}_m(k+1)$ should be generated as the projection of $\bar{w}_m(k)$ on $\bar{\mathcal{H}}_m(k)$. Thus, the F-SM-NLMS is characterized by the following recursion:

$$\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \bar{\boldsymbol{\mu}}(k) \circ \bar{\mathbf{e}}(k) \div \bar{\mathbf{x}}(k), \tag{5.9}$$

where $\bar{\boldsymbol{\mu}}(k) = [\bar{\mu}_0(k)\ \bar{\mu}_1(k)\ \ldots\ \bar{\mu}_{M-1}(k)]^T \in \mathbb{R}_+^M$ is the vector containing the step sizes corresponding to each frequency bin, which are defined as

$$\bar{\mu}_m(k) \triangleq \begin{cases} 1 - \frac{\overline{\gamma}_m}{|\bar{e}_m(k)|} & \text{if } |\bar{e}_m(k)|^2 > \overline{\gamma}_m^2, \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

Observe that if $\bar{w}_m(k) \notin \bar{\mathcal{H}}_m(k)$, then the error in the mth frequency bin after the update of the coefficient vector (a posteriori error) is given by

$$\begin{aligned} |\bar{d}_m(k) - \bar{w}_m(k+1)\bar{x}_m(k)|^2 &= \left| \bar{d}_m(k) - \left[\bar{w}_m(k) + \bar{\mu}_m(k) \frac{\bar{e}_m(k)}{\bar{x}_m(k)} \right] \bar{x}_m(k) \right|^2 \\ &= \left| \bar{d}_m(k) - \bar{w}_m(k)\bar{x}_m(k) - \bar{\mu}_m(k)\bar{e}_m(k) \right|^2 \\ &= |\bar{e}_m(k) - \bar{\mu}_m(k)\bar{e}_m(k)|^2 \\ &= \left| \bar{e}_m(k) - \left(1 - \frac{\overline{\gamma}_m}{|\bar{e}_m(k)|} \right) \bar{e}_m(k) \right|^2 = \overline{\gamma}_m^2, \end{aligned} \tag{5.11}$$

i.e., $\bar{w}_m(k+1) \in \bar{\mathcal{H}}_m(k)$.

It is worth mentioning that the recursion in (5.9) is very sensitive to $\bar{\mathbf{x}}(k)$, especially when one of its entries approaches 0. There are many ways of mitigating this issue. Here, we use a regularization $1 \gg \delta \in \mathbb{R}_+$ and vector $\mathbf{u} \triangleq [1\ \ldots\ 1]^T \in \mathbb{R}^M$, and the F-SM-NLMS recursion that we actually implement is

$$\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \bar{\boldsymbol{\mu}}(k) \circ \bar{\mathbf{e}}(k) \div (\bar{\mathbf{x}}(k) + \delta\mathbf{u}). \tag{5.12}$$

5.2 Frequency-Domain Data-Selective Algorithm for Semi-Blind Equalization of OFDM-based transmissions

In this section, we present a data-selective algorithm that operates in the frequency domain and is suitable to semi-blind equalization of communications systems based on the orthogonal frequency-division multiplexing (OFDM) employing the quadrature phase-shift keying (QPSK) as digital modulation. The proposed algorithm combines the standard equalizers used in OFDM with a data-selection algorithm based on the F-SM-NLMS presented in Subsection 5.1.2. Its data-selection scheme utilizes an idea that we proposed in [68].

In this section, a modest background in OFDM systems is assumed. Such a background can be found in Chapter 2 of [4].

5.2.1 Equalizers and OFDM-based Systems

Equalizers are widely used in communications and their task is to reverse the harmful effects of the channel on the transmitted signal. Basically, equalizers can be of the following three types: supervised, blind, and semi-blind.

In a *supervised equalization*, the equalizer is adjusted based on a reference or desired signal, which is known as *pilot* signal in the communication literature. These pilots correspond to data that are already known at the receiver side so that its transmission carries no information. Indeed, these pilots are used at the receiver to perform channel estimation, and this estimate is then used in the equalization processing of subsequent transmissions. Since the channel is usually time-varying, supervised equalizers require periodic transmissions of pilots so that it has a somewhat accurate information about the channel state. However, sending pilots decreases the system throughput since no new information is actually transmitted.

Unlike supervised equalizers, *blind equalization* (also called unsupervised equalization) algorithms do not use pilots and, therefore, they can be employed to increase the system throughput. In addition, these algorithms allow tracking of channel variations, which is a useful feature since real channels are usually time-varying, especially the wireless channel. On the other hand, blind algorithms usually require a proper initialization and can lose track in nonstationary environments, deteriorating the channel equalization process [2].

Algorithms for *semi-blind equalization* [29, 68] represent a trade-off between the gains of using blind equalization techniques and supervised (i.e., pilot-based) schemes. Indeed, such algorithms sometimes operate with pilot signals and sometimes operate without them. Therefore, semi-blind equalizers allow the communication system to send pilots less often, increasing the system throughput while keeping the channel equalization effective.

The majority of current communications systems use supervised equalization due to its improved results. By using pilot signals, the receiver is able to estimate the channel state, which is then used to set the equalizer coefficients properly, i.e., the equalizer is set so that it can reverse the channel effects on the next transmitted signals, assuming that the channel is time-invariant or slowly varying. However, as already explained, channels are time-varying and, as a consequence, these communications systems have to send pilots periodically, thus wasting radio resources.

Therefore, our aim is to develop an algorithm that can track channel variations even when no pilot is transmitted, i.e., we use the equalizer obtained when pilots are transmitted as the initial estimate (supervised part) and then we track small channel variations without using pilots (blind part) until a new pilot symbol is transmitted and the equalizer is re-initialized, as will be explained in Section 5.2.2. By doing so, we expect that pilots can be sent less frequently leading to an increase in system throughput.

We have chosen to work with OFDM-based systems due to many reasons. OFDM is the modulation scheme of some important communications standards such as the Wi-Fi [69], WiMAX [70], and the LTE (Long Term Evolution) [71]. OFDM already employs block processing so that latency is acceptable. In addition, by considering the standard implementation of OFDM, which employs a cyclic prefix, and also assuming that the length of such a prefix is properly set, then the mathematical model corresponding to an OFDM transmission is completely decoupled [4], as will be explained in the next subsection.

In addition, just to give an example of a current standard, the LTE transmits an entire OFDM symbol (data block) containing only pilots once at each 6 or 7 transmitted OFDM symbols, depending on the size of the cyclic prefix [71, 72].

5.2.2 Semi-Blind Equalization Scheme

Figure 5.2 illustrates a frequency-domain block representation of an OFDM-based system connected to the proposed semi-blind equalization (SBE) algorithm. In this figure, we considered perfect synchronization and that the cyclic-prefix length is greater than the channel memory so that we have a decoupled transmission, i.e., the symbols within a given OFDM symbol (block) do not interfere with each other. See [4] for further details on the mathematical modeling of OFDM systems.

In this section all vectors are in $\mathbb{C}^M$, where $M \in \mathbb{N}$ represents the number of subcarriers of the OFDM system. This number corresponds to the length of every vector in Figure 5.2. The kth transmitted OFDM symbol is denoted as $\bar{\mathbf{s}}(k)$, whose

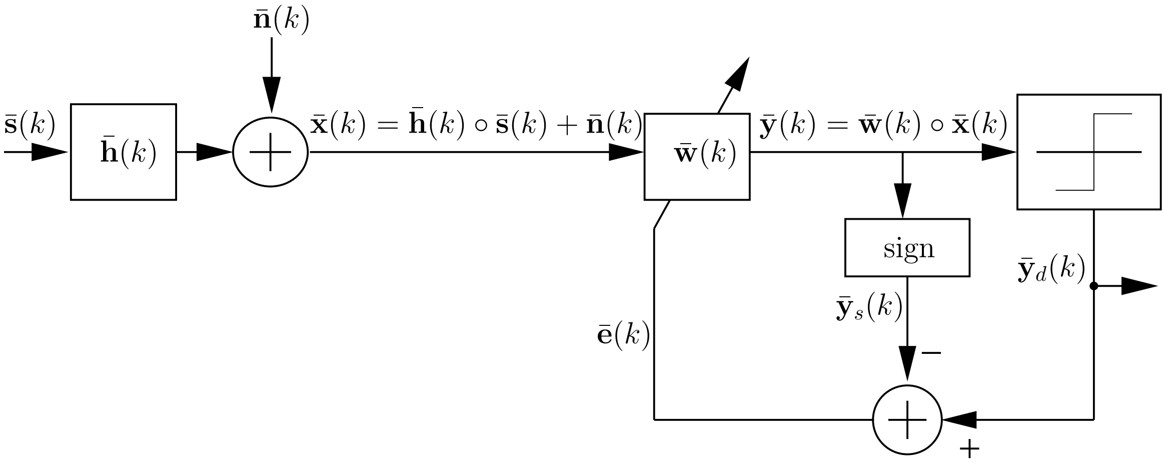


Figure 5.2: *Frequency-domain representation of an OFDM transmission using the proposed semi-blind equalization scheme at the receiver. The blocks perform element-wise operations.*

entries are symbols belonging to a QPSK modulation. The M-point FFT of the channel impulse response is represented by $\bar{\mathbf{h}}(k)$, and $\bar{\mathbf{n}}(k)$ is the additive white Gaussian noise (AWGN) vector with variance σ_n^2.

The vector $\bar{\mathbf{x}}(k)$ corresponds to the received version of the OFDM symbol $\bar{\mathbf{s}}(k)$, and they are related by:

$$\bar{\mathbf{x}}(k) = \bar{\mathbf{h}}(k) \circ \bar{\mathbf{s}}(k) + \bar{\mathbf{n}}(k), \tag{5.13}$$

i.e., a component of the received signal $\bar{\mathbf{x}}(k)$, say $\bar{x}_m(k)$ with $m = 0, 1, \ldots, M-1$, depends only on the mth symbol of the OFDM symbol $\bar{\mathbf{s}}(k)$, say $\bar{s}_m(k)$, and on the mth frequency bin of $\bar{\mathbf{h}}(k)$, say $\bar{h}_m(k)$.

The received vector $\bar{\mathbf{x}}(k)$ is used as the input of the adaptive filter $\bar{\mathbf{w}}(k)$ so that the output vector is $\bar{\mathbf{y}}(k) = \bar{\mathbf{w}}(k) \circ \bar{\mathbf{x}}(k)$, which contains the equalized received symbols. In addition, $\bar{\mathbf{y}}_d(k) = \text{QPSK decision}\{\bar{\mathbf{y}}(k)\}$ contains the QPSK symbols which are most likely to had been transmitted, whereas $\bar{\mathbf{y}}_s(k) = \text{sign}\{\mathbf{y}(k)\}$ contains the equalized received symbols projected on the unit circle, i.e., each of its entries corresponds to a projection of an entry of $\bar{\mathbf{y}}(k)$ (a complex number) on the unit-radius circle. The main idea of our proposal is to compensate only phase variations, since the magnitude is not important in the QPSK demodulation. The error vector, defined as $\bar{\mathbf{e}}(k) = \bar{\mathbf{y}}_d(k) - \bar{\mathbf{y}}_s(k)$, determines whenever an update in the adaptive filter $\bar{\mathbf{w}}(k)$ is needed.

The SBE algorithm is summarized in Table 5.1. Its updating scheme is based on the F-SM-NLMS algorithm, but its error signal $\bar{\mathbf{e}}(k)$ is defined in a different way since the desired signal is not available. The idea is very intuitive. We assume that $\bar{\mathbf{y}}_d(k)$ is correct, i.e., no errors were made in the QPSK hard decision, and we use $\bar{\mathbf{y}}_d(k)$ to replace the desired signal. In addition, since we are using a QPSK modulation, then

Table 5.1: *Semi-Blind Equalization (SBE) Algorithm.*

SBE Algorithm

```
Initialization
{
  Choose $\delta$ as a small constant
  Choose $\bar{\boldsymbol{\gamma}}$ (usually as a function of $\sigma_n^2$)
}
For $k = 1$:(Number of OFDM symbols) do
{
  If pilot was sent, then
  {
    Do channel estimation
    Initialize the equalizer, e.g., ZF equalizer: $\bar{\mathbf{w}}(k+1) = \mathbf{u} \div \big(\text{Estimated version of } \bar{\mathbf{h}}(k)\big)$
  }
  Else
  {
    $\bar{\mathbf{y}}(k) = \bar{\mathbf{w}}(k) \circ \bar{\mathbf{x}}(k)$
    $\bar{\mathbf{y}}_{\rm s}(k) = \text{sign}\{\bar{\mathbf{y}}(k)\}$
    $\bar{\mathbf{y}}_{\rm d}(k) = \text{QPSK decision}\{\bar{\mathbf{y}}(k)\}$
    $\bar{\mathbf{e}}(k) = \bar{\mathbf{y}}_{\rm d}(k) - \bar{\mathbf{y}}_{\rm s}(k)$
    For $m = 0 : M-1$ (i.e., for all subcarriers) do
    {
      $\bar{\mu}_m(k) = \begin{cases} \left(1 - \frac{\overline{\gamma}_m}{|\bar{e}_m(k)|}\right) & \text{if } |\bar{e}_m(k)| > \overline{\gamma}_m, \\ 0 & \text{otherwise.} \end{cases}$
    }
    $\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \bar{\boldsymbol{\mu}}(k) \circ [\bar{\mathbf{e}}(k) \div (\bar{\mathbf{x}}(k) + \delta\mathbf{u})]$
  }
    $\bar{\mathbf{w}}(k+1) = \dfrac{\bar{\mathbf{w}}(k+1)}{|\bar{\mathbf{w}}(k+1)|}$
}
```

the amplitude of the received signals is not important in the decision process. More specifically, we should design an equalizer that tracks the phase of the time-varying channel, whereas the modulus of the channel is not important. Therefore, we use $\bar{\mathbf{y}}_s(k)$ because both $\bar{\mathbf{y}}_s(k)$ and $\bar{\mathbf{y}}_d(k)$ have the same magnitude so that the error signal $\bar{\mathbf{e}}(k)$ can focus on the phase distortions introduced by the channel. In addition, the adaptive filter coefficients are re-initialized every time a pilot OFDM symbol is transmitted. Whenever this occurs, the receiver performs channel estimation and uses such an estimate to set $\bar{\mathbf{w}}(k)$. In Table 5.1, for example, the equalizer $\bar{\mathbf{w}}(k)$ is set according to a zero forcing (ZF) criterion [4].

The vector $\bar{\boldsymbol{\gamma}} \in \mathbb{R}_+^M$ contains decision thresholds for the errors associated with each subcarrier, i.e., the algorithm updates only if the modulus of the mth entry of

$\bar{\mathbf{e}}(k)$, viz. $|\bar{e}_m(k)|$, is greater than the mth entry of $\bar{\boldsymbol{\gamma}}$, viz. $\bar{\gamma}_m$. The vector $\bar{\boldsymbol{\mu}}(k)$ contains the step sizes for each of the filter coefficients and $\mathbf{u} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^M$. In addition, the regularization parameter δ must be chosen as a small constant used to prevent numerical instabilities due to small entries of $\bar{\mathbf{x}}(k)$. Most of these variables were already addressed in Subsection 5.1.2.

5.2.3 Results

Here, it is assumed that the length of the cyclic prefix is appropriately chosen in such a way that the transmission scheme in equation (5.13) is valid. The simulation setup follows closely the configuration of the physical-layer downlink connection of the LTE system [71, 73]. Each OFDM symbol is comprised of $M = 512$ subcarriers, each subcarrier corresponding to 15 kHz carries one symbol belonging to a QPSK constellation whose symbols are normalized to have unity variance.[1] The channel is time varying. For the first OFDM symbol of each simulation, the channel $\mathbf{h}(0)$ is generated according to the Extended Typical Urban-LTE (ETU-LTE) model [72]. The channels observed by the other OFDM symbols were generated following a *random-walk model* [2] applied only to the phase of the frequency response of the channel, i.e., given that $\bar{\mathbf{h}}(k) = |\bar{\mathbf{h}}(k)|e^{j\bar{\boldsymbol{\theta}}(k)}$, then $\bar{\mathbf{h}}(k+1)$ is generated as

$$\bar{\mathbf{h}}(k+1) = |\bar{\mathbf{h}}(k)|e^{j\bar{\boldsymbol{\theta}}(k+1)}, \tag{5.14}$$

$$\bar{\boldsymbol{\theta}}(k+1) = \lambda_c\bar{\boldsymbol{\theta}}(k) + \kappa\mathbf{n}_c(k), \tag{5.15}$$

where $\bar{\mathbf{h}}(k)$ is the frequency response of the channel during the kth OFDM symbol, $|\bar{\mathbf{h}}(k)|$ and $\bar{\boldsymbol{\theta}}(k)$ are vectors containing the magnitude and the phase components of $\bar{\mathbf{h}}(k)$, respectively. The other parameters are $\lambda_c = 0.99$, $\kappa = (1-\lambda_c)^{p/2}$, $p = 2$, and $\mathbf{n}_c(k)$ is a random vector drawn from a zero-mean Gaussian distribution with standard deviation $\sigma_{n_c} = 5$.

In addition, the regularization factor was chosen as $\delta = 10^{-12}$ and the upper bounds for the error in each subcarrier were set as $\bar{\gamma}_m = \sqrt{\tau\sigma_n^2}$, for $m = 0, 1, \dots, M-1$, with $\tau = 1$ and σ_n^2 is the variance of the additive white Gaussian measurement noise, which is used to control the signal-to-noise ratio (SNR). The bit-error rate (BER) results were generated considering 10^4 transmitted OFDM symbols and then averaging the results over 100 independent simulations.

Figure 5.3 depicts BER results for different values of SNR considering a channel that varies at every OFDM symbol. The magnitude of the channel frequency response is constant for each simulation and it was generated in such a way that its minimum value is 0.5. This particularization on the channel is justified by our

[1] The symbols must have unity variance so that $\bar{\mathbf{y}}_d$ (decided) and $\bar{\mathbf{y}}_s$ (projected on the unity-radius circle) have the same magnitude.

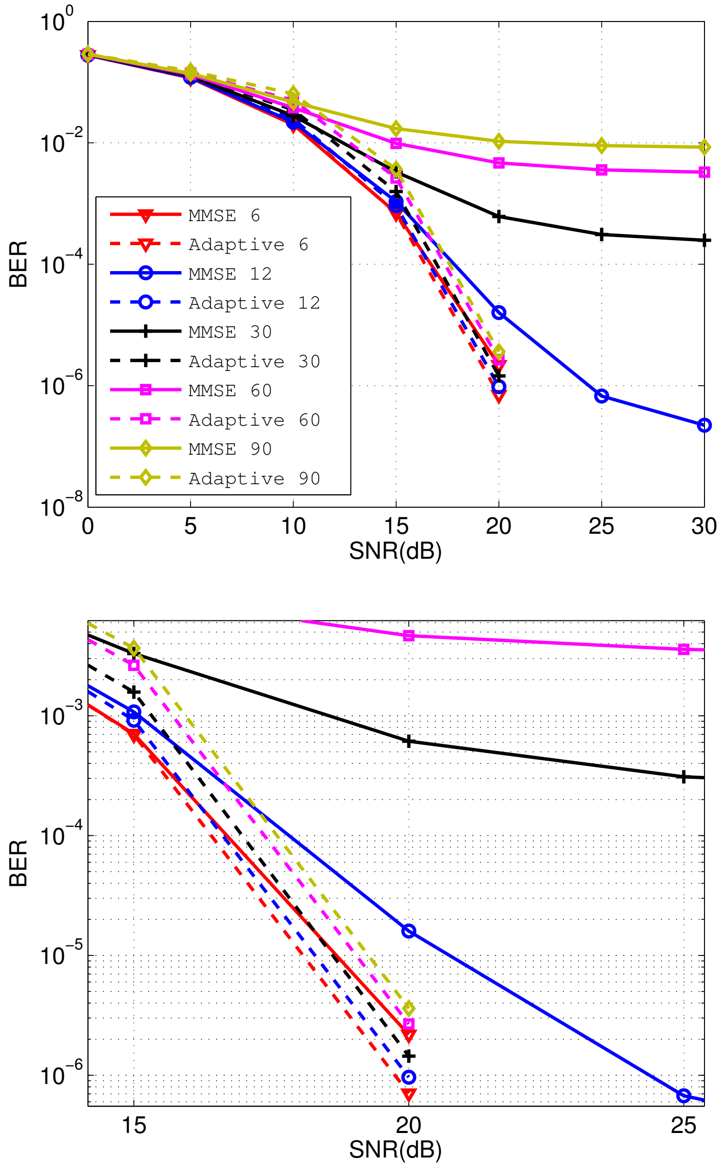


Figure 5.3: *BER versus SNR. In the first plot (top) we consider $|h_m(k)| > 0.5$ for all m (subcarriers) and k (OFDM symbols), whereas the second plot (bottom) is just a zoom of the previous one.*

SBE process, which is very sensitive to the *SNR at a given subcarrier*. Indeed, from the transmission scheme in (5.13) we observe that the symbol received at the mth subcarrier is given by $\bar{x}_m(k) = \bar{h}_m(k)\bar{s}_m(k) + \bar{n}_m(k)$, and the *SNR at this subcarrier* is given by $|\bar{h}_m(k)\bar{s}_m(k)|^2/|\bar{n}_m(k)|^2$, and thus, for low values of $|\bar{h}_m(k)|^2$ we might have a signal power much lower than the noise power at the receiver.

In Figure 5.3, there are two types of equalizers: the MMSE equalizer and the Adaptive equalizer. The MMSE equalizer is a supervised equalizer that uses the pilot-OFDM symbol to set $\bar{\mathbf{w}}(k)$ at a given iteration/block k and uses the same $\bar{\mathbf{w}}(k)$ to equalize the next OFDM-symbols transmitted until a new pilot-OFDM symbol is transmitted. The Adaptive equalizer uses the proposed SBE algorithm, i.e., it employs the MMSE equalizer when a pilot-OFDM symbol is transmitted and it tracks the channel variations when pilot-OFDM symbols are not transmitted. In addition, the number next to the type of equalizer corresponds to the frequency (how often) with which a pilot OFDM-symbol is transmitted. For instance, “MMSE 6” stands for the MMSE equalizer in which a pilot-OFDM symbol is transmitted once at every 6 OFDM-symbols, which corresponds to one of the actual configurations of the LTE system.

Figure 5.3 shows that the SBE algorithm can dramatically increase the system throughput provided a minimum SNR is guaranteed. For instance, at an SNR of 20 dB, the proposed Adaptive equalizer that transmits a pilot-OFDM symbol once at every 60 OFDM symbols achieves almost the same BER as the one achieved by the traditional OFDM retransmitting pilot-OFDM symbols at each 6 OFDM symbols. In this case, the usage of the SBE technique would yield an increase of 18% in the throughput as compared to the traditional OFDM transmission.

5.2.4 Final Remarks

In this section, we proposed a semi-blind equalization algorithm that can significantly improve the throughput of OFDM-based systems employing QPSK modulation. However, the proposed algorithm still lacks a mechanism to sense the magnitude of the channel at a given subcarrier $\bar{h}_m(k)$. Indeed, as previously explained, when $\bar{h}_m(k)$ is very small, the symbol $\bar{s}_m(k)$ is strongly attenuated so that it is better not to update our adaptive filter component $\bar{w}_m(k)$.

In addition, it is worth mentioning that there are other types of block transceivers that have the same transmission model of (5.13), such as the single-carrier with frequency-division equalization (SC-FD) [4]. The proposed semi-blind equalization algorithm is also applicable to such systems and that explains why we are using the term “OFDM-based systems” rather than just “OFDM systems”.

5.3 Perception-Based Acoustic Echo Cancellation

Perceptual signal processing is a paradigm within the general field of signal processing whose target is aligned with the goal of the so-called green technologies. Indeed, by processing only perceptible signals we gain some degrees of freedom which can be exploited in order to increase the efficiency with which the computational resources are used, thus enabling a reduction of energy consumption.

In acoustic echo cancellation systems, for example, there is room for further improvements if we take into account psychoacoustics criteria. Echo cancellers mitigate the acoustic echoes produced by a loudspeaker-enclosure-microphone (LEM) system by using an adaptive filter whose target is to estimate the impulse response of the LEM system [74, 75]. Nevertheless, such an approach does not exploit the fact that some distortions introduced by the LEM system may be inaudible. Hence, instead of searching for a unique adaptive filter whose coefficients match perfectly the LEM's impulse response, we can actually look for any adaptive filter that reduces the related echoes/distortions to levels at which one cannot perceive them anymore. This way, the set estimation framework is more appropriate than the traditional point estimation.

In this section, a frequency-domain set-membership filtering algorithm for acoustic echo cancellation is proposed. This algorithm takes psychoacoustics effects into account in order to update the filter coefficients only in the cases where the residual echo becomes audible.

5.3.1 Echo Cancellation Setup

Acoustic echoes appear when there exists an acoustic coupling between loudspeaker(s) and microphone(s), e.g., in hands-free communications and teleconference systems. Indeed, in such cases, the signal emitted by a loudspeaker reflects in some objects and walls and reaches a microphone, forming the loudspeaker-enclosure-microphone (LEM) system.

The state-of-the-art solution to this problem is to use an acoustic echo canceller based on simple adaptive filters, such as the NLMS and BNLMS [74, 75]. The inconvenience is that such an application usually requires long-length adaptive filters and, consequently, the computational complexity required by these algorithms may turn their use prohibitive. In addition, as mentioned in Chapter 4, the convergence speed of such algorithms deteriorate as the filter order increases.

On the other hand, frequency-domain adaptive filters have lower computational complexity and higher convergence speed, as compared to their time-domain counterparts. However, since they utilize block processing, then latency is always introduced.

The reader should refer to [74] for more details about echo cancellations systems.

5.3.2 Set-Membership Constrained Frequency-Domain Algorithm

The F-SMF concept presented in Section 5.1 uses frequency-domain processing so that it requires M-point FFT/IFFT modules, whereas the computation of the adaptive filter output requires only a multiplication between two scalar variables per frequency bin, totaling M scalar multiplications. At a given block k, the output signal represented in the frequency domain $\bar{\mathbf{y}}(k)$ corresponds to the multiplication of the DFT representations of the input signal $\mathbf{x}(k)$ and the impulse response of the adaptive filter $\mathbf{w}(k)$, i.e., $\bar{\mathbf{y}}(k)$ corresponds to the element-wise multiplication of two DFT representations, viz. $\bar{\mathbf{x}}(k)$ and $\bar{\mathbf{w}}(k)$. From the digital signal processing (DSP) literature, we know that the multiplication of two length-M DFT representations associated with two length-M signals correspond to the circular convolution between such signals in time domain [10]. Thus, the F-SMF concept is suitable to applications involving circular convolutions, i.e., it assumes that the desired vector $\mathbf{d}(k)$ originates from a circular convolution.

For applications involving linear convolutions, such as echo cancellation, we need to adapt the F-SMF concept. Indeed, in order to emulate a linear convolution using a circular convolution, we can employ one of the following widely-used approaches: the *overlap-and-add* and the *overlap-and-save* [10]. Such approaches have already been used in the adaptive filtering literature, see [76] and the books [2, 74].

Here, we follow the overlap-and-save approach and we use the constrained frequency-domain (CFD) algorithm, see Chapter 12 of [2], as the basis for the proposed algorithm. In addition, we incorporate a data-selection scheme following the same reasoning used when the F-SM-NLMS algorithm was introduced. This combination leads to the proposed algorithm, which we call *set-membership constrained frequency-domain* (SM-CFD). The SM-CFD algorithm is capable of taking psychoacoustics effects into account, as will be explained in the next subsection.

The proposed algorithm still follows the configuration depicted in Figure 5.1, but with some minor changes due to the overlap-and-save, as we now explain. First, instead of having a single block $\mathbf{x}(k) \in \mathbb{C}^M$, we have an augmented vector $\mathbf{x}_{\text{aug}}(k) \in \mathbb{C}^{2M}$ defined as

$$\mathbf{x}_{\text{aug}}(k) \triangleq \begin{bmatrix} \mathbf{I}_M \\ z^{-1}\mathbf{I}_M \end{bmatrix} \mathbf{x}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k-1) \end{bmatrix}, \tag{5.16}$$

i.e., $\mathbf{x}_{\text{aug}}(k)$ is the concatenation of the current block $\mathbf{x}(k)$ with the previous block $\mathbf{x}(k-1)$.[2] Then, a $(2M)$-point FFT is applied to $\mathbf{x}_{\text{aug}}(k)$ yielding $\bar{\mathbf{x}}(k) = \mathbf{F}_{2M}\mathbf{x}_{\text{aug}}(k) \in \mathbb{C}^{2M}$. Next, the adaptive filter $\bar{\mathbf{w}}_c(k) \in \mathbb{C}^{2M}$ is applied to $\bar{\mathbf{x}}(k)$ generating as output $\bar{\mathbf{y}}(k) = \bar{\mathbf{w}}_c(k) \circ \bar{\mathbf{x}}(k) \in \mathbb{C}^{2M}$. This output is transformed back to the time domain, but only half of its entries actually correspond to the linear convolution. Therefore, we generate the augmented output signal $\mathbf{y}_{\text{aug}}(k) \in \mathbb{C}^{2M}$ by applying a $(2M)$-point IFFT to $\bar{\mathbf{y}}(k)$, and then we discard the undesired entries

[2] The subscript of a matrix indicates its size. For example, $\mathbf{I}_M$ stands for the $M \times M$ identity matrix, whereas $\mathbf{F}_{2M}$ represents the $(2M) \times (2M)$ DFT matrix.

generating $\mathbf{y}(k) \in \mathbb{C}^M$ as

$$\begin{aligned}\mathbf{y}(k) &\triangleq [\mathbf{I}_M \;\; \mathbf{0}_M]\, \mathbf{y}_{\text{aug}}(k) \\ &= [\mathbf{I}_M \;\; \mathbf{0}_M]\, \mathbf{F}_{2M}^H \bar{\mathbf{y}}(k). \end{aligned} \tag{5.17}$$

Then, the error signal is computed as $\mathbf{e}(k) \triangleq \mathbf{d}(k) - \mathbf{y}(k) \in \mathbb{C}^M$. Since we are interested in updating the coefficients $\bar{w}_{c,m}(k)$ for $m = 0, 1, \ldots, 2M-1$, i.e., for each frequency bin, then we must take a $(2M)$-point FFT of the error signal $\mathbf{e}(k)$, leading to $\bar{\mathbf{e}}(k) \in \mathbb{C}^{2M}$ defined as

$$\begin{aligned}\bar{\mathbf{e}}(k) &\triangleq \mathbf{F}_{2M} \;\mathbf{e}_{\text{aug}}(k) \\ &= \mathbf{F}_{2M} \begin{bmatrix} \mathbf{I}_M \\ \mathbf{0}_M \end{bmatrix} \mathbf{e}(k). \end{aligned} \tag{5.18}$$

This $\bar{\mathbf{e}}(k)$ is fed back to the adaptive filter and used in its updating process.

In addition, the updating scheme of the SM-CFD algorithm is very similar to the F-SM-NLMS algorithm. In the SM-CFD algorithm we have $\bar{\mathbf{w}}_c(k) \in \mathbb{C}^{2M}$, which is a vector containing the adaptive filter coefficients in the frequency domain, and an auxiliary vector $\bar{\mathbf{w}}(k) \in \mathbb{C}^{2M}$, which is the vector that we actually update. The updating rule of $\bar{\mathbf{w}}(k)$ is given by

$$\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \boldsymbol{\Lambda}(k)\boldsymbol{\Sigma}^{-2}(k)\text{diag}\left(\bar{\mathbf{e}}(k)\right)\bar{\mathbf{x}}^*(k), \tag{5.19}$$

where $\boldsymbol{\Lambda}(k), \boldsymbol{\Sigma}^{-2}(k), \text{diag}\left(\bar{\mathbf{e}}(k)\right) \in \mathbb{C}^{2M \times 2M}$ are diagonal matrices. $\boldsymbol{\Lambda}(k)$ is a matrix containing the step sizes corresponding to each frequency bin, i.e.,

$$\boldsymbol{\Lambda}(k) \triangleq \text{diag}\left(\bar{\boldsymbol{\mu}}(k)\right), \tag{5.20}$$

with $\bar{\boldsymbol{\mu}}(k) = [\bar{\mu}_0(k) \;\; \bar{\mu}_1(k) \;\; \cdots \;\; \bar{\mu}_{2M-1}(k)]^T$. The matrix $\boldsymbol{\Sigma}^2(k)$ is defined as

$$\boldsymbol{\Sigma}^2(k) \triangleq \text{diag}\left(\boldsymbol{\sigma}^2(k)\right), \tag{5.21}$$

where $\boldsymbol{\sigma}^2(k) = \left[\delta + \sigma_0^2(k) \;\; \cdots \;\; \delta + \sigma_{2M-1}^2(k)\right]^T$, and $\sigma_m^2(k) = (1-\alpha)\sigma_m^2(k-1) + \alpha|\bar{x}_m(k)|^2$, for $m = 0, 1, \ldots, 2M-1$.

Using (5.20) and (5.21), one can rewrite (5.19) as

$$\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \bar{\boldsymbol{\mu}}(k) \circ \bar{\mathbf{e}}(k) \circ \left(\bar{\mathbf{x}}^*(k) \div \boldsymbol{\sigma}^2(k)\right) \tag{5.22}$$

$$\approx \bar{\mathbf{w}}(k) + \bar{\boldsymbol{\mu}}(k) \circ \bar{\mathbf{e}}(k) \div \bar{\mathbf{x}}(k). \tag{5.23}$$

It is straightforward to obtain (5.22) from (5.19). In addition, by considering $\delta \longrightarrow 0$ and $\alpha \longrightarrow 1$, i.e., by eliminating two parameters that are introduced to increase the

Table 5.2: *Set-Membership Constrained Frequency-Domain (SM-CFD) Algorithm.*

```
SM-CFD Algorithm

Initialization
{
  Choose $\delta$ as a small constant
  Choose $\eta$ such that $0 < \eta \leq 1$
  Choose $\alpha$ such that $0 < \alpha \leq 0.1$
  Compute $K$ (number of blocks of length $M$ to be processed)
  $\mathbf{x}(0) = \mathbf{0}$
  $\sigma_m^2(0) = 0$, for $m = 0, 1, \cdots, 2M-1$
}
For $k = 1 : K$ (i.e., for each block of length $M$) do
{
  $\bar{\mathbf{x}}(k) = \mathbf{F}_{2M}\ \mathbf{x}_{\text{aug}}(k) = \mathbf{F}_{2M}\begin{bmatrix}\mathbf{I}_M\\ z^{-1}\mathbf{I}_M\end{bmatrix}\mathbf{x}(k) = \mathbf{F}_{2M}\begin{bmatrix}\mathbf{x}(k)\\ \mathbf{x}(k-1)\end{bmatrix}$
  $\mathbf{y}(k) = [\mathbf{I}_M\ \ \mathbf{0}_M]\,\mathbf{F}_{2M}^H\bar{\mathbf{y}}(k) = [\mathbf{I}_M\ \ \mathbf{0}_M]\,\mathbf{F}_{2M}^H\begin{bmatrix}\bar{w}_{c,0}(k)\bar{x}_0(k)\\ \bar{w}_{c,1}(k)\bar{x}_1(k)\\ \vdots\\ \bar{w}_{c,2M-1}(k)\bar{x}_{2M-1}(k)\end{bmatrix}$
  $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{y}(k)$
  $\bar{\mathbf{e}}(k) = \mathbf{F}_{2M}\ \mathbf{e}_{\text{aug}}(k) = \mathbf{F}_{2M}\begin{bmatrix}\mathbf{I}_M\\ \mathbf{0}_M\end{bmatrix}\mathbf{e}(k)$
  For $m = 0 : (2M-1)$ (i.e., for each frequency bin) do
  {
    Compute $\bar{\Gamma}_m$ and $\Gamma_m^\dagger$ according to the desired data-selection mechanism (to be explained)
    $\bar{\mu}_m(k) = \begin{cases}\eta\left(1 - \dfrac{\Gamma_m^\dagger}{|\bar{e}_m(k)|}\right) & \text{if } |\bar{e}_m(k)| > \bar{\Gamma}_m,\\ 0 & \text{otherwise.}\end{cases}$
    $\sigma_m^2(k) = (1-\alpha)\sigma_m^2(k-1) + \alpha|\bar{x}_m(k)|^2$
  }
  $\bar{\boldsymbol{\mu}}(k) = [\bar{\mu}_0(k)\ \ \bar{\mu}_1(k)\ \ \cdots\ \ \bar{\mu}_{2M-1}(k)]^T$
  $\boldsymbol{\Lambda}(k) = \text{diag}\left(\bar{\boldsymbol{\mu}}(k)\right)$
  $\boldsymbol{\sigma}^2(k) = \left[\delta + \sigma_0^2(k)\ \ \cdots\ \ \delta + \sigma_{2M-1}^2(k)\right]^T$
  $\boldsymbol{\Sigma}^2(k) = \text{diag}\left(\boldsymbol{\sigma}^2(k)\right)$
  $\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) + \boldsymbol{\Lambda}(k)\boldsymbol{\Sigma}^{-2}(k)\text{diag}\left(\bar{\mathbf{e}}(k)\right)\bar{\mathbf{x}}^*(k)$
  $\bar{\mathbf{w}}_c(k+1) = \mathbf{F}_{2M}\begin{bmatrix}\mathbf{I}_M\\ \mathbf{0}_M\end{bmatrix}[\mathbf{I}_M\ \ \mathbf{0}_M]\,\mathbf{F}_{2M}^H\bar{\mathbf{w}}(k+1)$
}
```

robustness of the CFD algorithm against small entries of $\bar{\mathbf{x}}(k)$, then we obtain (5.23). Observe that (5.23) is essentially the same updating equation given in (5.9).

Finally, since we are interested in performing linear convolutions using FFTs, then we should constrain our adaptive filter generating $\bar{\mathbf{w}}_c(k+1)$ as follows [2]:

$$\bar{\mathbf{w}}_c(k+1) = \mathbf{F}_{2M}\begin{bmatrix}\mathbf{I}_M\\ \mathbf{0}_M\end{bmatrix}[\mathbf{I}_M\ \ \mathbf{0}_M]\,\mathbf{F}_{2M}^H\bar{\mathbf{w}}(k+1). \tag{5.24}$$

Table 5.2 summarizes the proposed SM-CFD algorithm. In this algorithm, δ is a regularization factor used to avoid ill-conditioning of the matrix $\mathbf{\Sigma}^{-2}(k)$, $(1-\alpha)$ can be seen as a forgetting factor for the variance $\sigma_m^2(k)$ of $\bar{x}_m(k)$, $\mathbf{I}_M$ and $\mathbf{0}_M$ stand for the identity and all-zero matrices of size M. The step size relative to the mth frequency bin is denoted as $\bar{\mu}_m(k)$.

In addition, $\bar{\Gamma}_m$ represents the upper bound in which the magnitude of the error corresponding to the mth bin, namely $|\bar{e}_m(k)|$, is acceptable, i.e., no update is performed for errors lower than $\bar{\Gamma}_m$. The threshold $\Gamma_m^\dagger$, usually chosen as a function of the noise standard deviation, is related to the level of residual error that will be left after update (*a posteriori* error).

The possible choices for the data-selection mechanisms $\bar{\Gamma}_m$ and $\Gamma_m^\dagger$ are addressed in the following section.

5.3.3 Data-Selection Mechanisms

Here, we present three different data-selection mechanisms for the proposed SM-CFD algorithm, viz. the *constant bound*, the *perceptual bound*, and the *hybrid bound*.

Constant Bound

For the constant bound, we have $\bar{\Gamma}_m = \Gamma_m^\dagger$, for $m = 0, 1, \ldots, 2M-1$, and $\bar{\Gamma}_m$ should be chosen as a function of the noise variance σ_n^2. Thus, like in time-domain SM algorithms, we choose $\bar{\Gamma}_m = \sqrt{\tau\sigma_n^2}$, where $\tau \in \mathbb{R}_+$. Observe that this bound is constant for all frequency bins within a block. We denominate the SM-CFD algorithm using the constant bound as C-SM-CFD algorithm.

Perceptual Bound

For the perceptual bound we also have $\bar{\Gamma}_m = \Gamma_m^\dagger$, for $m = 0, 1, \ldots, 2M-1$. Unlike the constant bound, the perceptual bound may use different thresholds for different frequency bins, i.e., we can have $\bar{\Gamma}_m \neq \Gamma_l$ for $k \neq l$. The entries $\bar{\Gamma}_m$ are chosen based on psychoacoustics criteria, such as *masking* and *just-noticeable changes* [77]. Figure 5.4 depicts the perceptual model used as well as the phenomena that are considered. The inputs of the perceptual model are the input vector $\mathbf{x}(k)$ and desired vector $\mathbf{d}(k)$, both represented in time domain. These signals are transformed to the frequency domain, then they are weighted with a transfer function that corresponds to the human ear (the so-called loudness curve). Next, these signals are transformed to a domain in which frequency masking can be taken into consideration. Finally, these signals are compared in a bin-by-bin basis, where the resulting bins related to the input signal form the bound $\bar{\Gamma}_m$. The perceptual model in Figure 5.4, is a simplified version of the psychoacoustics models used in some codecs and perceptual

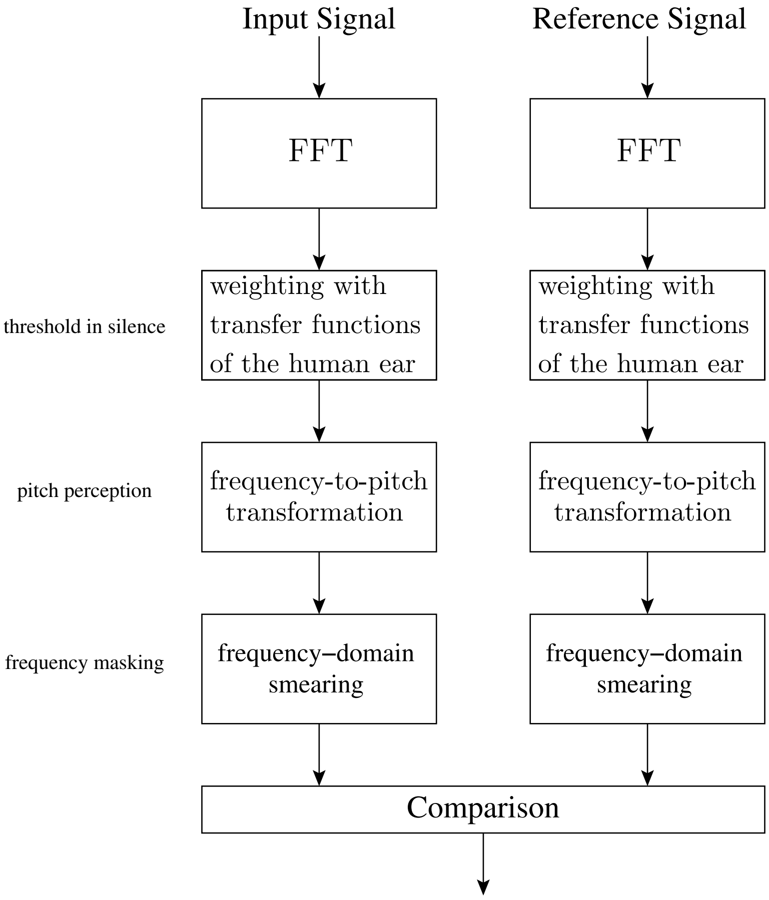

Input Signal
Reference Signal
FFT
FFT
threshold in silence
weighting with transfer functions of the human ear
weighting with transfer functions of the human ear
pitch perception
frequency-to-pitch transformation
frequency-to-pitch transformation
frequency masking
frequency–domain smearing
frequency–domain smearing
Comparison

Figure 5.4: *Perceptual model.*

evaluators, such as the perceptual evaluation of speech quality (PESQ). The SM-CFD algorithm employing the perceptual bound is denoted as P-SM-CFD algorithm.

Hybrid Bound

The hybrid bound exploits the best of the two bounds previously mentioned. Thus, the hybrid bound uses two distinct bounds, i.e., we have $\bar{\Gamma}_m \neq \Gamma^\dagger_m$, for $m = 0, 1, \ldots, 2M - 1$, which are chosen as:

1. $\bar{\Gamma}_m$ is chosen as the perceptual bound in order to have a reduced number of updates;

2. $\Gamma^\dagger_m$ is chosen as the constant bound in order to accelerate convergence of the proposed algorithm.

The SM-CFD algorithm employing the hybrid bound is denoted as H-SM-CFD algorithm.

5.3.4 Results

Scenario Description

In order to evaluate the quality of the echo cancellers, a set of 24 signals was prepared, each of them containing two distinct phonetic balanced sentences [78] with a silence gap of 300 ms between them as recommended in [79]. The sentences are in Brazilian Portuguese and were recorded by native people (2 males and 1 female), resulting in a database of 8 signals recorded by each person. The duration of the signals ranges from 4 to 7 seconds.

Since the duration of the signals is insufficient to achieve the convergence for all tested adaptive filtering algorithms, each of the 24 signals was repeated 32 times inserting a gap of 500 ms of silence between repetitions. This procedure generated the 24 extended signals with durations ranging from 100 to 170 seconds that were further contaminated with echo and white Gaussian noise.[3]

The echo signals were generated by using the gain levels of $\{-20, -30, -40, -50\}$ dB with respect to the reference signal and delays of $\{50, 100, 150, 200, 250, 300\}$ ms, in all possible combinations. The contamination model includes the *room impulse responses* (RIRs) that were randomly chosen from a set of RIRs obtained from [80]. After convolving the extended signals with the RIRs, a white Gaussian noise was added to the resulting signal so that the SNR assumes one of the following three values $\{30, 40, 50\}$ dB, with respect to the level of the echo signal.

Table 5.3 summarizes the signal characteristics. In this table m1, m2, and f1 are related to the persons who recorded the sentences: male 1, male 2, and female 1, respectively. Studio identifiers are the ones presented in [80] and the location identifiers are related to the position of the loudspeaker used to measure the RIR. The loudspeaker locations were represented in that table as 'R' for right position, 'L' for left position, and 'C' for center position. Moreover, the 'r' letter is used to identify if a rear speaker were used instead of a front speaker. In Table 5.3, the signals are organized in ascending order of the noise variance σ_n^2.

Results

Here, we provide simulation results for the C-SM-CFD, P-SM-CFD, and H-SM-CFD in order to compare the different data-selection mechanisms. More specifically, we are interested in comparing the number of times they update. In addition, the perceived results using the C-SM-CFD, P-SM-CFD, H-SM-CFD, NLMS, and BNLMS algorithms are identical during the steady-state, i.e., all of these algorithms reduce the echoes to levels at which one could not hear them.

[3] These extended signals allow us to take averages of some variables after the transient part, i.e., considering only the steady-state.

Table 5.3: *Characteristics of each degraded signal*

ID	speaker	Delay (ms)	Gain (dB)	SNR (dB)	Studio	Location	σ_n^2 (dB)
1	m2	150	-50	50	B	rR	-126.2
2	m2	300	-50	50	B	rL	-124.2
3	m1	150	-40	50	B	R	-115.2
4	m2	100	-50	40	A	rL	-114.7
5	m1	300	-40	50	D1	rL	-113.6
6	f1	250	-50	40	A	rL	-112.8
7	m1	100	-40	40	B	R	-105.6
8	m2	200	-50	30	D1	rR	-104.3
9	m2	250	-40	40	A	R	-102.7
10	f1	300	-30	50	B	R	-102.2
11	f1	50	-50	30	B	rR	-100.4
12	f1	150	-30	50	B	rL	-98.0
13	m1	200	-40	30	D1	R	-94.2
14	m2	300	-20	50	D1	rL	-93.3
15	m1	150	-20	50	A	rL	-92.8
16	f1	100	-30	40	B	rR	-92.8
17	f1	250	-30	40	A	C	-92.5
18	f1	50	-40	30	D1	C	-90.7
19	m1	250	-20	40	A	C	-84.9
20	m1	100	-20	40	B	R	-83.8
21	m2	200	-30	30	879	rL	-83.5
22	f1	50	-30	30	B	rL	-80.2
23	m1	50	-20	30	D1	rL	-75.1
24	m2	200	-20	30	B	R	-71.8

Figure 5.5 depicts the squared error averaged over time, considering only the steady-state, for each of the 24 signals. In addition to the three proposed techniques, we also plot the corresponding curves for the BNLMS algorithm and for the noise variance associated with each signal. It is interesting to observe that the curves corresponding to the BNLMS and the C-SM-CFD are similar to the curve representing the noise variance. On the other hand, the P-SM-CFD and H-SM-CFD yielded an error level that is almost constant, i.e., that do not depend on the noise variance.

Intuitively, Figure 5.5 indicates that the P-SM-CFD and H-SM-CFD accept higher values of error, as compared to the C-SM-CFD. As a consequence, such a result suggests that the former two techniques are capable of performing fewer updates than the C-SM-CFD. Indeed, Figures 5.6 and 5.7, which depict the percentage of updates per signal considering the entire signal and considering only the steady-state, respectively, corroborate our expectations. In these figures we observe that the P-SM-CFD and H-SM-CFD perform much fewer updates, when compared to the C-SM-CFD. Moreover, the H-SM-CFD perform even fewer updates than the

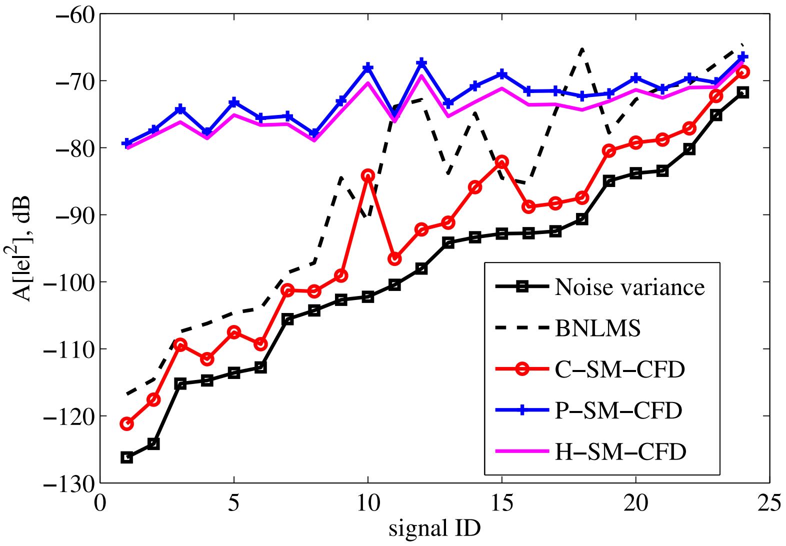


Figure 5.5: A[$|e|^2$] *represents the squared error* $|e|^2$ *averaged over time considering only the steady-state, for each of the 24 signals.*

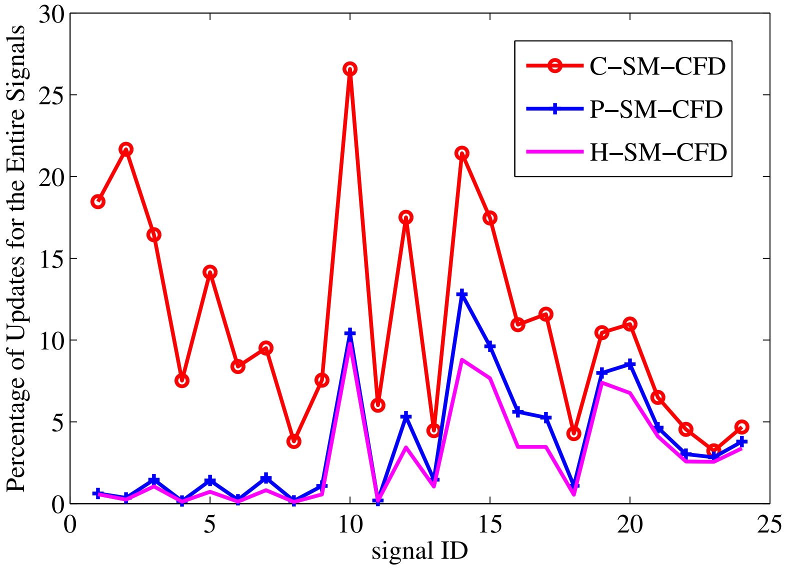


Figure 5.6: *Percentage of updates per signal considering the entire signals.*

P-SM-CFD.

Figure 5.8 illustrates how the percentage of updates varies with time, i.e., per block. Once again we can observe that the P-SM-CFD and H-SM-CFD are much better than the C-SM-CFD. Indeed, their percentage of updates are approximately half of the percentage of updates corresponding to the C-SM-CFD.

5.3.5 Final Remarks

In this section we proposed a data-selective algorithm for acoustic echo cancellation that can employ three different data-selection mechanisms. The simulation results show that the proposed algorithm can reduce significantly the number of updates

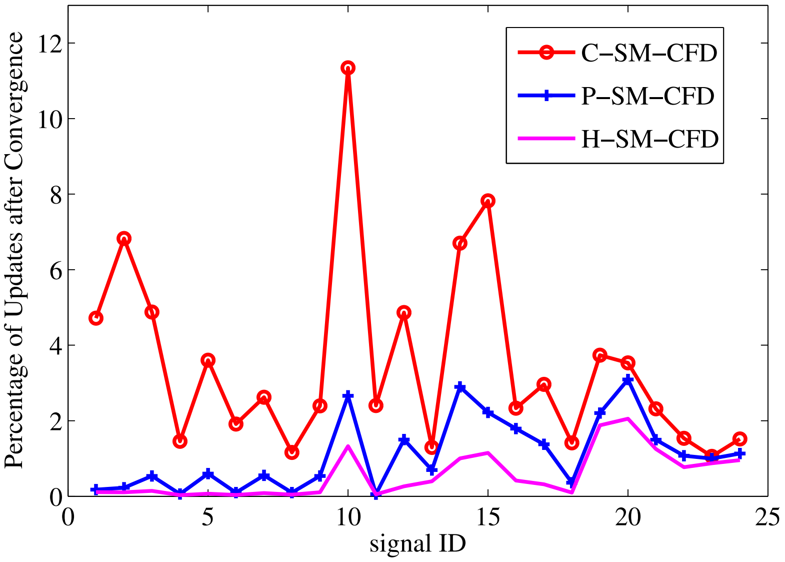


Figure 5.7: *Percentage of updates per signal after convergence.*

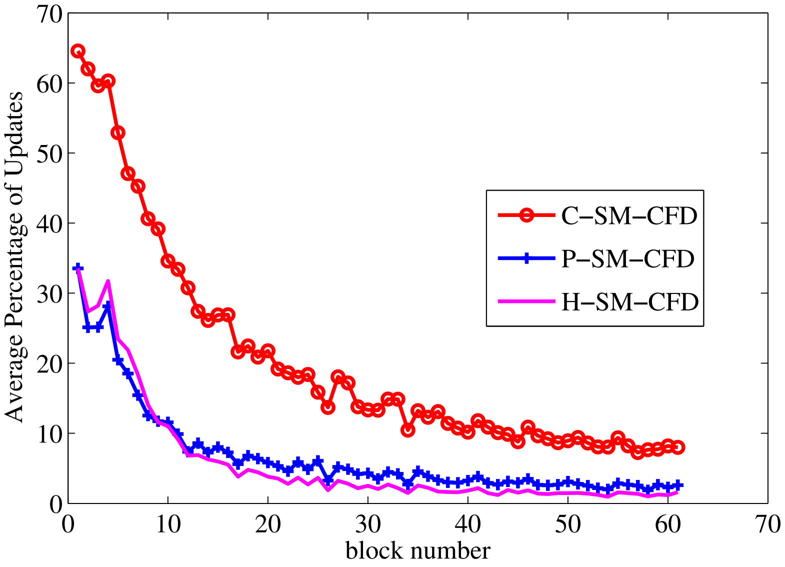


Figure 5.8: *Percentage of updates per block averaged over the* 24 *signals (computed up to the last block of the shortest signal).*

performed, thus reducing energy consumption, while being perceptually equivalent to competing techniques. In addition, we observed that the hybrid bound constitutes the best data-selection mechanism, among the three that we proposed.

5.4 Conclusion

In this chapter we proposed two algorithms, one for semi-blind equalization, and the other for acoustic echo cancellation. The semi-blind equalization algorithm is able to reduce significantly the amount of pilots transmitted by OFDM-based communications systems. However, this algorithm is very sensitive to channels that strongly attenuate some frequency bins. Therefore, there is room for the development of mechanisms to sense the channel in order to make the proposed algorithm more robust.

Regarding the acoustic echo cancellation, the proposed algorithm innovates by taking the perception/psychoacoustics into account. This strategy enables an even further reduction of the number of updates performed. In addition, such an algorithm calls for simpler psychoacoustics models so that these models do not impact the computational burden of frequency-domain algorithms.

Chapter 6

Conclusions, Contributions, and Future Works

In this thesis, energy-efficient adaptive filters were investigated. The combination of set estimation theory and data-selection mechanisms through the so-called set-membership filtering (SMF) concept was key to the design of such algorithms. Indeed, set estimation theory allows one to take into consideration prior information about the involved signals. In addition, since most applications usually present some source of uncertainty, it makes more sense to search for acceptable adaptive filters rather than searching for a single solution that is optimal with respect to a given criterion/optimization problem. Following this reasoning, data-selective adaptive filters are able to significantly reduce the computational burden, and thus energy consumption, by evaluating the input data and updating the adaptive filter only in the cases where the input signal conveys enough innovation.

6.1 Summary of Contributions

In this work, SMF-based algorithms for adaptive filtering problems were addressed. We revisited the SMF concept as well as the set-membership affine projection (SM-AP) algorithm, which is one of the most general SMF-based algorithms. As a result, some questions related to the SM-AP algorithm were answered. Indeed, we explained why and how to set the parameters of the SM-AP, in particular the constraint vector (CV), in order to obtain accurate estimates. In addition, we proposed two new CVs, among which the proposed exponential-decay CV (ED-CV) is worth of mention because it yields faster convergence speed, as compared to the already published simple-choice CV (SC-CV), while also achieving low computational burden and low steady-state mean-squared error (MSE). Besides, like the SC-CV, the ED-CV allows one to use high values of $\overline{\gamma}$, which means that the percentage of updates of the

SM-AP algorithm employing the ED-CV is very low. Moreover, we analyzed the steady-state MSE performance of the SM-AP algorithm, resulting in closed-form expressions for its excess MSE and misadjustment.

After a detailed study of the SMF concept and the SM-AP algorithm, new algorithms were proposed. We started considering problems involving sparse signals, which led us to propose two data-selective algorithms for applications involving sparse signals and systems. Such algorithms promote sparsity at each iteration through an approximation to the l^0 norm, rather than minimizing the widely used l^1 norm. The advantages of this approach were explained and verified via simulation. In addition, the proposed algorithms exhibited better MSE and misalignment results, as compared to the state-of-the-art algorithms designed to exploit sparsity, considering both sparse and compressible signals. Besides, the computational complexity and percentage of updates related to the proposed algorithms are also lower.

Finally, we also proposed two frequency-domain data-selective algorithms, one for semi-blind equalization of OFDM-based systems employing a given modulation scheme, and the other is an acoustic echo cancellation that takes psychoacoustics criteria into consideration in order to eliminate only the residual echoes that are audible.

6.2 Future Works

In this section, we list some of the possible future works:

- Improve the robustness of the semi-blind equalization algorithm proposed in Chapter 5 by introducing some kind of “channel sensing” so that the proposed algorithm could be employed even in channels with high attenuation at a given subcarrier.

- Verify the possibility of implementing some of the proposed algorithms in a distributed fashion, possibly using a diffusion or consensus average scheme.

In addition to the aforementioned future works, which are still related to adaptive filtering, we would like to apply the set estimation paradigm in other contexts. For instance, considering the problem of *sound source localization using microphone arrays*, it is widely known that computational complexity is a major issue. Indeed, the family of localization methods known as steered-response power (SRP) provide techniques that are more robust against reverberation than other families of methods. However, the accuracy of SRP techniques depend on the number of microphones and spatial sampling and, therefore, its required computational burden may be prohibitive. The works [81, 82] reduce the complexity of the SRP by using a different functional or by using a different search method (hierarchical search, instead of performing an exhaustive search across all points of the solution space as usually happens). Since complexity is the major problem, we feel that if we could add a data-selection scheme to such methods we could dramatically decrease their complexity, as we did with adaptive filters.

Bibliography

[1] GOLLAMUDI, S., NAGARAJ, S., KAPOOR, S., et al. “Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size,” *IEEE Signal Processing Letters*, v. 5, n. 5, pp. 111–114, May 1998. doi: 10.1109/97.668945.

[2] DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation.* 4th ed. New York, USA, Springer, 2013.

[3] ANTONIOU, A., LU, W.-S. *Practical Optimization: Algorithms and Engineering Applications.* New York, NY, Springer, 2007.

[4] DINIZ, P. S. R., MARTINS, W. A., LIMA, M. V. S. *Block Transceivers: OFDM and Beyond.* USA, Morgan & Claypool, 2012.

[5] LEHMANN, E. L., CASELLA, G. *Theory of Point Estimation.* 2nd ed. New York, USA, Springer, 2003.

[6] HAYKIN, S. *Adaptive Filter Theory.* 4th ed. Englewood Cliffs, NJ, Prentice Hall, 2002.

[7] COMBETTES, P. L. “The foundations of set theoretic estimation,” *Proceedings of the IEEE*, v. 81, n. 2, pp. 182–208, February 1993. doi: 10.1109/5.214546.

[8] SAYED, A. H. *Fundamentals of Adaptive Filtering.* New York, USA, Wiley-IEEE, 2003.

[9] WIDROW, B., STEARNS, S. D. *Adaptive Signal Processing.* Englewood Cliffs, NJ, Prentice Hall, 1985.

[10] DINIZ, P. S. R., DA SILVA, E. A. B., NETTO, S. L. *Digital Signal Processing: System Analysis and Design.* 2nd ed. Cambridge, UK, Cambridge University Press, 2010.

[11] DINIZ, P. S. R., WERNER, S. "Set-membership binormalized data reusing LMS algorithms," *IEEE Transactions on Signal Processing*, v. 51, n. 1, pp. 124–134, January 2003. doi: 10.1109/TSP.2002.806562.

[12] WERNER, S., DINIZ, P. S. R. "Set-membership affine projection algorithm," *IEEE Signal Processing Letters*, v. 8, n. 8, pp. 231–235, August 2001. doi: 10.1109/97.935739.

[13] LIMA, M. V. S., DINIZ, P. S. R. "Steady-state MSE performance of the set-membership affine projection algorithm," *Circuits, Systems, and Signal Processing*, v. 32, n. 4, pp. 1811–1837, January 2013. doi: 10.1007/s00034-012-9545-4.

[14] WIDROW, B., HOFF, M. E. "Adaptive switching circuits," *IRE WESCOM Conv. Rec.*, v. 4, pp. 96–104, 1960.

[15] NAGUMO, J. I., NODA, A. "A learning method for system identification," *IEEE Transactions on Automatic Control*, v. AC-12, pp. 282–287, 1967.

[16] ALBERT, A. E., GARDNER JR., L. S. *Stochastic approximation and nonlinear regression*. Cambridge, MA, MIT Press, 1967.

[17] BOYD, S., VANDENBERGHE, L. *Convex Optimization*. Cambridge, UK, Cambridge University Press, 2004.

[18] PLACKETT, R. L. "Some theorems in least squares," *Biometrika*, v. 37, pp. 149, 1950.

[19] OZEKI, K., UMEDA, T. "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, v. 67-A, n. 5, pp. 19–27, 1984.

[20] SHIN, H.-C., SAYED, A. H. "Mean-square performance of a family of affine projection algorithms," *IEEE Transactions on Signal Processing*, v. 52, n. 1, pp. 90–102, January 2004. doi: 10.1109/TSP.2003.820077.

[21] GAY, S. L., TAVATHIA, S. "The fast affine projection algorithm." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-95)*, v. 5, pp. 3023–3026, Detroit, USA, May 1995. doi: 10.1109/ICASSP.1995.479482.

[22] SANKARAN, S. G., BEEX, A. A. L. "Convergence behavior of affine projection algorithms," *IEEE Transactions on Signal Processing*, v. 48, n. 4, pp. 1086–1096, April 2000. doi: 10.1109/78.827542.

[23] SCHWEPPE, F. “Recursive state estimation: Unknown but bounded errors and system inputs,” *IEEE Transactions on Automatic Control*, v. 13, n. 1, pp. 22–28, February 1968. doi: 10.1109/TAC.1968.1098790.

[24] FOGEL, E., HUANG, Y.-F. “On the value of information in system identification–bounded noise case,” *Automatica*, v. 18, n. 2, pp. 229–238, March 1982. doi: http://dx.doi.org/10.1016/0005-1098(82)90110-8.

[25] DELLER, J. R. “Set-membership identification in digital signal processing,” *IEEE ASSP Magazine*, v. 6, n. 4, pp. 4–20, October 1989. doi: 10.1109/53.41661.

[26] NAGARAJ, S., GOLLAMUDI, S., KAPOOR, S., et al. “BEACON: An adaptive set-membership filtering technique with sparse updates,” *IEEE Transactions on Signal Processing*, v. 47, n. 11, pp. 2928–2941, November 1999. doi: 10.1109/78.796429.

[27] GOLLAMUDI, S., KAPOOR, S., NAGARAJ, S., et al. “Set-membership adaptive equalization and updator-shared implementation for multiple channel communications systems,” *IEEE Transactions on Signal Processing*, v. 46, n. 9, pp. 2372–2385, September 1998. doi: 10.1109/78.709523.

[28] GUO, L., HUANG, Y.-F. “Frequency-domain set-membership filtering and its applications,” *IEEE Transactions on Signal Processing*, v. 55, n. 4, pp. 1326–1338, April 2007. doi: 10.1109/TSP.2006.888890.

[29] MARTINS, W. A., LIMA, M. V. S., DINIZ, P. S. R. “Semi-blind data-selective equalizers for QAM.” In: *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2008)*, pp. 501–505, Recife, Brazil, July 2008. doi: 10.1109/SPAWC.2008.4641658.

[30] LIMA, M. V. S., DINIZ, P. S. R. “Fast learning set theoretic estimation.” In: *21st European Signal Processing Conference (EUSIPCO 2013)*, pp. 1–5, Marrakech, Morocco, September 2013.

[31] LIMA, M. V. S., MARTINS, W. A., FERREIRA, T. N., et al. “Sparsity-aware data-selective adaptive filters,” *IEEE Transactions on Signal Processing*, 2013 (submitted).

[32] LIMA, M. V. S., DINIZ, P. S. R. “Steady-state analysis of the set-membership affine projection algorithm.” In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pp. 3802–3805, Dallas, USA, March 2010. doi: 10.1109/ICASSP.2010.5495836.

[33] DINIZ, P. S. R. “Convergence performance of the simplified set-membership affine projection algorithm,” *Circuits, Systems, and Signal Processing*, v. 30, n. 2, pp. 439–462, April 2011. doi: 10.1007/s00034-010-9219-z.

[34] LIMA, M. V. S., DINIZ, P. S. R. “On the steady-state MSE performance of the set-membership NLMS algorithm.” In: *7th International Symposium on Wireless Communication Systems (ISWCS 2010)*, pp. 389–393, York, UK, September 2010. doi: 10.1109/ISWCS.2010.5624323.

[35] SAYED, A. H., RUPP, M. “Error-energy bounds for adaptive gradient algorithms,” *IEEE Transactions on Signal Processing*, v. 44, n. 8, pp. 1982–1989, August 1996. doi: 10.1109/78.533719.

[36] YOUSEF, N. R., SAYED, A. H. “A unified approach to the steady-state and tracking analyses of adaptive filters,” *IEEE Transactions on Signal Processing*, v. 49, n. 2, pp. 314–324, February 2001. doi: 10.1109/78.902113.

[37] PRICE, R. “A useful theorem for nonlinear devises having Gaussian inputs,” *IRE Transactions on Information Theory*, v. 4, n. 2, pp. 69–72, June 1958. doi: 10.1109/TIT.1958.1057444.

[38] PAPOULIS, A. *Probability, Random Variables, and Stochastic Processes*. 3rd ed. New York, USA, McGraw Hill, 1991.

[39] PETRAGLIA, M. R., HADDAD, D. B. “New adaptive algorithms for identification of sparse impulse responses – analysis and comparisons.” In: *7th International Symposium on Wireless Communication Systems (ISWCS 2010*, pp. 384–388, York, UK, September 2010. doi: 10.1109/ISWCS.2010.5624301.

[40] MENG, R., DE LAMARE, R. C., NASCIMENTO, V. H. “Sparsity-aware affine projection adaptive algorithms for system identification.” In: *Sensor Signal Processing for Defence (SSPD 2011)*, pp. 1–5, London, U.K., September 2011. doi: 10.1049/ic.2011.0144.

[41] DUTTWEILER, D. L. “Proportionate normalized least-mean-squares adaptation in echo cancelers,” *IEEE Transactions on Speech and Audio Processing*, v. 8, n. 5, pp. 508–518, September 2000. doi: 10.1109/89.861368.

[42] GAY, S. L. “An efficient, fast converging adaptive filter for network echo cancellation.” In: *Thirty-Second Asilomar Conference on Signals, Systems amp, Computers (ACSSC 1998)*, v. 1, pp. 394–398, 1998. doi: 10.1109/ACSSC.1998.750893.

[43] BENESTY, J., GAY, S. L. "An improved PNLMS algorithm." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, v. 2, pp. 1881–1884, Dallas, USA, May 2002. doi: 10.1109/ICASSP.2002.5744994.

[44] GU, Y., JIN, J., MEI, S. "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, v. 16, n. 9, pp. 774–777, September 2009. doi: 10.1109/LSP.2009.2024736.

[45] LIU, L., FUKUMOTO, M., SAIKI, S. "An improved mu-law proportionate NLMS algorithm." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pp. 3797–3800, 2008. doi: 10.1109/ICASSP.2008.4518480.

[46] NAYLOR, P. A., CUI, J., BROOKES, M. "Adaptive algorithms for sparse echo cancellation," *Signal Processing*, v. 86, n. 6, pp. 1182–1192, June 2006. doi: 10.1016/j.sigpro.2005.09.015.

[47] KHONG, A. W. H., NAYLOR, P. A. "Efficient use of sparse adaptive filters." In: *Fortieth Asilomar Conference on Signals, Systems and Computers (ACSSC 2006)*, pp. 1375–1379, 2006. doi: 10.1109/ACSSC.2006.354982.

[48] WERNER, S., APOLINÁRIO JR., J. A., DINIZ, P. S. R., et al. "A set-membership approach to normalized proportionate adaptation algorithms." In: *European Signal Processing Conference (EUSIPCO 2005)*, pp. 1–4, Antalya, Turkey, September 2005.

[49] WERNER, S., APOLINÁRIO JR., J. A., DINIZ, P. S. R. "Set-membership proportionate affine projection algorithms," *EURASIP Journal on Audio, Speech, and Music Processing*, v. 2007, n. 1, pp. 1–10, 2007. doi: 10.1155/2007/34242.

[50] HOSHUYAMA, O., GOUBRAN, R. A., SUGIYAMA, A. "A generalized proportionate variable step-size algorithm for fast changing acoustic environments." In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, v. 4, pp. 161–164, 2004. doi: 10.1109/ICASSP.2004.1326788.

[51] CANDES, E. J., WAKIN, M. B., BOYD, S. P. "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, v. 14, n. 5, pp. 877–905, December 2008. doi: 10.1007/s00041-008-9045-x.

[52] LIMA, M. V. S., MARTINS, W. A., DINIZ, P. S. R. “Affine projection algorithms for sparse system identification.” In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 5666–5670, Vancouver, Canada, May 2013.

[53] LIMA, M. V. S., SOBRON, I., MARTINS, W. A., et al. “Stability and MSE analyses of affine projection algorithms for sparse system identification.” In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, Florence, Italy, May 2014 (submitted).

[54] ELDAR, Y., KUTYNIOK, G. *Compressed Sensing: Theory and Applications.* Cambridge, UK, Cambridge University Press, 2012.

[55] MOHIMANI, H., BABAIE-ZADEH, M., JUTTEN, C. “A fast approach for overcomplete sparse decomposition based on smoothed l^0 norm,” *IEEE Transactions on Signal Processing*, v. 57, n. 1, pp. 289–301, January 2009. ISSN: 1053-587X. doi: 10.1109/TSP.2008.2007606.

[56] TRZASKO, J., MANDUCA, A. “Highly undersampled magnetic resonance image reconstruction via homotopic l_0-minimization,” *IEEE Transactions on Medical Imaging*, v. 28, n. 1, pp. 106–121, January 2009. doi: 10.1109/TMI.2008.927346.

[57] SING-LONG, C. A., TEJOS, C. A., IRARRAZAVAL, P. “Evaluation of continuous approximation functions for the l_0-norm for compressed sensing.” In: *International Soc. Mag. Reson. Med. (ISMRM 2009)*, v. 17, p. 4585, 2009.

[58] HUBER, P. *Robust Statistics.* New York, USA, Wiley, 1981.

[59] GEMAN, D., REYNOLDS, G. “Nonlinear image recovery with half-quadratic regularization,” *IEEE Transactions on Image Processing*, v. 4, n. 7, pp. 932–946, July 1995. doi: 10.1109/83.392335.

[60] KOPSINIS, Y., SLAVAKIS, K., THEODORIDIS, S., et al. “Reduced complexity online sparse signal reconstruction using projections onto weighted l_1 balls.” In: *International Conference on Digital Signal Processing (DSP 2011)*, pp. 1–8, July 2011. doi: 10.1109/ICDSP.2011.6005005.

[61] SLAVAKIS, K., KOPSINIS, Y., THEODORIDIS, S. “Adaptive algorithm for sparse system identification using projections onto weighted l_1 balls.” In: *International Conference on Acoustics Speech and Signal Processing (ICASSP 2010)*, v. 17, pp. 3742–3745, Dallas, USA, March 2010.

[62] KOPSINIS, Y., SLAVAKIS, K., THEODORIDIS, S. "Online sparse system identification and signal reconstruction using projections onto weighted l_1 balls," *IEEE Transactions on Signal Processing*, v. 59, n. 3, pp. 936–952, March 2011. doi: 10.1109/TSP.2010.2090874.

[63] CHOUVARDAS, S., SLAVAKIS, K., KOPSINIS, Y., et al. "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, v. 60, n. 10, pp. 5412–5425, October 2012. doi: 10.1109/TSP.2012.2204987.

[64] MANCERA, L., PORTILLA, J. "L0-norm-based sparse representation through alternate projections." In: *IEEE International Conference on Image Processing (ICIP 2006)*, pp. 2089–2092, Atlanta, USA, October 2006. doi: 10.1109/ICIP.2006.312819.

[65] CHEN, Y., GU, Y., HERO, A. O. "Sparse LMS for system identification." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pp. 3125–3128, Taipei, Taiwan, April 2009. doi: 10.1109/ICASSP.2009.4960286.

[66] LIMA, M. V. S., FREELAND, F. P., DINIZ, P. S. R. "Perception-based acoustic echo cancellation," *IEEE Signal Processing Letters*, 2013 (in preparation).

[67] LIMA, M. V. S., ESPINDOLA, B. N., FREELAND, F. P., et al. "Applications of data-selective adaptive filters." In: *Simpósio Brasileiro de Telecomunicações (SBrT 2011)*, pp. 1–5, Curitiba, Brazil, September 2011.

[68] DINIZ, P. S. R., LIMA, M. V. S., MARTINS, W. A. "Semi-blind data-selective algorithms for channel equalization." In: *IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, pp. 53–56, Seattle, USA, May 2008. doi: 10.1109/ISCAS.2008.4541352.

[69] IEEE. *IEEE 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Relatório Técnico Std 802.11n-2009, IEEE, October 2009.

[70] IEEE. *IEEE 802 Part 16: Air Interface for Broadband Wireless Access Systems*. Relatório Técnico Std 802.16-2009, IEEE, May 2009.

[71] 3GPP. *E-UTRA; physical channels and modulation*. Relatório Técnico 36.211 (Version 9.1.0), 3GPP, March 2010.

[72] 3GPP. *E-UTRA; user equipment radio transmission and reception.* Relatório Técnico 36.101 (Version 9.8.0), 3GPP, June 2011.

[73] LIMA, M. V. S., GUSSEN, C. M. G., ESPINDOLA, B. N., et al. "Open-source physical-layer simulator for LTE systems." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pp. 2781–2784, Kyoto, Japan, March 2012. doi: 10.1109/ICASSP.2012.6288494.

[74] HANSLER, E., SCHMIDT, G. *Acoustic Echo and Noise Control: A Practical Approach.* Hoboken, NJ, USA, Wiley, 2004.

[75] BENESTY, J., GANSLER, T., MORGAN, D. R., et al. *Advances in Network and Acoustic Echo Cancellation.* Berlin Heidelberg, Germany, Springer, 2010.

[76] CLARK, G. A., PARKER, S. R., MITRA, S. K. "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. ASSP-31, n. 5, pp. 1073–1083, October 1983.

[77] FASTL, H., ZWICKER, E. *Psychoacoustics: Facts and Models.* 3rd ed. Berlin Heidelberg, Germany, Springer, 2007.

[78] ALCAIM, A., SOLEWICZ, J. A., DE MORAES, J. A. "Freqüência de ocorrência dos fones e listas de frases foneticamente balanceadas no português falado no Rio de Janeiro," *Revista da Sociedade Brasileira de Telecomunicações*, v. 7, pp. 23–41, December 1992 (in Portuguese).

[79] ITU. "Methods for subjective determination of transmission quality." In: *ITU-T Rec. P.800*, International Telecommunications Union, Geneva, Switzerland, 1996.

[80] STUDIO, M. T. "Impulse Responses." available at: `https://files.nyu.edu/ar137/public/research_mtechirs.html`, 2008.

[81] LIMA, M. V. S., MARTINS, W. A., NUNES, L. O., et al. "Efficient steered-response power methods for sound source localization using microphone arrays," *IEEE Transactions on Audio, Speech, and Language Processing*, 2013 (submitted).

[82] NUNES, L. O., MARTINS, W. A., LIMA, M. V. S., et al. "A steered-response power algorithm employing hierarchical search for acoustic source localization using microphone arrays," *IEEE Transactions on Signal Processing*, 2013 (accepted).

Appendix A

Proofs Related to Chapter 3

This appendix contains most of the mathematical derivations related to Section 3.4 of Chapter 3. In addition to the mathematical derivations, it also shows how to model some of the variables. In Section A.6 we present and discuss the assumptions and statements invoked during the analysis of the steady-state MSE of the SM-AP algorithm.

A.1 Proof of Proposition 1

Proof. Squaring the Euclidean norm of both sides of Eq. (3.36), we have

$$\begin{aligned}&\left[\Delta\mathbf{w}(k+1)-\mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)\right]^T\left[\Delta\mathbf{w}(k+1)-\mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)\right]=\\&\left[\Delta\mathbf{w}(k)-\mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)\right]^T\left[\Delta\mathbf{w}(k)-\mathbf{X}(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)\right],\end{aligned}$$

and since $\tilde{\boldsymbol{\varepsilon}}(k)=-\mathbf{X}^T(k)\Delta\mathbf{w}(k+1)$ and $\tilde{\mathbf{e}}(k)=-\mathbf{X}^T(k)\Delta\mathbf{w}(k)$, it follows that

$$\begin{aligned}&\|\Delta\mathbf{w}(k+1)\|^2+\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)+\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)+\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)=\\&\|\Delta\mathbf{w}(k)\|^2+\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k)+\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)+\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k).\end{aligned}$$

Then, removing the equal terms on both sides of the last equation we get

$$\|\Delta\mathbf{w}(k+1)\|^2+\tilde{\mathbf{e}}^T(k)\mathbf{R}^{-1}(k)\tilde{\mathbf{e}}(k)=\|\Delta\mathbf{w}(k)\|^2+\tilde{\boldsymbol{\varepsilon}}^T(k)\mathbf{R}^{-1}(k)\tilde{\boldsymbol{\varepsilon}}(k).$$

□

A.2 Correlation Expression

Utilizing Eq. (3.32) we can eliminate $\tilde{\boldsymbol{\varepsilon}}(k)$ from Eq. (3.38), and since $\mathbf{R}(k)$ and $\mathbf{S}(k)$ are symmetric matrices, after some manipulations, it follows that

$$\begin{aligned} P_{\mathrm{up}}(k)\mu \mathrm{E}\left[[\mathbf{e}(k)-\boldsymbol{\gamma}(k)]^T\mathbf{S}(k)\mathbf{R}(k)\mathbf{S}(k)[\mathbf{e}(k)-\boldsymbol{\gamma}(k)]\right] = \\ 2\mathrm{E}\left[\tilde{\mathbf{e}}^T(k)\mathbf{S}(k)[\mathbf{e}(k)-\boldsymbol{\gamma}(k)]\right]. \end{aligned} \tag{A.1}$$

Using $\mathbf{e}(k) = \tilde{\mathbf{e}}(k) + \mathbf{n}(k)$, see Eq. (3.34), and defining

$$\underline{\tilde{\mathbf{e}}}(k) = \tilde{\mathbf{e}}(k) - \boldsymbol{\gamma}(k), \tag{A.2}$$

Eq. (A.1) can be rewritten as

$$\begin{aligned} P_{\mathrm{up}}(k)\mu \mathrm{E}\left[\left[\underline{\tilde{\mathbf{e}}}(k)+\mathbf{n}(k)\right]^T\mathbf{S}(k)\mathbf{R}(k)\mathbf{S}(k)\left[\underline{\tilde{\mathbf{e}}}(k)+\mathbf{n}(k)\right]\right] = \\ 2\mathrm{E}\left[\left[\underline{\tilde{\mathbf{e}}}(k)+\boldsymbol{\gamma}(k)\right]^T\mathbf{S}(k)\left[\underline{\tilde{\mathbf{e}}}(k)+\mathbf{n}(k)\right]\right]. \end{aligned} \tag{A.3}$$

Expanding the equation above and considering $\mathbf{S}(k) \approx \mathbf{R}^{-1}(k)$, see statement *St-4*, we get

$$\begin{aligned} & P_{\mathrm{up}}(k)\mu \mathrm{E}\left[\underline{\tilde{\mathbf{e}}}^T(k)\mathbf{S}(k)\underline{\tilde{\mathbf{e}}}(k) + \mathbf{n}^T(k)\mathbf{S}(k)\mathbf{n}(k) + 2\mathbf{n}^T(k)\mathbf{S}(k)\underline{\tilde{\mathbf{e}}}(k)\right] \\ = & 2\mathrm{E}\left[\underline{\tilde{\mathbf{e}}}^T(k)\mathbf{S}(k)\underline{\tilde{\mathbf{e}}}(k) + \underline{\tilde{\mathbf{e}}}^T(k)\mathbf{S}(k)\mathbf{n}(k) + \boldsymbol{\gamma}^T(k)\mathbf{S}(k)\underline{\tilde{\mathbf{e}}}(k) + \boldsymbol{\gamma}^T(k)\mathbf{S}(k)\mathbf{n}(k)\right]. \end{aligned}$$

Using the relation $\underline{\tilde{\mathbf{e}}}^T(k)\mathbf{S}(k)\mathbf{n}(k) = \mathbf{n}^T(k)\mathbf{S}(k)\underline{\tilde{\mathbf{e}}}(k)$, applying the trace to the equation above, and using the property $\mathrm{tr}\{\mathbf{AB}\} = \mathrm{tr}\{\mathbf{BA}\}$, we can write

$$\begin{aligned} & (2-P_{\mathrm{up}}(k)\mu)\mathrm{E}\left[\mathrm{tr}\left\{\underline{\tilde{\mathbf{e}}}(k)\underline{\tilde{\mathbf{e}}}^T(k)\mathbf{S}(k)\right\}\right] + 2(1-P_{\mathrm{up}}(k)\mu)\mathrm{E}\left[\mathrm{tr}\left\{\underline{\tilde{\mathbf{e}}}(k)\mathbf{n}^T(k)\mathbf{S}(k)\right\}\right] \\ & + 2\mathrm{E}\left[\mathrm{tr}\left\{\underline{\tilde{\mathbf{e}}}(k)\boldsymbol{\gamma}^T(k)\mathbf{S}(k)\right\}\right] \\ = & P_{\mathrm{up}}(k)\mu \mathrm{E}\left[\mathrm{tr}\left\{\mathbf{n}(k)\mathbf{n}^T(k)\mathbf{S}(k)\right\}\right] - 2\mathrm{E}\left[\mathrm{tr}\left\{\mathbf{n}(k)\boldsymbol{\gamma}^T(k)\mathbf{S}(k)\right\}\right]. \end{aligned} \tag{A.4}$$

Assuming that at steady-state $\mathbf{S}(k)$ is uncorrelated with the random matrices $\underline{\tilde{\mathbf{e}}}(k)\underline{\tilde{\mathbf{e}}}^T(k)$, $\underline{\tilde{\mathbf{e}}}(k)\mathbf{n}^T(k)$, $\underline{\tilde{\mathbf{e}}}(k)\boldsymbol{\gamma}^T(k)$, $\mathbf{n}(k)\mathbf{n}^T(k)$, and $\mathbf{n}(k)\boldsymbol{\gamma}^T(k)$, see assumption *As-2*, we get

$$\begin{aligned} & (2-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\underline{\tilde{\mathbf{e}}}(k)\underline{\tilde{\mathbf{e}}}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\} + 2(1-P_{\mathrm{up}}(k)\mu)\mathrm{tr}\left\{\mathrm{E}\left[\underline{\tilde{\mathbf{e}}}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\} \\ & + 2\mathrm{tr}\left\{\mathrm{E}\left[\underline{\tilde{\mathbf{e}}}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\} \\ = & P_{\mathrm{up}}(k)\mu\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\} - 2\mathrm{tr}\left\{\mathrm{E}\left[\mathbf{n}(k)\boldsymbol{\gamma}^T(k)\right]\mathrm{E}\left[\mathbf{S}(k)\right]\right\}. \end{aligned} \tag{A.5}$$

It is possible to eliminate the dependence on $\underline{\tilde{\mathbf{e}}}(k)$ by substituting (A.2) in the equation above as follows

$$\begin{aligned}&(2-P_{\text{up}}(k)\mu)\text{tr}\left\{\text{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}+2(1-P_{\text{up}}(k)\mu)\text{tr}\left\{\text{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}\\&-P_{\text{up}}(k)\mu\text{tr}\left\{\text{E}\left[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}+P_{\text{up}}(k)\mu\text{tr}\left\{\text{E}\left[\tilde{\mathbf{e}}(k)\boldsymbol{\gamma}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}\\&-(2-P_{\text{up}}(k)\mu)\text{tr}\left\{\text{E}\left[\boldsymbol{\gamma}(k)\tilde{\mathbf{e}}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}-2(1-P_{\text{up}}(k)\mu)\text{tr}\left\{\text{E}\left[\boldsymbol{\gamma}(k)\mathbf{n}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}\\=&P_{\text{up}}(k)\mu\text{tr}\left\{\text{E}\left[\mathbf{n}(k)\mathbf{n}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}-2\text{tr}\left\{\text{E}\left[\mathbf{n}(k)\boldsymbol{\gamma}^T(k)\right]\text{E}\left[\mathbf{S}(k)\right]\right\}.\end{aligned}\tag{A.6}$$

A.3 Calculating $\text{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]$

Examining the $(i+1)$th row of Eq. (3.33) and Eq. (3.34) we have

$$\tilde{\varepsilon}_i(k)=-\mathbf{x}^T(k-i)\Delta\mathbf{w}(k+1)\tag{A.7}$$

$$\tilde{e}_i(k)=-\mathbf{x}^T(k-i)\Delta\mathbf{w}(k)=e_i(k)-n(k-i)\tag{A.8}$$

for $i=0,\ldots,L$. Since $\mathbf{R}(k)\mathbf{S}(k)\approx\mathbf{I}$, see statement *St-4*, the $(i+1)$th row of Eq. (3.32) is

$$\tilde{\varepsilon}_i(k)=\tilde{e}_i(k)-P_{\text{up}}(k)\mu\left[e_i(k)-\gamma_i(k)\right].\tag{A.9}$$

Using Eq. (A.8) to replace $e_i(k)$ in the equation above it follows that

$$\tilde{\varepsilon}_i(k)=(1-P_{\text{up}}(k)\mu)\tilde{e}_i(k)-P_{\text{up}}(k)\mu\left(n(k-i)-\gamma_i(k)\right).\tag{A.10}$$

Squaring the equation above, we get

$$\begin{aligned}\tilde{\varepsilon}_i^2(k)=&(1-P_{\text{up}}(k)\mu)^2\tilde{e}_i^2(k)+(P_{\text{up}}(k)\mu)^2\left(n(k-i)-\gamma_i(k)\right)^2\\&-2P_{\text{up}}(k)\mu(1-P_{\text{up}}(k)\mu)\tilde{e}_i(k)\left(n(k-i)-\gamma_i(k)\right).\end{aligned}\tag{A.11}$$

Note that the noiseless *a posteriori* error vector is related to the noiseless *a priori* error vector through the following relation

$$\tilde{\varepsilon}_i(k-1)=-\mathbf{x}^T(k-(i+1))\Delta\mathbf{w}(k)=\tilde{e}_{i+1}(k).\tag{A.12}$$

Now, considering Eq. (A.11) at iteration $k-1$, substituting Eq. (A.12) in Eq. (A.11), and taking the expected value, we get

$$\begin{aligned}\text{E}\left[\tilde{e}_{i+1}^2(k)\right]=&(1-P_{\text{up}}\mu)^2\text{E}\left[\tilde{e}_i^2(k-1)\right]+(P_{\text{up}}\mu)^2\text{E}\left[\left(n(k-1-i)-\gamma_i(k-1)\right)^2\right]\\&-2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\text{E}\left[\tilde{e}_i(k-1)\left(n(k-1-i)-\gamma_i(k-1)\right)\right],\end{aligned}\tag{A.13}$$

where we considered that at steady-state $P_{\text{up}}(k)$ is a constant P_{up}.

Expanding Eq. (A.13) we get

$$\begin{aligned}\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right] =&(1-P_{\text{up}}\mu)^2\mathrm{E}\big[\tilde{e}_i^2(k-1)\big]+(P_{\text{up}}\mu)^2\mathrm{E}\big[n^2(k-1-i)\big]+(P_{\text{up}}\mu)^2\mathrm{E}\big[\gamma_i^2(k-1)\big]\\ &-2(P_{\text{up}}\mu)^2\mathrm{E}\left[n(k-1-i)\gamma_i(k-1)\right]\\ &-2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\mathrm{E}\left[\tilde{e}_i(k-1)n(k-1-i)\right]\\ &+2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\mathrm{E}\left[\tilde{e}_i(k-1)\gamma_i(k-1)\right] \end{aligned} \tag{A.14}$$

which can be simplified using (3.11), the relation $\mathrm{E}\left[\gamma_i^2(k-1)\right] = \overline{\gamma}^2$, for $i = 0,1,\ldots,L$, and $\mathrm{E}\left[n^2(k)\right]=\sigma_n^2, \forall k$ (see Definition 4), leading to

$$\begin{aligned}\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right] =&(1-P_{\text{up}}\mu)^2\mathrm{E}\left[\tilde{e}_i^2(k-1)\right]+(P_{\text{up}}\mu)^2\sigma_n^2+(P_{\text{up}}\mu)^2\overline{\gamma}^2\\ &-2(P_{\text{up}}\mu)^2\overline{\gamma}\mathrm{E}\big[n(k-1-i)\text{sign}[e_i(k-1)]\big]\\ &-2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\mathrm{E}\left[\tilde{e}_i(k-1)n(k-1-i)\right]\\ &+2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\overline{\gamma}\mathrm{E}\big[\tilde{e}_i(k-1)\text{sign}[e_i(k-1)]\big]. \end{aligned} \tag{A.15}$$

Assuming that at steady-state $\tilde{e}_i(k-1)$ is a zero-mean Gaussian RV, see assumption *As-4*, we can apply Result 2 to Eq. (A.15) leading to the following relation

$$\begin{aligned}\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right] =&(1-P_{\text{up}}\mu)^2\mathrm{E}\left[\tilde{e}_i^2(k-1)\right]+(P_{\text{up}}\mu)^2\sigma_n^2+(P_{\text{up}}\mu)^2\overline{\gamma}^2\\ &-2(P_{\text{up}}\mu)^2\overline{\gamma}\rho_i(k-1)\mathrm{E}\left[n(k-1-i)e_i(k-1)\right]\\ &-2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\mathrm{E}\left[\tilde{e}_i(k-1)n(k-1-i)\right]\\ &+2P_{\text{up}}\mu(1-P_{\text{up}}\mu)\overline{\gamma}\rho_i(k-1)\mathrm{E}\left[\tilde{e}_i(k-1)e_i(k-1)\right], \end{aligned} \tag{A.16}$$

where

$$\rho_i(k)=\sqrt{\frac{2}{\pi\mathrm{E}[e_i^2(k)]}}. \tag{A.17}$$

Utilizing the relation $e_i(k-1)=\tilde{e}_i(k-1)+n(k-1-i)$ in order to remove the dependence on the *a priori* error signal, and rearranging the terms we get

$$\begin{aligned}\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right] =&\left[1-P_{\text{up}}\mu+2P_{\text{up}}\mu\overline{\gamma}\rho_i(k)\right](1-P_{\text{up}}\mu)\,\mathrm{E}\left[\tilde{e}_i^2(k)\right]\\ &+\left[\sigma_n^2+\overline{\gamma}^2-2\overline{\gamma}\rho_i(k)\sigma_n^2\right](P_{\text{up}}\mu)^2, \end{aligned} \tag{A.18}$$

where we have used $\mathrm{E}\left[e_i^2(k-1)\right]=\mathrm{E}\left[e_i^2(k)\right]$ and $\mathrm{E}\left[\tilde{e}_i^2(k-1)\right]=\mathrm{E}\left[\tilde{e}_i^2(k)\right]$, see statement *St-2*, and we neglected the terms depending on $\mathrm{E}\left[n(k-1-i)\tilde{e}_i(k-1)\right]$, see statement *St-3*.

Assuming $\rho_i(k)\approx\rho_0(k)$, for $i=0,1,\ldots,L$, see assumption *As-5*, in order to simplify the mathematical manipulations, we can rewrite the recursion given by (A.18) as

$$\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right]=a\mathrm{E}\left[\tilde{e}_i^2(k)\right]+b\left(P_{\text{up}}\mu\right)^2, \tag{A.19}$$

where

$$a = [1 - P_{\text{up}}\mu + 2P_{\text{up}}\mu\overline{\gamma}\rho_0(k)]\,(1 - P_{\text{up}}\mu) \tag{A.20}$$

$$b = \left[\sigma_n^2 + \overline{\gamma}^2 - 2\overline{\gamma}\rho_0(k)\sigma_n^2\right]. \tag{A.21}$$

By induction, for $0 \leq i \leq L-1$ one can prove that

$$\mathrm{E}\left[\tilde{e}_{i+1}^2(k)\right] = a^{(i+1)}\mathrm{E}\left[\tilde{e}_0^2(k)\right] + \left(\sum_{l=0}^{i} a^l\right) b\,(P_{\text{up}}\mu)^2. \tag{A.22}$$

Assuming $\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right]$ is diagonally dominant, see assumption *As-3*, we can write

$$\mathrm{E}\left[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\right] = \mathbf{A}_1\mathrm{E}\left[\tilde{e}_0^2(k)\right] + \mathbf{A}_2 b\,(P_{\text{up}}\mu)^2 \tag{A.23}$$

with $\mathbf{A}_1 = \text{diag}\left\{1, a, a^2, \cdots, a^L\right\}$, and $\mathbf{A}_2 = \text{diag}\left\{0, 1, 1+a, \cdots, \sum_{l=0}^{L-1} a^l\right\}$.

A.4 Modeling $\rho_0(k)$

Eq. (A.17) shows that $\rho_0(k)$ is completely specified by $\mathrm{E}[e_0^2(k)]$. Therefore, we propose the following approximation for $\mathrm{E}[e_0^2(k)]$

$$\mathrm{E}[e_0^2(k)] = \alpha\sigma_n^2 + \beta\frac{1}{L+1}\overline{\gamma}^2, \tag{A.24}$$

where

$$\alpha = \begin{cases} 10 & \text{if } L = 0, \\ 2 & \text{otherwise.} \end{cases} \qquad \beta = \begin{cases} 10 & \text{if } L = 0, \\ 1 & \text{otherwise.} \end{cases}$$

The approximation above originates from experimental observations, and is a refinement of the approximation used in [33] for a simplified version of SM-AP algorithm, the SM-AP algorithm with *simple choice constraint vector* (SC-CV). Observing the SC-CV, see [2, 12], it is clear that the SM-AP algorithm with *fixed modulus error-based constraint vector* (FMEB-CV) is more sensitive to the data reuse factor L, since the SC-CV is chosen in such a way that many degrees of freedom are discarded. This dependence is represented in the approximation for $\mathrm{E}[e_0^2(k)]$.

A.5 Modeling P_{up}

Using arguments based on the central limit theorem [38] we can consider that, for a sufficiently large k, the error signal $e_0(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ will have a Gaussian distribution. This can also be observed in practice by plotting the histogram of

$e_0(k)$, as done in [8]. In addition, due to the signal model, see Definition 4, the error signal is a zero-mean random variable.

The probability of updating the filter coefficients at a certain iteration k is given by $P_{\rm up}(k) = P[|e_0(k)| > \overline{\gamma}]$. After convergence it can be written as

$$P_{\rm up} = 2Q\left(\frac{\overline{\gamma}}{\sigma_e}\right), \tag{A.25}$$

where σ_e^2 is the variance of $e_0(k)$, and $Q\left(\cdot\right)$ is the Gaussian complementary function, defined as $Q\left(x\right) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$.

If we use the independence assumption, the variance of the error could be written as $\sigma_e^2 = \sigma_n^2 + {\rm E}\left[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)\right]$, and utilizing the Rayleigh quotient [2] we can determine the region of possible values for σ_e^2

$$\sigma_n^2 + \lambda_{\rm min}{\rm E}\left[\|\Delta\mathbf{w}(k)\|^2\right] \leq \sigma_e^2 \leq \sigma_n^2 + \lambda_{\rm max}{\rm E}\left[\|\Delta\mathbf{w}(k)\|^2\right], \tag{A.26}$$

where $\lambda_{\rm min}$ and $\lambda_{\rm max}$ are the smallest and the largest eigenvalues of the autocorrelation matrix $\mathbf{R}$, respectively. Since both $\lambda_{\rm min}, \lambda_{\rm max} \geq 0$ [2], and at steady-state we expect that ${\rm E}\left[\|\Delta\mathbf{w}(k)\|^2\right]$ is small, then we can consider σ_e^2 to be close to σ_n^2. So, the following model seems reasonable

$$\sigma_e^2 = (1+\eta)\sigma_n^2, \tag{A.27}$$

where η is a small positive constant that must be chosen as

$$\eta = \begin{cases} 0.10 & \text{if input signal is uncorrelated,} \\ 0.25 & \text{otherwise.} \end{cases} \tag{A.28}$$

These values of η were empirically determined and tested in many simulation scenarios. In fact, the results are not very sensitive to the choice of η. Any choice of $\eta \in [0.1, 0.3]$ leads to good results. If some information about the autocorrelation of the input signal is available, one should choose η as in Eq. (A.28) in order to enhance the theoretical approximations.

Figure A.1 depicts the steady-state probability of update $P_{\rm up}$ for different values of L, considering the Basic Scenario of Subsection 3.4.7. As can be observed the theoretical curve follows closely the experimental one, for $L = 0$. When $L \neq 0$, however, we observe that the minimum value of $P_{\rm up}$ stabilizes at a value different from 0.

The formula for $P_{\rm up}$ given in Eq. (A.25) does not take into account that the Gaussian assumption is not valid for $L > 0$. As can be seen in Figure A.1, the steady-state probability of updating of the SM-AP algorithm increases with L, and for $L > 0$ the tail of the curve does not fall to 0. So, in order to properly estimate P_{up} we need to add a constant, leading to the following expression

$$P_{\text{up}} = 2Q\left(\frac{\gamma}{\sigma_e}\right) + P_{\text{min}}, \tag{A.29}$$

where P_{min} is a rough estimate of the smallest value that P_{up} assumes, as a function of $\overline{\gamma}$. Table A.1 summarizes the values of P_{min} that were used in our experiments. These values provided good steady-state MSE results in different scenarios, especially for values of τ that yield low steady-state probability of update, which agrees with assumption *As-6* used to simplify the expression for the EMSE of the SM-AP algorithm. Moreover, the approximation given by Eq. (A.29) has the attractive feature of not being very sensitive to small variations in P_{min}.

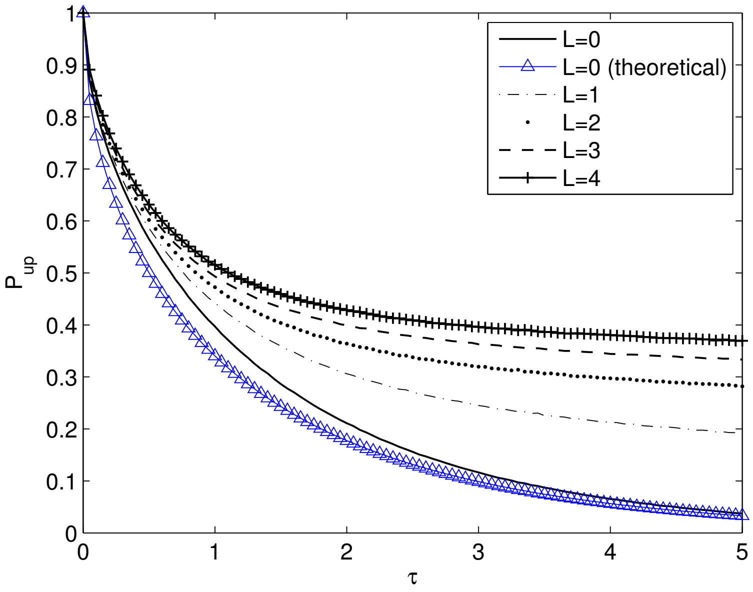


Figure A.1: *Probability of updating vs. τ, where $\overline{\gamma} = \sqrt{\tau\sigma_n^2}$, for different values of L, considering the Basic Scenario (BS).*

Table A.1: *Values of P_{min} as a function of L.*

$L \setminus$ input signal	uncorrelated	correlated
0	0	0
1	0.20	0.30
2	0.30	0.40
3	0.35	0.45
4	0.40	0.50

A.6 Assumptions and Statements

Now we discuss the assumptions and statements used in the steady-state MSE analysis. The assumptions are:

As-1 *The RV* $\mathbf{p}(k)$ *is independent of the event* $\{|e_0(k)| > \overline{\gamma}\}$: This assumption is reasonable at steady-state, when it is expected that the algorithm updates in different directions but always maintaining $\mathbf{w}(k)$ close to $\mathbf{w}_o$, no matter the value of $|e_0(k)|$.

As-2 *At steady-state* $\mathbf{S}(k)$ *is uncorrelated with the random matrices* $\underline{\tilde{\mathbf{e}}}(k)\underline{\tilde{\mathbf{e}}}^T(k)$, $\underline{\tilde{\mathbf{e}}}(k)\mathbf{n}^T(k)$, $\underline{\tilde{\mathbf{e}}}(k)\boldsymbol{\gamma}^T(k)$, $\mathbf{n}(k)\mathbf{n}^T(k)$, *and* $\mathbf{n}(k)\boldsymbol{\gamma}^T(k)$: Uncorrelation assumptions are required in all adaptive filtering analyses in order to maintain the mathematical tractability. The RVs $\mathbf{S}(k)$ and $\mathbf{n}(k)\mathbf{n}^T(k)$ are uncorrelated due to Definition 4. The other uncorrelation assumptions are motivated by the fact that $\mathbf{S}(k)$ varies slowly with k, especially for high values of N.

As-3 *Diagonally dominant assumption*: Since $\mathrm{E}\big[\tilde{\mathbf{e}}(k)\tilde{\mathbf{e}}^T(k)\big]$, $\mathrm{E}\big[\boldsymbol{\gamma}(k)\boldsymbol{\gamma}^T(k)\big]$, and $\mathrm{E}\big[\mathbf{n}(k)\mathbf{n}^T(k)\big]$ are autocorrelation matrices, they have higher values on the main diagonal. In addition, from Definition 2 and Result 2, the cross-correlation matrices presented in Eq. (3.39) can be written as a sum of one of the autocorrelation matrices with the cross-correlation matrix $\mathrm{E}\left[\tilde{\mathbf{e}}(k)\mathbf{n}^T(k)\right] \approx \mathbf{0}$, see Eq. (3.44).

As-4 *At steady-state* $\tilde{e}_i(k-1)$ *is a zero-mean Gaussian RV*: By using this assumption together with the distribution of $n(k)$ given in Definition 4, then $n(k-1-i)$ and $e_i(k-1)$ as well as $\tilde{e}_i(k-1)$ and $e_i(k-1)$ are jointly Gaussian RVs [38]. So, we can apply Result 2 to Eq. (A.15).

As-5 *At steady-state* $\rho_i(k) \approx \rho_0(k)$, *for* $i = 0, \ldots, L$: This is reasonable for small values of L, e.g., for $L = 0$, the relation given by assumption *As-5* is an equality rather than an approximation.

As-6 $P_{\text{up}} \ll 1$, *or* $\mu \ll 1$, *or* $P_{\text{up}}\mu \ll 1$: Since $\mu = 1$ for the SM-AP algorithm, then we must have $P_{\text{up}} \ll 1$. This is not true for small values of $\overline{\gamma}$. For example, in the limiting case where $\overline{\gamma} \to 0$ we have $P_{\text{up}} \to 1$. This implies that the proposed theoretical MSE expressions of the SM-AP algorithm are not so accurate for small values of $\overline{\gamma}$. However, from Chebyshev's inequality [8, 38] we know that $P[\mathbf{w}_o \in \mathcal{H}(k)] = P[|d(k) - \mathbf{w}_o^T\mathbf{x}(k)| \leq \overline{\gamma}] = 1 - P[|n(k)| > \overline{\gamma}] \geq 1 - (\sigma_n/\overline{\gamma})^2$, i.e., in order to have $\mathbf{w}_o \in \mathcal{H}(k)$ with high probability we must choose $\overline{\gamma} \gg \sigma_n$ (in fact, we know that Chebyshev's inequality provides a conservative lower bound; for $n(k)$ satisfying Definition4(c), e.g., choosing $\overline{\gamma} = 2\sigma_n$ makes $P[\mathbf{w}_o \in \mathcal{H}(k)] \approx 95\%$). On the other hand, for too large values of $\overline{\gamma}$, the algorithm may not update at all, thus not converging to a point close to $\mathbf{w}_o$.

As-7 *The elements on the main diagonal of* $\mathrm{E}\left[\mathbf{S}(k)\right]$ *are equal*: This assumption on the input-signal model is required to maintain the mathematical tractability of the problem at hand. Note that, for $L = 0$ this is always true (i.e., not an assumption), and for $L = 1$ this is equivalent to satisfy $\mathrm{E}\left[\|\mathbf{x}(k)\|^2\right] = \mathrm{E}\left[\|\mathbf{x}(k-1)\|^2\right]$, which is very likely to be a good approximation, especially for long vectors (since the difference between the terms on the left-hand side and right-hand side corresponds to just one sample/element), or for well behaved input signals (e.g., stationary signals). In addition, since $\mathbf{A}_1$ is a diagonal matrix, the ratio $\frac{\mathrm{tr}\{\mathbf{A}_1 \mathrm{E}[\mathbf{S}(k)]\}}{\mathrm{tr}\{\mathrm{E}[\mathbf{S}(k)]\}}$ represents a weighted mean of the elements on the main diagonal of $\mathbf{A}_1$, whose weights are the diagonal elements of $\mathrm{E}\left[\mathbf{S}(k)\right]$. This assumption enables us to exchange the weighted mean by an arithmetic mean, which is much easier to solve and also avoids the problem of determining the weights.

The statements used in the analysis are:

St-1 *$\mathbf{X}(k)$ has full column rank*: This guarantees the existence of $\mathbf{R}^{-1}(k)$ and is usually true for a tapped-delay-line structure with a random input signal during steady-state. An example of exception would be a signal that is constant during a long interval, but this is not likely to occur with random input signals.

St-2 *At steady-state $\mathrm{E}\left[e_i^2(k-1)\right] = \mathrm{E}\left[e_i^2(k)\right]$ and $\mathrm{E}\left[\tilde{e}_i^2(k-1)\right] = \mathrm{E}\left[\tilde{e}_i^2(k)\right]$*: Since the starting point of the analysis is to assume that the algorithm converged in order to analyze its steady-state behavior, then the sequence $\{\mathrm{E}\left[e_i^2(k)\right]\}_{k\in\mathbb{N}}$ converges and, therefore, the first equality in *St-2* always holds at steady-state. The second equality follows by a combination of the first equality and the noise model given in Definition 4.

St-3 *At steady-state $\mathrm{E}\left[n(k-1-i)\tilde{e}_i(k-1)\right]$ can be neglected*: This statement implies that $\mathrm{E}\left[n(k-1-i)\tilde{e}_i(k-1)\right] \approx 0$, for all i. Recalling Result 1 and Definition 4, we know that, for $i = 0$, $\mathrm{E}\left[n(k-1)\tilde{e}_0(k-1)\right] = 0$. Note that St-3 becomes less accurate as i grows (or equivalently, for large values of L).

St-4 *$\mathbf{S}(k) \approx \mathbf{R}^{-1}(k)$*: Follows from *St-1* and the fact that δ is chosen as a small constant used to avoid numerical instability problems that may occur especially in the first iterations, i.e., $\delta \ll 1$ as described in Subsection 3.2.2.

Appendix B

Proofs Related to Chapter 4

In this appendix we provide demonstration to some of the facts and theorems invoked in Chapter 4.

B.1 Proof that $\mathbf{p}_1$ is orthogonal to $\mathbf{p}_2$

Here we show that $\mathbf{p}_1$ is orthogonal to $\mathbf{p}_2$. One must remember that the regularization factor δ in $\mathbf{S}(k) = [\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}]^{-1}$ was artificially added just to avoid numerical issues and, therefore, should be chosen as $0 < \delta \ll 1$, which means that $\delta\mathbf{I} \approx \mathbf{0}$. In the following derivation we consider $\delta = 0$:

$$\begin{aligned}
\mathbf{p}_1^T\mathbf{p}_2 &= [\mathbf{e}(k) - \boldsymbol{\gamma}(k)]^T\, \mathbf{S}^T(k)\mathbf{X}^T(k)\frac{\alpha}{2}\left[\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}\right]\mathbf{f}_\beta(\mathbf{w}(k)) \\
&= [\mathbf{e}(k) - \boldsymbol{\gamma}(k)]^T\, \mathbf{S}^T(k)\frac{\alpha}{2}\left[\mathbf{X}^T(k)\mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{X}^T(k)\right]\mathbf{f}_\beta(\mathbf{w}(k)) \\
&= [\mathbf{e}(k) - \boldsymbol{\gamma}(k)]^T\, \mathbf{S}^T(k)\frac{\alpha}{2}\left[\mathbf{X}^T(k) - \mathbf{X}^T(k)\right]\mathbf{f}_\beta(\mathbf{w}(k)) = 0. \qquad \text{(B.1)}
\end{aligned}$$

B.2 Proof of Theorem 1

We prove Theorem 1 using an approach that is somewhat indirect, but has the advantages of highlighting geometric interpretation while avoiding cumbersome expressions. In what follows we prove two lemmas that facilitate the proof of the theorem.

First, let us define $\mathbf{w}_\alpha$ as the $\mathbf{w}(k+1)$ corresponding to the SSM-AP algorithm, i.e., $\mathbf{w}_\alpha \triangleq \mathbf{w}(k+1)$ given in Eq. (4.20). Thus, $\mathbf{w}_0$ represents the SSM-AP's $\mathbf{w}(k+1)$ using $\alpha = 0$. In addition, $\|\cdot\|$ denotes the 2-norm and recall that $\mathbf{w}_*$ is the impulse response of the unknown system.

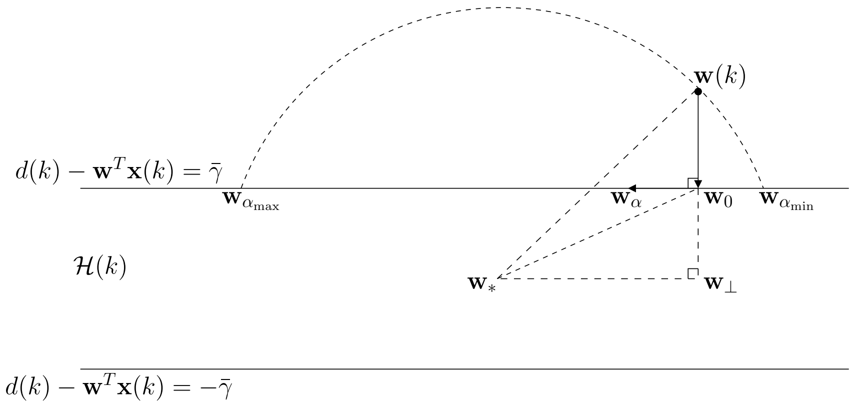

$\mathbf{w}(k)$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = \bar{\gamma}$
$\mathbf{w}_{\alpha_{\max}}$
$\mathbf{w}_\alpha$
$\mathbf{w}_0$
$\mathbf{w}_{\alpha_{\min}}$
$\mathcal{H}(k)$
$\mathbf{w}_*$
$\mathbf{w}_\perp$
$d(k) - \mathbf{w}^T\mathbf{x}(k) = -\bar{\gamma}$

Figure B.1: *Geometric interpretation of the SSM-AP algorithm considering $L = 0$ aiming at explaining how to choose α properly. The choice of $L = 0$ allows for a clear figure, but we highlight that the same relations and angles are also valid for general L (the only difference is that instead of a single hyperplane, when $L \neq 0$ we have an intersection of the last $L + 1$ hyperplanes).*

Lemma 1. $\|\mathbf{w}_* - \mathbf{w}_0\|^2 \leq \|\mathbf{w}_* - \mathbf{w}(k)\|^2$.

Proof. Consider the geometric construction shown in Figure B.1, which depicts the updating process of the SSM-AP algorithm. Observe that $\|\mathbf{w}_* - \mathbf{w}_0\|$ and $\|\mathbf{w}_* - \mathbf{w}(k)\|$ are the lengths of the hypotenuses of the two right triangles exhibited in this figure. Using the Pythagorean theorem it follows that

$$\begin{aligned}\|\mathbf{w}_* - \mathbf{w}_0\|^2 &= \|\mathbf{w}_* - \mathbf{w}_\perp\|^2 + \|\mathbf{w}_\perp - \mathbf{w}_0\|^2 \\ &< \|\mathbf{w}_* - \mathbf{w}_\perp\|^2 + \|\mathbf{w}_\perp - \mathbf{w}(k)\|^2 \\ &= \|\mathbf{w}_* - \mathbf{w}(k)\|^2, \qquad \text{(B.2)}\end{aligned}$$

where the inequality follows from the fact that we are assuming that an update occurs, i.e., $|e_0(k)| = |d(k) - \mathbf{w}^T(k)\mathbf{x}(k)| > \overline{\gamma}$ implying that $\mathbf{w}(k) \notin \mathcal{H}(k)$ and hence does not belong to the intersection of the last $L + 1$ constraint sets. Therefore, $\|\mathbf{w}_* - \mathbf{w}_0\|^2 < \|\mathbf{w}_* - \mathbf{w}(k)\|^2$ in such cases. On the other hand, when the algorithm does not update we have $\mathbf{w}_0 = \mathbf{w}(k)$, thus $\|\mathbf{w}_* - \mathbf{w}_0\|^2 = \|\mathbf{w}_* - \mathbf{w}(k)\|^2$, completing the proof. □

Lemma 2. *Let us define* $c \triangleq [\mathbf{e}(k) - \boldsymbol{\gamma}(k)]^T \mathbf{S}(k) [\mathbf{e}(k) - \boldsymbol{\gamma}(k)] + 2\Delta\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k) [\mathbf{e}(k) - \boldsymbol{\gamma}(k)]$*, where* $\Delta\mathbf{w}(k) \triangleq \mathbf{w}(k) - \mathbf{w}_*$*. Then, Lemma 1 implies* $c \leq 0$.

Proof. From Lemma 1 we know that $\|\mathbf{w}_* - \mathbf{w}_0\|^2 - \|\mathbf{w}_* - \mathbf{w}(k)\|^2 \leq 0$, where the equality holds only when there is no update, i.e., $\mathbf{w}_0 = \mathbf{w}(k)$. Otherwise, an update takes place and $\mathbf{w}_0$ can be related to $\mathbf{w}(k)$ via Eq. (4.20) employing $\alpha = 0$.

Expanding the first term of the inequality $\|\mathbf{w}_* - \mathbf{w}_0\|^2 - \|\mathbf{w}_* - \mathbf{w}(k)\|^2 < 0$ we obtain

$$\begin{aligned}
\|\mathbf{w}_* - \mathbf{w}_0\|^2 =& \|\mathbf{w}_*\|^2 - 2\mathbf{w}_*^T\mathbf{w}_0 + \|\mathbf{w}_0\|^2 \\
=& \|\mathbf{w}_*\|^2 - 2\mathbf{w}_*^T \left\{\mathbf{w}(k) + \mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]\right\} \\
& + \Big\{\|\mathbf{w}(k)\|^2 + 2\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] \\
& \quad + \left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}^T(k)\mathbf{X}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]\Big\} \\
=& \|\mathbf{w}_* - \mathbf{w}(k)\|^2 - 2\mathbf{w}_*^T\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] \\
& + 2\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] + \left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] \\
=& \|\mathbf{w}_* - \mathbf{w}(k)\|^2 + 2\Delta\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] \\
& + \left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right], \\
=& \|\mathbf{w}_* - \mathbf{w}(k)\|^2 + c\,, \qquad \text{(B.3)}
\end{aligned}$$

where in the third equality we used $\mathbf{S}(k) = \left[\mathbf{X}^T(k)\mathbf{X}(k)\right]^{-1}$, (i.e., $\delta = 0$, as explained in B.1), in the fourth equality we used the definition of $\Delta\mathbf{w}(k)$, and in the last equality the definition of c was used. By substituting Eq. (B.3) into $\|\mathbf{w}_* - \mathbf{w}_0\|^2 - \|\mathbf{w}_* - \mathbf{w}(k)\|^2 < 0$ it follows that $c < 0$. In addition, $c = 0$ when no update occurs. Therefore, $c \leq 0$. □

Now we are ready to prove the theorem by obtaining the values of α that satisfy the inequality $\|\mathbf{w}_* - \mathbf{w}_\alpha\|^2 \leq \|\mathbf{w}_* - \mathbf{w}(k)\|^2$. First, we define $f(\alpha) \triangleq \|\mathbf{w}_* - \mathbf{w}_\alpha\|^2 - \|\mathbf{w}_* - \mathbf{w}(k)\|^2$. Then, similarly to what we have done in Lemma 2, we expand $\mathbf{w}_\alpha$ using Eq. (4.20), but now considering any real-valued α. As a result we obtain:

$$f(\alpha) = a\alpha^2 + b\alpha + c, \qquad \text{(B.4)}$$

where

$$\begin{aligned}
a &= \frac{\|\mathbf{P}^\perp \mathbf{f}_\beta(\mathbf{w}(k))\|^2}{4}, && \text{(B.5)} \\
b &= \Delta\mathbf{w}^T(k)\mathbf{P}^\perp\mathbf{f}_\beta(\mathbf{w}(k)), && \text{(B.6)} \\
c &= \left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] + 2\Delta\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right], && \text{(B.7)}
\end{aligned}$$

and $\mathbf{P}^\perp \triangleq \mathbf{X}(k)\mathbf{S}(k)\mathbf{X}^T(k) - \mathbf{I}$.

Eq. (B.4) shows that $f(\alpha)$ is a second-order polynomial in α. Since $a > 0$ (except for $\mathbf{P}^\perp\mathbf{f}_\beta(\mathbf{w}(k)) = \mathbf{0}$, which does not occur in practice due to floating point arithmetic), $f(\alpha)$ corresponds to a convex parabola. In addition, its discriminant is

$d \geq 0$ because

$$d = \underbrace{b^2}_{\geq 0} - \underbrace{4ac}_{\leq 0} \geq 0, \tag{B.8}$$

where we have used Lemma 2. In fact, if an update occurs and $\mathbf{P}^{\perp}\mathbf{f}_{\beta}(\mathbf{w}(k)) \neq \mathbf{0}$, then $a > 0$ and $c < 0$, implying that $ac < 0 \Rightarrow d > 0$. This means that $f(\alpha)$ has two distinct roots, viz. $\alpha_{\min}$ and $\alpha_{\max}$, which are given by $\alpha = (-b \pm \sqrt{d})/(2a)$. Therefore, we should choose α satisfying $\alpha_{\min} < \alpha < \alpha_{\max}$ in order for the SSM-AP algorithm to produce estimates $\mathbf{w}(k+1)$ (or equivalently, $\mathbf{w}_{\alpha}$) that are closer to $\mathbf{w}_*$ than $\mathbf{w}(k)$. This completes the proof of Theorem 1.

Finally, observe that this theorem matches our geometric intuition. Indeed, in Figure B.1 we draw part of the circle with center in $\mathbf{w}_*$ and radius $\|\mathbf{w}_* - \mathbf{w}(k)\|$. The points where this circle intersects the closest hyperplane are the $\mathbf{w}_{\alpha_{\min}}$ and $\mathbf{w}_{\alpha_{\max}}$.

B.3 Proof of Theorem 2

Here we use the same idea and Lemmas of Appendix B.2. First, let us define $\mathbf{w}_{\alpha}$ as the $\mathbf{w}(k+1)$ corresponding to the QSSM-AP algorithm, i.e., $\mathbf{w}_{\alpha} \triangleq \mathbf{w}(k+1)$ given in Eq. (4.21). Thus, $\mathbf{w}_0$ represents the QSSM-AP's $\mathbf{w}(k+1)$ using $\alpha = 0$ and, in this case, it coincides with the expression of the SSM-AP algorithm in the same condition. In addition, $\|\cdot\|$ denotes the 2-norm and recall that $\mathbf{w}_*$ is the impulse response of the unknown system.

We define $f_q(\alpha) \triangleq \|\mathbf{w}_* - \mathbf{w}_{\alpha}\|^2 - \|\mathbf{w}_* - \mathbf{w}(k)\|^2$. By expanding $\mathbf{w}_{\alpha}$ using Eq. (4.21) we obtain

$$f_q(\alpha) = a_q\alpha^2 + b_q\alpha + c_q, \tag{B.9}$$

where

$$a_q = \frac{\|\mathbf{f}_{\beta}(\mathbf{w}(k))\|^2}{4}, \tag{B.10}$$

$$b_q = -\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}(k)\mathbf{X}^T(k)\mathbf{f}_{\beta}(\mathbf{w}(k)), \tag{B.11}$$

$$c_q = \left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right]^T \mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] + 2\Delta\mathbf{w}^T(k)\mathbf{X}(k)\mathbf{S}(k)\left[\mathbf{e}(k) - \boldsymbol{\gamma}(k)\right] \triangleq c. \tag{B.12}$$

So, $f_q(\alpha)$ is a second-order polynomial in α. Since $a_q > 0$ (except for $\mathbf{f}_{\beta}(\mathbf{w}(k)) = \mathbf{0}$, or equivalently, $\mathbf{w}(k) = \mathbf{0}$, which does not occur in practice due to floating point arithmetic), $f_q(\alpha)$ corresponds to a convex parabola. In addition, its discriminant is $d_q \geq 0$ because

$$d_q = \underbrace{b_q^2}_{\geq 0} - \underbrace{4a_q c_q}_{\leq 0} \geq 0, \tag{B.13}$$

where we have used Lemma 2. In fact, if an update occurs and $\mathbf{w}(k) \neq \mathbf{0}$, then $a_q > 0$ and $c_q < 0$, implying that $a_q c_q < 0 \Rightarrow d_q > 0$. This means that $f_q(\alpha)$ has two distinct roots, viz. $\alpha_{q,\min}$ and $\alpha_{q,\max}$, which are given by $\alpha = (-b_q \pm \sqrt{d_q})/(2a_q)$. Therefore, we should choose α satisfying $\alpha_{q,\min} < \alpha < \alpha_{q,\max}$ in order for the QSSM-AP algorithm to produce estimates $\mathbf{w}(k+1)$ (or equivalently, $\mathbf{w}_\alpha$) that are closer to $\mathbf{w}_*$ than $\mathbf{w}(k)$. This completes the proof of Theorem 2.

List of Publications and Software

This section lists the scientific production (i.e., publications and software) of this D.Sc. candidate. In summary, such production corresponds to:

- One book;
- One chapter of book;
- Five articles in international journals;
- Nine articles in international congresses;
- Four articles in national congresses;
- Two software.

In order to facilitate the identification of the type of publication, we use the following tags: B = book; Ch = book chapter; J = journal; C = congress; S = software. The items in the scientific production are:

[B1] Diniz, P. S. R.; Martins, W. A.; **Lima, M. V. S.**, "Block Transceivers: OFDM and Beyond," Editor: Morgan & Claypool, 2012, 206 pages.

[Ch1] Biscainho, L. W. P.; Martins, W. A.; **Lima, M. V. S.**; Nunes, L. O., "Solution Manual of the book: Adaptive Filtering: Algorithms and Practical Implementation (Third Edition) - P. S. R. Diniz," Chapters 6 and 13.

[J1] **Lima, M. V. S.**; Diniz, P. S. R., "Steady-State MSE Performance of the Set-Membership Affine Projection Algorithm," Circuits, Systems, and Signal Processing, v. 32, n. 4, pp. 1811-1837, 2013.

[J2] Nunes, L. O.; Martins, W. A.; **Lima, M. V. S.**; Biscainho, L. W. P.; Costa, M. V. M.; Gonçalves, F. M.; Said, A.; Lee, B., "A Steered-Response Power Algorithm Employing Hierarchical Search for Acoustic Source Localization Using Microphone Arrays," accepted for publication in IEEE Transactions on Signal Processing.

[J3] **Lima, M. V. S.**; Martins, W. A.; Nunes, L. O.; Biscainho, L. W. P.; Ferreira, T. N.; Costa, M. V. M.; Gonçalves, F. M.; Said, A.; Lee, B., "Efficient Steered-Response Power Methods for Sound Source Localization Using Microphone Arrays," submitted for publication in IEEE Transactions on Audio, Speech, and Language Processing.

[J4] **Lima, M. V. S.**; Martins, W. A.; Ferreira, T. N.; Diniz, P. S. R., "Sparsity-Aware Data-Selective Adaptive Filters," submitted for publication in IEEE Transactions on Signal Processing.

[J5] **Lima, M. V. S.**; Freeland, F. P.; Diniz, P. S. R., "Perception-Based Acoustic Echo Cancellation," to be submitted to IEEE Signal Processing Letters.

[C01] Martins, W. A.; **Lima, M. V. S.**; Diniz, P. S. R., "Semi-blind data-selective equalizers for QAM," The 9th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2008), Recife, Brazil, pp. 501-505, 2008.

[C02] Diniz, P. S. R.; **Lima, M. V. S.**; Martins, W. A., "Semi-blind data-selective algorithms for channel equalization," IEEE International Symposium on Circuits and Systems (ISCAS 2008), Seattle, USA, pp. 53-56, 2008.

[C03] Espindola, B. N.; **Lima, M. V. S.**; Diniz, P. S. R., "Toolbox de filtragem adaptativa para MATLAB," Simpósio Brasileiro de Telecomunicações (SBrT 2009), Blumenau, Brazil, pp. 1-2, 2009.

[C04] **Lima, M. V. S.**; Diniz, P. S. R., "On the steady-state MSE performance of the set-membership NLMS algorithm," The 7th International Symposium on Wireless Communication Systems (ISWCS 2010), York, UK, pp. 389-393, 2010.

[C05] **Lima, M. V. S.**; Diniz, P. S. R., "Steady-state analysis of the set-membership affine projection algorithm," The 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2010), Dallas, USA, pp. 3802-3805, 2010.

[C06] **Lima, M. V. S.**; Espindola, B. N.; Freeland, F. P.; Diniz, P. S. R., "Applications of data-selective adaptive filters," Simpósio Brasileiro de Telecomunicações (SBrT 2011), Curitiba, Brazil, pp. 1-5, 2011.

[C07] Nunes, L. O.; Martins, W. A.; **Lima, M. V. S.**; Biscainho, L. W. P.; Lee, B.; Said, A.; Schafer, R., "Discriminability measure for microphone array source localization," International Workshop on Acoustic Signal Enhancement (IWAENC 2012), Aachen, Germany, pp. 1-5, 2012.

[C08] **Lima, M. V. S.**; Gussen, C. M. G.; Espindola, B. N.; Ferreira, T. N.; Martins, W. A.; Diniz, P. S. R., "Open-source physical-layer simulator for LTE systems," The 37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), Kyoto, Japan, pp. 2781-2784, 2012.

[C09] Martins, W. A.; Nunes, L. O.; Haddad, D. B.; Biscainho, L. W. P.; **Lima, M. V. S.**; Costa, M. V. M.; Lee, B., "Time-of-flight selection for improved acoustic sensor localization using multiple loudspeakers," Simpósio Brasileiro de Telecomunicações (SBrT 2013), Fortaleza, Brazil, pp. 1-5, 2013.

[C10] Fonini, P. A. M.; **Lima, M. V. S.**; Diniz, P. S. R., "Estudo da interferência entre sistemas de comunicações baseados em C-OFDM e W-CDMA," Simpósio Brasileiro de Telecomunicações (SBrT 2013), Fortaleza, Brazil, pp. 1-2, 2013.

[C11] **Lima, M. V. S.**; Diniz, P. S. R., "Fast learning set theoretic estimation," The 21st European Signal Processing Conference (EUSIPCO 2013), Marrakech, Morocco, pp. 1-5, 2013.

[C12] **Lima, M. V. S.**; Martins, W. A.; Diniz, P. S. R., "Affine projection algorithms for sparse system identification," The 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013), Vancouver, Canada, pp. 5666-5670, 2013.

[C13] **Lima, M. V. S.**; Sobron, I.; Martins, W. A.; Diniz, P. S. R., "Stability and MSE analyses of affine projection algorithms for sparse system identification," The 39th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014), Florence, Italy (submitted).

[S1] Pinto, G. O.; **Lima, M. V. S.**; Martins, W. A.; Biscainho, L. W. P.; Diniz, P. S. R., "Adaptive_Filtering_Toolbox_v4 (complete version)," 2011.

[S2] **Lima, M. V. S.**; Gussen, C. M. G.; Espindola, B. N.; Ferreira, T. N.; Martins, W. A.; Diniz, P. S. R., "LTE Physical-Layer Simulator (LTE-LPS), version 0.7," 2011.

Some of the aforementioned publications have been used in this thesis. This is the case of: [J1], [J4], [C05], and [C11], which were used in Chapter 3; [J4], [C12], and [C13], which were used in Chapter 4; [J5], [C06], and [C08] which were used in Chapter 5.