



## AVANÇOS EM ALGORITMOS ADAPTATIVOS CONSCIENTES DA ESTRUTURA DO SISTEMA A SER IDENTIFICADO

Leonardo Oliveira dos Santos

Dissertação de Mestrado 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 Mestre em Engenharia Elétrica.

Orientadores: Mariane Rembold Petraglia  
Diego Barreto Haddad

Rio de Janeiro  
Fevereiro de 2021

AVANÇOS EM ALGORITMOS ADAPTATIVOS CONSCIENTES DA  
ESTRUTURA DO SISTEMA A SER IDENTIFICADO

Leonardo Oliveira dos Santos

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

Orientadores: Mariane Rembold Petraglia  
Diego Barreto Haddad

Aprovada por: Prof<sup>a</sup>. Mariane Rembold Petraglia  
Prof. Diego Barreto Haddad  
Prof. Julio Cesar Boscher Torres  
Prof. Michel Pompeu Tcheou

RIO DE JANEIRO, RJ – BRASIL  
FEVEREIRO DE 2021

Oliveira dos Santos, Leonardo

Avanços em Algoritmos Adaptativos Conscientes da Estrutura do Sistema a Ser Identificado/Leonardo Oliveira dos Santos. – Rio de Janeiro: UFRJ/COPPE, 2021.

XII, 62 p.: il.; 29, 7cm.

Orientadores: Mariane Rembold Petraglia

Diego Barreto Haddad

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2021.

Referências Bibliográficas: p. 56 – 62.

1. Filtragem Adaptativa.
2. Sistemas Esparsos.
3. Análise Transiente. I. Rembold Petraglia, Mariane *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*À minha mãe Mônica.  
Ao meu pai Delcídes.  
À minha Família.  
Aos meus Amigos.*

# Agradecimentos

Eu não tenho nem palavras para agradecer o apoio e o carinho de todas as pessoas que percorreram essa jornada comigo, e que sempre tinham alguma palavra de consolo quando eu pensava em desistir pelo menos umas 3 vezes por dia. Por isso, eu gostaria muito de fazer um texto bonito pra mostrar o quão grato eu sou a todos que sempre davam um jeito de me colocar nos eixos de novo, mas essas pessoas sabem que me expressar não é uma das minhas qualidades. Então, eu queria apenas agradecer a toda minha família, aos meus amigos: Carlos Lordelo, Diego Freitas, Fábio Oliveira, Felipe Petraglia, Gabriel Torres, Gustavo Nunes, Henrique Dias, João Victor, Lucas Oliveira, Marcos Paulo, Michel Morais, e aos companheiros do IPQM: Augusto, Cátia, Corrêa, Fabiano, Gabriel, Izabel, Marcela, Monnerat, Sidney, William, por todo carinho e apoio que recebi desde o meu primeiro dia no instituto.

Gostaria ainda de fazer um agradecimento em especial aos meus orientadores: prof. Diego Haddad e profa. Mariane Petraglia por todo apoio, disponibilidade e paciência. Sem eles essa dissertação não teria sido possível.

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

## AVANÇOS EM ALGORITMOS ADAPTATIVOS CONSCIENTES DA ESTRUTURA DO SISTEMA A SER IDENTIFICADO

Leonardo Oliveira dos Santos

Fevereiro/2021

Orientadores: Mariane Rembold Petraglia

Diego Barreto Haddad

Programa: Engenharia Elétrica

Sistemas que apresentam uma determinada estrutura, dentre os quais os esparsos, costumam ser elencados nas mais diversas áreas de aplicação da filtragem adaptativa, dentre elas controle, engenharia biomédica e cancelamento de eco. Algoritmos capazes de usar essa propriedade para acelerar a convergência e/ou melhorar o erro em regime permanente vêm ganhando cada vez mais destaque na comunidade acadêmica.

Motivado por essas experiências positivas, este trabalho propõe um arcabouço matemático capaz de gerar diferentes algoritmos adaptativos, sendo eles conscientes de esparsidade ou não, através do método de multiplicadores de Lagrange. Utilizando este arcabouço matemático, é proposto o algoritmo  $\ell_2$ -norm Feature LMS, o qual é capaz de penalizar soluções que não adiram a uma determinada estrutura. As simulações apresentadas neste trabalho mostram que o algoritmo aqui proposto é capaz de superar o desempenho do algoritmo LMS, o qual é usado para comparação.

Neste trabalho, também é proposto um modelo estocástico do algoritmo  $\ell_2$ -norm Feature LMS, tanto para o caso em que o filtro adaptativo tem o mesmo comprimento do filtro ótimo, como para o caso em que o filtro adaptativo possui um comprimento menor devido a alguma restrição. As simulações apresentadas neste trabalho mostram que o modelo teórico é capaz de fornecer estimativas fiéis das curvas de aprendizado tanto do desvio médio quadrático (MSD) do vetor de coeficientes quanto do erro médio quadrático (MSE).

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## ADVANCES IN STRUCTURE-AWARE ADAPTIVE ALGORITHMS

Leonardo Oliveira dos Santos

February/2021

Advisors: Mariane Rembold Petraglia

Diego Barreto Haddad

Department: Electrical Engineering

Sparse systems are common in several areas of application of adaptive filtering, including control, biomedical engineering and echo cancellation. Algorithms that are able to use such system property to accelerate convergence and/or to improve the steady-state error have gained increasing prominence in the academic community.

Motivated by these positive experiences, this work presents a mathematical framework capable of generating different adaptive algorithms, conscious of sparsity or not, by the Lagrange multipliers method. Using this mathematical framework, the  $\ell_2$ -norm Feature LMS algorithm is devised, which is capable of exploiting the hidden structure of the system that the adaptive filter intends to emulate. The simulations presented in this work show that the algorithm proposed here outperforms the LMS algorithm, the benchmark for adaptive filtering algorithms.

In this work, a stochastic model for the  $\ell_2$ -norm Feature LMS algorithm is also proposed, for both cases when the adaptive filter has the same length as the optimum filter and when the adaptive filter presents deficient length due to some restriction. The simulations presented in this work show that the theoretical model is capable of providing accurate predictions for the mean-square deviation (MSD) of the coefficient vector and mean-square error (MSE) learning curves.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Métricas de Esparsidade</b>	<b>4</b>
2.1 Introdução . . . . .	4
2.2 Derivação de Algoritmos Conscientes de Esparsidade Normalizados . . . . .	5
2.3 Estudo de caso . . . . .	8
2.4 Resultados . . . . .	9
2.5 Comentários Finais . . . . .	14
<b>3 <math>\ell_2</math>-norm Feature LMS</b>	<b>20</b>
3.1 Introdução . . . . .	20
3.2 Derivação do Algoritmo $\ell_2$ -norm Feature LMS . . . . .	21
3.3 Resultados . . . . .	23
3.4 Comentários Finais . . . . .	25
<b>4 Análise Estatística <math>\ell_2</math>-norm Feature LMS</b>	<b>26</b>
4.1 Introdução . . . . .	26
4.2 Métodos de Análise . . . . .	26
4.3 Alguns Comentários Sobre as Hipóteses . . . . .	27
4.4 Análise de Primeira Ordem . . . . .	28
4.5 Análise de Segunda Ordem . . . . .	30
4.6 Identificação de Sistemas Variantes no Tempo . . . . .	34
4.7 Resultados . . . . .	37
4.8 Comentários Finais . . . . .	39
<b>5 <math>\ell_2</math>-norm Feature LMS Submodelado</b>	<b>42</b>
5.1 Introdução . . . . .	42
5.2 Considerações Iniciais . . . . .	43



5.3	Análise de Primeira Ordem . . . . .	43
5.4	Análise de Segunda Ordem . . . . .	45
5.5	Resultados . . . . .	50
5.6	Comentários Finais . . . . .	51
<b>6</b>	<b>Conclusão</b>	<b>54</b>
6.1	Publicações Correlatas . . . . .	55
	<b>Referências Bibliográficas</b>	<b>56</b>

# Lista de Figuras

1.1	Diagrama de blocos de um algoritmo adaptativo. . . . .	2
2.1	(a) Resposta ao impulso da planta 1. (b) Resposta ao impulso da planta 2. . . . .	8
2.2	(a) Evolução do MSD do algoritmo adaptativo usando $G_3$ para $\kappa = 10^{-7}$ . (b) Evolução do MSD do algoritmo adaptativo usando $G_3$ para $\kappa = 10^{-6}$ . . . . .	11
2.3	(a) Evolução do MSD do algoritmo adaptativo usando $G_7$ . (b) Evolução do MSD do algoritmo adaptativo usando $G_9$ . . . . .	12
2.4	(a) Evolução do MSD do algoritmo adaptativo usando $G_6$ . (b) Evolução do MSD do algoritmo adaptativo usando $G_8$ . . . . .	13
2.5	(a) Evolução do MSD do algoritmo adaptativo usando $F_1$ . (b) Evolução do MSD do algoritmo adaptativo usando $F_4$ . . . . .	14
2.6	(a) Evolução do MSD do algoritmo adaptativo usando $F_2$ . (b) Evolução do MSD do algoritmo adaptativo usando $F_5$ . . . . .	15
2.7	Evolução do MSD para todos os algoritmos otimizados para desempenho máximo, assim como do NLMS, para a planta 1. . . . .	16
2.8	Evolução do MSD para todos os algoritmos mais o NLMS para a planta 2. . . . .	17
2.9	Média do MSD em regime permanente para todos os algoritmos mais o NLMS para a planta 1. . . . .	18
2.10	Evolução do MSD para todos os algoritmos cada qual com seus parâmetros específicos, mais o NLMS para a planta 1. . . . .	19
3.1	Comparação da evolução do MSD simulado dos algoritmos LMS (em vermelho), $\ell_2$ -norm-F-LMS (em verde) e F-LMS, utilizando o filtro ótimo $w_t^*$ . (a) Utilizando $\beta = 0.015$ para todos. (b) Utilizando $\beta = 0.0083$ , $\beta = 0.015$ e $\beta = 0.014$ , respectivamente. . . . .	24

3.2	Comparação da evolução do MSD simulado dos algoritmos LMS (em vermelho), $\ell_2$ -norm-F-LMS (em verde) e F-LMS, utilizando o filtro ótimo $\mathbf{w}_h^*$ . (a) Utilizando $\beta = 0.015$ para todos. (b) Utilizando $\beta = 0.0135$ , $\beta = 0.015$ e $\beta = 0.0145$ , respectivamente. . . . .	25
4.1	Comparação da evolução de alguns coeficientes $w_i(k)$ , na média, entre simulado (em azul) e teórico (em vermelho), com $i \in 11, 12, 19$ . (a) Utilizando $\mathbf{w}_l^*$ . (b) Utilizando $\mathbf{w}_h^*$ . . . . .	38
4.2	Comparação do MSD, em regime permanente, simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS. A performance, na média, do algoritmo LMS é mostrada em verde. (a) Utilizando $\mathbf{w}_l^*$ . (b) Utilizando $\mathbf{w}_h^*$ . . . . .	39
4.3	Comparação da evolução do MSD simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS. A performance, na média, do algoritmo LMS é mostrada em verde. (a) Utilizando $\mathbf{w}_l^*$ . (b) Utilizando $\mathbf{w}_h^*$ . . . . .	40
4.4	Comparação do MSE, em regime permanente, simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS, com planta variante no tempo. (a) Utilizando $\mathbf{w}_l^*$ . (b) Utilizando $\mathbf{w}_h^*$ . . . . .	41
4.5	Comparação da evolução do MSD simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS, com ruído aditivo não-estacionário. (a) Utilizando $\mathbf{w}_l^*$ . (b) Utilizando $\mathbf{w}_h^*$ . . . . .	41
5.1	Comparação da evolução de alguns coeficientes $w_i(k)$ , na média, entre simulado (em azul) e teórico (em vermelho), com $i \in 11, 12, 16$ , utilizando $\mathbf{w}_l^*$ . (a) Com $P = 1$ . (b) Com $P = 4$ . . . . .	51
5.2	Comparação da evolução do MSE simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS, utilizando $\mathbf{w}_l^*$ e $P = 1$ . (a) Com $\beta = 0.015$ . (b) Com $\beta = 0.0015$ . . . . .	52
5.3	Comparação da evolução do MSE simulado (em azul) e teórico (em vermelho) do algoritmo $\ell_2$ -F-LMS, utilizando $\mathbf{w}_l^*$ e $P = 4$ . (a) Com $\beta = 0.015$ . (b) Com $\beta = 0.0015$ . . . . .	53

# Lista de Tabelas

2.1	Possíveis escolhas para $G[\mathbf{w}(k)]$ . . . . .	6
2.2	$i$ -ésimo elemento das funções $\mathbf{g}[\mathbf{w}(k)] \triangleq -\frac{\nabla G[\mathbf{w}(k)]}{\nabla \mathbf{w}(k)}$ relativos às funções $G[\mathbf{w}(k)]$ elencadas na Tab. 2.1. . . . .	7

# Capítulo 1

## Introdução

Dada a sua versatilidade e devido à presença ubíqua de sistemas variantes no tempo e/ou desconhecidos pelo projetista, técnicas de filtragem adaptativa vêm sendo empregadas em diversas aplicações, tais como: cancelamento de eco acústico, equalização de canal, controle de ruído ativo e identificação de sistemas [1, 2]. Com o crescente número de pesquisas no ramo, bem como de repercussões em ambientes microprocessados, muitos autores notaram que sistemas reais, recorrentemente, apresentam respostas ao impulso esparsas [1–5]. A esparsidade é um tipo de conhecimento *a priori* sobre o sistema a ser identificado o qual pode, frequentemente, ser incorporado ao algoritmo adaptativo, fazendo com que este apresente menor erro em regime permanente e/ou maior taxa de convergência [3, 6–8]. De modo geral, o advento de quaisquer informações *a priori* acerca do sistema que cumpre ao filtro adaptativo emular pode ser empregado para melhorar suas características de aprendizado.

Neste diapasão, esta dissertação se concentra em três objetivos principais:

1. descrever um arcabouço matemático capaz de gerar algoritmos adaptativos conscientes da estrutura, utilizando o método dos multiplicadores de Lagrange e, então, desenvolvendo um algoritmo consciente da estrutura;
2. desenvolver um modelo estocástico do algoritmo proposto no item 1, que seja capaz de fornecer informações precisas ao projetista acerca do comportamento do algoritmo.

Em um algoritmo de filtragem adaptativa, os coeficientes são atualizados recursivamente de sorte que, após um número suficiente de iterações, o filtro possua uma estimativa adequada do sistema de referência. Tal atualização costuma ser regida pelo cálculo do gradiente de uma função custo estocástica, o qual é capaz de fornecer a direção de atualização engendrada pelo algoritmo adaptativo. Cabe notar que o gradiente aponta para a direção de maior crescimento da função, e a atualização é

feita no sentido contrário ao do gradiente<sup>1</sup>, já que, afinal, o objetivo consiste em minimizar uma função custo.

A Figura 1.1 mostra a estrutura de um algoritmo adaptativo (aplicado à identificação de sistemas, foco deste projeto), cujos coeficientes do filtro de comprimento  $N$  podem ser coletados no vetor  $\mathbf{w}(k) \in \mathbb{R}^N$  definido por:

$$\mathbf{w}(k) \triangleq [w_0(k) \quad w_1(k) \quad \dots \quad w_{N-1}(k)]^T, \quad (1.1)$$

sendo a saída do filtro no instante  $k$  definida por  $y(k) \triangleq \mathbf{w}^T(k)\mathbf{x}(k)$ , com o vetor de entrada  $\mathbf{x}(k)$  descrito por:

$$\mathbf{x}(k) \triangleq [x(k) \quad x(k-1) \quad \dots \quad x(k-N+1)]^T. \quad (1.2)$$

Após um número suficiente de iterações, é esperado que  $y(k)$  seja similar ao sinal de referência

$$d(k) = (\mathbf{w}^*)^T \mathbf{x}(k) + \nu(k), \quad (1.3)$$

sendo  $\mathbf{w}^* \in \mathbb{R}^N$  o vetor que contém os coeficientes a serem identificados e definido por:

$$\mathbf{w}^* \triangleq [w_0^* \quad w_1^* \quad \dots \quad w_{N-1}^*]^T, \quad (1.4)$$

e  $\nu(k)$  um ruído branco com variância  $\sigma_\nu^2(k)$  somado ao sinal de referência, cuja origem é normalmente atribuída a erros de medição e/ou de modelagem. Denotamos por  $e(k) \triangleq d(k) - y(k)$  o sinal de realimentação, o qual consiste no erro entre o sinal de referência e a saída do filtro adaptativo.

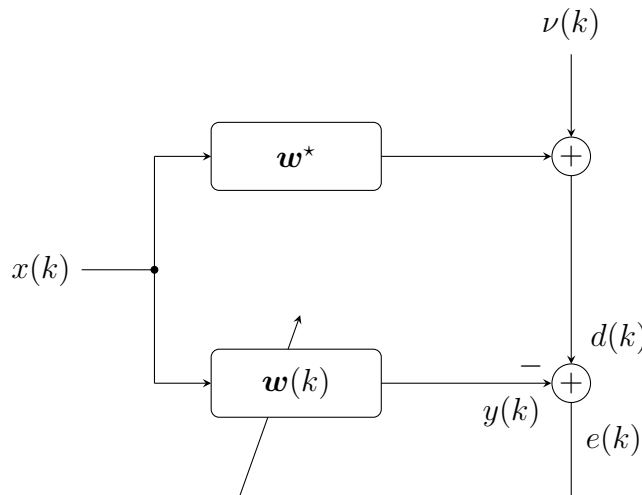


Figura 1.1: Diagrama de blocos de um algoritmo adaptativo.

<sup>1</sup>Uma exceção a essa regra reside nos algoritmos que se valem de funções objetivo baseadas na correntropia, as quais cumpre maximizar [9].

Em termos gerais, esta dissertação estrutura-se do seguinte modo: o Capítulo 2 concentra-se em mostrar diversas métricas de esparsidade (um tipo de estrutura) propostas na literatura científica correlata, apresentando um estudo de caso que indica quais destas métricas seriam mais promissoras em filtragem adaptativa. Ademais, um procedimento para geração de algoritmos conscientes da esparsidade (ou, em termos gerais, da estrutura) que difere do gradiente estocástico é descrita. O Capítulo 3 descreve um algoritmo adaptativo que penaliza a energia de uma *transformação linear* dos coeficientes adaptativos, de modo a evitar soluções que não adiram a uma estrutura pré-estabelecida. O Capítulo 4 provê um modelo estocástico capaz de prever o desempenho do algoritmo proposto no Capítulo 3, a qual é estendida para a identificação de sistemas variantes no tempo. Tal extensão é importante, para se ter garantias de desempenho no que tange à rastreabilidade do algoritmo. O Capítulo 5 estende tais análises para o caso submodelado, o qual se caracteriza pelo fato de o filtro adaptativo apresentar um tamanho inferior ao do filtro FIR a identificar. Por fim, o Capítulo 6 elenca os comentários conclusivos desta dissertação, bem como apresenta alguns futuros desdobramentos considerados promissores.

# Capítulo 2

## Métricas de Esparsidade

### 2.1 Introdução

Um sistema esparso é um sistema cuja resposta ao impulso possui muitos coeficientes de magnitude nula ou quase nula<sup>1</sup>. Por se tratar de uma propriedade recorrente, a capacidade de incorporá-la no processo de aprendizagem do filtro adaptativo vem ganhando notoriedade, seja por acelerar a convergência dos algoritmos, seja por diminuir o erro em regime permanente [1–4]. Como nenhum dos algoritmos tradicionais da família LMS faziam uso dessa propriedade na identificação de sistemas (ou sejam, eram *agnósticos* com relação à esparsidade), algoritmos conscientes da esparsidade foram desenvolvidos. Tais algoritmos são capazes de aproveitar a esparsidade dos sistemas de diferentes formas, seja identificando e não atualizando regiões inativas (isto é, com coeficientes muito próximos de zero) [10, 11], seja propondo uma atualização específica para cada coeficiente com base em sua magnitude (algoritmos proporcionais) [3, 12, 13], ou ainda aplicando uma penalização aos coeficientes, de forma a evitar o encontro de soluções pouco esparsas [1, 4, 7, 8], dentre outras formas propostas na literatura [5].

Neste capítulo, é mostrado o desenvolvimento de uma versão do algoritmo NLMS consciente de esparsidade, com uma função de penalização genérica, através do método de multiplicadores de Lagrange. O arcabouço proposto é poderoso o suficiente para gerar uma família de algoritmos. Ademais, um estudo de caso é empreendido para efetuar uma avaliação das diferentes funções penalizadoras de soluções não esparsas, para indicar quais as variantes se revelam mais promissoras.

---

<sup>1</sup>Alguns autores efetuam uma distinção entre sistemas esparsos e sistemas compressivos, a qual não é relevante no escopo desta dissertação.



## 2.2 Derivação de Algoritmos Conscientes de Esparsidade Normalizados

Esta seção apresenta um paradigma unificado capaz de engendrar toda uma família de algoritmos conscientes de esparsidade. Tal paradigma se vale da técnica dos multiplicadores de Lagrange. Note que, diferentemente da abordagem tradicional - a qual adota o método do gradiente estocástico para derivação de novos algoritmos - a equação de atualização do algoritmo é obtida resolvendo-se um problema de otimização com restrição.

O método dos multiplicadores de Lagrange permite encontrar uma solução para um problema de otimização determinístico. Portanto, considere o seguinte problema de otimização:

$$\begin{aligned} \min \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \kappa G[\mathbf{w}(k+1)] \\ \text{sujeito a } e_p(k) = [1 - \gamma(k)] e(k), \end{aligned} \quad (2.1)$$

onde  $\kappa$  é um parâmetro ajustável responsável por controlar a magnitude da penalização e  $e_p(k)$  é o erro *a posteriori*, definido por:

$$e_p(k) \triangleq d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k), \quad (2.2)$$

enquanto que  $\gamma(k)$  e  $G[\mathbf{w}(k+1)]$  são funções genéricas, ficando a cargo do projetista escolher a que melhor atende às idiossincrasias de seu problema particular. Importa expor que é esperado que  $\mathbb{E}[|1 - \gamma(k)|] < 1$ , de modo a evitar instabilidade do algoritmo. Isso ocorre porque deve-se esperar que em geral  $|e_p(k)| < |e(k)|$ , o que pode ser garantido impondo-se que a restrição em (2.1) efetue um mapeamento contrativo (*contractive mapping*).

Vale notar que, com relação a  $\gamma(k)$  e  $G[\mathbf{w}(k+1)]$ , existem funções amplamente difundidas e comumente usadas na literatura. Por exemplo, o projetista, ao escolher  $\gamma(k) = \beta \|\mathbf{x}(k)\|^2$  é capaz de criar algoritmos não normalizados (apropriados à família de algoritmos LMS). Já se o projetista optar por escolher  $\gamma(k) = \beta$ , os algoritmos resultantes serão normalizados (ou seja, da família do NLMS). Do mesmo modo, ao escolher  $G[\mathbf{w}(k+1)]$ , o projetista pode optar por diversas funções que penalizam soluções esparsas, tais como, por exemplo, as normas  $\ell_0$  (na realidade, uma aproximação da norma  $\ell_0$ ) e  $\ell_1$ . Ambas as possibilidades são bastante usadas na literatura, com especial destaque para a norma  $\ell_0$  (a qual não é uma norma em sentido restrito, sendo mais apropriado denominá-la de *pseudo-norma*).

Motivado justamente por essa preferência pelas funções citadas anteriormente como opções de penalização de esparsidade, essa seção foi criada na esperança de confirmar o que vem sido frequentemente adotado na literatura ou mostrar funções que não são muito utilizadas, mas que possam obter melhores resultados. Portanto,

nesta seção é feita uma análise comparativa do algoritmo NLMS (*benchmark*), ou seja, agnóstico com relação à esparsidade, e os conscientes de esparsidade, utilizando 9 funções diferentes, as quais encontram-se discriminadas na Tabela 2.1.

Tabela 2.1: Possíveis escolhas para  $G[\mathbf{w}(k)]$ .

$j$	$G^{(j)}[\mathbf{w}(k)]$	
1	$\sum_i  w_i(k) $	[14]
2	$-\frac{\sqrt{\sum_i w_i^2(k)}}{\sum_i  w_i(k) }$	[15]
3	$\sum_i \tanh( aw_i(k) ^b), a > 0, b > 1$	[16]
4	$\sum_i \log(1 +  w_i(k) )$	[16]
5	$-\frac{\sum_i w_i^4(k)}{(\sum_i w_i^2(k))^2}$	[15]
6	$\sum_i (1 - e^{-a w_i(k) })$	[4]
7	$\sum_i (1 - e^{-\frac{1}{2}a^2 w_i^2(k)})$	[17]
8	$\sum_i \left(1 - \frac{1}{1+a w_i(k) }\right)$	[18]
9	$\sum_i \left(1 - \frac{1}{1+a^2 w_i^2(k)}\right)$	[18]

Evitando-se neste momento a adoção de alguma função  $G[\mathbf{w}(k)]$  particular constante na Tab. 2.1 e resolvendo o problema de otimização com restrição proposto pela equação (2.1) pelo método dos multiplicadores de Lagrange, é possível reescrever a equação (2.1) como um problema de otimização equivalente (ou seja, que preserve a solução) sem restrições, da seguinte forma:

$$\begin{aligned} \min \mathcal{F}[\mathbf{w}(k+1)] = & \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ & + \kappa G[\mathbf{w}(k+1)] \\ & + \lambda \{e_p(k) - [1 - \gamma(k)]e(k)\}, \end{aligned} \quad (2.3)$$

onde  $\lambda$  denota o multiplicador de Lagrange.

Para minimizar a função custo da equação (2.3), basta igualar ao vetor nulo o gradiente da função custo. Assim:

$$\frac{\partial \mathcal{F}[\mathbf{w}(k+1)]}{\partial \mathbf{w}(k+1)} = \mathbf{w}(k+1) - \mathbf{w}(k) - \kappa \mathbf{g}[\mathbf{w}(k+1)] - \lambda \mathbf{x}(k) = \mathbf{0}, \quad (2.4)$$

onde  $\mathbf{g}[\mathbf{w}(k+1)] \triangleq -\frac{\partial G[\mathbf{w}(k+1)]}{\partial \mathbf{w}(k+1)}$ .

Pode-se observar que o termo  $\mathbf{w}(k+1)$  não é facilmente isolável. Porém, aplicando a seguinte aproximação  $\mathbf{g}[\mathbf{w}(k+1)] \approx \mathbf{g}[\mathbf{w}(k)]$  [19], é possível isolá-lo. Cumpre ressaltar que esta aproximação costuma ser razoável, já que o algoritmo de filtragem adaptativa aplica um mapeamento do tipo passa baixas no sinal de entrada, de modo que os coeficientes adaptativos evoluem numa taxa muito mais lenta do que

a do sinal de entrada [20]. Assim, obtém-se que:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \kappa \mathbf{g}[\mathbf{w}(k)] + \lambda \mathbf{x}(k). \quad (2.5)$$

Fica claro que a equação (2.5) ainda não é a equação de atualização dos coeficientes do filtro adaptativo, uma vez que importa ainda calcular o multiplicador de Lagrange  $\lambda$ . Para isso, é necessário substituir a equação (2.5) na restrição da equação (2.1) e assim obter:

$$d(k) - \mathbf{w}^T(k)\mathbf{x}(k) - \kappa \mathbf{g}^T[\mathbf{w}(k)]\mathbf{x}(k) - \lambda \mathbf{x}^T(k)\mathbf{x}(k) = e(k) - \gamma(k)e(k). \quad (2.6)$$

Note que  $d(k) - \mathbf{w}^T(k)\mathbf{x}(k) = e(k)$  e que  $\mathbf{x}^T(k)\mathbf{x}(k) = \|\mathbf{x}(k)\|^2$ . Portanto, a equação (2.6) pode ser reescrita da seguinte forma:

$$\lambda = \frac{\gamma(k)e(k)}{\|\mathbf{x}(k)\|^2} - \kappa \frac{\mathbf{g}^T[\mathbf{w}(k)]\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2}. \quad (2.7)$$

Uma vez calculado  $\lambda$ , e escolhendo-se  $\gamma(k) = \beta$ , basta substituir a equação (2.7) na equação (2.5). Por fim, obtém-se a seguinte equação de atualização do filtro adaptativo:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\beta \mathbf{x}(k)e(k)}{\|\mathbf{x}(k)\|^2} - \kappa \frac{\mathbf{x}(k)\mathbf{g}^T[\mathbf{w}(k)]\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} + \kappa \mathbf{g}[\mathbf{w}(k)]. \quad (2.8)$$

Tabela 2.2:  $i$ -ésimo elemento das funções  $\mathbf{g}[\mathbf{w}(k)] \triangleq -\frac{\nabla G[\mathbf{w}(k)]}{\nabla \mathbf{w}(k)}$  relativos às funções  $G[\mathbf{w}(k)]$  elencadas na Tab. 2.1.

$j$	$g_i^{(j)}[\mathbf{w}(k)]$
1	$-\text{sign}[w_i(k)]$
2	$\frac{w_i(k)}{[\sqrt{\sum_i w_i^2(k)}][\sum_i  w_i(k)  + \delta_s]} - \frac{\text{sign}[w_i(k)]\sqrt{\sum_i w_i^2(k)}}{[\sum_i  w_i(k) ]^2 + \delta_s}$
3	$a^b b  w_i(k) ^{b-1} \text{sign}[aw_i(k)] \left[ \tanh( aw_i(k) ^b)^2 - 1 \right]$
4	$-\frac{\text{sign}[w_i(k)]}{ w_i(k) +1}$
5	$\frac{4w_i^3(k)}{[\sum_i w_i^2(k)]^2 + \delta_s} - \frac{4w_i(k)\sum_i w_i^4(k)}{[\sum_i w_i^2(k)]^3 + \delta_s}$
6	$-a \text{sign}[w_i(k)] e^{-a w_i(k) }$
7	$-a^2 w_i(k) e^{-\frac{1}{2}a^2 w_i^2(k)}$
8	$-\frac{a \text{sign}[w_i(k)]}{(1+a w_i(k) )^2}$
9	$-\frac{2a^2 w_i(k)}{(1+a^2 w_i^2(k))^2}$

## 2.3 Estudo de caso

O paradigma descrito de derivação de algoritmos é compatível com todas as funções descritas na Tab. 2.1. O fato de tal compatibilidade gerar *teoricamente* diversos algoritmos inéditos de filtragem adaptativa não significa que tais algoritmos sejam funcionais. E, o sendo, caberia ainda a questão acerca de quais algoritmos se mostram mais promissores para aplicações futuras. Para delinear respostas a estas perguntas importantes, foram escolhidos dois sistemas com resposta ao impulso reais de duas salas de aula. Como estudo de caso, a identificação de ambos os sistemas será avaliada em simulações, sendo comparados todos os nove algoritmos que compõe a família de algoritmos proposta.

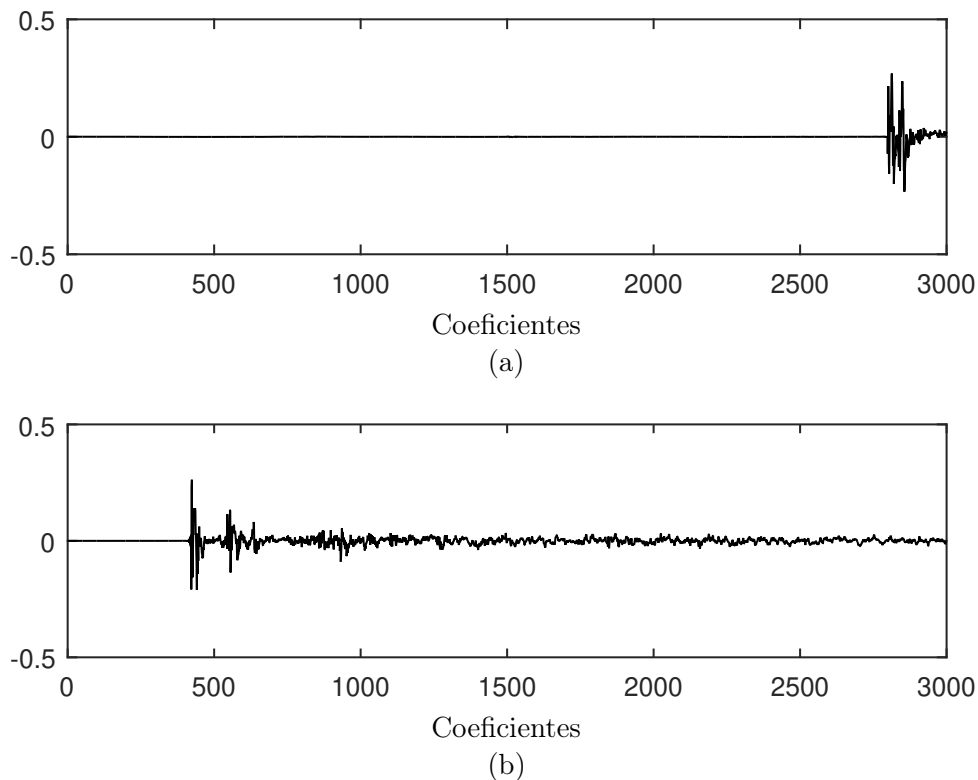


Figura 2.1: (a) Resposta ao impulso da planta 1. (b) Resposta ao impulso da planta 2.

A Figura 2.1 mostra as respostas ao impulso dos dois sistemas utilizados; ambas foram normalizadas para uma energia unitária. É possível observar que a planta 1 é mais esparsa que a planta 2, e não só para corroborar essa observação, mas também para ter uma estimativa do quão esparsas são as plantas utilizadas, foram usadas duas métricas que têm como objetivo estimar a esparsidade de uma resposta ao impulso [15].

A primeira métrica usada é a de Hoyer [21]. Tal métrica é uma normalização da

razão  $\frac{\ell_2}{\ell_1}$ , como pode ser observado na sua equação:

$$H = \left( \sqrt{N} - \frac{\sum_i w_i^*}{\sqrt{\sum_i (w_i^*)^2}} \right) (\sqrt{N} - 1)^{-1}. \quad (2.9)$$

A segunda métrica é o índice de Gini [22], calculado pela equação:

$$\bar{G} = 1 - 2 \sum_{i=1}^N \frac{w_i^*}{\|\mathbf{w}^*\|_1} \left( \frac{N - i + \frac{1}{2}}{N} \right). \quad (2.10)$$

Um fato curioso em relação ao índice de Gini, é que ele foi proposto originalmente em pesquisa econômica como uma métrica de aferição de desigualdade econômica [22–25], e ainda é estudado tanto na área de distribuição de renda quanto em outras áreas [26–29]. Na linguagem de processamento de sinais, “desigualdade econômica” quer dizer “eficiência de representação” ou “esparsidade” [15]. E a utilidade do índice de Gini como uma maneira de medir esparsidade tem sido demonstrada em [30–33].

Segundo [15], essas são duas das métricas que apresentaram melhores resultados. Aplicando, então, a fórmula de Hoyer e o índice de Gini nas respostas ao impulso mostradas na Figura 2.1 o que se obtém para a planta 1 é:

$$\begin{aligned} H &= 84.8532\% \\ \bar{G} &= 95.5519\% \end{aligned}$$

e para a planta 2:

$$\begin{aligned} H &= 47.8541\% \\ \bar{G} &= 58.0641\% \end{aligned}$$

Tais cálculos corroboram a impressão de que a resposta ao impulso do sistema 1 é mais esparsa que a resposta ao impulso do sistema 2, obtida ao observar a Figura 2.1.

## 2.4 Resultados

Como dito anteriormente, nesta seção é feita uma análise comparativa entre o algoritmo NLMS e algoritmos conscientes de esparsidade derivados das funções mostradas na Tabela 2.1, cujos gradientes encontram-se explicitados na Tabela 2.2. Porém, para fazer uma análise comparativa adequada, é necessário escolher os parâmetros

de cada algoritmo de forma que seja possível observá-los em seu rendimento ótimo ou quase ótimo, para as condições de análise definidas nesta dissertação.

O desempenho dos algoritmos será avaliado através da evolução do MSD (do inglês, “*Mean Square Deviation*”) para as plantas 1 e 2, discutidas na seção anterior,  $\beta = 1$  e  $\sigma_v^2 = 10^{-3}$ . O MSD na  $k$ -ésima iteração é definido por

$$\text{MSD}(k) \triangleq \mathbb{E} [\|\mathbf{w}^* - \mathbf{w}(k)\|^2]. \quad (2.11)$$

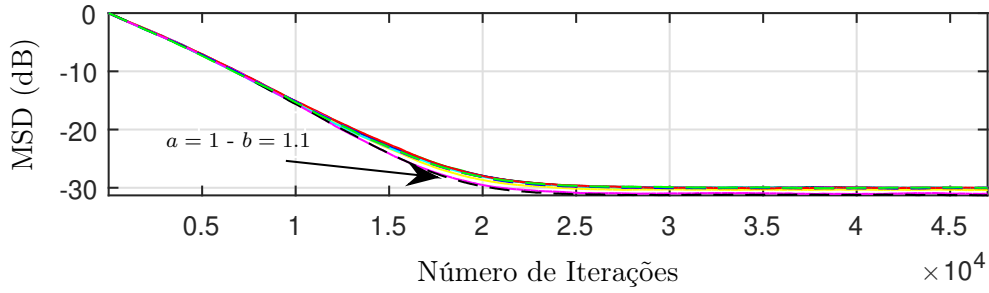
Contudo, nessa primeira fase da análise, a qual pode ser vista como uma fase de sintonização (“*tuning*”) dos parâmetros dos algoritmos, somente serão levados em consideração os resultados obtidos para a planta 1, que é a planta considerada esparsa. Note que usar apenas a planta esparsa nessa fase não prejudica a análise, uma vez que o uso desses algoritmos é indicado quando as plantas subjacentes são efetivamente esparsas, sendo possível detectar, ao longo das iterações, se efetivamente esta hipótese é válida.

Como pode ser visto na Tabela 2.1 e na Tabela 2.2, algumas funções possuem parâmetros adicionais, além do  $\kappa$ . Por isso, as funções são separadas em três grupos, sendo o grupo 1 formado somente pela função 3 que possui os parâmetros adicionais  $a$  e  $b$ , o grupo 2 formado pelas funções de 6 a 9 que possui o parâmetro adicional  $a$  e o grupo 3, formado pelas funções 1, 2, 4, 5 que não possuem parâmetros adicionais, apenas o parâmetro comum  $\kappa$ .

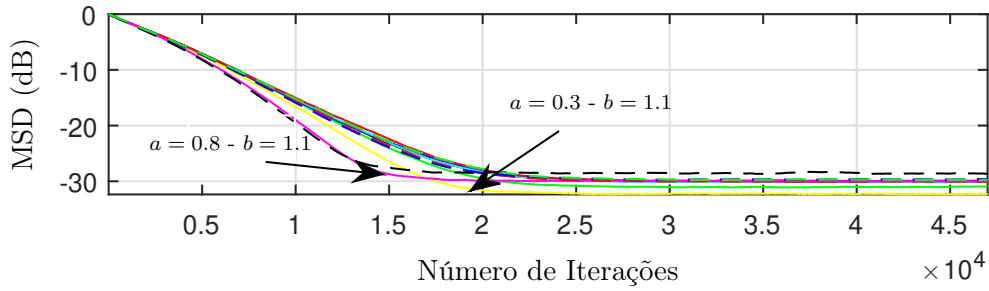
A Figura 2.2 corresponde à análise para identificar os parâmetros do grupo 1 e mostra a evolução do MSD em função do número de iterações do algoritmo adaptativo. Foram realizadas simulações utilizando diversos valores de “ $b$ ”, “ $a$ ” e  $\kappa$ , cujos valores eram, respectivamente,  $\{1.1, 1.5, 2\}$ ,  $\{0.1, 0.3, 0.5, 0.8, 1\}$  e  $\{10^{-8}, 10^{-7}, 10^{-6}\}$ . Na Figura 2.2, não foram mostrados os resultados obtidos para  $\kappa = 10^{-8}$  porque não haviam diferenças significativas entre as curvas, nem em relação à convergência nem relação ao regime permanente. Também não foram colocadas legendas, ressaltando apenas as melhores curvas, de forma que o resultado pudesse ser mostrado em apenas uma figura.

Ao observar a Figura 2.2, é possível perceber que a curva demarcada pelos parâmetros  $a = 0.8, b = 1.1$  e  $\kappa = 10^{-6}$  parece ser a mais adequada para se escolher, devido à sua alta taxa de convergência quando comparada às outras curvas presentes nos itens (a) e (b). Porém, essa curva não é capaz de atingir o valor mínimo de MSD que algumas outras curvas são capazes de atingir, como será mostrado mais a frente nessa seção, mesmo variando o passo de aprendizagem  $\beta$ . Por isso, a curva escolhida para a análise comparativa é a curva demarcada pelos parâmetros  $a = 0.3, b = 1.1, \kappa = 10^{-6}$ .

As Figuras 2.3 e 2.4 correspondem à análise para identificar os parâmetros do



(a)



(b)

Figura 2.2: (a) Evolução do MSD do algoritmo adaptativo usando  $G_3$  para  $\kappa = 10^{-7}$ . (b) Evolução do MSD do algoritmo adaptativo usando  $G_3$  para  $\kappa = 10^{-6}$

grupo 2 e mostram a evolução do MSD em função do número de iterações do algoritmo adaptativo. Foram realizadas simulações utilizando diversos valores de  $a$  e  $\kappa$ , cujos valores eram, respectivamente,  $[0.1, 0.3, 0.8, 1]$ ,  $[10^{-8}, 10^{-7}, 10^{-6}]$ . Não foram colocadas legendas, ressaltando apenas as melhores curvas, de forma que o resultado pudesse ser mostrado em duas figuras.

Na Figura 2.3, são mostrados os resultados obtidos para as funções 7 e 9. Note que a mudança dos parâmetros não impacta significativamente o desempenho dos algoritmos. Portanto, foram escolhidos, arbitrariamente, os valores de 0.1 e  $10^{-6}$  para  $a$  e  $\kappa$ , respectivamente, para ambos os casos. Já na Figura 2.4, são mostrados os resultados obtidos para as funções 6 e 8, sendo possível observar que as curvas destacadas em ambos os itens apresentam alta taxa de convergência, caso as comparamos às demais. Porém, a curva com parâmetros  $a = 0.5$  e  $\kappa = 10^{-6}$  não é capaz de atingir o MSD mínimo alcançado por outras curvas. Logo, foram escolhidos os valores de 0.3 e  $10^{-6}$  para  $a$  e  $\kappa$ , respectivamente, em ambos os casos.

As Figuras 2.5 e 2.6 correspondem à análise para identificar os parâmetros do grupo 3 e mostram a evolução do MSD em função do número de iterações do algoritmo adaptativo. Foram realizadas simulações utilizando diversos valores de  $\kappa$ , cujos valores eram  $\{10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}\}$ . Não foram colocadas legendas, ressal-

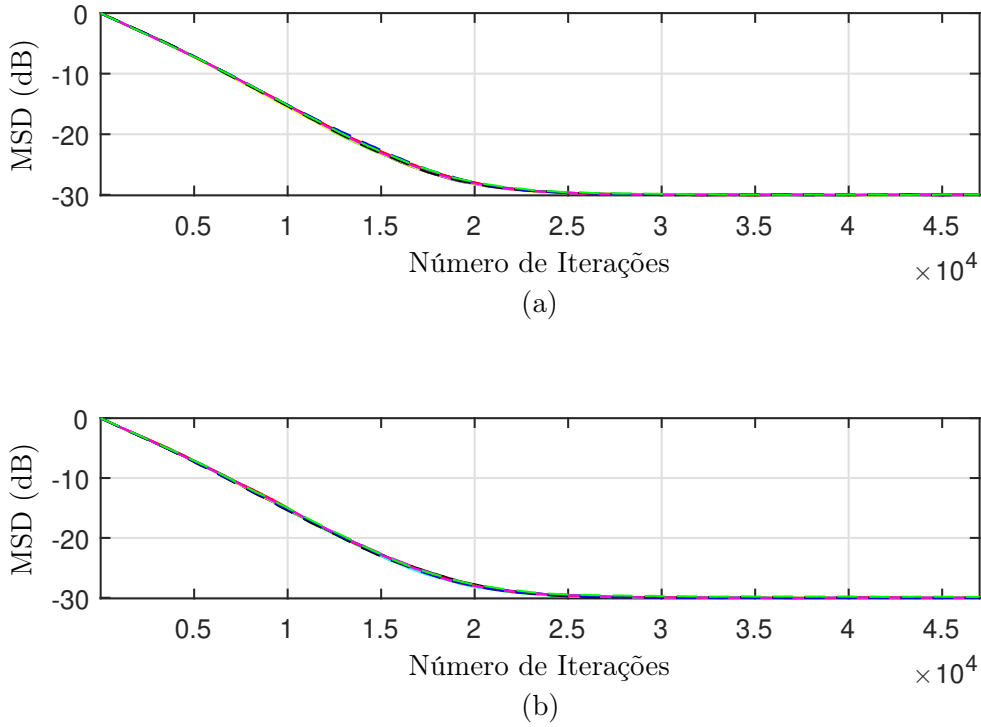


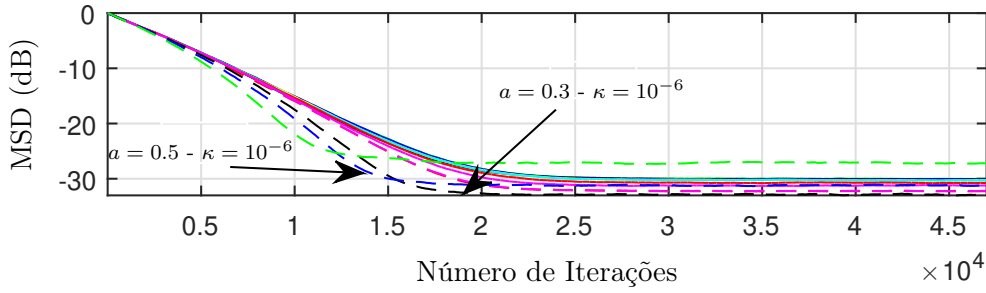
Figura 2.3: (a) Evolução do MSD do algoritmo adaptativo usando  $G_7$ . (b) Evolução do MSD do algoritmo adaptativo usando  $G_9$ .

tando apenas as melhores curvas, de forma que o resultado pudesse ser mostrado em duas figuras.

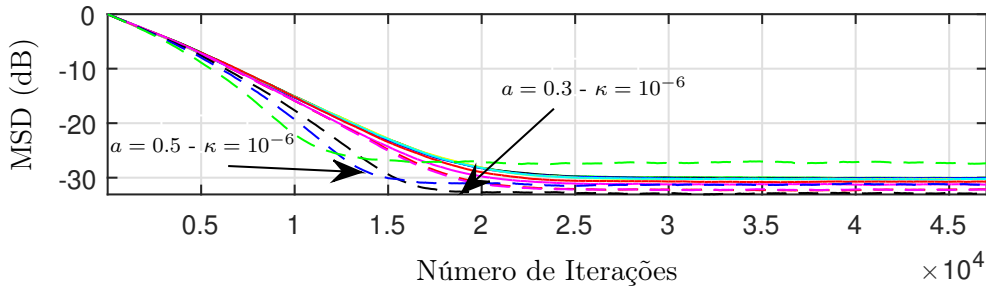
Na Figura 2.5 são mostrados os resultados obtidos para as funções 1 e 4. Observe que, para ambos os algoritmos, o valor de  $\kappa$  mais adequado é de  $10^{-7}$ , uma vez que apresenta uma alta taxa de convergência e um valor de MSD em regime permanente razoável. Já na Figura 2.6 são mostrados os resultados obtidos para as funções 2 e 5, sendo possível observar que a curva com  $\kappa = 10^{-5}$  para o algoritmo com a função 2, item (a), apresenta um desempenho bem superior às demais curvas. Já para o algoritmo com a função 5, a curva com  $\kappa = 10^{-5}$  apresenta o pior resultado, enquanto que as curvas com  $\kappa = 10^{-6}$ ,  $\kappa = 10^{-7}$  e  $\kappa = 10^{-8}$  apresentam um desempenho melhor que a outra curva, mas não apresentam uma diferença significativa entre si. Então, foi escolhido, arbitrariamente dentre as três melhores curvas mencionadas, um valor de  $\kappa = 10^{-6}$ .

A Figura 2.7 mostra a evolução do MSD para os diversos algoritmos adaptativos utilizando todas as funções da Tabela 2.1 para a planta 1. A simulação foi feita utilizando os parâmetros definidos para cada algoritmo individualmente, de forma que, como discutido anteriormente, cada algoritmo tivesse o melhor desempenho possível para a configuração do problema em questão. Essa figura é interessante





(a)



(b)

Figura 2.4: (a) Evolução do MSD do algoritmo adaptativo usando  $G_6$ . (b) Evolução do MSD do algoritmo adaptativo usando  $G_8$ .

porque permite observar as diferenças de MSD em regime permanente para  $\beta = 1$ .

A Figura 2.8 mostra a evolução do MSD para os diversos algoritmos adaptativos utilizando todas as funções da Tabela 2.1. A simulação foi feita utilizando os parâmetros definidos para cada algoritmo individualmente, de forma que, como discutido anteriormente, cada algoritmo tivesse o melhor desempenho possível para a configuração do problema em questão. Essa figura é interessante porque permite observar que os algoritmos com menor MSD na simulação com a planta 1, apresentam maior MSD na simulação com a planta 2. Vale notar também que a inclusão da penalização para soluções esparsas parece não influenciar na convergência inicial dos algoritmos, diferentemente do que é observado na Figura 2.7.

A Figura 2.9 mostra a média do MSD em regime permanente para os diversos algoritmos adaptativos utilizando todas as funções da Tabela 2.1. A simulação foi feita variando o parâmetro  $\beta$  de 0.5 a 1 com passo de 0.1. Esses dados permitem avaliar a variação do MSD em regime permanente em função de  $\beta$ , permitindo a escolha de valores desse parâmetro para cada algoritmo, de forma que seja possível igualar o MSD em regime permanente de todos e avaliar suas respectivas taxas de convergência.

A Figura 2.10 mostra a evolução do MSD para os diversos algoritmos adapta-

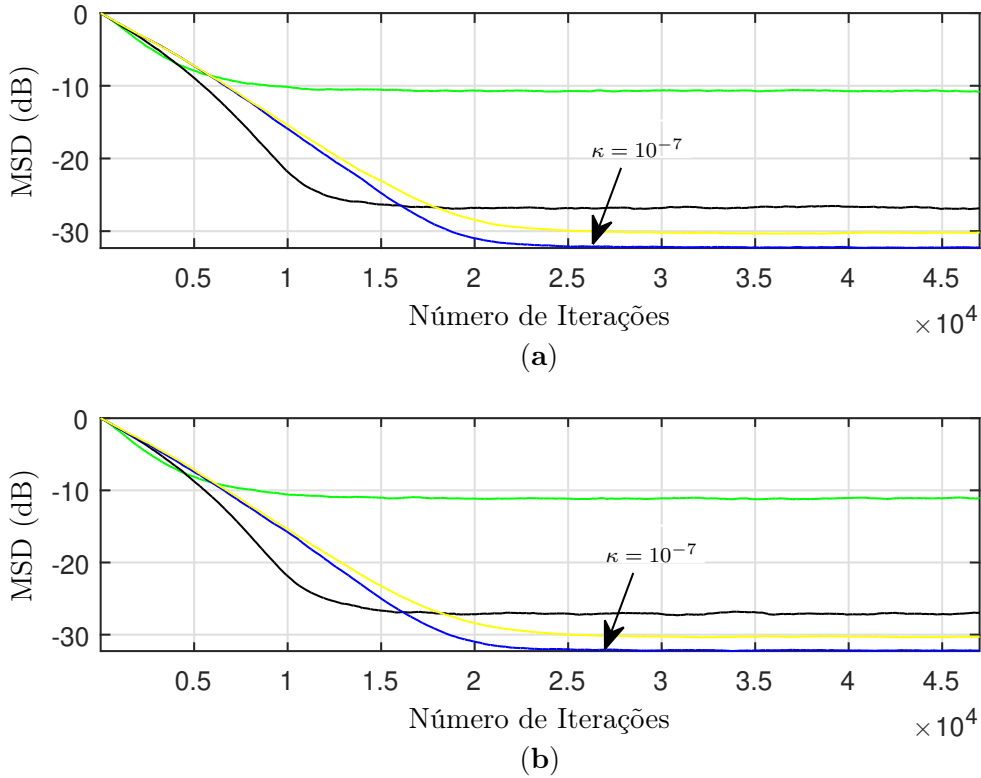


Figura 2.5: (a) Evolução do MSD do algoritmo adaptativo usando  $F_1$ . (b) Evolução do MSD do algoritmo adaptativo usando  $F_4$

tivos utilizando todas as funções da Tabela 2.1. A simulação foi feita utilizando os parâmetros definidos para cada algoritmo individualmente, inclusive  $\beta$ , de modo que todos os algoritmos possuísem valores de MSD em regime permanente suficientemente próximos.

Observando as Figuras 2.7 e 2.10, é possível concluir que os algoritmos que usam as funções de penalização 6 e 8, funções que aproximam a norma  $\ell_0$  utilizando o módulo dos coeficientes, possuem melhor desempenho, tanto em regime permanente quanto em taxa de convergência.

## 2.5 Comentários Finais

Como principal contribuição, este capítulo descreve um arcabouço capaz de gerar inúmeros algoritmos adaptativos conscientes da esparsidade, alguns dos quais sendo casos particulares que já foram objeto de atenção em alguns artigos.

Diferentes funções penalizadoras foram mobilizadas, a partir de uma varredura bibliográfica. Ao fim, é avaliado quais destas funções apresentam o melhor desempenho num estudo de caso. Como conclusão preliminar, verifica-se que a adoção de penalizações baseadas na pseudo-norma  $\ell_0$  tende a gerar um processo de aprendiza-

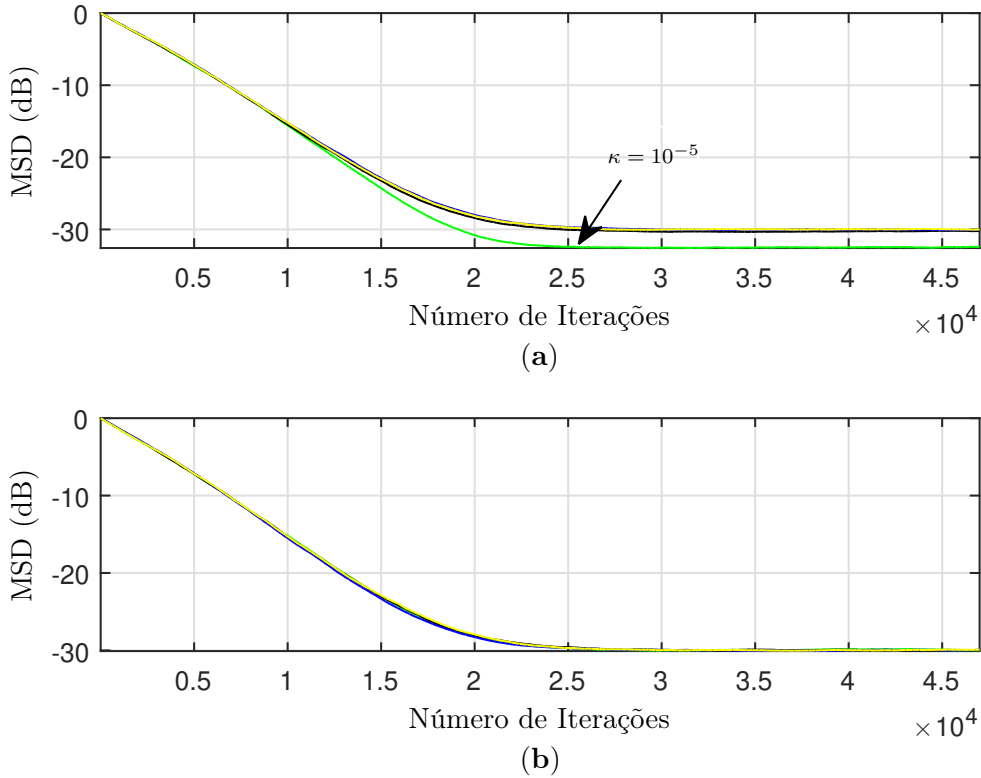


Figura 2.6: (a) Evolução do MSD do algoritmo adaptativo usando  $F_2$ . (b) Evolução do MSD do algoritmo adaptativo usando  $F_5$ .

gem superior, o que é compatível com as conclusões de [34], um artigo que compara a eficácia das técnicas proporcionais e de regularização de norma. Infelizmente, esta conclusão empírica não provê maiores garantias de que tal vantagem é geral. Isso se deve ao fato de que os algoritmos em foco não serem facilmente aptos a uma modelagem estocástica de seu desempenho.

No próximo capítulo, é proposto um novo algoritmo que penaliza a energia de transformações lineares nos coeficientes a identificar. Contrariamente aos algoritmos tratados neste capítulo, o processo de aprendizagem deste novo algoritmo será posteriormente alvo de um modelo estocástico, capaz de prover garantias mais robustas de desempenho ao projetista. Cabe ademais aduzir que tal algoritmo penalizará soluções não *estruturadas*, como por exemplo, as que não se revestem de uma característica passa-baixa ou passa-faixa. Importa enfatizar que tal tipo de conhecimento *a priori*, relacionado à característica ou estrutura da função de transferência a identificar, é mais geral do que a esparsidade, característica mais explorada na literatura de algoritmos adaptativos não agnósticos.

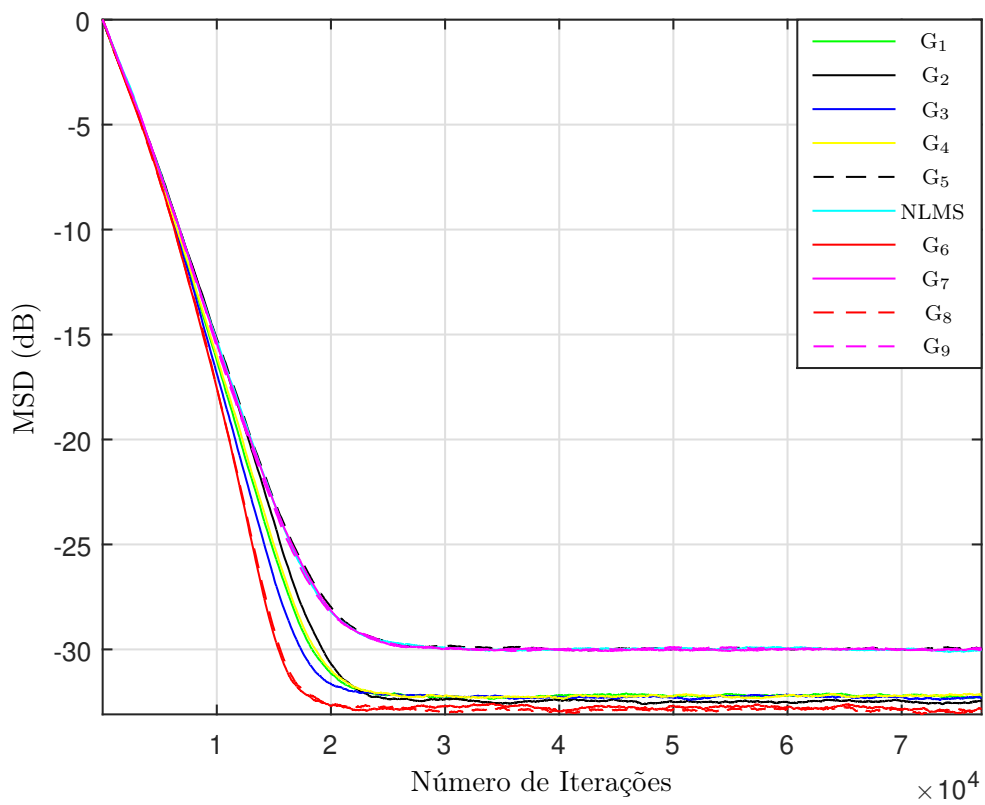


Figura 2.7: Evolução do MSD para todos os algoritmos otimizados para desempenho máximo, assim como do NLMS, para a planta 1.

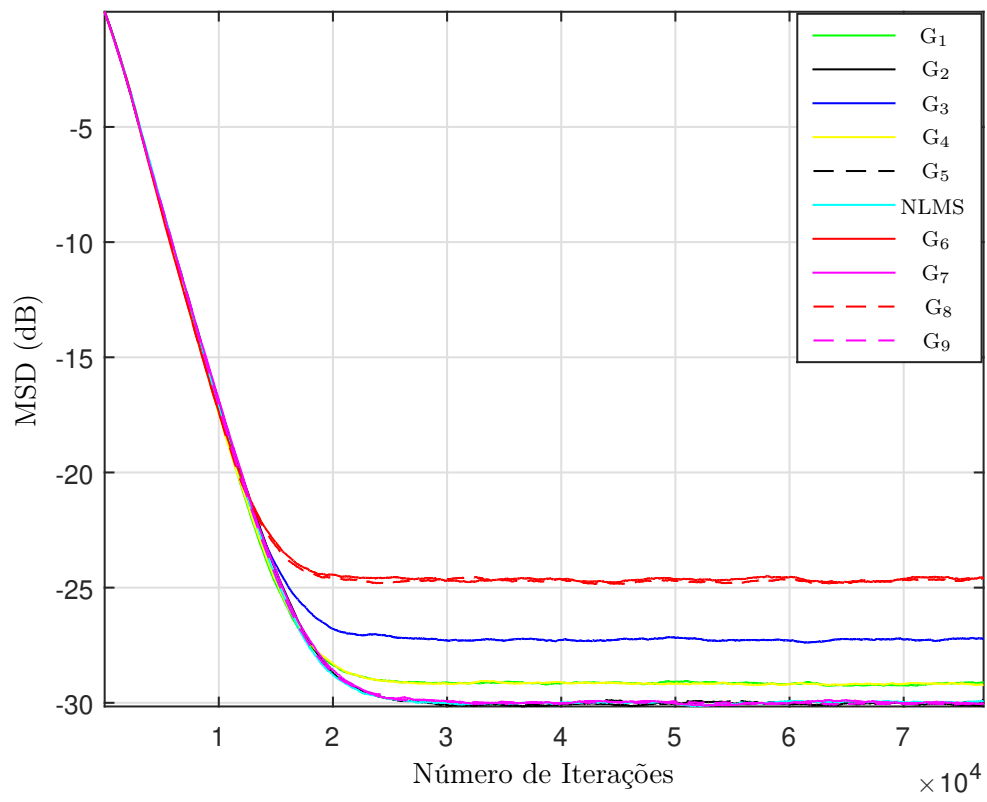


Figura 2.8: Evolução do MSD para todos os algoritmos mais o NLMS para a planta 2.

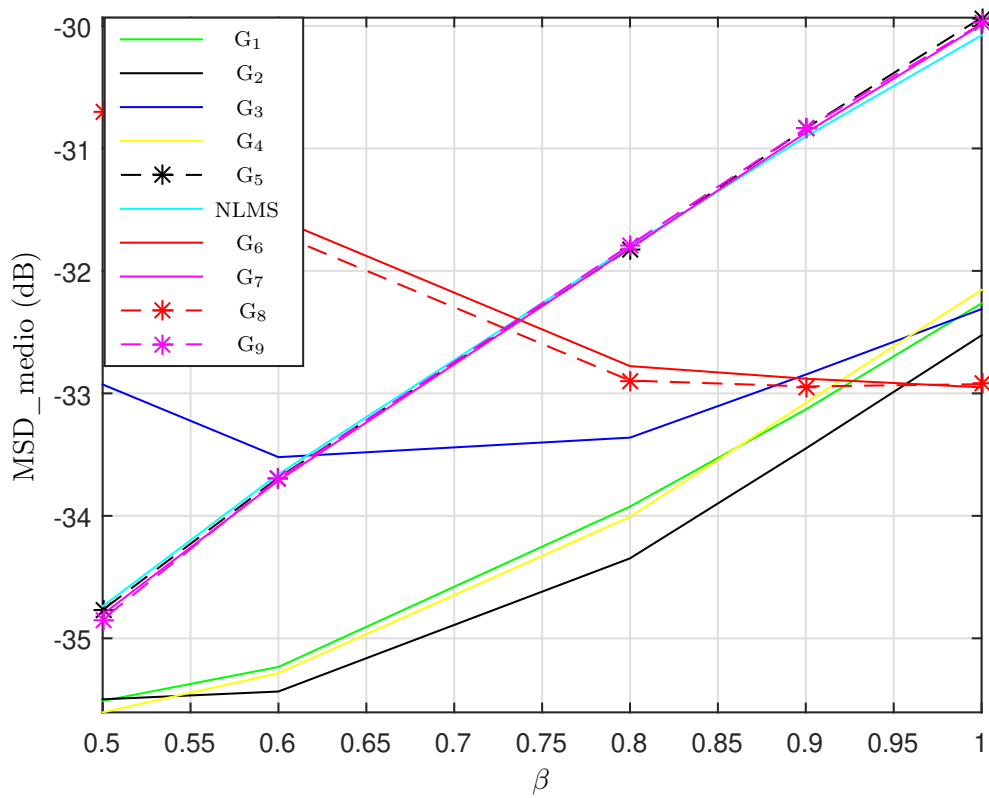


Figura 2.9: Média do MSD em regime permanente para todos os algoritmos mais o NLMS para a planta 1.

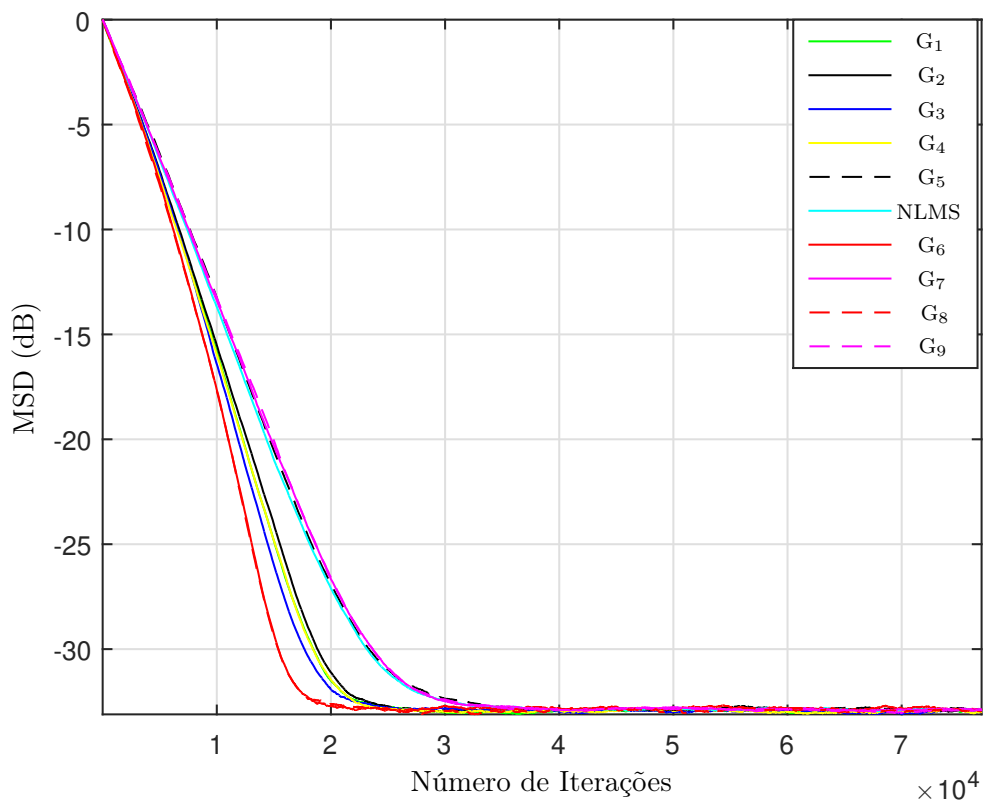


Figura 2.10: Evolução do MSD para todos os algoritmos cada qual com seus parâmetros específicos, mais o NLMS para a planta 1.

# Capítulo 3

## $\ell_2$ -norm Feature LMS

### 3.1 Introdução

Como vem sido dito ao longo deste trabalho, muitos sistemas reais possuem resposta ao impulso esparsa. Diversos algoritmos recentes conscientes desta propriedade exploram-na de sorte a obter ganhos, como na taxa de convergência. Porém, nos últimos anos, pesquisadores vêm criando algoritmos adaptativos capazes de explorar um outro tipo de esparsidade, denominada de esparsidade implícita [2, 6, 35].

Atualmente, o conceito de esparsidade pode ser classificado de duas formas: esparsidade explícita e esparsidade implícita. A esparsidade explícita é aquela em que a resposta ao impulso do sistema possui um grande número de coeficientes nulos ou quase nulos, de modo que a energia do sistema fica concentrada em poucos coeficientes. Essa classificação de esparsidade é mais tradicionalmente explorada nos algoritmos apresentados na literatura, bem como pelos algoritmos do capítulo 2 [1, 3–5, 7, 8, 10, 12, 13, 19, 36–38]. Já a esparsidade implícita tem haver com o comportamento dos coeficientes da resposta ao impulso de um sistema, comportamento esse que faz com que os sistemas possuam certas características [2, 6]. Por exemplo, se o sistema a ser identificado possuir características de um filtro passa-baixa, os coeficientes da resposta ao impulso do sistema tendem a variar suavemente. Diferentemente de sistemas que possuem características de filtros passa-alta, cujos coeficientes podem, por exemplo, não variar muito de amplitude dentro de uma dada vizinhança, mas podem trocar de sinal.

Portanto, declarar que um sistema é esparso normalmente equivale dizer que ele apresenta esparsidade *explícita*. Já quando se diz que um sistema passa-baixa, ou passa-alta, é esparso, deve-se entender que ele apresenta esparsidade implícita (e possivelmente explícita também). É natural que o projetista explore as características do sistema que lhe são conhecidas, de modo a otimizar o desempenho dos algoritmos de filtragem adaptativa empregados.



Neste capítulo, é feita a derivação do algoritmo  $\ell_2$ -norm *Feature* LMS (cujo acrônimo é  $\ell_2$ -F-LMS), seguindo o arcabouço matemático desenvolvido no capítulo 2, bem como uma análise comparativa com o algoritmo LMS (que é tido como *benchmark* para algoritmos adaptativos não normalizados) e o algoritmo *Feature* LMS proposto em [2]. Cabe ressaltar que o algoritmo  $\ell_2$ -F-LMS explora um eventual conhecimento *a priori* acerca de uma característica implícita da função de transferência, a qual não se confunde com a esparsidade. Afinal, algoritmos não agnósticos à esparsidade costumam se valer de normas  $\ell_p$  com  $p \in [0, 1]$ , enquanto que o algoritmo  $\ell_2$ -F-LMS emprega a norma  $\ell_2$ , de modo a penalizar a *energia* de soluções não aderentes a uma determinada estrutura, conhecida *a priori*.

## 3.2 Derivação do Algoritmo $\ell_2$ -norm *Feature* LMS

Nesta seção, é feita a derivação da equação de atualização do algoritmo  $\ell_2$ -F-LMS. No capítulo 2, foi apresentada uma forma de se obter uma família de algoritmos conscientes de esparsidade através da técnica dos multiplicadores de Lagrange. Por também explorar características da função de transferência, o algoritmo  $\ell_2$ -norm *Feature* LMS pode ser facilmente derivado a partir das equações presentes no capítulo 2, quando:

$$G[\mathbf{w}(k+1)] = \|\mathbf{F}\mathbf{w}(k+1)\|^2. \quad (3.1)$$

Logo, tem-se o seguinte problema de otimização:

$$\begin{aligned} \min \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \kappa \|\mathbf{F}\mathbf{w}(k+1)\|^2 \\ \text{sujeito a } e_p(k) = [1 - \gamma(k)] e(k), \end{aligned} \quad (3.2)$$

onde  $\kappa$  é um parâmetro ajustável para controlar o módulo da penalização,  $\gamma(k)$  é um parâmetro escolhido pelo projetista, como visto no capítulo 2, e  $e_p(k)$  é o erro *a posteriori* definido por:

$$e_p(k) \triangleq d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k). \quad (3.3)$$

Vale a pena fazer um comentário acerca da matriz  $\mathbf{F}$ . Note que se for de conhecimento do projetista que o sistema a ser identificado possuir uma resposta ao impulso que favorece baixas frequências, uma escolha possível para a matriz  $\mathbf{F}$  seria [2, 6, 35]:

$$\mathbf{F}_\ell = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix}, \quad (3.4)$$

onde  $\mathbf{F}_\ell$  penaliza soluções com características de um filtro passa-alta. Já se for um sistema cuja resposta ao impulso favorece altas frequências [2, 6, 35]:

$$\mathbf{F}_h = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \end{bmatrix}. \quad (3.5)$$

Analogamente ao capítulo 2, esse problema de otimização é resolvido utilizando-se a técnica dos multiplicadores de Lagrange. Para tal, a equação (3.2) deve ser reescrita no seguinte problema de otimização sem restrição:

$$\begin{aligned} \min \mathcal{F}[\mathbf{w}(k+1)] = & \frac{1}{2} \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ & + \kappa \|\mathbf{F}\mathbf{w}(k+1)\|^2 \\ & + \lambda \{e_p(k) - [1 - \gamma(k)]e(k)\}, \end{aligned} \quad (3.6)$$

onde  $\lambda$  é o multiplicador de Lagrange.

Para minimizar a função custo da equação (3.6), cabe calcular o gradiente da função e igualá-lo a zero. Portanto:

$$\frac{\partial \mathcal{F}[\mathbf{w}(k+1)]}{\partial \mathbf{w}(k+1)} = \mathbf{w}(k+1) - \mathbf{w}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}(k+1) - \lambda \mathbf{x}(k) = \mathbf{0}. \quad (3.7)$$

Pode-se observar que o termo  $\mathbf{w}(k+1)$  não é facilmente isolável. Porém, aplicando a aproximação  $\mathbf{F}^T \mathbf{F} \mathbf{w}(k+1) \approx \mathbf{F}^T \mathbf{F} \mathbf{w}(k)$  [19], o termo é facilmente isolado. Assim, temos que:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}(k) + \lambda \mathbf{x}(k). \quad (3.8)$$

Fica claro que a equação (3.8) ainda não é a equação de atualização dos coeficientes do filtro adaptativo, uma vez que cabe ainda calcular o multiplicador de Lagrange  $\lambda$ . Para isso, é necessário substituir a equação (3.8) na restrição da equação (3.2) e assim obter:

$$d(k) - \mathbf{w}^T(k) \mathbf{x}(k) - \kappa \mathbf{w}^T(k) \mathbf{F}^T \mathbf{F} \mathbf{x}(k) - \lambda \mathbf{x}^T(k) \mathbf{x}(k) = e(k) - \gamma(k)e(k). \quad (3.9)$$

Note que  $d(k) - \mathbf{w}^T(k) \mathbf{x}(k) = e(k)$  e que  $\mathbf{x}^T(k) \mathbf{x}(k) = \|\mathbf{x}(k)\|^2$ . Portanto, a equação (3.9) pode ser reescrita da seguinte forma:

$$\lambda = \frac{\gamma(k)e(k)}{\|\mathbf{x}(k)\|^2} - \kappa \frac{\mathbf{w}^T(k) \mathbf{F}^T \mathbf{F} \mathbf{x}(k)}{\|\mathbf{x}(k)\|^2}. \quad (3.10)$$

Porém, o termo  $\kappa \frac{\mathbf{w}^T(k) \mathbf{F}^T \mathbf{F} \mathbf{x}(k)}{\|\mathbf{x}(k)\|^2}$  é negligenciável [9] (um motivo para isso é a sua ordem de grandeza quando comparado ao outro termo penalizador, por conta da normalização pela energia do sinal de entrada), dando origem a

$$\lambda = \frac{\gamma(k)e(k)}{\|\mathbf{x}(k)\|^2} \quad (3.11)$$

Uma vez calculado  $\lambda$ , basta substituir a equação (3.11) na equação (3.8). Note que todo o desenvolvimento até aqui, não faz distinção quanto à normalização do algoritmo. Porém, o foco desta seção é a proposta do algoritmo  $\ell_2$ -norm-F-LMS, e como o termo  $\gamma(k)$  fica a cargo do projetista, basta que este termo seja definido como  $\gamma(k) \triangleq \beta \|\mathbf{x}(k)\|^2$ . Permitindo obter a seguinte equação de atualização do filtro adaptativo:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta \mathbf{x}(k)e(k) - \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}(k). \quad (3.12)$$

Escrevendo a equação (3.12) de forma mais conveniente, obtém-se:

$$\mathbf{w}(k+1) = [\mathbf{I} - \kappa \mathbf{F}^T \mathbf{F}] \mathbf{w}(k) + \beta \mathbf{x}(k)e(k). \quad (3.13)$$

### 3.3 Resultados

Nesta seção, são feitas simulações acerca da performance do algoritmo proposto em comparação com os algoritmos LMS e F-LMS [2]. Tais simulações consistem em identificar dois sistemas, como na configuração proposta neste trabalho, com 20 coeficientes e cujas respostas ao impulso são definidas por:

$$w_{i,\ell}^* = \begin{cases} 0, & \text{if } 0 \leq i \leq 9 \\ 0.05(i-9), & \text{if } 10 \leq i \leq 14 \\ 0.3, & \text{if } 15 \leq i \leq 19 \end{cases}, \quad (3.14)$$

para o sistema passa-baixas a ser identificado, e

$$w_{i,h}^* = (-1)^i w_{i,\ell}^*, \quad (3.15)$$

para o sistema passa-altas. O sinal de entrada é obtido filtrando-se um sinal branco, gaussiano, e de variância unitária pela função de transferência  $H(z) = 1 + 0.8z^{-1} - 0.2z^{-2}$ . É assumido que o ruído de medição é um sinal aditivo branco e gaussiano de variância  $\sigma_v^2 = 10^{-1}$ . Nas simulações utilizando o filtro ótimo  $w_\ell^*$ , foram utilizados  $\kappa = 0.0316$  e  $\kappa = 0.0008$  para os algoritmos  $\ell_2$ -F-LMS e F-LMS, respectivamente. Já nas simulações utilizando o filtro ótimo  $w_h^*$ , foram utilizados  $\kappa = 0.0032$  e  $\kappa = 0.000075$  para os algoritmos  $\ell_2$ -F-LMS e F-LMS, respectivamente.

As curvas de aprendizado são obtidas fazendo a média dos resultados provenientes de 1000 simulações independentes de Monte Carlo.

A Figura 3.1 mostra a evolução do MSD para os 3 algoritmos propostos quando se utiliza o filtro ótimo  $w_\ell^*$ . A Figura 3.1(a) mostra os resultados utilizando  $\beta = 0.015$  para todos os algoritmos, enquanto que a Figura 3.1(b) mostra os resultados utilizando um passo de aprendizagem específico de 0.0083, 0.015 e 0.014 para os algoritmos LMS,  $\ell_2$ -norm-F-LMS e F-LMS, respectivamente, de forma que o MSD tivesse o mesmo valor, e a taxa de convergência dos algoritmos pudesse ser comparada. Sendo assim, é possível observar que os algoritmos da família "Feature" LMS conseguem atingir um valor de MSD menor em regime permanente, de mais de 2dB nesse caso, e também possuem maior taxa de convergência. Sendo ainda possível observar que o algoritmo proposto neste capítulo, alcança valores menores de MSD e possui maior taxa de convergência no caso estudado.

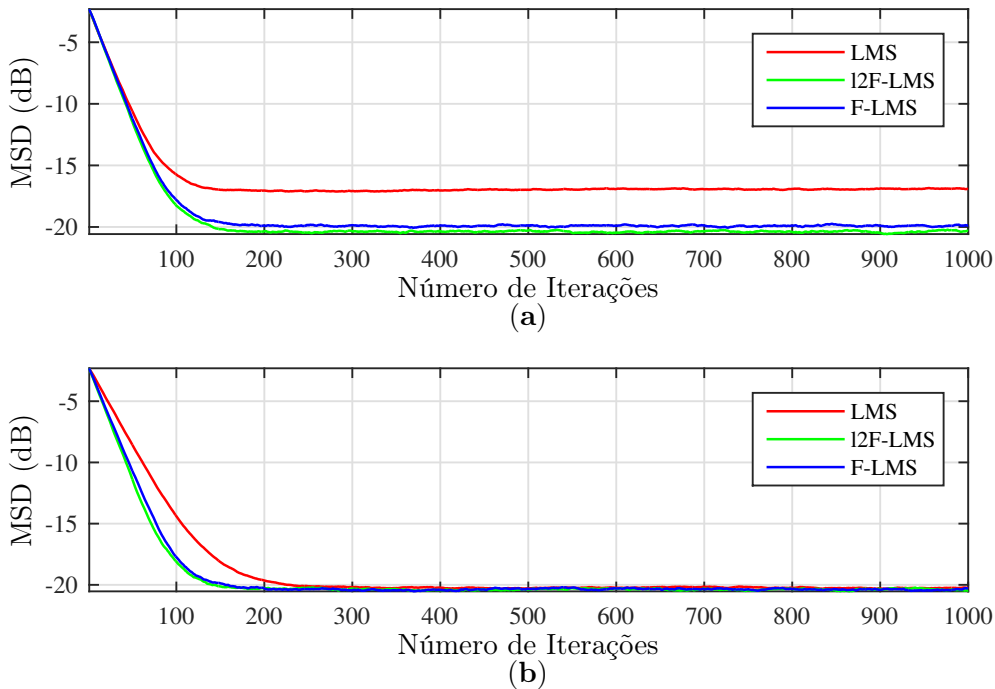


Figura 3.1: Comparação da evolução do MSD simulado dos algoritmos LMS (em vermelho),  $\ell_2$ -norm-F-LMS (em verde) e F-LMS, utilizando o filtro ótimo  $w_\ell^*$ . (a) Utilizando  $\beta = 0.015$  para todos. (b) Utilizando  $\beta = 0.0083$ ,  $\beta = 0.015$  e  $\beta = 0.014$ , respectivamente.

A Figura 3.2 mostra a evolução do MSD para os 3 algoritmos quando se utiliza o filtro ótimo  $w_h^*$ . A Figura 3.2(a) mostra os resultados utilizando  $\beta = 0.015$  para todos os algoritmos, enquanto que a Figura 3.2(b) mostra os resultados utilizando um passo de aprendizagem específico de 0.0135, 0.015 e 0.0145 para os algoritmos LMS,  $\ell_2$ -norm-F-LMS e F-LMS, respectivamente, de forma que o MSD tivesse o mesmo

valor, e a taxa de convergência dos algoritmos pudesse ser comparada. Sendo assim, é possível observar que os algoritmos da família *Feature LMS* conseguem atingir um valor de MSD menor em regime permanente, (de mais de 2dB nesse caso), e também possuem maior taxa de convergência. Sendo ainda possível observar que o algoritmo proposto neste capítulo alcança valores menores de MSD e possui maior taxa de convergência no caso estudado.

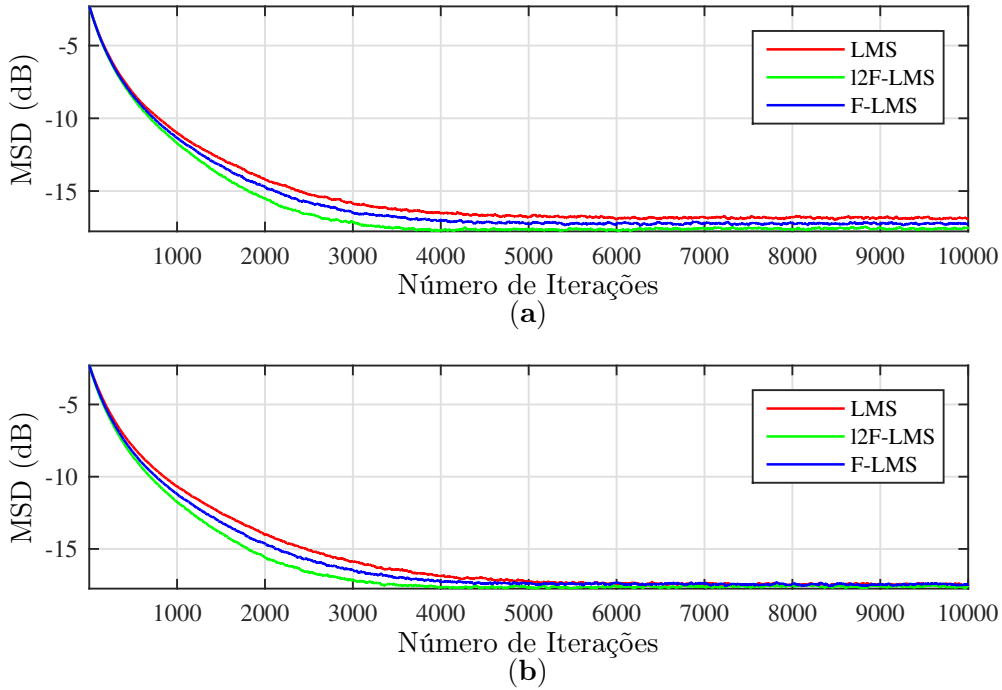


Figura 3.2: Comparação da evolução do MSD simulado dos algoritmos LMS (em vermelho),  $\ell_2$ -norm-F-LMS (em verde) e F-LMS, utilizando o filtro ótimo  $w_h^*$ . (a) Utilizando  $\beta = 0.015$  para todos. (b) Utilizando  $\beta = 0.0135$ ,  $\beta = 0.015$  e  $\beta = 0.0145$ , respectivamente.

### 3.4 Comentários Finais

Como pôde ser visto, os algoritmos da família *Feature LMS* são capazes de obter melhores resultados que o algoritmo LMS, para a classe de funções de transferências utilizadas. Vale a pena destacar o desempenho do algoritmo proposto ( $\ell_2$ -F-LMS) em comparação aos outros dois algoritmos, alcançando menores valores de MSD em regime permanente e apresentando maior taxa de convergência.

# Capítulo 4

## Análise Estatística $\ell_2$ -norm Feature LMS

### 4.1 Introdução

A elaboração de modelos determinísticos ou estocásticos para estimar a performance de filtros adaptativos é de alta importância, pois estes modelos fornecem garantias de performance, diretrizes para os projetistas, limites de estabilidade, estimativas de comportamento em regime transiente e regime permanente ou até esclarecem se o algoritmo é robusto perante perturbações [39].

Neste capítulo, é feita uma análise de transiente, de regime permanente e de rastreamento do algoritmo  $\ell_2$ -norm Feature LMS. Tal algoritmo penaliza soluções *filtradas* que apresentem alta energia. Ele é portanto capaz de explorar a relação entre os coeficientes da resposta ao impulso do sistema a ser identificado, que, como dito anteriormente, com algum conhecimento *a priori* dessa relação, pode melhorar o comportamento do filtro adaptativo tanto em regime transiente quanto em regime permanente.

### 4.2 Métodos de Análise

A análise teórica de algoritmos adaptativos é de grande importância, pois cabe a ela fornecer de antemão informações sobre as características e o comportamento do algoritmo ao projetista. Diversas técnicas de análises foram empregadas na literatura ao longo dos anos, dentre as quais cumpre destacar: análise de esperança estatística exata [40], balanceamento de energia [41] e cálculo recursivo da matriz de autocorrelação dos desvios dos coeficientes [1]. A maior parte das técnicas tentam estimar de maneira confiável a evolução de duas métricas principais de desempenho em algoritmos adaptativos, o MSE e o MSD.

Neste trabalho, é feita uma análise do algoritmo  $\ell_2$ -norm Feature LMS, empregando a técnica do cálculo recursivo da matriz de autocorrelação dos desvios. Tal técnica consiste em determinar equações a diferenças que consigam prever o comportamento de  $\mathbf{R}_{\tilde{\mathbf{w}}}(k) \triangleq \mathbb{E}[\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)]$ , onde  $\tilde{\mathbf{w}}(k) = \mathbf{w}^* - \mathbf{w}(k)$  é o vetor de desvio dos coeficientes adaptativos na  $k$ -ésima iteração. Vale notar que o cálculo de tais parâmetros exigem o cálculo de estatísticas de segunda ordem que, por sua vez, dependem do cálculo de estatísticas de primeira ordem. Além disso, a análise de primeira ordem permite obter informações importantes acerca das estimativas dos coeficientes. Principalmente se tal estimativa possui com algum tipo de viés ou não.

Para que a análise seja possível, é necessário que algumas hipóteses sejam feitas:

**H1:**  $\mathbf{x}(k)$ ,  $\tilde{\beta}(k)$  e  $\tilde{\mathbf{w}}(k)$  são mutuamente independentes.

**H2:** O ruído aditivo  $\nu(k)$ , de média zero e variância finita, é branco e independente de  $\mathbf{x}(k)$ . Sua variância no instante  $k$  é descrita por  $\sigma_\nu^2(k)$ .

**H3:** A distribuição dos coeficientes  $\tilde{\mathbf{w}}_i(k)$  é Gaussiana.

**H4:** O sinal de entrada é estacionário, gaussiano e todos os autovalores da sua matriz de autocorrelação são não nulos.

Com as hipóteses **H1** e **H2**, podemos calcular o MSE segundo [42]:

$$\xi(k) \triangleq \mathbb{E}[e^2(k)] \approx \sigma_\nu^2(k) + \text{Tr}\{\mathbf{R}_{\mathbf{x}}\mathbb{E}[\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)]\}, \quad (4.1)$$

onde  $\mathbf{R}_{\mathbf{x}} = \mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)]$  é a autocorrelação do sinal de entrada,  $\text{Tr}\{\mathbf{X}\}$  é o traço da matriz  $\mathbf{X}$  e  $\sigma_\nu^2(k)$  é a variância do ruído no instante  $k$ .

Já o MSD pode ser calculado pela seguinte equação:

$$\text{MSD}(k) \triangleq \mathbb{E}[\|\mathbf{w}^* - \mathbf{w}(k)\|^2] = \text{Tr}\{\mathbb{E}[\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)]\}. \quad (4.2)$$

### 4.3 Alguns Comentários Sobre as Hipóteses

Vale fazer algumas observações acerca das hipóteses descritas e que serão utilizadas ao longo deste capítulo. Em H1 é suposto que  $\mathbf{x}(k)$ ,  $\tilde{\beta}(k)$  e  $\tilde{\mathbf{w}}(k)$  são mutuamente independentes. A independência entre  $\mathbf{x}(k)$  e  $\tilde{\mathbf{w}}(k)$  é uma propriedade conhecida como “*independence assumption*” e amplamente encontrada na literatura aberta [42–45]. Já no que tange a independência de  $\tilde{\beta}(k)$  com relação às outras variáveis aleatórias, é fácil notar que para o caso do algoritmo LMS a hipótese é verdadeira, uma vez que  $\beta$  é uma constante que não depende do valor assumido por  $x(k)$ . Porém, no caso do algoritmo NLMS é preciso aplicar o “*Averaging Principle*” [46] na expressão  $\mathbb{E}[\tilde{\beta}(k)A]$ , onde  $\tilde{\beta}(k)$  e  $A$  são variáveis aleatórias ditas conjuntamente estacionárias [20, 47, 48]. Tal princípio defende que  $\tilde{\beta}(k)$  varia lentamente com relação ao sinal de entrada, permitindo que  $\mathbb{E}[\tilde{\beta}(k)A]$  seja aproximado por  $\mathbb{E}[\tilde{\beta}(k)]\mathbb{E}[A]$ .

Sobre a hipótese H2, vale notar que ela reforça a linearidade entre  $\mathbf{x}(k)$  e  $d(k)$ , o que é razoável em diversas aplicações [1]. Este trabalho não fica restrito a uma variância do ruído constante como é comum na literatura, aceitando que ela varie ao longo do tempo.

Assim como H1, H3 é uma hipótese muito encontrada na literatura e que pode ser justificada pelo teorema central do limite [20]. Resultados acerca da distribuição dos coeficientes de  $\mathbf{w}(k)$ , e que corroboram a utilização desta hipótese, podem ser encontrados também em [1].

A hipótese H4 não assume que o sinal de entrada seja branco, tornando o estudo aqui feito mais abrangente do que é comumente encontrado a este respeito. Porém, não engloba o caso de todos os sinais coloridos, uma vez que assume-se que todos os autovalores de  $\mathbf{R}_x$  são não-nulos.

## 4.4 Análise de Primeira Ordem

Apesar de definida a equação de atualização do algoritmo, a equação (3.13) não é suficiente para se fazer a análise de transiente. Para que a análise de transiente seja possível, é necessário encontrar uma equação de atualização recursiva capaz de estimar a evolução dos desvios. Portanto, considere que os desvios são definidos da seguinte forma:

$$\tilde{\mathbf{w}}(k+1) \triangleq \mathbf{w}^* - \mathbf{w}(k+1). \quad (4.3)$$

Logo, substituindo a equação (3.13) na equação (4.3) obtém-se:

$$\tilde{\mathbf{w}}(k+1) = \mathbf{w}^* - \mathbf{w}(k+1) = \mathbf{w}^* - [\mathbf{I} - \kappa \mathbf{F}^T \mathbf{F}] \mathbf{w}(k) - \tilde{\beta}(k) \mathbf{x}(k) e(k). \quad (4.4)$$

A equação (4.4) ainda não representa uma equação recursiva. Portanto, manipulando-a algebricamente e considerando a definição de  $e(k)$ :  $e(k) \triangleq d(k) - \mathbf{w}^T(k) \mathbf{x}(k)$  obtém-se:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) &= \mathbf{w}^* - \mathbf{w}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}(k) \\ &\quad - \tilde{\beta}(k) \mathbf{x}(k) [(\mathbf{w}^*)^T \mathbf{x}(k) + \nu(k) - \mathbf{w}^T(k) \mathbf{x}(k)] \\ &= \tilde{\mathbf{w}}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* - \kappa \mathbf{F}^T \mathbf{F} \tilde{\mathbf{w}}(k) \\ &\quad - \tilde{\beta}(k) \mathbf{x}(k) \tilde{\mathbf{w}}^T(k) \mathbf{x}(k) - \tilde{\beta}(k) \mathbf{x}(k) \nu(k) \\ &= \tilde{\mathbf{w}}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* - \kappa \mathbf{F}^T \mathbf{F} \tilde{\mathbf{w}}(k) \\ &\quad - \tilde{\beta}(k) \mathbf{x}(k) \mathbf{x}^T(k) \tilde{\mathbf{w}}(k) - \tilde{\beta}(k) \mathbf{x}(k) \nu(k). \end{aligned} \quad (4.5)$$

Por fim, a equação de atualização dos desvios pode ser escrita como:

$$\tilde{\mathbf{w}}(k+1) = [\mathbf{I} - \tilde{\beta}(k) \mathbf{x}(k) \mathbf{x}^T(k) - \kappa \mathbf{F}^T \mathbf{F}] \tilde{\mathbf{w}}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* - \tilde{\beta}(k) \mathbf{x}(k) \nu(k) \quad (4.6)$$



A técnica de cálculo da matriz de autocorrelação dos desvios permite uma análise estatística do comportamento do algoritmo, ou seja, permite estimar como o algoritmo se comporta na média. Então, para que seja possível utilizar a equação de atualização dos desvios, é necessário aplicar o operador estatístico do valor esperado à equação (4.6). Obtendo-se então:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{w}}(k+1)] = & \mathbb{E} \left\{ \left[ \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \right] \tilde{\mathbf{w}}(k) \right\} \\ & + \mathbb{E}[\kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*] \\ & - \mathbb{E}[\tilde{\beta}(k)\mathbf{x}(k)\nu(k)] \end{aligned} \quad (4.7)$$

Aplicando a propriedade de linearidade do operador do valor esperado e **H2**, é possível reescrever a equação (4.7) da seguinte forma:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{w}}(k+1)] = & \mathbb{E}[\tilde{\mathbf{w}}(k)] - \mathbb{E}[\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)] \\ & - \kappa\mathbf{F}^T\mathbf{F}\mathbb{E}[\tilde{\mathbf{w}}(k)] + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^* \end{aligned} \quad (4.8)$$

Em sequência, aplicando a hipótese **H1**, a qual diz que  $\tilde{\beta}(k)$ ,  $x(k)$  e  $\tilde{w}(k)$  são mutuamente independentes, a equação (4.8) pode ser reescrita da seguinte maneira:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{w}}(k+1)] = & \mathbb{E}[\tilde{\mathbf{w}}(k)] - \mathbb{E}[\tilde{\beta}(k)]\mathbf{R}_x\mathbb{E}[\tilde{\mathbf{w}}(k)] \\ & - \kappa\mathbf{F}^T\mathbf{F}\mathbb{E}[\tilde{\mathbf{w}}(k)] + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*, \end{aligned} \quad (4.9)$$

onde  $\mathbf{R}_x \triangleq \mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)]$  é a matriz de autocorrelação do sinal de entrada.

Pela equação (4.9) é possível perceber que o algoritmo faz uma estimativa enviesada do filtro ótimo. Tal viés pode ser calculado aplicando o operador de limite, bem como sua propriedade de linearidade, e reescrevendo a equação (4.9) de forma a se obter:

$$\begin{aligned} \mathbf{0} = & \lim_{k \rightarrow \infty} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] - \mathbb{E}[\tilde{\mathbf{w}}(k+1)] \} \\ & - \lim_{k \rightarrow \infty} \left\{ \mathbb{E}[\tilde{\beta}(k)]\mathbf{R}_x\mathbb{E}[\tilde{\mathbf{w}}(k)] + \kappa\mathbf{F}^T\mathbf{F}\mathbb{E}[\tilde{\mathbf{w}}(k)] \right\} \\ & + \lim_{k \rightarrow \infty} \{ \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^* \}, \end{aligned} \quad (4.10)$$

considerando que o algoritmo convirja quando  $k \rightarrow \infty$ , pode-se considerar que  $\mathbb{E}[\tilde{\mathbf{w}}(k+1)] \approx \mathbb{E}[\tilde{\mathbf{w}}(k)]$ . Logo, a equação (2.7) pode ser reescrita como:

$$\lim_{k \rightarrow \infty} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] \} = \kappa \left\{ \mathbb{E}[\tilde{\beta}(k)]\mathbf{R}_x + \kappa\mathbf{F}^T\mathbf{F} \right\}^{-1} \mathbf{F}^T\mathbf{F}\mathbf{w}^*, \quad (4.11)$$

o que significa que o algoritmo  $\ell_2$ -F-LMS, ao contrário do LMS, é assintoticamente enviesado. Devido ao compromisso entre viés e variância, espera-se que este viés seja compensado por uma menor variância dos desvios em regime permanente [4].

## 4.5 Análise de Segunda Ordem

Apesar da análise do momento de primeira ordem algoritmo adaptativo, a qual resultou na equação (4.10), permitir verificar que sua estimativa do filtro ótimo possui um viés e como esse viés é calculado, essa análise não permite verificar a evolução dos coeficientes do filtro adaptativo. Portanto, cabe ainda fazer a análise do momento de segunda ordem do algoritmo que permite verificar a evolução dos coeficientes.

Como o método escolhido para essa realizar tal análise foi o método do cálculo da matriz de autocorrelação dos desvios, deve-se calcular a evolução da matriz  $\mathbf{\Gamma}(k) \triangleq \tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)$  para, então, aplicar-se o operador do valor esperado como feito na análise do momento de primeira ordem.

Utilizando a equação recursiva de atualização dos desvios, encontrada na equação (4.6), para calcular  $\tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1)$ , obtém-se:

$$\begin{aligned}
\tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1) = & \tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) + \kappa\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\
& - \kappa\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} - \tilde{\beta}(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\
& - \tilde{\beta}(k)\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k) \\
& + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} - \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\
& - \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\
& - \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\mathbf{w}^*\mathbf{x}^T(k)\nu(k) - \kappa\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \\
& - \kappa^2\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\
& + \kappa^2\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\
& + \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\
& + \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) \\
& - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \\
& - \kappa\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\
& + \kappa\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\
& + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\
& + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) \\
& - \tilde{\beta}(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\nu(k) \\
& - \kappa\tilde{\beta}(k)\mathbf{x}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}\nu(k) \\
& + \kappa\tilde{\beta}(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F}\nu(k) \\
& + \tilde{\beta}^2(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu(k) \\
& + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k)
\end{aligned} \tag{4.12}$$

Pode-se observar que a equação (4.12) é uma equação muito grande e, portanto, pode ser difícil manuseá-la caso o operador do valor esperado fosse aplicado agora. Vale a pena notar também que, apesar de se tratar de uma equação recursiva, a equação (4.12) não estabelece uma relação de recursão clara entre a matriz de

desvios  $\Gamma$  no instante  $k + 1$  e no instante  $k$ . Portanto, a equação (4.12) será reescrita da seguinte forma:

$$\Gamma(k + 1) = \tilde{\mathcal{W}}[\Gamma(k)] + \mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \nu^2] + \mathcal{O}[\nu(k)], \quad (4.13)$$

onde  $\tilde{\mathcal{W}}[\Gamma(k)]$  contém todos os termos com  $\Gamma(k)$ ,  $\mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \nu^2]$  contém todos os termos com  $\tilde{\mathbf{w}}, \mathbf{w}^*$  e  $\nu^2$  que não dependem de termos de ruído de primeira ordem, pois estes estão contidos na matriz  $\mathcal{O}[\nu(k)]$ . Dessa forma obtém-se as seguintes equações:

$$\begin{aligned} \tilde{\mathcal{W}}[\Gamma(k)] = & \tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) - \kappa\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \\ & - \tilde{\beta}(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\ & + \kappa\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\ & + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ & + \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ & + \kappa^2\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \end{aligned} \quad (4.14)$$

$$\begin{aligned} \mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \nu^2] = & \kappa\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} - \kappa\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\ & - \kappa^2\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\ & + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k) - \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ & - \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\ & + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k) \end{aligned} \quad (4.15)$$

$$\begin{aligned} \mathcal{O}[\nu(k)] = & \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) - \tilde{\beta}(k)\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) \\ & - \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\mathbf{w}^*\mathbf{x}^T(k)\nu(k) \\ & + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\mathbf{x}^T(k)\nu(k) \\ & - \tilde{\beta}(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\nu(k) \\ & - \kappa\tilde{\beta}(k)\mathbf{x}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}\nu(k) \\ & + \kappa\tilde{\beta}(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F}\nu(k) \\ & + \tilde{\beta}^2(k)\mathbf{x}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu(k) \end{aligned} \quad (4.16)$$

Com a equação (4.12) fragmentada nas equações (4.14), (4.15) e (4.16), fica mais fácil fazer algumas manipulações algébricas para tornar a equação (4.12) mais coesa. O primeiro passo para atingir tal objetivo pode ser extraído a partir da observação da equação (4.15), reescrevendo-a da seguinte maneira:

$$\begin{aligned} \mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \nu^2] = & \kappa \left[ \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \right] \tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\ & + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k) \left[ \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \right] \\ & + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k) \end{aligned} \quad (4.17)$$

O segundo passo é extraído da equação (4.14). Note que a equação (4.14) possui

muitos termos em comum com a equação (4.15), logo pode-se fazer uma manipulação similar a da equação (4.17). Portanto, a equação (4.14) pode ser reescrita da seguinte forma:

$$\begin{aligned}\tilde{\mathcal{W}}[\boldsymbol{\Gamma}(k)] &= \begin{bmatrix} \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \\ \boldsymbol{\Gamma}(k) - \tilde{\beta}(k)\boldsymbol{\Gamma}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\boldsymbol{\Gamma}(k)\mathbf{F}^T\mathbf{F} \end{bmatrix} \\ &= \mathbf{A}(k, \beta, \kappa)\boldsymbol{\Gamma}(k)\mathbf{A}(k, \beta, \kappa),\end{aligned}\quad (4.18)$$

onde

$$\mathbf{A}(k, \beta, \kappa) = \begin{bmatrix} \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \\ \boldsymbol{\Gamma}(k) - \tilde{\beta}(k)\boldsymbol{\Gamma}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\boldsymbol{\Gamma}(k)\mathbf{F}^T\mathbf{F} \end{bmatrix} = \mathbf{A}^T(k, \beta, \kappa) \quad (4.19)$$

Não são necessárias manipulações algébricas na equação (4.16) por motivo que ficará claro um pouco mais adiante. Cabe ainda substituir as equações (4.18) e (4.17) na equação (4.13), de forma que a equação de atualização da matriz de desvios possa ser reescrita da seguinte forma:

$$\begin{aligned}\boldsymbol{\Gamma}(k+1) &= \mathbf{A}(k, \beta, \kappa)\boldsymbol{\Gamma}(k)\mathbf{A}(k, \beta, \kappa) + \mathcal{O}[\nu(k)] \\ &\quad + \kappa\mathbf{A}(k, \beta, \kappa)\tilde{\boldsymbol{\omega}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\boldsymbol{\omega}}^T(k)\mathbf{A}(k, \beta, \kappa) \\ &\quad + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k)\end{aligned}\quad (4.20)$$

Apesar das manipulações realizadas terem cumprido seu papel ao tornar a equação de atualização da matriz de desvios mais coesa, tais manipulações não resolvem o problema da equação não ser uma recursão característica. Um meio de contornar esse problema é aplicando o operador  $\text{vec}(\mathbf{X})$ , o qual empilha, consecutivamente, as colunas da matriz  $\mathbf{X}$ , formando um vetor coluna como resultado. Isso permite que a equação (4.20) seja reescrita da seguinte forma:

$$\begin{aligned}\text{vec}[\boldsymbol{\Gamma}(k+1)] &= \text{vec}[\mathbf{A}(k, \beta, \kappa)\boldsymbol{\Gamma}(k)\mathbf{A}(k, \beta, \kappa)] + \text{vec}\{\mathcal{O}[\nu(k)]\} \\ &\quad + \kappa\text{vec}[\mathbf{A}(k, \beta, \kappa)\tilde{\boldsymbol{\omega}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}] \\ &\quad + \kappa\text{vec}[\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\boldsymbol{\omega}}^T(k)\mathbf{A}(k, \beta, \kappa)] \\ &\quad + \kappa^2\text{vec}[\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}] \\ &\quad + \tilde{\beta}^2(k)\nu^2(k)\text{vec}[\mathbf{x}(k)\mathbf{x}^T(k)]\end{aligned}\quad (4.21)$$

A aplicação do operador  $\text{vec}(X)$  para tentar resolver o problema da equação (4.20), é justificada por este operador apresentar uma propriedade muito conveniente para este caso, a qual permite escrever:

$$\text{vec}(\mathbf{XYZ}) = [\mathbf{Z}^T \otimes \mathbf{X}] \text{vec}(\mathbf{Y}). \quad (4.22)$$

Usando a equação (4.22) na equação (4.21), pode ser mostrado que:

$$\begin{aligned} \text{vec} [\mathbf{\Gamma}(k+1)] &= \mathbf{\Theta}(k, \kappa, \beta) \text{vec} [\mathbf{\Gamma}(k)] + \text{vec} \{ \mathcal{O} [\nu(k)] \} \\ &\quad + \kappa \mathbf{\Lambda}_1(k) \text{vec} [\tilde{\mathbf{w}}(k) (\mathbf{w}^*)^T] \\ &\quad + \kappa \mathbf{\Lambda}_2(k) \text{vec} [\mathbf{w}^* \tilde{\mathbf{w}}^T(k)] \\ &\quad + \tilde{\beta}^2(k) \nu^2(k) \mathbf{\mathcal{X}}(k) + \kappa^2 \mathbf{\Psi}, \end{aligned} \quad (4.23)$$

onde

$$\mathbf{\Theta}(k, \kappa, \beta) \triangleq \mathbf{A}(k, \kappa, \beta) \otimes \mathbf{A}(k, \kappa, \beta), \quad (4.24)$$

$$\mathbf{\Lambda}_1(k) \triangleq [\mathbf{F}^T \mathbf{F}] \otimes \mathbf{A}(k, \kappa, \beta), \quad (4.25)$$

$$\mathbf{\Lambda}_2(k) \triangleq [\mathbf{A}(k, \kappa, \beta) \otimes \mathbf{F}^T \mathbf{F}], \quad (4.26)$$

$$\mathbf{\mathcal{X}}(k) \triangleq \text{vec} [\mathbf{x}(k) \mathbf{x}^T(k)], \quad (4.27)$$

$$\mathbf{\Psi} \triangleq [(\mathbf{F}^T \mathbf{F}) \otimes (\mathbf{F}^T \mathbf{F})] \text{vec} \{ \mathbf{w}^* [\mathbf{w}^*]^T \}. \quad (4.28)$$

Assim como no caso da análise de primeira ordem, cabe agora aplicar o operador do valor esperado à equação (4.23). Tal procedimento permite, então, estimar o comportamento do algoritmo adaptativo, na média, através da evolução da matriz de desvios dos coeficientes:

$$\begin{aligned} \mathbb{E} \{ \text{vec} [\mathbf{\Gamma}(k+1)] \} &= \mathbb{E} \{ \mathbf{\Theta}(k, \kappa, \beta) \text{vec} [\mathbf{\Gamma}(k)] \} + \mathbb{E} \{ \text{vec} \{ \mathcal{O} [\nu(k)] \} \} \\ &\quad + \kappa \mathbb{E} \{ \mathbf{\Lambda}_1(k) \text{vec} [\tilde{\mathbf{w}}(k) (\mathbf{w}^*)^T] \} \\ &\quad + \kappa \mathbb{E} \{ \mathbf{\Lambda}_2(k) \text{vec} [\mathbf{w}^* \tilde{\mathbf{w}}^T(k)] \} \\ &\quad + \mathbb{E} \{ \tilde{\beta}^2(k) \nu^2(k) \mathbf{\mathcal{X}}(k) \} + \kappa^2 \mathbb{E} \{ \mathbf{\Psi} \} \end{aligned} \quad (4.29)$$

Cabe agora aplicar as hipóteses **H1** e **H2**. **H1** diz que  $\tilde{\beta}(k)$ ,  $\mathbf{x}(k)$  e  $\tilde{\mathbf{w}}(k)$  são mutuamente independentes. Já **H2** permite inferir que  $\mathbb{E} \{ \text{vec} \{ \mathcal{O} [\nu(k)] \} \} = 0$ , tornando claro o motivo de não serem necessárias nenhum tipo de manipulação algébrica para  $\mathcal{O} [\nu(k)]$ . A aplicação dessas hipóteses permite que a equação (4.29) seja reescrita da seguinte maneira:

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbb{E} [\mathbf{\Theta}(k, \kappa, \beta)] \mathbf{v}(k) + \sigma_\nu^2 \mathbb{E} [\tilde{\beta}^2(k)] \mathbb{E} [\mathbf{\mathcal{X}}(k)] \\ &\quad + \kappa \mathbb{E} [\mathbf{\Lambda}_1(k)] \text{vec} \{ \mathbb{E} [\tilde{\mathbf{w}}(k)] (\mathbf{w}^*)^T \} \\ &\quad + \kappa \mathbb{E} [\mathbf{\Lambda}_2(k)] \text{vec} \{ \mathbf{w}^* \mathbb{E} [\tilde{\mathbf{w}}^T(k)] \} + \kappa^2 \mathbf{\Psi}, \end{aligned} \quad (4.30)$$

onde  $\mathbf{v}(k) \triangleq \mathbb{E} \{ \text{vec} [\mathbf{\Gamma}(k)] \}$ . Note que como  $\text{vec}(\mathbf{X})$  é um operador bijetor, a matriz  $\mathbf{R}_{\tilde{\mathbf{w}}}(k)$ , onde  $\mathbf{R}_{\tilde{\mathbf{w}}}(k) \triangleq \mathbb{E} [\tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k)]$ , pode ser facilmente recuperada a partir de  $\mathbf{v}(k)$ .

Vale notar que todo o desenvolvimento feito até agora é independente do algoritmo ser normalizado ou não. Então, cabe, por fim, chegar nas devidas equações de

atualização da matriz de autocorrelação dos desvios. Para o caso não normalizado, tem-se que  $\tilde{\beta}(k) = \beta$ , logo a equação (4.30) pode ser escrita da seguinte maneira:

$$\begin{aligned} \mathbf{v}(k+1) = & \mathbb{E}[\boldsymbol{\Theta}(k, \kappa, \beta)] \mathbf{v}(k) + \sigma_\nu^2 \beta^2 \mathbb{E}[\boldsymbol{\mathcal{X}}(k)] + \kappa^2 \boldsymbol{\Psi} \\ & + \kappa \mathbb{E}[\boldsymbol{\Lambda}_1(k)] \text{vec} \left\{ \mathbb{E}[\tilde{\mathbf{w}}(k)] (\mathbf{w}^*)^T \right\} \\ & + \kappa \mathbb{E}[\boldsymbol{\Lambda}_2(k)] \text{vec} \left\{ \mathbf{w}^* \mathbb{E}[\tilde{\mathbf{w}}^T(k)] \right\}. \end{aligned} \quad (4.31)$$

Note também que a equação (4.31) depende do termo  $\mathbb{E}[\tilde{\mathbf{w}}(k)]$ , o qual pode ser encontrado pela equação (4.9), a qual por sua vez, para o algoritmo não normalizado, pode ser reescrita da seguinte forma:

$$\begin{aligned} \mathbb{E}[\tilde{\mathbf{w}}(k+1)] = & \mathbb{E}[\tilde{\mathbf{w}}(k)] - \beta \mathbf{R}_x \mathbb{E}[\tilde{\mathbf{w}}(k)] \\ & - \kappa \mathbf{F}^T \mathbf{F} \mathbb{E}[\tilde{\mathbf{w}}(k)] + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^*. \end{aligned} \quad (4.32)$$

Com uma equação estatística acerca da evolução da matriz de autocorrelação dos desvios, é possível estimar a evolução do MSE e do MSD ao longo do tempo:

$$\text{MSE}(k) = \sigma_\nu^2 + \text{Tr}[\mathbf{R}_x \mathbf{R}_{\tilde{\mathbf{w}}}(k)] \quad (4.33)$$

$$\text{MSD}(k) = \text{Tr}[\mathbf{R}_{\tilde{\mathbf{w}}}(k)]. \quad (4.34)$$

## 4.6 Identificação de Sistemas Variantes no Tempo

Uma das principais motivações para o uso de filtros adaptativos é a capacidade que esses filtros têm de rastrear sistemas que variam com o tempo. Tornando-se, então, interessante fazer uma análise a respeito do comportamento desse algoritmo perante o cenário descrito.

É comum na literatura que essa análise seja feita considerando o sistema a ser identificado um processo de Markov de primeira ordem [42, 45, 49–51] de modo que:

$$\mathbf{w}^*(k+1) = \lambda_w \mathbf{w}^*(k) + \mathbf{q}(k), \quad (4.35)$$

onde  $\lambda_w < 1$  é um fator de esquecimento muito próximo de 1. Note que  $\lambda_w \rightarrow 1$  pode não representar um sistema real devido a  $\mathbb{E}[\mathbf{w}(k)\mathbf{w}^T(k)]$  apresentar elementos não limitados se, por exemplo,  $\mathbf{q}(k)$  não possuir média exatamente igual a zero [42]. Porém, considerar que  $\lambda_w \approx 1$  não compromete a análise proposta [42, 45, 50, 51]. Portanto,  $\lambda_w$  será considerado igual a 1 daqui em diante.

A equação (4.35) torna necessária a redefinição do vetor de desvios dos coeficientes, de modo que o mesmo seja descrito por:

$$\tilde{\mathbf{w}}(k) \triangleq \mathbf{w}^*(k) - \mathbf{w}(k), \quad (4.36)$$

e o sinal desejado seja descrito por:

$$d(k) = [\mathbf{w}^*(k)]^T \mathbf{x}(k) + \nu(k). \quad (4.37)$$

Visando tornar essa análise possível, a seguinte hipótese é considerada:

**H5:** o vetor aleatório  $\mathbf{q}(k) \in \mathbb{R}^N$  possui média 0, é i.i.d. (independente e igualmente distribuído) e independente das outras variáveis aleatórias. Portanto, a matriz de covariância  $\mathbf{Q} \triangleq \mathbb{E} [\mathbf{q}(k)\mathbf{q}^T(k)]$  é igual a  $\sigma_q^2 \mathbf{I}$ , onde  $\sigma_q^2$  é a variância de  $q_i(k)$ . Vale notar que **H5** é amplamente aceita na literatura e conhecida como *Tracking Assumption* [42, 45, 49].

Substituindo a equação (4.35) na equação (4.6), obtém-se para obter a nova equação de atualização dos desvios. Logo:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) = & [\mathbf{I} - \beta \mathbf{x}(k)\mathbf{x}^T(k) - \kappa \mathbf{F}^T \mathbf{F}] \tilde{\mathbf{w}}(k) + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^*(k) \\ & - \beta \mathbf{x}(k)\nu(k) + \mathbf{q}(k) \end{aligned} \quad (4.38)$$

Assim como na análise de identificação de sistemas invariantes no tempo, cabe agora aplicar o operador de valor esperado na equação (4.38). Sendo possível, então, obter as estatísticas de primeira ordem do algoritmo para o cenário estudado nesta seção e, conseqüentemente, obtendo informações acerca de seu comportamento em regime permanente. Aplicando-se **H1**, **H2** e **H5**, obtém-se como resultado:

$$\mathbb{E} [\tilde{\mathbf{w}}(k+1)] = [\mathbf{I} - \beta \mathbf{R}_x - \kappa \mathbf{F}^T \mathbf{F}] \mathbb{E} [\tilde{\mathbf{w}}(k)] + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}_m^*, \quad (4.39)$$

onde  $\mathbb{E} [\mathbf{w}^*(k)] = \mathbf{w}^*(0) = \mathbf{w}_m^*$ .

Supondo que o algoritmo convirja quando  $k \rightarrow \infty$  ( $\mathbb{E} [\tilde{\mathbf{w}}(k+1)] \approx \mathbb{E} [\tilde{\mathbf{w}}(k)]$ ), é possível calcular o viés do estimador proposto na equação (4.39). Logo:

$$\lim_{k \rightarrow \infty} \mathbb{E} [\tilde{\mathbf{w}}(k)] = [\beta \mathbf{R}_x - \kappa \mathbf{F}^T \mathbf{F}]^{-1} \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}_m^*, \quad (4.40)$$

note que este resultado é idêntico ao da equação (4.11). O que já era esperado, uma vez que a própria equação de atualização do valor esperado dos desvios é igual à equação (4.9).

Cabe ainda o cálculo das estatísticas de segunda ordem, para isso é necessário calcular  $\Gamma(k+1) = \tilde{\mathbf{w}}(k+1)\tilde{\mathbf{w}}^T(k+1)$ . Com base na equação de atualização dos desvios encontrada em (4.38), a fragmentação dos termos realizada na equação

(4.13) e a definição fetia na equação (4.19), obtém-se:

$$\begin{aligned}
\mathbf{\Gamma}(k+1) = & \mathbf{A}(k, \beta, \kappa)\mathbf{\Gamma}(k)\mathbf{A}(k, \beta, \kappa) + \mathcal{O}[\mathbf{q}(k)\nu(k)] \\
& + \kappa\mathbf{A}(k, \beta, \kappa)\tilde{\mathbf{w}}(k)(\mathbf{w}^*(k))^T\mathbf{F}^T\mathbf{F} \\
& + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)\mathbf{A}(k, \beta, \kappa) + \mathbf{q}(k)\mathbf{q}^T(k) \\
& + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(k)(\mathbf{w}^*(k))^T\mathbf{F}^T\mathbf{F} + \beta^2\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k)
\end{aligned} \tag{4.41}$$

onde  $\mathcal{O}[\mathbf{q}(k), \nu(k)]$  contém todos os termos de primeira ordem tanto do ruído de medição  $\nu(k)$  quanto do vetor aleatório  $\mathbf{q}(k)$ .

Assim como na análise de segunda ordem da seção anterior, é necessário aplicar o operador  $\text{vec}(\cdot)$  na equação (4.41) para que se tenha uma equação recursiva implementável. Portanto, usando a propriedade do operador  $\text{vec}(\cdot)$  descrita na equação (4.22), a equação (4.41) pode ser reescrita como:

$$\begin{aligned}
\text{vec}[\mathbf{\Gamma}(k+1)] = & \mathbf{\Theta}(k, \kappa, \beta)\text{vec}[\mathbf{\Gamma}(k)] + \text{vec}\{\mathcal{O}[\nu(k)]\} \\
& + \kappa\mathbf{\Lambda}_1(k)\text{vec}[\tilde{\mathbf{w}}(k)(\mathbf{w}^*(k))^T] \\
& + \kappa\mathbf{\Lambda}_2(k)\text{vec}[\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)] \\
& + \kappa^2\mathbf{\Psi}\text{vec}[\mathbf{w}^*(k)(\mathbf{w}^*(k))^T] \\
& + \beta^2\nu^2(k)\mathbf{\mathcal{X}}(k) + \text{vec}[\mathbf{q}(k)\mathbf{q}^T(k)],
\end{aligned} \tag{4.42}$$

onde

$$\mathbf{\Theta}(k, \kappa, \beta) \triangleq \mathbf{A}(k, \kappa, \beta) \otimes \mathbf{A}(k, \kappa, \beta), \tag{4.43}$$

$$\mathbf{\Lambda}_1(k) \triangleq [\mathbf{F}^T\mathbf{F}] \otimes \mathbf{A}(k, \kappa, \beta), \tag{4.44}$$

$$\mathbf{\Lambda}_2(k) \triangleq [\mathbf{A}(k, \kappa, \beta) \otimes \mathbf{F}^T\mathbf{F}], \tag{4.45}$$

$$\mathbf{\mathcal{X}}(k) \triangleq \text{vec}[\mathbf{x}(k)\mathbf{x}^T(k)], \tag{4.46}$$

$$\mathbf{\Psi} \triangleq [(\mathbf{F}^T\mathbf{F}) \otimes (\mathbf{F}^T\mathbf{F})]. \tag{4.47}$$

Para completar a análise da capacidade de rastreamento do algoritmo estudada nesta seção, falta ainda aplicar o operador do valor esperado à equação (4.42). A aplicação desse operador permite que se use as hipóteses **H1**, **H2** e **H5**. **H1** diz que  $\tilde{\beta}(k)$ ,  $\mathbf{x}(k)$  e  $\tilde{\mathbf{w}}(k)$  são mutuamente independentes. Já **H2** e **H5** permitem inferir que  $\mathbb{E}\{\text{vec}\{\mathcal{O}[\mathbf{q}(k), \nu(k)]\}\} = 0$ . A aplicação dessas hipóteses permite que a equação (4.42) seja reescrita da seguinte maneira:

$$\begin{aligned}
\mathbf{v}(k+1) = & \mathbf{\Theta}(k, \kappa, \beta)\mathbf{v}(k) + \beta^2\sigma^2\mathbb{E}[\mathbf{\mathcal{X}}(k)] + \text{vec}[\mathbf{Q}] \\
& + \kappa\mathbf{\Lambda}_1(k)\text{vec}\{\mathbb{E}[\tilde{\mathbf{w}}(k)(\mathbf{w}^*(k))^T]\} \\
& + \kappa\mathbf{\Lambda}_2(k)\text{vec}\{\mathbb{E}[\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)]\} \\
& + \kappa^2\mathbf{\Psi}\text{vec}\{\mathbb{E}[\mathbf{w}^*(k)(\mathbf{w}^*(k))^T]\},
\end{aligned} \tag{4.48}$$



onde  $\text{vec}[\mathbf{q}(k)\mathbf{q}^T(k)]$  é a matriz de autocorrelação da variável aleatória  $q(k)$ ,  $\mathbf{v}(k) = \text{vec}\{\mathbb{E}[\mathbf{\Gamma}(k)]\}$ ,  $\text{vec}\{\mathbb{E}[\tilde{\mathbf{w}}(k)(\mathbf{w}^*(k))^T]\}$  e  $\text{vec}\{\mathbb{E}[\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)]\}$  são as matrizes de correlação cruzada entre a planta a ser identificada e o vetor de desvios dos coeficientes (note que tais matrizes são iguais, uma vez que são matrizes simétricas e  $\tilde{\mathbf{w}}(k)(\mathbf{w}^*(k))^T = [\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)]^T$ ) e  $\text{vec}\{\mathbb{E}[\mathbf{w}^*(k)(\mathbf{w}^*(k))^T]\}$  é a autocorrelação da planta a ser identificada. Vale notar que, assim como a matriz de autocorrelação dos desvios, tais matrizes podem ser calculadas recursivamente. Nesse caso, para o cálculo da correlação cruzada, é necessário aplicar o operador de valor esperado na equação resultante do produto das equações (4.6) e (4.35). Obtendo-se então:

$$\begin{aligned} \mathbb{E}[\mathbf{w}^*(k+1)\tilde{\mathbf{w}}^T(k+1)] &= [\mathbf{I} - \beta\mathbf{R}_x - \kappa\mathbf{F}^T\mathbf{F}] \mathbb{E}[\mathbf{w}^*(k)\tilde{\mathbf{w}}^T(k)] \\ &+ \kappa\mathbf{F}^T\mathbf{F}\mathbb{E}[\mathbf{w}^*(k)(\mathbf{w}^*(k))^T]. \end{aligned} \quad (4.49)$$

Já o cálculo da matriz de autocorrelação do filtro ótimo, é facilmente obtido a partir da equação (4.35):

$$\mathbb{E}[\mathbf{w}^*(k+1)(\mathbf{w}^*(k+1))^T] = \mathbb{E}[\mathbf{w}^*(k)(\mathbf{w}^*(k))^T] + \mathbf{Q}; \quad (4.50)$$

## 4.7 Resultados

Nesta seção, são feitas simulações acerca da análise proposta neste capítulo, bem como uma comparação com o algoritmo LMS. Tais simulações consistem em identificar dois sistemas, como na configuração proposta neste trabalho, cujas funções de transferência foram definidas na seção de resultados do capítulo 3. O sinal de entrada é obtido filtrando-se um sinal branco, gaussiano, e de variância unitária pela função de transferência  $H(z) = 1 + 0.8z^{-1} - 0.2z^{-2}$ . As curvas de aprendizado são obtidas fazendo a média dos resultados provenientes de 1000 simulações independentes de Monte Carlo.

A Figura 4.1 mostra os resultados obtidos da análise de primeira ordem, sendo possível observar a evolução dos coeficientes na média. Nessas simulações, foram utilizados os parâmetros  $\beta = 0.01$ ,  $\sigma_v^2 = 0.1$  em ambos os casos. A Figura 4.1(a) mostra o resultado obtido utilizando o filtro ótimo  $w_\ell^*$  (3.14) e  $\kappa = 0.0316$ , enquanto que a Figura 4.1(b) mostra o resultado obtido utilizando o filtro ótimo  $w_h^*$  (3.15) e  $\kappa = 0.0032$ . Com os resultados apresentados, pode-se concluir que a análise de primeira ordem possui boa fidelidade com os resultados experimentais.

A Figura 4.2 mostra os resultados obtidos pelo modelo estocástico proposto para análise em regime permanente, sendo possível observar o MSD em regime permanente para diversos valores de  $\kappa$ . Nessas simulações, foram utilizados os parâmetros  $\beta = 0.015$ ,  $\sigma_v^2 = 0.1$  em ambos os casos. A Figura 4.2(a) mostra o resultado obtido utilizando o filtro ótimo  $w_\ell^*$  (3.14), enquanto que a Figura 4.2(b) mostra o resultado

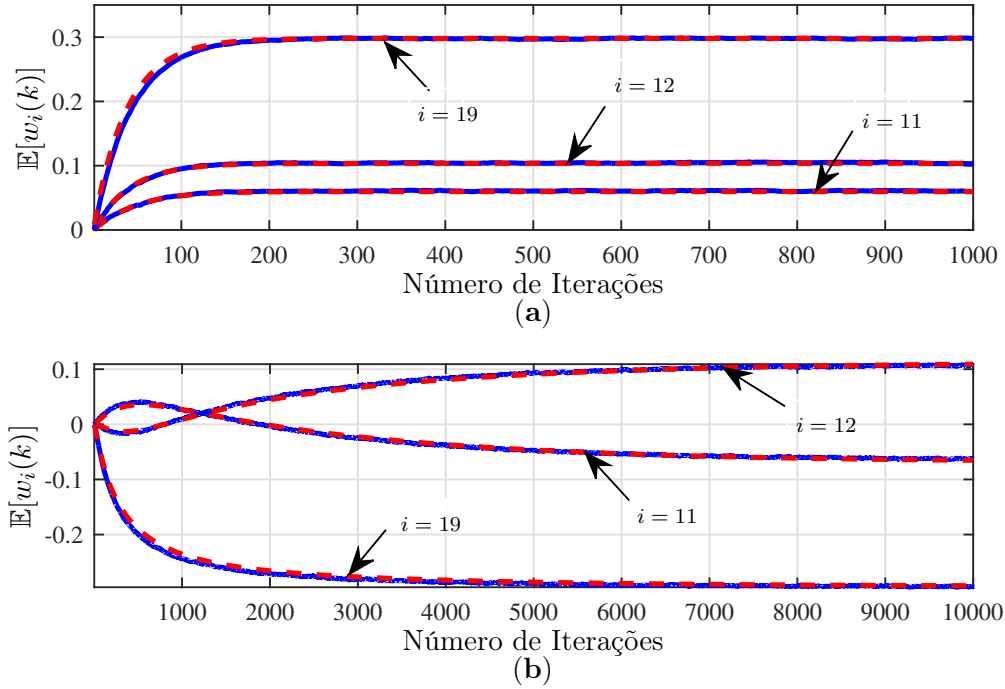


Figura 4.1: Comparação da evolução de alguns coeficientes  $w_i(k)$ , na média, entre simulado (em azul) e teórico (em vermelho), com  $i \in \{11, 12, 19\}$ . (a) Utilizando  $w_l^*$ . (b) Utilizando  $w_h^*$ .

obtido utilizando o filtro ótimo  $w_h^*$  (3.15). Pelos resultados apresentados na Figura 4.2 é possível concluir que o modelo teórico é capaz de prever o comportamento do algoritmo, em regime permanente, na prática, bem como é capaz de prever o valor de  $\kappa$  que otimiza a performance em regime permanente.

A Figura 4.3 mostra a evolução do MSD ao longo das iterações, permitindo avaliar a capacidade de previsão do modelo teórico para a métrica em questão. Nessas simulações, foram utilizados os parâmetros  $\beta = 0.015$ ,  $\sigma_\nu^2 = 0.1$  em ambos os casos. A Figura 4.3(a) mostra o desempenho obtido utilizando o filtro ótimo  $w_\ell^*$  e  $\kappa = 0.0316$ , enquanto que a Figura 4.3(b) mostra o resultado obtido utilizando o filtro ótimo  $w_h^*$  e  $\kappa = 0.0032$ . Vale notar que o modelo estocástico é capaz de prever o comportamento do algoritmo na prática, principalmente em regime permanente. A diferença nos valores do MSD entre o modelo teórico e o resultado experimental observada nas simulações acontece devido ao modelo teórico assumir a hipótese de independência entre o sinal de entrada e os coeficientes do filtro adaptativo. Tal diferença está fortemente relacionada com os valores de  $\beta$  e  $\kappa$ . Porém, note que isso só acontece porque o sinal de entrada não é branco.

A Figura 4.4 mostra os resultados obtidos pelas simulações utilizando os filtros ótimos variando no tempo e observando o MSE em regime permanente. Nessas simulações, foram utilizados os parâmetros  $\sigma_\nu^2 = 0.01$  e  $\sigma_q^2 = 10^{-6}$  e  $\kappa = 0.0001$  em

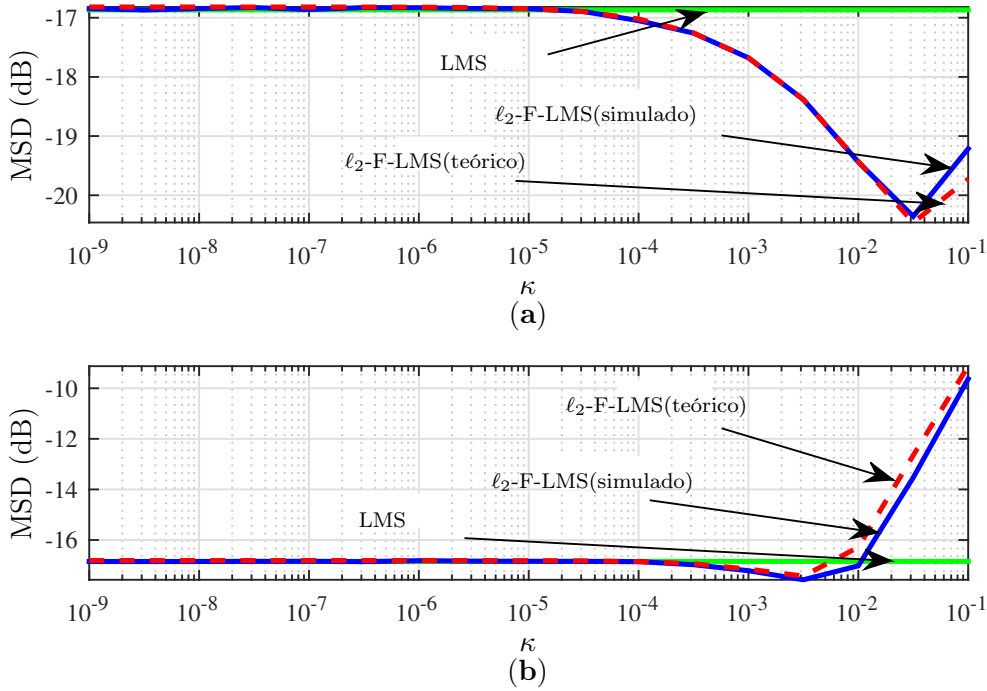


Figura 4.2: Comparação do MSD, em regime permanente, simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS. A performance, na média, do algoritmo LMS é mostrada em verde. (a) Utilizando  $w_l^*$ . (b) Utilizando  $w_h^*$ .

ambos os casos. A Figura 4.4(a) mostra o resultado obtido utilizando o filtro ótimo  $w_l^*$ , enquanto que a Figura 4.4(b) mostra o resultado obtido utilizando o filtro ótimo  $w_h^*$ . Pode-se concluir que o modelo teórico é capaz de prever o MSE em regime permanente adequadamente, bem como o valor de  $\beta$  ótimo. Uma vez que valores pequenos de  $\beta$  fazem com que o filtro não consiga acompanhar adequadamente as variações da planta, enquanto que valores altos de  $\beta$  prejudicam a estimativa, uma vez que aumentam a variância do estimador.

A Figura 4.5 mostra os resultados obtidos pelas simulações no caso em que o erro de medição não é estacionário ( $\sigma_v^2$  não é constante). Nessas simulações, é observado a evolução do MSD ao longo das iterações utilizando  $\beta = 0.015$ . A Figura 4.5(a) mostra o resultado obtido utilizando o filtro ótimo  $w_l^*$  e  $\kappa = 0.0316$ , enquanto que a Figura 4.5(b) mostra o resultado obtido utilizando o filtro ótimo  $w_h^*$  e  $\kappa = 0.0032$ . Pelos resultados observados, é possível concluir que o modelo teórico é capaz de prever o comportamento experimental com fidelidade.

## 4.8 Comentários Finais

Neste capítulo foi proposto um modelo estocástico com o objetivo de fornecer informações sobre o comportamento do algoritmo proposto neste trabalho. Também fo-

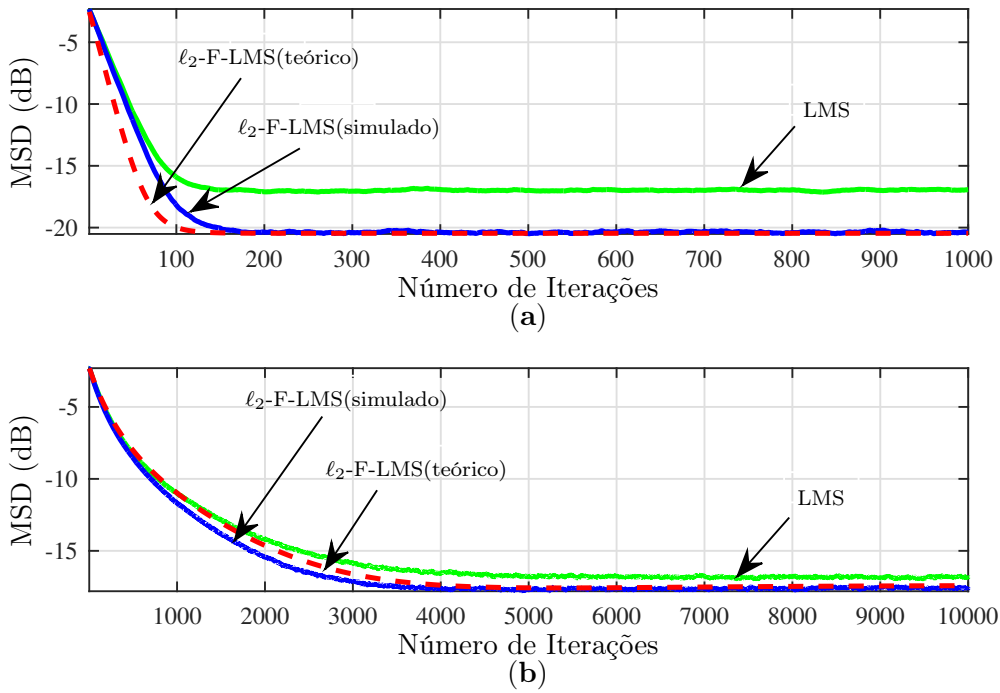


Figura 4.3: Comparação da evolução do MSD simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS. A performance, na média, do algoritmo LMS é mostrada em verde. (a) Utilizando  $w_h^*$ . (b) Utilizando  $w_h^*$ .

ram feitos experimentos para comparar a performance do algoritmo  $\ell_2$ -norm-F-LMS com a performance do algoritmo LMS, bem como a fidelidade do modelo teórico proposto. As simulações apresentadas na seção de resultados permitem concluir que o algoritmo proposto supera o desempenho do algoritmo LMS, que é agnóstico à esparsidade (ou à energia de vetores obtidos a partir de transformações lineares dos coeficientes adaptativos), e, também, permite concluir que o modelo teórico proposto é capaz de fornecer informações adequadas acerca do comportamento de primeira ordem, quadrático médio e de reastreamento do algoritmo. Ademais, tal modelo é capaz de prever as curvas de aprendizado com fidelidade adequada.

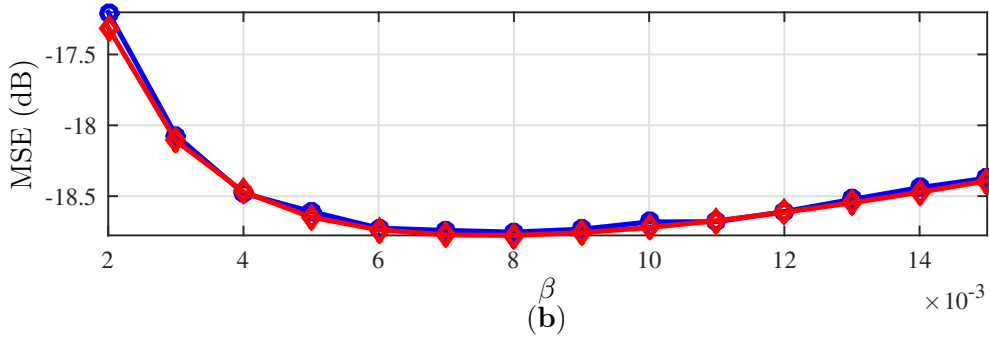
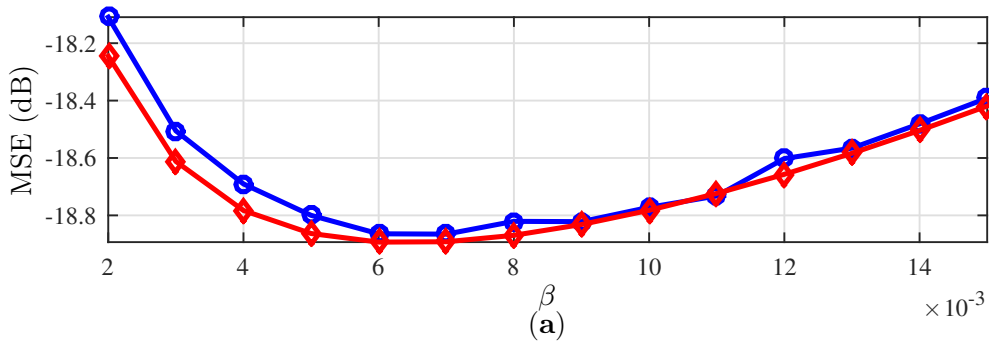


Figura 4.4: Comparação do MSE, em regime permanente, simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS, com planta variante no tempo. (a) Utilizando  $\mathbf{w}_l^*$ . (b) Utilizando  $\mathbf{w}_h^*$ .

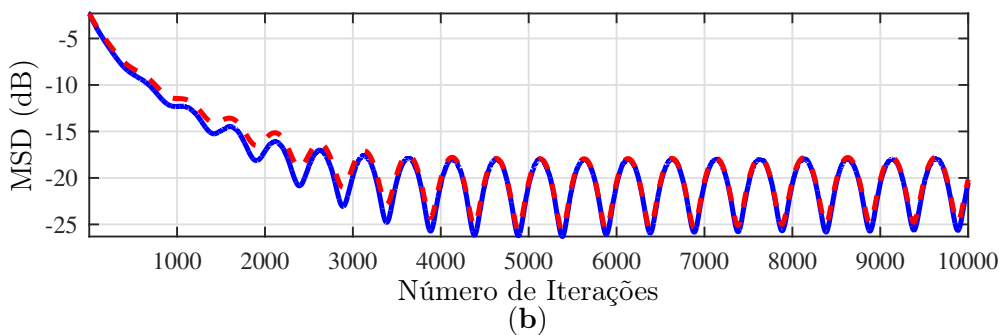
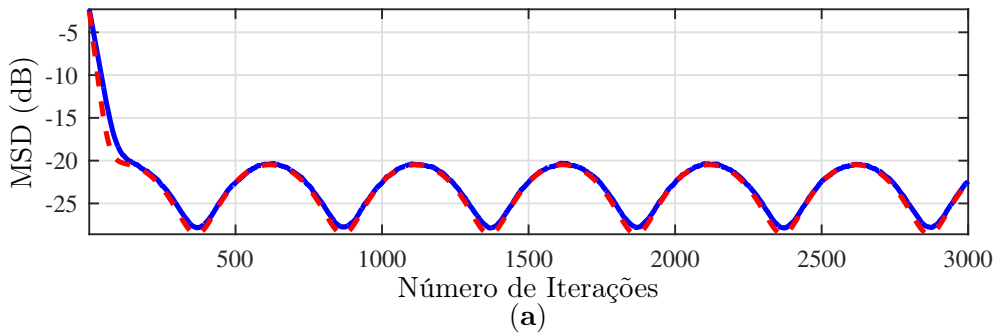


Figura 4.5: Comparação da evolução do MSD simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS, com ruído aditivo não-estacionário. (a) Utilizando  $\mathbf{w}_l^*$ . (b) Utilizando  $\mathbf{w}_h^*$ .

# Capítulo 5

## $\ell_2$ -norm Feature LMS Submodelado

### 5.1 Introdução

No capítulo 4, foi feita uma análise de transiente capaz de estimar o comportamento do algoritmo  $\ell_2$ -norm *Feature* LMS. Nessa análise, foi considerado que o filtro adaptativo possui o mesmo número de coeficientes da resposta ao impulso do sistema a ser identificado. Note que essa é uma suposição comum em diversas análises presentes na literatura, seja para algoritmos da família LMS agnósticos de esparsidade [52–58], seja para derivações desses algoritmos conscientes de esparsidade [1, 3, 7, 36], onde para esse caso, essas análises foram capazes de estimar o comportamento desses algoritmos [59]. Contudo, em situações práticas, é comum o projetista se deparar com situações em que o tamanho da resposta ao impulso do sistema a ser identificado é desconhecido, e em alguns casos muito grande [60], ou em que se tenha que lidar com limitações computacionais [61], resultando no uso de um filtro adaptativo com um número de coeficientes menor do que o sistema desconhecido.

Os resultados obtidos em análises que consideram que o filtro adaptativo possui tamanho suficientemente grande não necessariamente refletem o comportamento do filtro quando o mesmo possui uma limitação quanto ao número de coeficientes, fazendo, então, com que seja de grande importância uma análise estatística do comportamento dos filtros adaptativos nessas situações em que não é possível usar, por razões práticas, um número de coeficientes adequado [59]. Apesar da importância de tal análise para esses problemas reais, não é comum encontrar análises que levem em consideração um filtro adaptativo de tamanho menor que o sistema a ser identificado [59, 60, 62–65], sendo, inclusive, a primeira vez em que é feita uma análise de regimes transiente e permanente capaz de levar em consideração que o filtro adaptativo pode ter tamanho menor que a resposta ao impulso do sistema desconhecido para um algoritmo da família *Feature* LMS.

Neste capítulo, é feita uma análise de transiente para o algoritmo  $\ell_2$ -norm *Feature*

LMS, similar àquela feita no capítulo 4, porém, considerando que o filtro adaptativo é submodelado, ou seja, possui um número menor de coeficientes do que o sistema a ser identificado.

## 5.2 Considerações Iniciais

No capítulo 2, foram definidos os vetores  $\mathbf{x}(k)$ ,  $\mathbf{w}(k)$  e  $\mathbf{w}^*(k)$ , contendo os valores do sinal de entrada, os valores do filtro adaptativo e os valores do filtro a ser identificado, respectivamente. Porém, em um cenário onde não é possível usar um filtro adaptativo com o mesmo número de coeficientes que o filtro cuja resposta ao impulso deseja-se estimar, fazem-se necessárias as seguintes definições:

$$\bar{\mathbf{x}}(k) \triangleq \begin{bmatrix} x(k-N) & x(k-N-1) & \dots & x(k-N-P+1) \end{bmatrix}^T, \quad (5.1)$$

$$\bar{\mathbf{w}}^* \triangleq \begin{bmatrix} w_N^* & w_{N+1}^* & \dots & w_{N+P-1}^* \end{bmatrix}^T, \quad (5.2)$$

onde os vetores  $\bar{\mathbf{x}}(k)$ ,  $\bar{\mathbf{w}}^* \in \mathbb{R}^P$  representam a parte do sistema que o filtro adaptativo não é capaz de modelar. Note que essas definições acarretam em mudanças no cálculo do sinal de referência, que passa a ser escrito como:

$$d(k) = \mathbf{x}^T(k)\mathbf{w}^* + \bar{\mathbf{x}}^T(k)\bar{\mathbf{w}}^* + \nu(k) \quad (5.3)$$

onde  $\nu(k)$  representa um ruído aditivo com as mesmas características definidas no capítulo 2. Conseqüentemente, essa mudança no cálculo do sinal de referência acarreta em uma mudança no cálculo do erro, que passa a ser calculado da seguinte maneira:

$$e(k) = d(k) - y(k) = \mathbf{x}^T(k)\mathbf{w}^* + \bar{\mathbf{x}}^T(k)\bar{\mathbf{w}}^* + \nu(k) - \mathbf{x}^T(k)\mathbf{w}(k). \quad (5.4)$$

## 5.3 Análise de Primeira Ordem

A fim de realizar uma análise de primeira ordem e, com isso, obter informações sobre a convergência do algoritmo, serão derivadas equações recursivas capazes de estimar a evolução dos desvios, na média. Como já foi realizada o mesmo tipo de análise no capítulo 4, pode-se utilizar os resultados parciais obtidos de forma que ao substituir a equação (5.4) na equação (4.4), a seguinte equação seja obtida:

$$\begin{aligned} \tilde{\mathbf{w}}(k+1) = & \left[ \mathbf{I} - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\mathbf{F}^T\mathbf{F} \right] \tilde{\mathbf{w}}(k) + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^* \\ & - \tilde{\beta}(k)\mathbf{x}(k)\bar{\mathbf{x}}^T(k)\bar{\mathbf{w}}^* - \tilde{\beta}(k)\mathbf{x}(k)\nu(k) \end{aligned} \quad (5.5)$$

Para entender o comportamento do algoritmo na média, e consequentemente obter uma análise estocástica de primeira ordem, é necessário aplicar o operador do valor esperado à equação (5.5). Obtendo-se então:

$$\begin{aligned} \mathbb{E} [\tilde{\mathbf{w}}(k+1)] = & \mathbb{E} \left\{ \left[ \mathbf{I} - \tilde{\beta}(k) \mathbf{x}(k) \mathbf{x}^T(k) - \kappa \mathbf{F}^T \mathbf{F} \right] \tilde{\mathbf{w}}(k) \right\} \\ & + \mathbb{E} [\kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^*] - \mathbb{E} \left[ \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \right] \\ & - \mathbb{E} \left[ \tilde{\beta}(k) \mathbf{x}(k) \nu(k) \right] \end{aligned} \quad (5.6)$$

Aplicando as hipóteses H1 e H2, definidas no capítulo 4, é possível reescrever a equação (5.6) da seguinte maneira:

$$\begin{aligned} \mathbb{E} [\tilde{\mathbf{w}}(k+1)] = & \mathbb{E} [\tilde{\mathbf{w}}(k)] - \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_x \mathbb{E} [\tilde{\mathbf{w}}(k)] - \kappa \mathbf{F}^T \mathbf{F} \mathbb{E} [\tilde{\mathbf{w}}(k)] \\ & - \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_{x\bar{x}} \bar{\mathbf{w}}^* + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^*, \end{aligned} \quad (5.7)$$

onde  $\mathbf{R}_x \triangleq \mathbb{E} [\mathbf{x}(k) \mathbf{x}^T(k)]$  e  $\mathbf{R}_{x\bar{x}} \triangleq \mathbb{E} [\mathbf{x}(k) \bar{\mathbf{x}}^T(k)]$  são as matrizes de autocorrelação e correlação cruzada do sinal de entrada, respectivamente.

É possível observar que a equação (5.7) é igual à equação (4.9) mais o termo  $-\mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_x \bar{\mathbf{w}}^*$ , e assim como na análise do capítulo 4, o algoritmo apresenta um viés, o qual pode ser calculado aplicando o operador de limite, com  $k \rightarrow \infty$ , à equação (5.7) e obtendo:

$$\begin{aligned} \mathbf{0} = & \lim_{k \rightarrow \infty} \left\{ \mathbb{E} [\tilde{\mathbf{w}}(k)] - \mathbb{E} [\tilde{\mathbf{w}}(k+1)] \right\} \\ & - \lim_{k \rightarrow \infty} \left\{ \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_x \mathbb{E} [\tilde{\mathbf{w}}(k)] + \kappa \mathbf{F}^T \mathbf{F} \mathbb{E} [\tilde{\mathbf{w}}(k)] \right\} \\ & + \lim_{k \rightarrow \infty} \left\{ \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* - \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_{x\bar{x}} \bar{\mathbf{w}}^* \right\}, \end{aligned} \quad (5.8)$$

considerando que o algoritmo convirja quando  $k \rightarrow \infty$ , pode-se considerar que  $\mathbb{E} [\tilde{\mathbf{w}}(k+1)] \approx \mathbb{E} [\tilde{\mathbf{w}}(k)]$ . Logo, a equação (2.7) pode ser reescrita como:

$$\lim_{k \rightarrow \infty} \left\{ \mathbb{E} [\tilde{\mathbf{w}}(k)] \right\} = \left\{ \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_x + \kappa \mathbf{F}^T \mathbf{F} \right\}^{-1} \left\{ \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* - \mathbb{E} \left[ \tilde{\beta}(k) \right] \mathbf{R}_{x\bar{x}} \bar{\mathbf{w}}^* \right\} \quad (5.9)$$

Vale a pena notar que se o sinal de entrada for descorrelacionado,  $\mathbf{R}_{x\bar{x}} = 0$ , o filtro adaptativo, de tamanho reduzido, tende para o mesmo viés encontrado no capítulo 4. Já quando o sinal é colorido, aparece, somado a esse mesmo viés, um termo de perturbação proveniente da parte do sistema que o filtro adaptativo não é capaz de modelar.



## 5.4 Análise de Segunda Ordem

Apesar da análise de primeira ordem fornecer informações sobre o comportamento do algoritmo em regime permanente, faz-se necessária uma análise de segunda ordem de forma a se obter informações sobre a variância do vetor de pesos, uma vez que mesmo que se prove um desvio quadrático médio finito, o mesmo pode, na prática, divergir devido à variância infinita [53, 66].

Assim como na análise de primeira ordem, é possível partir dos resultados parciais obtidos no capítulo 4. Note que a equação encontrada em (5.5) difere da equação (4.6) apenas pelo termo  $-\tilde{\beta}(k)\mathbf{x}(k)\bar{\mathbf{x}}^T(k)\bar{\mathbf{w}}^*$ . Logo, pode-se utilizar a equação (4.13) e apenas “atualizar” seus termos. De forma que se obtenha:

$$\mathbf{\Gamma}(k+1) = \tilde{\mathcal{W}}[\mathbf{\Gamma}(k)] + \mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \bar{\mathbf{w}}^*, \nu^2] + \mathcal{O}[\nu(k)], \quad (5.10)$$

onde  $\tilde{\mathcal{W}}[\mathbf{\Gamma}(k)]$  contém todos os termos com  $\mathbf{\Gamma}(k)$ ,  $\mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \bar{\mathbf{w}}^*, \nu^2]$  contém todos os termos com  $\tilde{\mathbf{w}}, \mathbf{w}^*, \bar{\mathbf{w}}^*$  e  $\nu^2$  que não dependem de termos de ruído de primeira ordem, pois estes estão contidos na matriz  $\mathcal{O}[\nu(k)]$ . Dessa forma obtém-se as seguintes equações:

$$\begin{aligned} \tilde{\mathcal{W}}[\mathbf{\Gamma}(k)] = & \tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) - \tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) - \kappa\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k) \\ & - \tilde{\beta}(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) - \kappa\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\ & + \kappa\tilde{\beta}(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \\ & + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ & + \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ & + \kappa^2\mathbf{F}^T\mathbf{F}\tilde{\mathbf{w}}(k)\tilde{\mathbf{w}}^T(k)\mathbf{F}^T\mathbf{F} \end{aligned} \quad (5.11)$$

$$\begin{aligned}
\mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \bar{\mathbf{w}}^*, \nu^2] = & \kappa \tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} - \kappa \tilde{\beta}(k) \mathbf{x}(k) \mathbf{x}^T(k) \tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} \\
& - \kappa^2 \mathbf{F}^T \mathbf{F} \tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} \\
& + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* \tilde{\mathbf{w}}^T(k) - \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \mathbf{w}^* \tilde{\mathbf{w}}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \\
& - \kappa^2 \mathbf{F}^T \mathbf{F} \mathbf{w}^* \tilde{\mathbf{w}}^T(k) \mathbf{F}^T \mathbf{F} \\
& - \tilde{\beta}(k) \tilde{\mathbf{w}}(k)(\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \mathbf{x}^T(k) \tilde{\mathbf{w}}(k)(\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& + \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \tilde{\mathbf{w}}(k)(\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& - \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \tilde{\mathbf{w}}^T(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \tilde{\mathbf{w}}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \\
& + \kappa \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \tilde{\mathbf{w}}^T(k) \mathbf{F}^T \mathbf{F} \\
& - \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \mathbf{w}^* (\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& - \kappa \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* (\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* (\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& + \kappa^2 \mathbf{F}^T \mathbf{F} \mathbf{w}^* (\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} + \tilde{\beta}^2(k) \mathbf{x}(k) \mathbf{x}^T(k) \nu^2(k)
\end{aligned} \tag{5.12}$$

$$\begin{aligned}
\mathcal{O}[\nu(k)] = & \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \tilde{\mathbf{w}}(k) \mathbf{x}^T(k) \nu(k) - \tilde{\beta}(k) \tilde{\mathbf{w}}(k) \mathbf{x}^T(k) \nu(k) \\
& - \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \mathbf{w}^* \mathbf{x}^T(k) \nu(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \mathbf{x}^T(k) \tilde{\mathbf{w}}(k) \mathbf{x}^T(k) \nu(k) \\
& - \tilde{\beta}(k) \mathbf{x}(k) \tilde{\mathbf{w}}^T(k) \nu(k) \\
& - \kappa \tilde{\beta}(k) \mathbf{x}(k) (\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} \nu(k) \\
& + \kappa \tilde{\beta}(k) \mathbf{x}(k) \tilde{\mathbf{w}}^T(k) \mathbf{F}^T \mathbf{F} \nu(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \tilde{\mathbf{w}}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \nu(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \mathbf{x}(k) \nu(k) \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) (\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}^T(k) \mathbf{x}^T(k) \nu(k)
\end{aligned} \tag{5.13}$$

Nas equações (5.11), (5.12) e (5.13), é possível observar que a equação (5.11) é igual à encontrada no capítulo 4 e que, apesar dos termos adicionados nas outras duas equações, muitos termos são comuns aos que já existiam. Sendo possível escrevê-las da seguinte forma:

$$\tilde{\mathcal{W}}[\Gamma(k)] = \mathbf{A}(k, \beta, \kappa) \Gamma(k) \mathbf{A}(k, \beta, \kappa), \tag{5.14}$$

e

$$\begin{aligned}
\mathcal{V}[\tilde{\mathbf{w}}, \mathbf{w}^*, \bar{\mathbf{w}}^*, \nu^2] = & \kappa \mathbf{A}(k, \beta, \kappa) \tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^* \tilde{\mathbf{w}}^T(k) \mathbf{A}(k, \beta, \kappa) \\
& - \tilde{\beta}(k) \mathbf{A}(k, \beta, \kappa) \tilde{\mathbf{w}}(k)(\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& - \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* \tilde{\mathbf{w}}^T(k) \mathbf{A}(k, \beta, \kappa) \\
& - \kappa \tilde{\beta}(k) \mathbf{F}^T \mathbf{F} \mathbf{w}^* (\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& - \kappa \tilde{\beta}(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* (\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} \\
& + \tilde{\beta}^2(k) \mathbf{x}(k) \bar{\mathbf{x}}^T(k) \bar{\mathbf{w}}^* (\bar{\mathbf{w}}^*)^T \bar{\mathbf{x}}(k) \mathbf{x}^T(k) \\
& + \kappa^2 \mathbf{F}^T \mathbf{F} \mathbf{w}^* (\mathbf{w}^*)^T \mathbf{F}^T \mathbf{F} + \tilde{\beta}^2(k) \mathbf{x}(k) \mathbf{x}^T(k) \nu^2(k)
\end{aligned} \tag{5.15}$$

Cabe ainda substituir as equações (5.14) e (5.15) na equação (5.10), de forma que a equação de atualização da matriz de desvios possa ser reescrita da seguinte forma:

$$\begin{aligned}
\Gamma(k+1) = & \mathbf{A}(k, \beta, \kappa)\Gamma(k)\mathbf{A}(k, \beta, \kappa) + \mathcal{O}[\nu(k)] \\
& + \kappa\mathbf{A}(k, \beta, \kappa)\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \kappa\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{A}(k, \beta, \kappa) \\
& - \tilde{\beta}(k)\mathbf{A}(k, \beta, \kappa)\tilde{\mathbf{w}}(k)(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k) \\
& - \tilde{\beta}(k)\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*\tilde{\mathbf{w}}^T(k)\mathbf{A}(k, \beta, \kappa) \\
& - \kappa\tilde{\beta}(k)\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k) \\
& - \kappa\tilde{\beta}(k)\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} \\
& + \tilde{\beta}^2(k)\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k) \\
& + \kappa^2\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F} + \tilde{\beta}^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k)
\end{aligned} \tag{5.16}$$

Uma vez mais, as manipulações algébricas realizadas não são capazes de fazer com que a equação (5.16) seja uma equação recursiva característica. Fazendo-se, então, necessária a aplicação do operador  $\text{vec}(\mathbf{X})$ , e obtendo como resultado:

$$\begin{aligned}
\text{vec}[\Gamma(k+1)] = & \text{vec}[\mathbf{A}(k, \beta, \kappa)\Gamma(k)\mathbf{A}(k, \beta, \kappa)] + \text{vec}\{\mathcal{O}[\nu(k)]\} \\
& + \kappa\text{vec}[\mathbf{A}(k, \beta, \kappa)\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}] \\
& + \kappa\text{vec}[\mathbf{F}^T\mathbf{F}\mathbf{w}^*\tilde{\mathbf{w}}^T(k)\mathbf{A}(k, \beta, \kappa)] \\
& - \tilde{\beta}(k)\text{vec}[\mathbf{A}(k, \beta, \kappa)\tilde{\mathbf{w}}(k)(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k)] \\
& - \tilde{\beta}(k)\text{vec}[\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*\tilde{\mathbf{w}}^T(k)\mathbf{A}(k, \beta, \kappa)] \\
& - \kappa\tilde{\beta}(k)\text{vec}[\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k)] \\
& - \kappa\tilde{\beta}(k)\text{vec}[\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}] \\
& + \tilde{\beta}^2(k)\text{vec}[\mathbf{x}(k)\overline{\mathbf{x}}^T(k)\overline{\mathbf{w}}^*(\overline{\mathbf{w}}^*)^T\overline{\mathbf{x}}(k)\mathbf{x}^T(k)] \\
& + \kappa^2\text{vec}[\mathbf{F}^T\mathbf{F}\mathbf{w}^*(\mathbf{w}^*)^T\mathbf{F}^T\mathbf{F}] \\
& + \tilde{\beta}^2(k)\text{vec}[\mathbf{x}(k)\mathbf{x}^T(k)\nu^2(k)]
\end{aligned} \tag{5.17}$$

Cabe ainda aplicar a propriedade do operador mostrada em (4.22), permitindo reescrever a equação (5.17) da seguinte forma:

$$\begin{aligned}
\text{vec}[\Gamma(k+1)] = & \Theta(k, \kappa, \beta)\text{vec}[\Gamma(k)] + \text{vec}\{\mathcal{O}[\nu(k)]\} \\
& + \kappa\Lambda_1(k)\text{vec}[\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T] \\
& + \kappa\Lambda_2(k)\text{vec}[(\mathbf{w}^*)\tilde{\mathbf{w}}^T(k)] \\
& - \tilde{\beta}(k)\Lambda_3(k)\text{vec}[\tilde{\mathbf{w}}(k)(\overline{\mathbf{w}}^*)^T] \\
& - \tilde{\beta}(k)\Lambda_4(k)\text{vec}[\overline{\mathbf{w}}^*\tilde{\mathbf{w}}^T(k)] \\
& - \kappa\tilde{\beta}(k)\Lambda_5(k)\text{vec}[\mathbf{w}^*(\overline{\mathbf{w}}^*)^T] \\
& - \kappa\tilde{\beta}(k)\Lambda_6(k)\text{vec}[\overline{\mathbf{w}}^*(\mathbf{w}^*)^T] \\
& + \tilde{\beta}^2(k)\Lambda_7(k)\text{vec}[\overline{\mathbf{w}}^*(\overline{\mathbf{w}}^*)^T] \\
& + \tilde{\beta}^2(k)\nu^2(k)\mathcal{X}(k) + \kappa^2\Psi,
\end{aligned} \tag{5.18}$$

onde

$$\Theta(k, \kappa, \beta) \triangleq \mathbf{A}(k, \kappa, \beta) \otimes \mathbf{A}(k, \kappa, \beta), \quad (5.19)$$

$$\Lambda_1(k) \triangleq [\mathbf{F}^T \mathbf{F}] \otimes \mathbf{A}(k, \kappa, \beta), \quad (5.20)$$

$$\Lambda_2(k) \triangleq \mathbf{A}(k, \kappa, \beta) \otimes [\mathbf{F}^T \mathbf{F}], \quad (5.21)$$

$$\Lambda_3(k) \triangleq [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)] \otimes \mathbf{A}(k, \kappa, \beta), \quad (5.22)$$

$$\Lambda_4(k) \triangleq \mathbf{A}(k, \kappa, \beta) \otimes [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)], \quad (5.23)$$

$$\Lambda_5(k) \triangleq [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)] \otimes [\mathbf{F}^T \mathbf{F}], \quad (5.24)$$

$$\Lambda_6(k) \triangleq [\mathbf{F}^T \mathbf{F}] \otimes [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)], \quad (5.25)$$

$$\Lambda_7(k) \triangleq [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)] \otimes [\mathbf{x}(k)\bar{\mathbf{x}}^T(k)], \quad (5.26)$$

$$\mathcal{X}(k) \triangleq \text{vec} [\mathbf{x}(k)\mathbf{x}^T(k)], \quad (5.27)$$

$$\Psi \triangleq [(\mathbf{F}^T \mathbf{F}) \otimes (\mathbf{F}^T \mathbf{F})] \text{vec} \left\{ \mathbf{w}^* [\mathbf{w}^*]^T \right\}. \quad (5.28)$$

Uma vez encontrada a devida equação recursiva para atualização da matriz de desvios, é necessário aplicar o operador do valor esperado à equação (5.18) para que seja possível dar prosseguimento à análise estatística de segunda ordem, objetivo dessa seção. Obtendo-se assim:

$$\begin{aligned} \mathbb{E} \{ \text{vec} [\Gamma(k+1)] \} &= \mathbb{E} \{ \Theta(k, \kappa, \beta) \text{vec} [\Gamma(k)] \} + \mathbb{E} \{ \text{vec} \{ \mathcal{O} [\nu(k)] \} \} \\ &+ \kappa \mathbb{E} \left\{ \Lambda_1(k) \text{vec} [\tilde{\mathbf{w}}(k)(\mathbf{w}^*)^T] \right\} \\ &+ \kappa \mathbb{E} \left\{ \Lambda_2(k) \text{vec} [\mathbf{w}^* \tilde{\mathbf{w}}^T(k)] \right\} \\ &- \mathbb{E} \left\{ \tilde{\beta}(k) \Lambda_3(k) \text{vec} [\tilde{\mathbf{w}}(k)(\bar{\mathbf{w}}^*)^T] \right\} \\ &- \mathbb{E} \left\{ \tilde{\beta}(k) \Lambda_4(k) \text{vec} [\bar{\mathbf{w}}^* \tilde{\mathbf{w}}^T(k)] \right\} \\ &- \kappa \mathbb{E} \left\{ \tilde{\beta}(k) \Lambda_5(k) \text{vec} [\mathbf{w}^* (\bar{\mathbf{w}}^*)^T] \right\} \\ &- \kappa \mathbb{E} \left\{ \tilde{\beta}(k) \Lambda_6(k) \text{vec} [\bar{\mathbf{w}}^* (\mathbf{w}^*)^T] \right\} \\ &+ \mathbb{E} \left\{ \tilde{\beta}^2(k) \Lambda_7(k) \text{vec} [\bar{\mathbf{w}}^* (\bar{\mathbf{w}}^*)^T] \right\} \\ &+ \mathbb{E} \left\{ \tilde{\beta}^2(k) \nu^2(k) \mathcal{X}(k) \right\} + \kappa^2 \mathbb{E} \{ \Psi \} \end{aligned} \quad (5.29)$$

Cabe agora aplicar as hipóteses H1 e H2. H1 diz que  $\tilde{\beta}(k)$ ,  $\mathbf{x}(k)$  e  $\tilde{\mathbf{w}}(k)$  são mutuamente independentes. Já H2, permite inferir que  $\mathbb{E} \{ \text{vec} \{ \mathcal{O} [\nu(k)] \} \} = 0$ , tornando claro o motivo de não serem necessárias nenhum tipo de manipulação algébrica para  $\mathcal{O} [\nu(k)]$ . A aplicação dessas hipóteses permite que a equação (5.29)

seja reescrita da seguinte maneira:

$$\begin{aligned}
\mathbf{v}(k+1) = & \mathbb{E}[\Theta(k, \kappa, \beta)] \mathbf{v}(k) + \sigma_v^2 \mathbb{E}[\tilde{\beta}^2(k)] \mathbb{E}[\mathcal{X}(k)] + \kappa^2 \Psi \\
& + \kappa \mathbb{E}[\Lambda_1(k)] \text{vec} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] (\mathbf{w}^*)^T \} \\
& + \kappa \mathbb{E}[\Lambda_2(k)] \text{vec} \{ \mathbf{w}^* \mathbb{E}[\tilde{\mathbf{w}}^T(k)] \}, \\
& - \mathbb{E}[\tilde{\beta}(k)] \mathbb{E}[\Lambda_3(k)] \text{vec} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] (\bar{\mathbf{w}}^*)^T \} \\
& - \mathbb{E}[\tilde{\beta}(k)] \mathbb{E}[\Lambda_4(k)] \text{vec} \{ \bar{\mathbf{w}}^* \mathbb{E}[\tilde{\mathbf{w}}^T(k)] \} \\
& - \kappa \mathbb{E}[\tilde{\beta}(k)] \mathbb{E}[\Lambda_5(k)] \text{vec} [\mathbf{w}^* (\bar{\mathbf{w}}^*)^T] \\
& - \kappa \mathbb{E}[\tilde{\beta}(k)] \mathbb{E}[\Lambda_6(k)] \text{vec} [\bar{\mathbf{w}}^* (\mathbf{w}^*)^T] \\
& + \mathbb{E}[\tilde{\beta}^2(k)] \mathbb{E}[\Lambda_7(k)] \text{vec} [\bar{\mathbf{w}}^* (\bar{\mathbf{w}}^*)^T],
\end{aligned} \tag{5.30}$$

onde  $\mathbf{v}(k) \triangleq \mathbb{E} \{ \text{vec} [\Gamma(k)] \}$ . Note que como  $\text{vec}(\mathbf{X})$  é um operador bijetor, a matriz  $\mathbf{R}_{\tilde{\mathbf{w}}}(k)$ , onde  $\mathbf{R}_{\tilde{\mathbf{w}}}(k) \triangleq \mathbb{E} [\tilde{\mathbf{w}}(k) \tilde{\mathbf{w}}^T(k)]$ , pode ser facilmente recuperada a partir de  $\mathbf{v}(k)$ .

Vale notar que todo o desenvolvimento feito até agora é independente do algoritmo ser normalizado ou não. Então, cabe, por fim, chegar nas devidas equações de atualização da matriz de autocorrelação dos desvios. Para o caso não normalizado, tem-se que  $\tilde{\beta}(k) = \beta$ , logo a equação (4.30) pode ser escrita da seguinte maneira:

$$\begin{aligned}
\mathbf{v}(k+1) = & \mathbb{E}[\Theta(k, \kappa, \beta)] \mathbf{v}(k) + \sigma_v^2 \beta^2 \mathbb{E}[\mathcal{X}(k)] + \kappa^2 \Psi \\
& + \kappa \mathbb{E}[\Lambda_1(k)] \text{vec} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] (\mathbf{w}^*)^T \} \\
& + \kappa \mathbb{E}[\Lambda_2(k)] \text{vec} \{ \mathbf{w}^* \mathbb{E}[\tilde{\mathbf{w}}^T(k)] \}, \\
& - \beta \mathbb{E}[\Lambda_3(k)] \text{vec} \{ \mathbb{E}[\tilde{\mathbf{w}}(k)] (\bar{\mathbf{w}}^*)^T \} \\
& - \beta \mathbb{E}[\Lambda_4(k)] \text{vec} \{ \bar{\mathbf{w}}^* \mathbb{E}[\tilde{\mathbf{w}}^T(k)] \} \\
& - \kappa \beta \mathbb{E}[\Lambda_5(k)] \text{vec} [\mathbf{w}^* (\bar{\mathbf{w}}^*)^T] \\
& - \kappa \beta \mathbb{E}[\Lambda_6(k)] \text{vec} [\bar{\mathbf{w}}^* (\mathbf{w}^*)^T] \\
& + \beta^2 \mathbb{E}[\Lambda_7(k)] \text{vec} [\bar{\mathbf{w}}^* (\bar{\mathbf{w}}^*)^T],
\end{aligned} \tag{5.31}$$

Note também que a equação (5.31) depende do termo  $\mathbb{E}[\tilde{\mathbf{w}}(k)]$ , o qual pode ser encontrado pela equação (5.7), a qual por sua vez, para o algoritmo não normalizado, pode ser reescrita da seguinte forma:

$$\begin{aligned}
\mathbb{E}[\tilde{\mathbf{w}}(k+1)] = & \mathbb{E}[\tilde{\mathbf{w}}(k)] - \beta \mathbf{R}_{\mathbf{x}} \mathbb{E}[\tilde{\mathbf{w}}(k)] - \kappa \mathbf{F}^T \mathbf{F} \mathbb{E}[\tilde{\mathbf{w}}(k)] \\
& - \beta \mathbf{R}_{\mathbf{x}\bar{\mathbf{x}}} \bar{\mathbf{w}}^* + \kappa \mathbf{F}^T \mathbf{F} \mathbf{w}^*.
\end{aligned} \tag{5.32}$$

Com uma equação estatística acerca da evolução da matriz de autocorrelação

dos desvios, é possível estimar a evolução do MSE e do MSD ao longo do tempo:

$$\text{MSE}(k) = \sigma_\nu^2 + \text{Tr}[\mathbf{R}_x \mathbf{R}_{\tilde{\mathbf{w}}}(k)] + (\bar{\mathbf{w}}^*)^T \mathbf{R}_{\bar{\mathbf{x}}} \bar{\mathbf{w}}^* + 2\mathbb{E}[\tilde{\mathbf{w}}(k)] \mathbf{R}_{\bar{\mathbf{x}}} \bar{\mathbf{w}}^* \quad (5.33)$$

$$\text{MSD}(k) = \text{Tr}[\mathbf{R}_{\tilde{\mathbf{w}}}(k)], \quad (5.34)$$

onde  $\mathbf{R}_{\bar{\mathbf{x}}} = \mathbb{E}[\bar{\mathbf{x}}(k)\bar{\mathbf{x}}^T(k)]$  é a matriz de autocorrelação do sinal de entrada atrasado em  $N$  amostras.

## 5.5 Resultados

Nesta seção, são feitas simulações acerca da análise proposta neste capítulo. Tais simulações consistem em identificar um sistema, como na configuração proposta neste trabalho, cuja função de transferência foi definida na seção de resultados do capítulo 3. O sinal de entrada é obtido filtrando-se um sinal branco, gaussiano, e de variância unitária pela função de transferência  $H(z) = 1 + 0.8z^{-1} - 0.2z^{-2}$ . As curvas de aprendizado são obtidas fazendo a média dos resultados provenientes de 1000 simulações independentes de Monte Carlo.

A Figura 5.1 mostra os resultados obtidos da análise de primeira ordem, sendo possível observar a evolução dos coeficientes na média. Nessas simulações, foram utilizados os parâmetros  $\beta = 0.015$ ,  $\kappa = 0.0316$ ,  $\sigma_\nu^2 = 0.1$  e ótimo  $w_\ell^*$  em ambos os casos. A Figura 5.1(a) mostra o resultado obtido utilizando  $P = 1$ , enquanto que a Figura 5.1(b) mostra o resultado obtido utilizando  $P = 4$ . Com os resultados apresentados, pode-se concluir que a análise de primeira ordem possui boa fidelidade com os resultados experimentais.

A Figura 5.2 mostra a evolução do MSE ao longo das iterações, permitindo avaliar a capacidade de predição do modelo teórico para a métrica em questão. Nessas simulações, foram utilizados os parâmetros  $P = 1$ ,  $\kappa = 0.0316$ ,  $\sigma_\nu^2 = 0.1$  e ótimo  $w_\ell^*$  em ambos os casos. A Figura 5.2(a) mostra o resultado obtido utilizando  $\beta = 0.015$ , enquanto que a Figura 5.2(b) mostra o resultado obtido utilizando  $\beta = 0.0015$ . Com os resultados apresentados, pode-se concluir que o modelo teórico é capaz de prever a curva de aprendizado de maneira fidedigna com os resultados experimentais.

A Figura 5.3 mostra a evolução do MSE ao longo das iterações, permitindo avaliar a capacidade de predição do modelo teórico para a métrica em questão. Nessas simulações, foram utilizados os parâmetros  $P = 4$ ,  $\kappa = 0.0316$ ,  $\sigma_\nu^2 = 0.1$  e ótimo  $w_\ell^*$  em ambos os casos. A Figura 5.3(a) mostra o resultado obtido utilizando  $\beta = 0.015$ , enquanto que a Figura 5.3(b) mostra o resultado obtido utilizando  $\beta = 0.0015$ . Com os resultados apresentados, pode-se concluir que o modelo teórico é capaz de prever a curva de aprendizado de maneira fidedigna com os resultados experimentais.

Tanto na Figura 5.2 quanto na Figura 5.3, é possível observar um descolamento

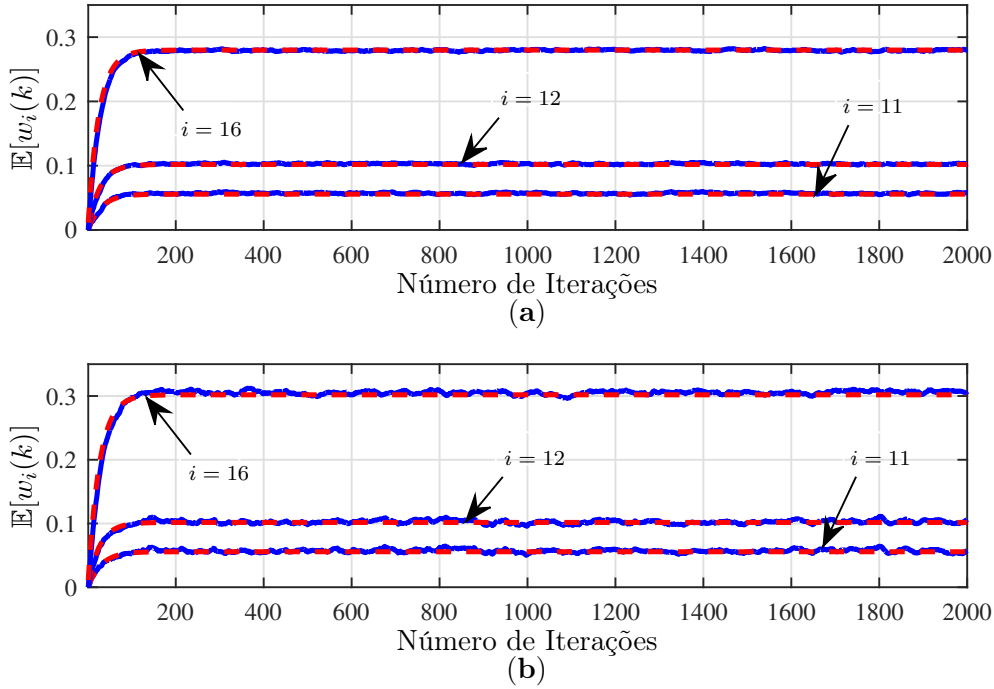


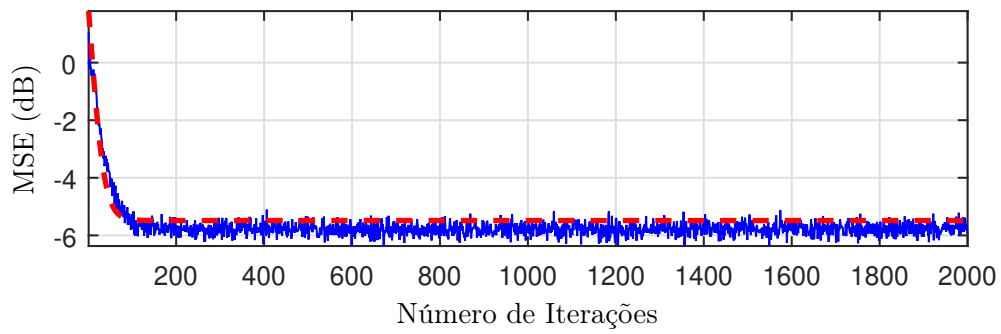
Figura 5.1: Comparação da evolução de alguns coeficientes  $w_i(k)$ , na média, entre simulado (em azul) e teórico (em vermelho), com  $i \in 11, 12, 16$ , utilizando  $\mathbf{w}_i^*$ . (a) Com  $P = 1$ . (b) Com  $P = 4$ .

entre o simulado e o teórico quando beta aumenta. Isso acontece, principalmente, por conta da hipótese **H1**, onde é assumido que  $\mathbf{x}(k)$  e  $\tilde{\mathbf{w}}(k)$  são independentes. Isso claramente não é verdade, já que a atualização de  $\tilde{\mathbf{w}}(k)$  depende de  $\mathbf{x}(k)$ , mas como essa atualização é feita de forma gradual e sem grandes saltos, essa hipótese é amplamente utilizada como uma ferramenta para facilitar o desenvolvimento do modelo estatístico nesse tipo de análise.

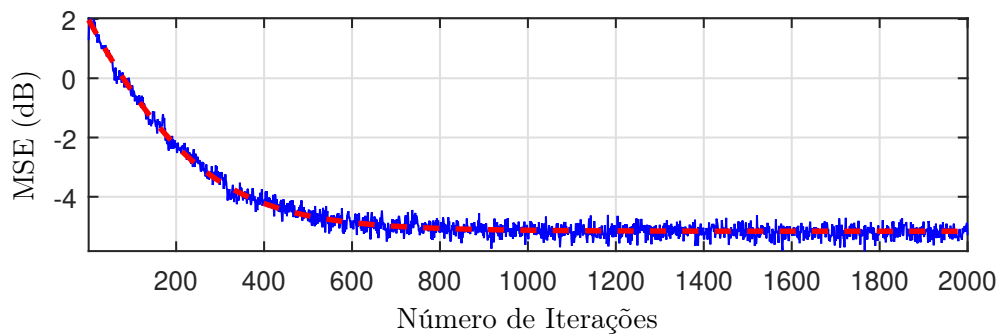
Há outros tipos de análise como, por exemplo, a análise exata [39] que não utilizam essa hipótese e, portanto, esse descolamento das curvas não é observado. Porém, são análises extremamente complexas e com um custo computacional muito elevado.

## 5.6 Comentários Finais

Neste capítulo, foi proposto um modelo estocástico com o objetivo de fornecer informações sobre o comportamento do algoritmo proposto neste trabalho quando há algum tipo de limitação quanto ao tamanho do filtro adaptativo, fazendo com que este não possua o mesmo comprimento que o filtro ótimo. Também foram feitos experimentos para avaliar a fidelidade do modelo teórico proposto. As simulações apresentadas na seção de resultados permitem concluir que o modelo teórico



(a)

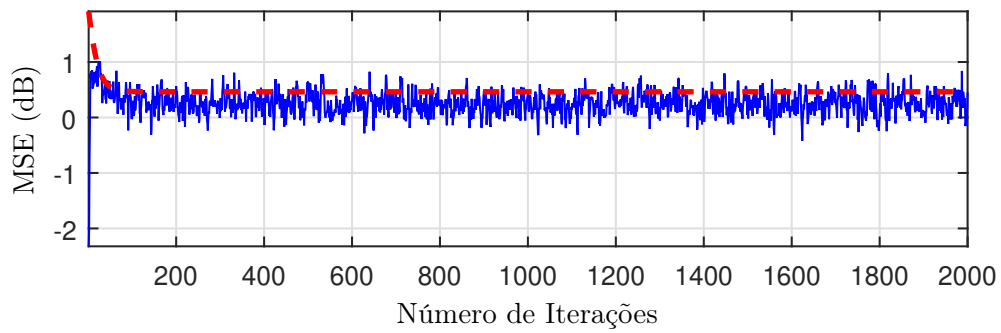


(b)

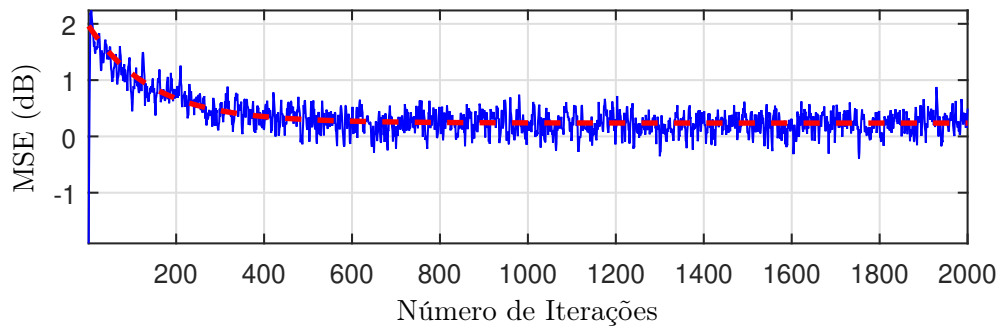
Figura 5.2: Comparação da evolução do MSE simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS, utilizando  $\mathbf{w}_l^*$  e  $P = 1$ . (a) Com  $\beta = 0.015$ . (b) Com  $\beta = 0.0015$ .

proposto é capaz de fornecer informações adequadas acerca do comportamento de primeira ordem e quadrático médio do algoritmo, sendo capaz de prever as curvas de aprendizado com fidelidade adequada.





(a)



(b)

Figura 5.3: Comparação da evolução do MSE simulado (em azul) e teórico (em vermelho) do algoritmo  $\ell_2$ -F-LMS, utilizando  $\mathbf{w}_i^*$  e  $P = 4$ . (a) Com  $\beta = 0.015$ . (b) Com  $\beta = 0.0015$ .

# Capítulo 6

## Conclusão

Neste trabalho, foram propostos um arcabouço matemático capaz de gerar diversos algoritmos adaptativos, o algoritmo  $\ell_2$ -F-LMS capaz de explorar esparsidade através da penalização de soluções cujos vetores  $\mathbf{F}\mathbf{w}(k)$  possuam alta energia associada, e um modelo teórico capaz de prever as curvas de aprendizado com fidelidade adequada e fornecer informações ao projetista acerca do comportamento do algoritmo  $\ell_2$ -F-LMS, tanto para o caso sem restrição quanto para o caso com restrição do comprimento do filtro.

No Capítulo 2, foi proposto o arcabouço matemático supracitado capaz de gerar diversos algoritmos, no caso, conscientes de esparsidade, bem como um estudo de caso de um apanhado de funções encontradas na literatura, sendo avaliados seus desempenhos utilizando duas funções de transferências reais de duas salas de aula.

No Capítulo 3, utiliza-se a formulação matemática proposta no Capítulo 2 para desenvolver o algoritmo  $\ell_2$ -F-LMS, e comparar com um algoritmo da mesma família proposto em [2] e com o algoritmo LMS. Através dos experimentos realizados neste capítulo, foi possível observar que ambos os algoritmos conscientes de esparsidade são capazes de superar o desempenho do algoritmo LMS, tanto em regime permanente como em taxa de convergência, ressaltando-se o desempenho ligeiramente superior do algoritmo proposto neste trabalho nas configurações simuladas.

No Capítulo 4, foi proposto um modelo estocástico, onde se estudou a capacidade do modelo de prever o comportamento de primeira ordem, quadrático médio e de rastreamento do algoritmo, enquanto se fazia uma comparação do desempenho do algoritmo proposto com o do algoritmo LMS. As simulações realizadas neste capítulo permitiram corroborar os resultados obtidos no Capítulo 3 com relação ao algoritmo LMS, e ao mesmo tempo, mostrar que o modelo teórico proposto é capaz de fazer previsões adequadas acerca do comportamento, fornecendo curvas de aprendizado fiéis às experimentais.

No Capítulo 5, foi proposto um modelo estocástico, assim como no Capítulo 4, mas com o objetivo de estudar o comportamento do algoritmo quando há restri-

ções quanto ao comprimento do filtro adaptativo, tendo este tamanho menor que o do filtro ótimo. Pelos experimentos realizados neste capítulo, é possível concluir que o modelo teórico é capaz de prever o comportamento do algoritmo de maneira satisfatória e que as curvas de aprendizado são fidedignas às experimentais.

Existem extensões das propostas descritas nesta dissertação que podem ser consideradas promissoras. Por exemplo, a adição do termo penalizador do algoritmo  $\ell_2$ -norm LMS em diversos outros algoritmos na literatura obtidos mediante a técnica de multiplicadores de Lagrange, tais como os algoritmos proporcionais.

## 6.1 Publicações Correlatas

Um artigo de tema correlato, oriundo de um trabalho anterior, foi publicado em conferência nacional [36]. Também, durante este trabalho, um artigo sobre o algoritmo proposto no Capítulo 3 e a análise desenvolvida no Capítulo 4 foi publicado em periódico internacional [35].

# Referências Bibliográficas

- [1] DA S. OLINTO, K., HADDAD, D. B., PETRAGLIA, M. R. “Transient analysis of  $l_0$ -LMS and  $l_0$ -NLMS algorithms”, *Signal Processing*, v. 127, pp. 217 – 226, Jan 2016. ISSN: 0165-1684.
- [2] DINIZ, P. S. R., YAZDANPANA, H., LIMA, M. V. S. “Feature LMS Algorithms”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4144–4148, April 2018. doi: 10.1109/ICASSP.2018.8461674.
- [3] HADDAD, D. B., PETRAGLIA, M. R. “Transient and steady-state MSE analysis of the IMPNLMS algorithm”, *Digital Signal Processing*, v. 33, pp. 50 – 59, July 2014. ISSN: 1051-2004.
- [4] 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, Sept 2009. ISSN: 1070-9908.
- [5] KALOUPSIDIS, N., MILEOUNIS, G., BABADI, B., et al. “Adaptive algorithms for sparse system identification”, *Signal Processing*, v. 91, n. 8, pp. 1910 – 1919, Feb 2011. ISSN: 0165-1684.
- [6] YAZDANPANA, H., APOLINARIO, J., DINIZ, P., et al. “ $l_0$ -NORM FEATURE LMS ALGORITHMS”. In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 311–315. IEEE, 11 2018. doi: 10.1109/GlobalSIP.2018.8646465.
- [7] SU, G., JIN, J., GU, Y., et al. “Performance Analysis of  $l_0$  Norm Constraint Least Mean Square Algorithm”, *IEEE Transactions on Signal Processing*, v. 60, n. 5, pp. 2223–2235, May 2012. ISSN: 1053-587X.
- [8] YU, Y., ZHAO, H., CHEN, B. “Sparse normalized subband adaptive filter algorithm with  $l_0$ -norm constraint”, *Journal of the Franklin Institute*, v. 353, n. 18, pp. 5121 – 5136, Sept 2016. ISSN: 0016-0032.

- [9] HADDAD, D. B., PETRAGLIA, M. R., PETRAGLIA, A. “A unified approach for sparsity-aware and maximum correntropy adaptive filters”. In: *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 170–174, Aug 2016. doi: 10.1109/EUSIPCO.2016.7760232.
- [10] MARTIN, R. K., SETHARES, W. A., WILLIAMSON, R. C., et al. “Exploiting sparsity in adaptive filters”, *IEEE Transactions on Signal Processing*, v. 50, n. 8, pp. 1883–1894, Aug 2002. ISSN: 1053-587X.
- [11] HOMER, J. “Detection guided NLMS estimation of sparsely parametrized channels”, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 47, n. 12, pp. 1437–1442, Dec 2000. ISSN: 1057-7130.
- [12] 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, Sep 2000. ISSN: 1063-6676.
- [13] VEGA, L. R., REY, H., BENESTY, J., et al. “A Family of Robust Algorithms Exploiting Sparsity in Adaptive Filters”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 17, n. 4, pp. 572–581, May 2009. ISSN: 1558-7916.
- [14] CANDES, E., TAO, T. “Decoding by Linear Programming”, *Information Theory, IEEE Transactions on*, v. 51, pp. 4203 – 4215, 01 2006. doi: 10.1109/TIT.2005.858979.
- [15] HURLEY, N., RICKARD, S. “Comparing Measures of Sparsity”, *IEEE Transactions on Information Theory*, v. 55, n. 10, pp. 4723–4741, Oct 2009. ISSN: 1557-9654. doi: 10.1109/TIT.2009.2027527.
- [16] KARVANEN, J. “Measuring Sparseness Of Noisy Signals”, 03 2003.
- [17] LIMA, M., MARTINS, W., DINIZ, P. “Affine projection algorithms for sparse system identification”. pp. 5666–5670, 10 2013. doi: 10.1109/ICASSP.2013.6638749.
- [18] SING-LONG, C., TEJOS, C., IRARRAZAVAL, P. “Evaluation of continuous approximation functions for the l0-norm for compressed sensing”, *International Soc. Mag. Reson. Med.(ISMRM 2009)*, v. 17, pp. 4585, 2009.
- [19] LIMA, M., SOBRON, I., MARTINS, W., et al. “Stability and MSE analyses of affine projection algorithms for sparse system identification”, *ICASSP*,

- IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 6399–6403, 05 2014. doi: 10.1109/ICASSP.2014.6854836.
- [20] WAGNER, K. T., DOROSLOVACKI, M. I. “Towards analytical convergence analysis of proportionate-type NLMS algorithms”. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 3825–3828. IEEE, March 2008.
- [21] HOYER, P. O. “Non-negative matrix factorization with sparseness constraints”. 2004.
- [22] GINI, C. “Measurement of inequality of incomes”, *The Economic Journal*, v. 31, n. 121, pp. 124–126, 1921.
- [23] DALTON, H. “The measurement of the inequality of incomes”, *The Economic Journal*, v. 30, n. 119, pp. 348–361, 1920.
- [24] ARNOLD, B. C. *Majorization and the Lorenz order: A brief introduction*, v. 43. Springer Science & Business Media, 2012.
- [25] LORENZ, M. O. “Methods of measuring the concentration of wealth”, *Publications of the American statistical association*, v. 9, n. 70, pp. 209–219, 1905.
- [26] SHALIT, H., YITZHAKI, S. “The mean-gini efficient portfolio frontier”, *Journal of Financial Research*, v. 28, n. 1, pp. 59–75, 2005.
- [27] AABERGE, R. “Axiomatic characterization of the Gini coefficient and Lorenz curve orderings”, *Journal of Economic Theory*, v. 101, n. 1, pp. 115–132, 2001.
- [28] AABERGE, R. “Erratum to “Axiomatic characterization of the Gini coefficient and Lorenz curve orderings” [J. Econ. Theory 101 (2001) 115–132]”, *Journal of Economic Theory*, v. 140, pp. e1, 05 2008. doi: 10.1016/j.jet.2004.07.004.
- [29] MILANOVIC, B. “A simple way to calculate the Gini coefficient, and some implications”, *Economics Letters*, v. 56, n. 1, pp. 45–49, 1997.
- [30] RICKARD, S., FALLON, M. “The Gini index of speech”. In: *Proceedings of the 38th Conference on Information Science and Systems (CISS’04)*, 2004.
- [31] RICKARD, S. “Sparse sources are separated sources”. In: *2006 14th European Signal Processing Conference*, pp. 1–5. IEEE, 2006.

- [32] HURLEY, N., RICKARD, S., CURRAN, P. “Parameterized lifting for sparse signal representations using the Gini index”, *Signal Processing with Adaptive Sparse Structured Representations (SPARS05)*, Rennes, France, 2005.
- [33] HURLEY, N., RICKARD, S., CURRAN, P., et al. “Maximizing sparsity of wavelet representations via parameterized lifting”. In: *2007 15th International Conference on Digital Signal Processing*, pp. 631–634. IEEE, 2007.
- [34] DAS, B. K., AZPICUETA-RUIZ, L. A., CHAKRABORTY, M., et al. “A comparative study of two popular families of sparsity-aware adaptive filters”. In: *2014 4th International Workshop on Cognitive Information Processing (CIP)*, pp. 1–6, May 2014.
- [35] HADDAD, D., SANTOS, L., ALMEIDA, L., et al. “l<sub>2</sub>-norm Feature LMS Algorithm”, *Electronics Letters*, 02 2020. doi: 10.1049/el.2019.3939.
- [36] SANTOS, L. O., HADDAD, D. B., PETRAGLIA, M. R. “Análise de Transiente do Algoritmo l<sub>0</sub>-sign-LMS”, *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT)*, 2017.
- [37] DENG, H., DOROSLOVACKI, M. “Proportionate adaptive algorithms for network echo cancellation”, *IEEE Transactions on Signal Processing*, v. 54, n. 5, pp. 1794–1803, Apr 2006.
- [38] PETRAGLIA, M. R., HADDAD, D. B. “New adaptive algorithms for identification of sparse impulse responses—Analysis and comparisons”. In: *2010 7th International Symposium on Wireless Communication Systems*, pp. 384–388. IEEE, 2010.
- [39] LARA, P., TARRATACA, L., HADDAD, D. “Exact Expectation Analysis of the Deficient-Length LMS Algorithm”, *Signal Processing*, v. 162, 04 2019. doi: 10.1016/j.sigpro.2019.04.009.
- [40] DOUGLAS, S. C., PAN, W. “Exact expectation analysis of the LMS adaptive filter”, *IEEE Transactions on Signal Processing*, v. 43, n. 12, pp. 2863–2871, Dec 1995. ISSN: 1053-587X.
- [41] AL-NAFFOURI, T. Y., SAYED, A. H. “Transient analysis of adaptive filters with error nonlinearities”, *IEEE Transactions on Signal Processing*, v. 51, n. 3, pp. 653–663, March 2003. ISSN: 1053-587X.
- [42] DINIZ, P. S. R. “Adaptive filtering: algorithms and practical implementation”, *The international series in Engineering and Computer Scienc*, 2008.

- [43] RICKARD, J., ZEIDLER, J. “Second-order output statistics of the adaptive line enhancer”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 27, n. 1, pp. 31–39, Feb 1979.
- [44] WIDROW, B., STEARNS, S. D. “Adaptive signal processing”, *Printice Hall, New Jersey*, 1985.
- [45] HAYKIN, S. S. *Adaptive filter theory*. India, Pearson Education India, 2008.
- [46] SAMSON, C., REDDY, V. “Fixed point error analysis of the normalized ladder algorithm”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 31, n. 5, pp. 1177–1191, Oct 1983. ISSN: 0096-3518.
- [47] WERNER, S., DE CAMPOS, M. L., DINIZ, P. S. “Partial-update NLMS algorithms with data-selective updating”, *IEEE Transactions on Signal Processing*, v. 52, n. 4, pp. 938–949, April 2004.
- [48] BERSHAD, N. J., EWEDA, E., BERMUDEZ, J. C. “Stochastic analysis of the LMS and NLMS algorithms for cyclostationary white Gaussian inputs”, *IEEE Transactions on Signal Processing*, v. 62, n. 9, pp. 2238–2249, May 2014.
- [49] FARHANG-BOROJENY, B., GAZOR, S. “Performance of LMS-based adaptive filters in tracking a time-varying plant”, *IEEE Transactions on Signal Processing*, v. 44, n. 11, pp. 2868–2871, 1996.
- [50] EWEDA, E. “Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels”, *IEEE Transactions on Signal Processing*, v. 42, n. 11, pp. 2937–2944, Nov 1994. ISSN: 1941-0476. doi: 10.1109/78.330354.
- [51] SILVA, M., NASCIMENTO, V. “Improving the Tracking Capability of Adaptive Filters via Convex Combination”, *Signal Processing, IEEE Transactions on*, v. 56, pp. 3137 – 3149, 08 2008. doi: 10.1109/TSP.2008.919105.
- [52] HOROWITZ, L., SENNE, K. “Performance Advantage of Complex LMS for Controlling Narrow-Band Adaptive Arrays”, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 29, pp. 722– 736, 07 1981. doi: 10.1109/TCS.1981.1085024.
- [53] TARRAB, M., FEUER, A. “Convergence and performance analysis of the normalized LMS algorithm with uncorrelated Gaussian data”, *Information Theory, IEEE Transactions on*, v. 34, pp. 680 – 691, 08 1988. doi: 10.1109/18.9768.



- [54] FOLEY, J., BOLAND, F. “A Note on the Convergence Analysis of LMS Adaptive Filters with Gaussian Data”, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 36, pp. 1087 – 1089, 08 1988. doi: 10.1109/29.1632.
- [55] EWEDA, E. “Transient performance degradation of the LMS, RLS, sign, signed regressor, and sign-sign algorithms with data correlation”, *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, v. 46, pp. 1055 – 1062, 09 1999. doi: 10.1109/82.782049.
- [56] SAYED, A., RUPP, M. “Error-energy bounds for adaptive gradient algorithms”, *Signal Processing, IEEE Transactions on*, v. 44, pp. 1982 – 1989, 09 1996. doi: 10.1109/78.533719.
- [57] YOUSEF, N., SAYED, A. “A unified approach to the steady-state and tracking analyses of adaptive filters”, *Signal Processing, IEEE Transactions on*, v. 49, pp. 314 – 324, 03 2001. doi: 10.1109/78.902113.
- [58] BUTTERWECK, H. “A wave theory of long LMS adaptive filters”, *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, v. 48, pp. 739 – 747, 07 2001. doi: 10.1109/81.928156.
- [59] MAYYAS, K. “Performance Analysis of the Deficient Length LMS Adaptive Algorithm”, *Signal Processing, IEEE Transactions on*, v. 53, pp. 2727 – 2734, 09 2005. doi: 10.1109/TSP.2005.850347.
- [60] WU, M., YANG, J. “A Step Size Control Method for Deficient Length FBLMS Algorithm”, *IEEE Signal Processing Letters*, v. 22, pp. 1–1, 01 2014. doi: 10.1109/LSP.2014.2382113.
- [61] DEALMEIDA, S., COSTA, M., BERMUDEZ, J. “A stochastic model for the deficient length pseudo affine projection adaptive algorithm”, *European Signal Processing Conference*, pp. 1715–1719, 01 2009.
- [62] BILCU, R., KUOSMANEN, P., EGIAZARIAN, K. “A new variable length LMS algorithm: Theoretical analysis and implementations”. In: *9th International conference on Electronics, Circuits and Systems*, v. 3, pp. 1031 – 1034 vol.3. IEEE, 02 2002. ISBN: 0-7803-7596-3. doi: 10.1109/ICECS.2002.1046426.
- [63] GU, Y., TANG, K., CUI, H., et al. “Convergence analysis of a deficient-length LMS filter and optimal-length sequence to model exponential decay impulse response”, *Signal Processing Letters, IEEE*, v. 10, pp. 4 – 7, 02 2003. doi: 10.1109/LSP.2002.806704.

- [64] XIA, Y., DOUGLAS, S., MANDIC, D. “Performance Analysis of the Deficient Length Augmented CLMS Algorithm for Second Order Non-circular Complex Signals”, *Signal Processing*, v. 144, 10 2017. doi: 10.1016/j.sigpro.2017.10.021.
- [65] PALEOLOGU, C., CIOCHINA, S., BENESTY, J. “Variable Step-Size NLMS Algorithm for Under-Modeling Acoustic Echo Cancellation”, *Signal Processing Letters, IEEE*, v. 15, pp. 5 – 8, 02 2008. doi: 10.1109/LSP.2007.910276.
- [66] FEUER, A., WEINSTEIN, E. “Convergence analysis of LMS filters with uncorrelated Gaussian data”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 33, n. 1, pp. 222–230, 1985.