

Eduardo Beck

**ALGORITMOS ADAPTATIVOS BASEADOS EM PROJEÇÕES E  
RESTRICÇÃO DE NORMA PARA IDENTIFICAÇÃO DE SISTEMAS  
ESPARSOS:  
NOVA ABORDAGEM E MODELAGEM ESTOCÁSTICA**

Tese submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do Grau de Doutor em Engenharia Elétrica.

Orientador: Prof. Dr. Rui Seara

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor através do  
Programa de Geração Automática da Biblioteca Universitária da UFSC.

Beck, Eduardo

Algoritmos Adaptativos Baseados em Projeções e Restrição  
de Norma para Identificação de Sistemas Esparsos : Nova  
Abordagem e Modelagem Estocástica / Eduardo Beck ;  
orientador, Rui Seara - Florianópolis, SC, 2016.  
166 p.

Tese (doutorado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação  
em Engenharia Elétrica.

Inclui referências

1. Engenharia Elétrica. 2. Algoritmos adaptativos.  
3. Sistemas esparsos. 4. Normas vetoriais. 5. Modelagem  
estocástica. I. Seara, Rui. II. Universidade Federal de  
Santa Catarina. Programa de Pós-Graduação em Engenharia  
Elétrica. III. Título.

Eduardo Beck

**ALGORITMOS ADAPTATIVOS BASEADOS EM PROJEÇÕES E  
RESTRICÇÃO DE NORMA PARA IDENTIFICAÇÃO DE SISTEMAS  
ESPARSOS: NOVA ABORDAGEM E MODELAGEM ESTOCÁSTICA**

Esta Tese foi julgada adequada para a obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração Comunicações e Processamento de Sinais, e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Florianópolis, 24 de fevereiro 2016.

---

Prof. Dr. Carlos Galup Montoro  
Coordenador

**Banca Examinadora:**

---

Prof. Dr. Rui Seara  
Orientador

---

Primeiro membro  
Universidade ...

---

Segundo membro  
Universidade ...

---

Terceiro membro  
Universidade ...

---

Quarto membro  
Universidade ...

Dedico este trabalho a minha querida esposa Milene e minhas filhas e neto, Katiuscia, Gabrielle e Kaio. Dedico ainda à memória de meus pais, Arno e Ruth, e de meus irmãos Cláudio e Letícia.



## Agradecimentos

Inicialmente, agradeço a meu orientador, Prof. Rui Seara, por me receber como seu orientado (após longo tempo afastado do Linse) permitindo que eu compartilhasse de seus preciosos ensinamentos.

Agradeço também ao Prof. Eduardo Batista (o Dudu) por sua essencial participação no desenvolvimento da pesquisa que fundamenta esta tese e sua perseverança na busca do melhor resultado.

Devo agradecer ainda aos amigos Elton (e sua veia artística com as figuras) e Walter (o prático incansável), que sempre estiveram a disposição para colaborar com a construção deste trabalho, bem como para as boas conversas nos intervalos para o cafezinho.

Agradeço a meus colegas e amigos de sala e empreitada, Eduardo Kuhn (que foi fundamental na revisão deste trabalho), Marcos Matsuo (grande colega de estudos e trabalhos de aula), os veteranos de guerra Fábio e Nonato, e ainda Ciro, Zilli, Ênio, Barbosa, Douglas, Gabriel, Lariana, Bernardo, Bina, Francisco, Javier e outros colegas que por aqui passaram, por sua compreensão, atenção, colaboração e bons papos.

Não posso deixar de agradecer também aos demais pesquisadores e amigos do Linse, pela agradável convivência e grande ajuda, destacando as amigas Simone, Sandra, Carol, Dominique e Profa. Izabel, e os amigos Augusto, Neco, Guilherme, Rafael, Rodrigo, João Paulo, Feijão, Prof. Sidnei e o saudoso Orlando. Sem esquecer os nerds da sala do professor Walter.

Agradeço as valorosas contribuições que enriqueceram muito este trabalho, trazidas pelos professores que se dispuseram a participar de minhas bancas de avaliação. A Profa. Mariane e os professores Renato, Walter, Bartolomeu, Eduardo, Meloni e Hans.

Agradeço ainda a meus amigos do Curso Técnico em Meteorologia, Márcia, Mário, Daniel, Michel e Yoshi, por apoiarem meu afastamento da instituição para o desenvolvimento deste trabalho.

Agradeço muito à minha esposa Milene, minhas filhas e neto, Katiuscia, Gabrielle e Kaio, meus irmãos Amauri, Cláudio, Letícia, Milton e Sérgio, e também a todos os meus amigos e familiares que me apoiaram nessa árdua caminhada.

Por fim, agradeço ao IFSC - Instituto Federal de Santa Catarina, instituição da qual faço parte a mais de 25 anos, por me propiciar o tempo e a dedicação para a execução da pesquisa que culminou nesta tese.





## Resumo

Neste trabalho de pesquisa, novos algoritmos de filtragem adaptativa orientados à identificação de sistemas esparsos são desenvolvidos. Tais algoritmos baseiam-se no uso de restrição de norma aplicada ao vetor de coeficientes, de forma semelhante a outros algoritmos voltados a sistemas esparsos, destacando-se os algoritmos baseados nas projeções em esferas de norma  $\ell_1$  ( $\ell_1$  norm ball), recentemente propostos. Em contraste com tais algoritmos, a abordagem aqui proposta incorpora uma restrição de norma ao processo de otimização simultaneamente à restrição afim, que caracteriza o algoritmo NLMS (*normalized least-mean-square algorithm*). Com uma concepção mais simples do que alguns importantes algoritmos da literatura, essa nova abordagem leva a algoritmos eficazes com menor complexidade computacional e com parâmetros de controle de fácil ajuste. Uma formulação geral para obtenção dos algoritmos é proposta de maneira a permitir a utilização de diferentes tipos de normas em sua concepção. A partir de tal formulação, duas famílias de algoritmos são derivadas, a saber: os algoritmos com restrição de norma  $\ell_1$  [ $\ell_1$ -norm constrained ( $\ell_1$ NC) algorithms] e os algoritmos com restrição de norma  $\ell_0$  [ $\ell_0$ -norm constrained ( $\ell_0$ NC) algorithms], sendo que diferentes versões desses algoritmos são apresentadas e discutidas. Através de simulações numéricas, os algoritmos propostos são avaliados, exibindo resultados promissores quando comparados com outros algoritmos da literatura para aplicações em identificação de sistemas esparsos. Modelos estocásticos para os algoritmos propostos são também derivados, buscando prever seu comportamento em diferentes condições de operação.

**Palavras-chave:** Algoritmo adaptativo. Norma  $\ell_0$ . Norma  $\ell_1$ . Otimização. Restrições. Sistemas esparsos.



## Abstract

In this research work, novel adaptive algorithms for sparse system identification are developed. Such algorithms are based on norm constraint applied to the weight vector, similarly to other algorithms aimed at sparse systems, especially the projection-based algorithms using  $\ell_1$  norm ball recently proposed. In contrast to such algorithms, the proposed approach incorporates a norm constraint into the optimization process along with the affine constraint, which characterizes the normalized least-mean-square (NLMS) algorithm. With a simpler conception than some important algorithms from the literature, this new approach leads to efficient algorithms with a lower computational complexity and an easy adjustment of their control parameters. A general formulation is proposed allowing the use of different weight-vector norms in the algorithm development. From this formulation, two algorithm families have been derived, namely: the  $\ell_1$ -norm constrained ( $\ell_1$ NC) algorithm and the  $\ell_0$ -norm constrained ( $\ell_0$ NC) one. Different versions of these algorithms are presented and discussed. Through numerical simulations, the proposed algorithms are assessed. These algorithms have shown promising results as compared with other algorithms from the literature for applications in sparse system identification. Stochastic models for the proposed algorithms are also derived, aiming to predict their behavior in different operating conditions.

**Keywords:** Adaptive algorithm.  $\ell_0$ -norm.  $\ell_1$ -norm. Optimization. Constraints. Sparse systems.



## Lista de Figuras

|           |  |    |
|-----------|--|----|
| Figura 1  | Diagrama representativo de um problema geral de filtragem adaptativa.....  | 29 |
| Figura 2  | Diagrama de blocos de um problema de identificação de sistemas.....  | 30 |
| Figura 3  | Diagrama em blocos de uma estrutura típica de um sistema de filtragem adaptativa para cancelamento de eco.....   | 31 |
| Figura 4  | Resposta ao impulso típica de um caminho de eco em sistema de comunicação.....   | 33 |
| Figura 5  | Representação de uma função com restrição, apresentando as direções dos vetores de gradiente e a direção de descida.....                               | 41 |
| Figura 6  | Representação gráfica de estimação baseada na abordagem da teoria de conjuntos.....  | 45 |
| Figura 7  | Princípio de operação do método serial de projeções sobre conjuntos convexos proposto por Kaczmarz.....  | 47 |
| Figura 8  | Princípio de operação do método paralelo de projeções sobre conjuntos convexos proposto por Cimmino.....   | 47 |
| Figura 9  | Processo iterativo de projeções sucessivas nas superfícies de soluções geradas por uma mesma restrição função de $n$ .....                             | 48 |
| Figura 10 | Princípio de operação do algoritmo APL1.....   | 58 |
| Figura 11 | Princípio de operação do algoritmo APWL1, apresentando modificação da esfera de norma $\ell_1$ .....   | 58 |
| Figura 12 | Projeção na interseção entre as superfícies $S_\varepsilon(n)$ e $S'_{\ell_1}(n)$ ....   | 67 |
| Figura 13 | Exemplo 3.1. Desempenho em regime permanente do algoritmo $\ell_1$ NC para $\mu = 1$ em função de $\delta_{\ell_1}$ .....                              | 73 |
| Figura 14 | Exemplo 3.1. Desempenho em regime permanente do algoritmo $\ell_1$ NC para $\mu = 1$ , $\mu = 0,1$ e $\mu = 0,01$ em função de $\delta_{\ell_1}$ ..... | 73 |
| Figura 15 | Exemplo 3.2. Comparação de desempenho entre os algoritmos NLMS, $\ell_1$ NC e V- $\ell_1$ NC.....  | 74 |
| Figura 16 | Comportamento da função (3.41), que representa cada elemento do vetor na aproximação da norma $\ell_0$ , para diferentes valores de $\beta$ .....      | 76 |
| Figura 17 | Curvas de nível com valor igual a 1 da função (3.42), que aproxima a norma $\ell_0$ , para diferentes valores de $\beta$ .....                         | 76 |
| Figura 18 | Comportamento da função (3.44), que representa cada ele-   |    |

|   |     |
|---|-----|
| mento do vetor na aproximação da norma $\ell_0$ , para diferentes valores de $\beta$ .....  | 78  |
| Figura 19 Curvas de nível com valor igual a 1 da função (3.45), que aproxima a norma $\ell_0$ , para diferentes valores de $\beta$ .....                    | 78  |
| Figura 20 Comportamento da função (3.69), que representa cada elemento do vetor na aproximação da norma $\ell_0$ , para diferentes valores de $\beta$ ..... | 86  |
| Figura 21 Curvas de nível com valor igual a 1 da função (3.73), que aproxima a norma $\ell_0$ , para diferentes valores de $\beta$ .....                    | 86  |
| Figura 22 Representação das derivadas de (3.69) em dois pontos distintos.....   | 87  |
| Figura 23 Princípio de operação do algoritmo Z- $\ell_0$ NC.....  | 90  |
| Figura 24 Exemplo 3.3. Desempenho em regime permanente do algoritmo $\ell_0$ NC em função de $\delta_{\ell_0}$ .....  | 92  |
| Figura 25 Exemplo 3.4. Desempenho em regime permanente do algoritmo $\ell_0$ NC em função de $\beta$ .....  | 93  |
| Figura 26 Exemplo 3.5. Desempenho em regime permanente do algoritmo V- $\ell_0$ NC em função de $\beta$ .....   | 94  |
| Figura 27 Exemplo 3.6. Comparação de desempenho entre os algoritmos NLMS, $\ell_0$ NC e V- $\ell_0$ NC.....   | 95  |
| Figura 28 Exemplo 3.7. Desempenho em regime permanente dos algoritmos V- $\ell_0$ NC e VS- $\ell_0$ NC em função de $\beta$ .....                           | 96  |
| Figura 29 Exemplo 3.8. Comparação de desempenho entre os algoritmos V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC.....                                   | 97  |
| Figura 30 Exemplo 3.8. Evolução da adaptação do coeficiente não significativo da posição 90 para o algoritmo V- $\ell_0$ NC.....                            | 98  |
| Figura 31 Exemplo 3.8. Evolução da adaptação de um coeficiente não significativo da posição 90 para o algoritmo Z- $\ell_0$ NC.....                         | 98  |
| Figura 32 Etapas do processo de adaptação do algoritmo $\ell_1$ NC.....   | 100 |
| Figura 33 Resposta ao impulso representativa de um caminho de eco definida pela Recomendação ITU-T G.168, modelo #1, com 512 coeficientes.....              | 107 |
| Figura 34 Resposta ao impulso representativa de um caminho de eco definida pela Recomendação ITU-T G.168, modelo #4, com 512 coeficientes.....              | 107 |
| Figura 35 Exemplo 4.1. Curvas de desalinhamento normalizado considerando a planta $\mathbf{h}_1$ .....  | 109 |
| Figura 36 Exemplo 4.2. Curvas de desalinhamento normalizado consi-  |     |

|  |     |
|--|-----|
| derando a planta $\mathbf{h}_4$ . . . . .  | 110 |
| Figura 37 Exemplo 4.3. Curvas de desalinhamento normalizado considerando a planta $\mathbf{h}_5$ . . . . .                                       | 111 |
| Figura 38 Exemplo 4.4. Curvas de desalinhamento normalizado (perturbação na iteração $n = 6000$ ). . . . .                                       | 113 |
| Figura 39 Exemplo 4.4. Curvas de desalinhamento normalizado (perturbação na iteração $n = 1000$ ). . . . .                                       | 113 |
| Figura 40 Exemplo 4.5. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_1$ . . . . .                                      | 119 |
| Figura 41 Exemplo 4.5. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_2$ . . . . .                                      | 120 |
| Figura 42 Exemplo 4.6. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_4$ e sinal de entrada gaussiano branco. . . . .   | 121 |
| Figura 43 Exemplo 4.6. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_4$ e sinal de entrada gaussiano colorido. . . . . | 122 |
| Figura 44 Exemplo 4.7. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_5$ e sinal de entrada gaussiano branco. . . . .   | 123 |
| Figura 45 Exemplo 4.7. Curvas de desalinhamento normalizado, considerando a planta $\mathbf{h}_5$ e sinal de entrada gaussiano colorido. . . . . | 124 |
| Figura 46 Exemplo 5.1. MSD em regime permanente do algoritmo $\ell_1$ NC, comparando o modelo com simulações de Monte Carlo. . . . .             | 143 |
| Figura 47 Exemplo 5.2. Comportamento médio dos coeficientes significativos para os algoritmos NLMS, $\ell_1$ NC e $V\text{-}\ell_1$ NC. . . . .  | 144 |
| Figura 48 Exemplo 5.2. MSD dos algoritmos NLMS, $\ell_1$ NC e $V\text{-}\ell_1$ NC. . . . .  | 144 |
| Figura 49 Exemplo 5.2. Comportamento médio dos parâmetros $\lambda_{\ell_1}(n)$ e $\lambda_{V\ell_1}(n)$ . . . . .                               | 145 |
| Figura 50 Exemplo 5.3. MSD em regime permanente do algoritmo $\ell_0$ NC, comparando o modelo com simulações de Monte Carlo. . . . .             | 152 |
| Figura 51 Exemplo 5.4. Comportamento médio dos coeficientes significativos para os algoritmos NLMS, $\ell_0$ NC e $V\text{-}\ell_0$ NC. . . . .  | 153 |
| Figura 52 Exemplo 5.4. MSD dos algoritmos NLMS, $\ell_0$ NC e $V\text{-}\ell_0$ NC. . . . .  | 153 |
| Figura 53 Exemplo 5.4. Valores esperados de $\lambda_{\ell_0}(n)$ e $\lambda_{V\ell_0}(n)$ . . . . .   | 154 |





## Lista de Tabelas

|           |   |     |
|-----------|---|-----|
| Tabela 1  | Sumário do algoritmo $\ell_1$ NC  | 69  |
| Tabela 2  | Sumário do algoritmo V- $\ell_1$ NC   | 71  |
| Tabela 3  | Sumário do problema de identificação de sistemas utilizado nos exemplos   | 72  |
| Tabela 4  | Sumário do algoritmo $\ell_0$ NC  | 80  |
| Tabela 5  | Sumário do algoritmo V- $\ell_0$ NC   | 83  |
| Tabela 6  | Sumário do algoritmo VS- $\ell_0$ NC  | 88  |
| Tabela 7  | Sumário do algoritmo Z- $\ell_0$ NC   | 91  |
| Tabela 8  | Complexidade computacional de alguns algoritmos para sistemas esparsos, em ordem de menor para maior número de multiplicações | 101 |
| Tabela 9  | Exemplo 4.1. Parâmetros de configuração dos algoritmos, considerando a planta $\mathbf{h}_1$                                  | 109 |
| Tabela 10 | Exemplo 4.2. Parâmetros de configuração dos algoritmos, considerando a planta $\mathbf{h}_4$                                  | 110 |
| Tabela 11 | Exemplo 4.3. Parâmetros de configuração dos algoritmos, considerando a planta $\mathbf{h}_5$                                  | 110 |
| Tabela 12 | Exemplo 4.4. Parâmetros de configuração dos algoritmos  | 112 |
| Tabela 13 | Exemplo 4.5. Parâmetros de configuração dos algoritmos  | 115 |
| Tabela 14 | Exemplo 4.6. Parâmetros de configuração dos algoritmos para a planta $\mathbf{h}_4$ e sinal de entrada gaussiano branco       | 116 |
| Tabela 15 | Exemplo 4.6. Parâmetros de configuração dos algoritmos para a planta $\mathbf{h}_4$ e sinal de entrada gaussiano colorido     | 116 |
| Tabela 16 | Exemplo 4.7. Parâmetros de configuração dos algoritmos para a planta $\mathbf{h}_5$ e sinal de entrada gaussiano branco       | 117 |
| Tabela 17 | Exemplo 4.7. Parâmetros de configuração dos algoritmos para a planta $\mathbf{h}_5$ e sinal de entrada gaussiano colorido     | 118 |
| Tabela 18 | Sumário do problema de identificação de sistemas utilizado nos exemplos   | 142 |



## Lista de Abreviaturas e Siglas

|                  |   |    |
|------------------|---|----|
| FIR              | finite impulse response . . . . .   | 28 |
| LMS              | least-mean-square . . . . .   | 31 |
| NLMS             | normalized least-mean-square . . . . .  | 32 |
| ITU-T            | International Telecommunication Union - Telecommunica-<br>tion Standardization Sector . . . . . | 32 |
| PNLMS            | proportionate normalized least-mean-square . . . . .  | 33 |
| ZA-LMS           | zero-attracting least-mean-square . . . . .   | 33 |
| RZA-LMS          | reweighted zero-attracting least-mean-square . . . . .  | 33 |
| APL1             | adaptive projection-based algorithm using $\ell_1$ balls . . . . .                              | 33 |
| APWL1            | adaptive projection-based algorithm using weighted $\ell_1$ balls                               | 33 |
| LMMSE            | linear minimum mean-square error . . . . .  | 42 |
| POCS             | projections onto convex sets . . . . .  | 46 |
| APA              | affine projection algorithm . . . . .   | 49 |
| PNLMS++          | proportionate normalized least-mean-squares ++ . . . . .  | 53 |
| IPNLMS           | improved proportionate normalized least-mean-squares . . .                                      | 53 |
| MPNLMS           | $\mu$ -law proportionate normalized least-mean-squares . . . . .                                | 53 |
| SPNLMS           | segment proportionate normalized least-mean-squares . . . .                                     | 54 |
| IIPNLMS          | improved improved proportionate normalized least-mean-<br>squares . . . . .                     | 54 |
| IPNLMS- $\ell_0$ | improved proportionate normalized least-mean-squares ba-<br>sed on $\ell_0$ norm . . . . .      | 54 |
| IAF-PNLMS        | individual-activation-factor proportionate normalized least-<br>mean-squares . . . . .          | 54 |
| SC-PNLMS         | sparseness-controlled proportionate normalized least-mean-<br>squares . . . . .                 | 54 |
| ZA-NLMS          | zero-attracting normalized least-mean-square . . . . .  | 55 |
| RZA-NLMS         | reweighted zero-attracting normalized least-mean-square . .                                     | 55 |
| $\ell_0$ -LMS    | $\ell_0$ norm constraint least-mean-square . . . . .  | 55 |
| $\ell_0$ -NLMS   | $\ell_0$ norm constraint normalized least-mean-square . . . . .                                 | 56 |
| $\ell_p$ NC      | $\ell_p$ -norm constrained algorithm . . . . .  | 61 |
| $\ell_1$ NC      | $\ell_1$ -norm constrained algorithm . . . . .  | 61 |
| $\ell_0$ NC      | $\ell_0$ -norm constrained algorithm . . . . .  | 61 |

|                 |  |     |
|-----------------|--|-----|
| V- $\ell_1$ NC  | variable $\ell_1$ -norm constrained algorithm . . . . .            | 70  |
| V- $\ell_0$ NC  | variable $\ell_0$ -norm constrained algorithm . . . . .            | 82  |
| S- $\ell_0$ NC  | smoothed $\ell_0$ -norm constrained algorithm . . . . .            | 84  |
| VS- $\ell_0$ NC | variable smoothed $\ell_0$ -norm constrained algorithm . . . . .   | 87  |
| Z- $\ell_0$ NC  | zeroed-coefficients $\ell_0$ -norm constrained algorithm . . . . . | 90  |
| MC              | Monte Carlo . . . . .  | 125 |
| MSE             | mean-square error . . . . .  | 127 |
| MSD             | mean-square deviation . . . . .                                    | 127 |

## Lista de Símbolos

|                      |   |    |
|----------------------|---|----|
| $y(n)$               | sinal de saída do filtro adaptativo . . . . .                                     | 28 |
| $d(n)$               | sinal desejado ou de referência . . . . .   | 28 |
| $x(n)$               | sinal de entrada do filtro . . . . .  | 28 |
| $\mathbf{w}(n)$      | vetor de coeficientes do filtro adaptativo . . . . .                              | 28 |
| $e(n)$               | sinal de erro instantâneo . . . . .   | 28 |
| $\mathbf{w}_0$       | vetor de coeficientes do sistema a ser identificado . . . . .                     | 29 |
| $r(n)$               | ruído de medição . . . . .  | 29 |
| $\kappa(n)$          | desalinhamento normalizado . . . . .  | 31 |
| $\mu$                | passo de adaptação . . . . .  | 31 |
| $f(\cdot)$           | função custo . . . . .  | 38 |
| $c_i(\cdot)$         | função de restrição $i$ . . . . .   | 38 |
| $\nabla f(\cdot)$    | gradiente da função custo . . . . .   | 38 |
| $\mathbf{d}(n)$      | vetor de direção da atualização no instante $n$ . . . . .                         | 38 |
| $\nabla' f(\cdot)$   | matriz hessiana da função custo . . . . .   | 39 |
| $\nabla c(\cdot)$    | gradiente da função de restrição . . . . .  | 40 |
| $\lambda_i$          | multiplicador de Lagrange referente à restrição $i$ . . . . .                     | 40 |
| $\mathcal{L}(\cdot)$ | função de Lagrange . . . . .  | 40 |
| $q(\cdot)$           | função dual de Lagrange . . . . .   | 41 |
| $\mathbf{p}$         | vetor de correlação cruzada entre o sinal desejado e o sinal de entrada . . . . . | 43 |
| $\mathbf{R}$         | matriz de autocorrelação do sinal de entrada . . . . .                            | 43 |
| $\mathbf{w}^*$       | vetor de coeficientes ótimo do filtro adaptativo . . . . .                        | 43 |
| $S$                  | conjunto de soluções . . . . .  | 45 |
| $P_{i_n}(\cdot)$     | função de projeção na superfície $i$ na iteração $n$ . . . . .                    | 46 |
| $\varepsilon(n)$     | sinal de erro <i>a posteriori</i> . . . . .                                       | 48 |
| $\ \mathbf{w}\ _p$   | norma $\ell_p$ do vetor $\mathbf{w}$ . . . . .                                    | 50 |
| $\mathcal{S}(\cdot)$ | grau de esparsidade de um vetor . . . . .   | 51 |
| $\mathbf{G}(n)$      | matriz de ganhos . . . . .  | 52 |
| $\gamma_i(n)$        | função de proporcionalidade . . . . .   | 52 |
| $\varrho$            | parâmetro de proporcionalidade . . . . .  | 52 |
| $\delta_p$           | parâmetro de inicialização . . . . .  | 52 |
| $\text{sgn}[\cdot]$  | vetor de sinais . . . . .   | 55 |

|                          |  |     |
|--------------------------|--|-----|
| $\rho$                   | parâmetro que controla o grau de atração da norma $\ell_1$ . . . . .                                 | 55  |
| $\kappa$                 | fator de penalização . . . . .   | 56  |
| $\delta_{\ell_p}$        | parâmetro de limitação de norma . . . . .  | 62  |
| $f_{\ell_p}[\cdot]$      | função afim que aproxima a norma $\ell_p$ . . . . .  | 63  |
| $\mathbf{g}_{\ell_p}(n)$ | gradiente da função que representa a norma $\ell_p$ . . . . .  | 64  |
| $\mathbf{h}_i$           | vetor de coeficientes da planta $i$ . . . . .  | 105 |
| $E[\cdot]$               | valor esperado . . . . .   | 126 |
| $\mathbf{R}_1$           | matriz de autocorrelação normalizada . . . . .   | 126 |
| $\mathbf{Q}$             | matriz de autovetores obtida através da autodecomposição de $\mathbf{R}$ . . . . .                   | 126 |
| $J(n)$                   | erro quadrático médio . . . . .  | 127 |
| $D(n)$                   | desvio quadrático médio . . . . .  | 127 |
| $D_N(n)$                 | desvio quadrático médio normalizado . . . . .  | 127 |
| $J_{\min}$               | erro quadrático médio mínimo . . . . .   | 127 |
| $J_{\text{ex}}$          | erro quadrático médio em excesso . . . . .   | 127 |
| $\mathbf{K}(n)$          | matriz de correlação do vetor de erro nos coeficientes . . . . .                                     | 127 |
| $\Lambda$                | matriz de autovalores de $\mathbf{R}$ . . . . .  | 128 |
| $\mathbf{v}'(n)$         | vetor de erro nos coeficientes nas coordenadas definidas pelos autovetores de $\mathbf{R}$ . . . . . | 128 |
| $\mathbf{K}'(n)$         | matriz de correlação do vetor $\mathbf{v}'(n)$ . . . . .   | 128 |
| $\mathbf{k}'(n)$         | vetor com os elementos da diagonal de $\mathbf{K}'(n)$ . . . . .                                     | 128 |

# Sumário

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>   | <b>27</b> |
| 1.1      | Filtragem adaptativa  | 27        |
| 1.2      | Filtragem adaptativa em sistemas esparsos                                       | 32        |
| 1.3      | Objetivos do trabalho   | 34        |
| 1.4      | Organização do trabalho   | 35        |
| <b>2</b> | <b>Fundamentação Teórica e Revisão Bibliográfica</b>                            | <b>37</b> |
| 2.1      | Métodos de otimização numérica aplicados à estimação                            | 37        |
| 2.1.1    | Métodos de otimização sem restrições  | 38        |
| 2.1.2    | Otimização com restrições baseada no método dos multiplicadores de Lagrange     | 39        |
| 2.2      | Estimação de sinais   | 42        |
| 2.2.1    | Estimação baseada na abordagem da minimização de uma função custo               | 42        |
| 2.2.2    | Estimação baseada na abordagem da teoria de conjuntos                           | 45        |
| 2.2.3    | O algoritmo NLMS como um problema de otimização com restrições                  | 48        |
| 2.3      | Métrica vetorial e esparsidade  | 50        |
| 2.4      | Revisão sobre algoritmos adaptativos para sistemas esparsos                     | 51        |
| 2.4.1    | Algoritmos de adaptação proporcional  | 52        |
| 2.4.2    | Algoritmos baseados na minimização de uma função custo com penalização de norma | 54        |
| 2.4.3    | Algoritmos baseados nas projeções em esferas de normas                          | 56        |
| 2.5      | Conclusões  | 59        |
| <b>3</b> | <b>Algoritmos Adaptativos com Restrição de Norma Propostos</b>                  | <b>61</b> |
| 3.1      | Formulação geral baseada em otimização com restrição de norma                   | 62        |
| 3.1.1    | Problema de otimização  | 62        |
| 3.1.2    | Atualização do vetor de coeficientes  | 64        |
| 3.2      | Algoritmos com restrição de norma $\ell_1$                                      | 65        |
| 3.2.1    | Algoritmo $\ell_1$ NC com parâmetro de limitação de norma variável              | 69        |
| 3.2.2    | Exemplos  | 71        |
| 3.3      | Algoritmos com restrição de norma $\ell_0$                                      | 74        |

|          |   |            |
|----------|---|------------|
| 3.3.1    | Considerações sobre o comportamento do algoritmo $\ell_0$ NC .....                                    | 80         |
| 3.3.2    | Algoritmo $\ell_0$ NC com parâmetro de limitação de norma variável .....                              | 81         |
| 3.3.3    | Algoritmo $\ell_0$ NC com aproximação suavizada .....   | 82         |
| 3.3.4    | Algoritmo com restrição de norma $\ell_0$ com coeficientes iguais a zero .....                        | 88         |
| 3.3.5    | Exemplos .....  | 92         |
| 3.4      | Considerações sobre os algoritmos propostos .....   | 99         |
| 3.4.1    | Sobre a complexidade computacional dos algoritmos propostos .....                                     | 100        |
| 3.4.2    | Sobre a convergência dos algoritmos $\ell_p$ NC .....   | 102        |
| 3.5      | Conclusões .....  | 103        |
| <b>4</b> | <b>Avaliação de Desempenho dos Algoritmos Propostos .....</b>   | <b>105</b> |
| 4.1      | Definição dos cenários para as simulações .....   | 105        |
| 4.2      | Resultados de simulações .....  | 108        |
| 4.2.1    | Algoritmos baseados na norma $\ell_1$ .....   | 108        |
| 4.2.2    | Variação abrupta do grau de esparsidade .....   | 111        |
| 4.2.3    | Identificação de plantas com ruído agregado .....   | 114        |
| 4.2.4    | Identificação de plantas realísticas .....  | 115        |
| 4.3      | Conclusões .....  | 118        |
| <b>5</b> | <b>Modelagem Estocástica dos Algoritmos Propostos .....</b>   | <b>125</b> |
| 5.1      | Modelagem estocástica do algoritmo NLMS .....   | 125        |
| 5.1.1    | Comportamento médio do vetor de coeficientes .....  | 125        |
| 5.1.2    | Curva de aprendizagem .....   | 126        |
| 5.1.2.1  | Curva de MSE .....  | 127        |
| 5.1.2.2  | Curva de MSD .....  | 128        |
| 5.1.3    | Matriz de correlação do vetor de erro nos coeficientes .....  | 129        |
| 5.1.4    | Análise em regime permanente .....  | 130        |
| 5.2      | Modelagem estocástica do algoritmo $\ell_p$ NC .....  | 131        |
| 5.2.1    | Comportamento médio do vetor de coeficientes .....  | 131        |
| 5.2.2    | Matriz de correlação do vetor de erro nos coeficientes .....  | 132        |
| 5.2.3    | Curva de aprendizagem .....   | 135        |
| 5.2.4    | Análise em regime permanente .....  | 136        |
| 5.3      | Modelagem estocástica dos algoritmos $\ell_1$ NC .....  | 136        |
| 5.3.1    | Considerações sobre $E[\mathbf{g}_{\ell_1}(n)]$ , $\mathbf{k}'_{sv}(n)$ e $\mathbf{k}'_{ss}(n)$ ..... | 136        |
| 5.3.2    | Valor esperado de $\lambda_{\ell_1}(n)$ .....   | 138        |
| 5.3.3    | Valor esperado de $\lambda_{v\ell_1}(n)$ .....  | 139        |
| 5.3.4    | Sumário do modelo do algoritmo $\ell_1$ NC .....  | 140        |



|          |   |            |
|----------|---|------------|
| 5.3.5    | Sumário do modelo do algoritmo V- $\ell_1$ NC   | 141        |
| 5.3.6    | Exemplos  | 142        |
| 5.4      | Modelagem estocástica do algoritmo $\ell_0$ NC  | 145        |
| 5.4.1    | Considerações sobre $E[\mathbf{g}_{\ell_0}(n+1)]$ , $\mathbf{k}'_{ss}(n)$ e $\mathbf{k}'_{sv}(n)$ | 145        |
| 5.4.2    | Valor esperado de $\lambda_{\ell_0}(n)$   | 146        |
| 5.4.3    | Sumário do modelo do algoritmo $\ell_0$ NC  | 147        |
| 5.4.4    | Valor esperado de $\lambda_{v\ell_0}(n)$  | 148        |
| 5.4.5    | Análise em regime permanente para V- $\ell_0$ NC  | 148        |
| 5.4.6    | Sumário do modelo do algoritmo V- $\ell_0$ NC   | 150        |
| 5.4.7    | Especificidades do modelo para o algoritmo VS- $\ell_0$ NC  | 150        |
| 5.4.8    | Exemplos  | 151        |
| 5.5      | Conclusões  | 154        |
| <b>6</b> | <b>Comentários e Conclusões Finais</b>  | <b>155</b> |
| 6.1      | Sumário   | 156        |
| 6.2      | Discussão dos Resultados  | 157        |
| 6.3      | Trabalhos publicados  | 160        |
| 6.4      | Sugestões para trabalhos futuros  | 160        |



# Capítulo 1

## Introdução

O fácil acesso às tecnologias de informação e comunicação que estamos experimentado desde o final do século passado caracteriza um período de grandes inovações chamado “era da informação” [1]. A automatização e informatização de atividades do dia-a-dia e o uso massivo de sistemas de comunicação multimídia vêm criando uma verdadeira revolução no cotidiano das pessoas. Fatores importantes, como a convergência da base tecnológica para a tecnologia digital; a dinâmica industrial da microeletrônica [2], que tem disponibilizado produtos cada vez mais potentes e, ao mesmo tempo, com menor custo; e o crescimento acelerado da internet e da infraestrutura de redes de comunicação, têm contribuído para essa rápida mudança [3]. Nesse contexto, o processamento digital de sinais torna-se fundamental. Desde a geração da informação até sua utilização pelo usuário final, o sinal digital, que representa a informação, passa por diversas etapas de processamento, comumente denominadas filtragem [4], [5], visando corrigir imperfeições ou alterar características dessa informação, disponibilizando-a ao usuário de acordo com requisitos especificados. Nem sempre é possível, no entanto, projetar um sistema de filtragem que atenda a todas as situações exigidas por determinada aplicação, demandando sistemas que possam se adaptar a cada condição. Tais sistemas, denominados filtros adaptativos [4]-[6], são requeridos em situações em que não é possível determinar completamente suas especificações ou elas são variantes no tempo. Um exemplo típico de aplicação que requer filtragem adaptativa são os sistemas de telefonia móvel celular. Nesses sistemas, os meios de transmissão, as distâncias entre os interlocutores e as antenas de recepção geralmente são variáveis a cada conexão, estando a conversação sujeita a diferentes condições de comunicação.

### 1.1 Filtragem adaptativa

Filtros adaptativos são sistemas cujos parâmetros são adaptados iterativamente por algoritmos matemáticos de estimação levando em conta o sinal que está sendo tratado [4]-[6]. Aplicações de processamento de sinais nas mais diversas áreas (telecomunicações, multimídia, engenharia biomédica, radar, sonar, sismologia, controle de processos, dentre outras) requerem filtragem adaptativa [7], podendo-se destacar sua utilização em cancelamento

de interferências, ruídos e eco; equalização de canais; tratamento de voz e imagens; arranjo de antenas de comunicação; predição e codificação da informação. A característica adaptativa desse tipo de sistema lhe confere a flexibilidade de se ajustar à situação de uso durante sua operação ou através de um período de treinamento, não sendo necessária a especificação completa do filtro. No entanto, alguns aspectos fundamentais definem o sistema de filtragem adaptativa, devendo-se considerá-los em sua implementação [8], a saber:

- o sinal que está sendo processado pelo filtro e o sistema no qual o filtro está inserido;
- a estrutura de filtragem, que define como o sinal de saída é calculado em função do sinal de entrada;
- as variáveis e os parâmetros que definem o filtro e que são modificados pelo processo adaptativo;
- o algoritmo adaptativo que descreve como as variáveis do filtro são ajustadas a cada iteração, sendo este último, geralmente, dependente dos demais aspectos.

Definidas tais características, a determinação do filtro de forma adaptativa deve levar em conta algum tipo de parâmetro de comparação que determine a qualidade do processo e o grau de proximidade à solução desejada. Dessa forma, um problema geral de adaptação, como ilustrado pela Figura 1, tem por objetivo obter um sinal  $y(n)$  na saída do filtro que se aproxime ao máximo de uma referência desejada  $d(n)$ , sendo o sinal de saída o parâmetro estimado, enquanto o filtro é seu estimador. Utilizando um sistema digital com resposta ao impulso finita (*finite impulse response* - FIR) como estrutura de filtragem adaptativa, o sinal de saída estará relacionado com o sinal de entrada  $x(n)$  através da combinação linear  $y(n) = \mathbf{w}^T(n) \mathbf{x}(n)$ , sendo  $\mathbf{w}(n)$  o vetor de coeficientes do filtro FIR, que será atualizado pelo algoritmo de adaptação a cada instante  $n$ , e  $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]$ , sendo  $N$  o número de elementos do vetor  $\mathbf{w}(n)$ . A diferença entre o sinal desejado e o sinal estimado produz um sinal de erro

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= d(n) - \mathbf{w}^T(n) \mathbf{x}(n) \end{aligned} \quad (1.1)$$

que será o indicador da eficácia do filtro, o qual é utilizado pelo algoritmo como parâmetro de comando. Tais estruturas, que caracterizam os “sistemas adaptativos lineares” [7], são bastante utilizadas em filtragem adapta-

tiva por apresentarem características que tornam mais simples seu projeto e implementação.

Um importante exemplo de arranjo utilizado em aplicações de filtragem adaptativa é o problema de identificação de sistemas (veja a Figura 2), onde  $\mathbf{w}_o$  representa o vetor de coeficientes do sistema a ser identificado. Para esse tipo de arranjo, o sinal de erro  $e(n)$  pode ser escrito como

$$e(n) = [\mathbf{w}_o - \mathbf{w}(n)]^T \mathbf{x}(n) + r(n). \quad (1.2)$$

Nessa configuração,  $r(n)$  é um ruído de medição que, usualmente, é uma variável independente de  $\mathbf{x}(n)$ . Uma aplicação típica de filtragem adaptativa em identificação de sistemas é o cancelamento de eco em sistemas de comunicação [7], na qual o filtro é adaptado para reproduzir a resposta ao impulso do caminho de eco de forma a cancelar o sinal indesejado de eco que retorna ao emissor, conforme ilustrado na Figura 3.

Definidos os aspectos relativos à aplicação de filtragem adaptativa, como os sinais envolvidos e o tipo de resposta ao impulso do sistema, e os aspectos construtivos, como a estrutura de filtragem e as variáveis a serem adaptadas, busca-se, então, definir o algoritmo adaptativo que será utilizado para se obter o melhor desempenho do processo de adaptação. Para tal, alguns aspectos de desempenho devem ser observados, tais como [4]:

- Velocidade de convergência. Um algoritmo possui melhor velocidade de convergência quando o número de iterações necessárias para se atingir o regime permanente é menor.

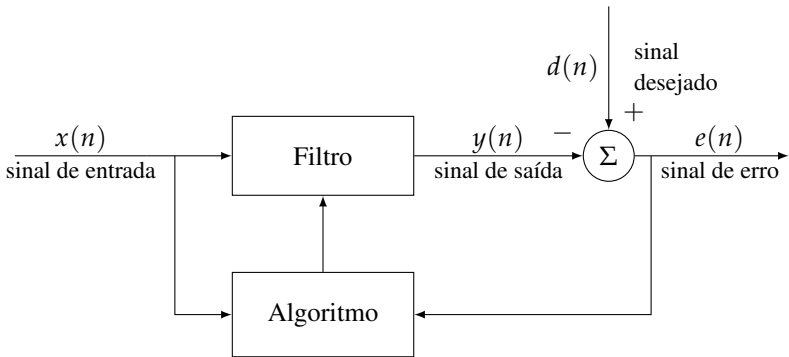


Figura 1: Diagrama representativo de um problema geral de filtragem adaptativa.

- Desajuste. É uma medida adimensional que define o quão próximo o vetor de coeficientes do filtro adaptativo está de seu valor ótimo após o processo de adaptação, isto é, em seu regime permanente.
- Complexidade computacional. Um algoritmo adaptativo desperta maior interesse em suas diversas aplicações quando possui menor complexidade computacional. Usualmente, essa complexidade está relacionada ao número de operações realizadas e à quantidade de memória requerida a cada iteração.

Outros aspectos também são geralmente observados no desempenho do algoritmo, tais como robustez, que é a insensibilidade a distúrbios, e a habilidade de rastreamento, caracterizada pela capacidade de realizar a adaptação em ambientes não estacionários. A avaliação dos principais aspectos de desempenho dos algoritmos normalmente se vale de critérios de medição que demonstrem sua proximidade ao resultado desejado, podendo-se destacar os critérios de erro quadrático médio (*mean-square error*) e de desalinhamento (*misalignment*) [9]. Neste trabalho, a avaliação do desempenho dos algoritmos é realizada através do desalinhamento normalizado (em dB) [10], defi-

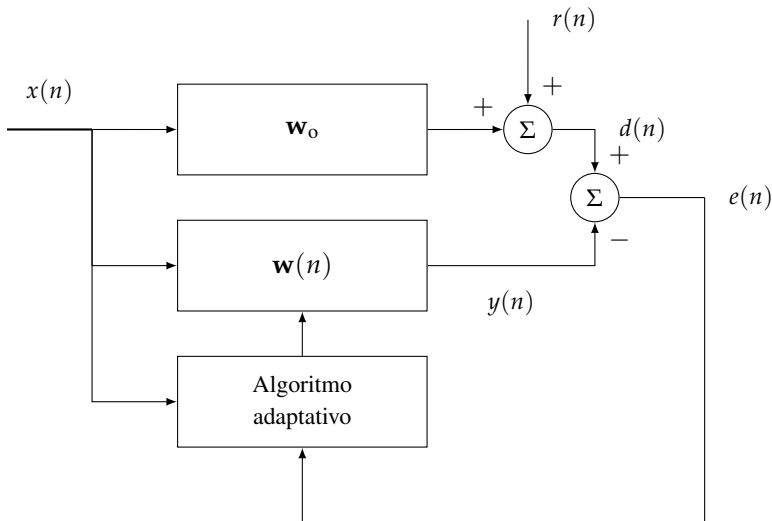


Figura 2: Diagrama de blocos de um problema de identificação de sistemas.

nido por

$$\kappa(n) = 10 \log_{10} \frac{\|\mathbf{w}_o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}_o\|_2^2}. \quad (1.3)$$

Nas simulações realizadas para obter os resultados apresentados neste trabalho, é obtido um valor médio de  $\mathbf{w}(n)$  através de simulações de Monte Carlo, com um determinado número de realizações independentes.

Dois importantes algoritmos se destacam em filtragem adaptativa, servindo de base para o desenvolvimento de vários outros algoritmos: o algoritmo LMS (*least-mean-square algorithm*) e o algoritmo NLMS (LMS normalizado). O algoritmo LMS foi proposto em 1959 por Widrow e Hoff [11] como uma alternativa simplificada aos métodos do gradiente baseados na teoria de estimação de Wiener. Aproximando o gradiente do erro quadrático médio pelo valor instantâneo de erro quadrático, a atualização do vetor de coeficientes do filtro adaptativo pelo algoritmo LMS é obtida por [7]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}(n) \quad (1.4)$$

onde  $\mu$  é o passo de adaptação. O algoritmo LMS tem a seu favor a reduzida complexidade computacional; no entanto, é penalizado em muitas aplicações

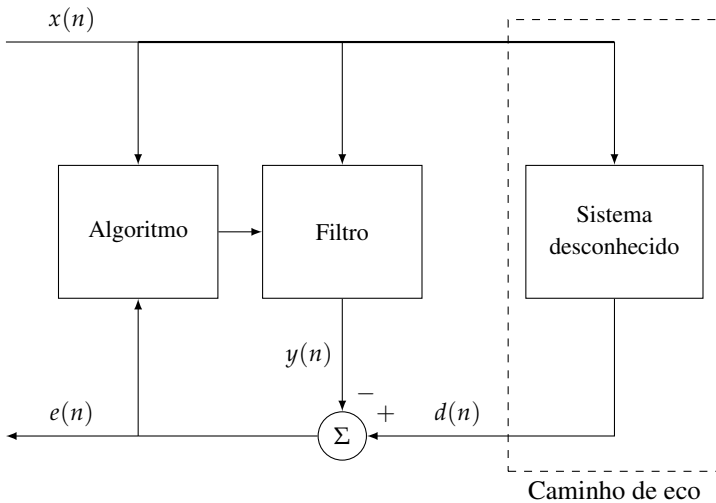


Figura 3: Diagrama em blocos de uma estrutura típica de um sistema de filtragem adaptativa para cancelamento de eco.

pela baixa velocidade de adaptação. Sendo a atualização do vetor de coeficientes proporcional à magnitude do vetor de entrada, valores elevados do sinal de entrada podem gerar o chamado “problema da amplificação do ruído do gradiente” [4], podendo levar o algoritmo à divergência. Esse problema impõe a redução do passo de adaptação, controlado por  $\mu$ , e, consequentemente, a redução da velocidade de convergência. Para minimizar a dependência do algoritmo LMS com respeito à magnitude do vetor de entrada, em 1967 uma proposta alternativa de algoritmo adaptativo é apresentada. Tal algoritmo, denominado LMS normalizado (*normalized LMS* - NLMS) [12], [13], é dado por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \mathbf{x}(n) \quad (1.5)$$

onde  $\mu$  é o parâmetro de controle de passo. Nota-se, a partir de (1.5), que a atualização do algoritmo NLMS é normalizada por  $\mathbf{x}^T(n)\mathbf{x}(n)$ , que representa o quadrado da norma euclidiana (descrita na Seção 2.3) do vetor de entrada. Essa normalização contorna o problema gerado pela variação de magnitude de  $\mathbf{x}(n)$ , não sendo mais necessária a redução do passo de adaptação para garantir a convergência. Dessa forma, o algoritmo NLMS apresenta desempenho geralmente superior ao algoritmo LMS, tornando-se um dos algoritmos mais utilizados em filtragem adaptativa em uma vasta gama de aplicações.

## 1.2 Filtragem adaptativa em sistemas esparsos

Diversas aplicações de filtragem adaptativa possuem sistemas com resposta ao impulso esparsa. Um sistema é dito esparsa quando grande parte da energia de sua resposta ao impulso está concentrada em uma pequena região, tendo assim uma parte significativa da resposta com valores iguais a zero ou próximos de zero [14], [15]. Exemplos típicos de sistemas com resposta ao impulso esparsa são os caminhos de eco em sistemas de comunicação [16]. A Figura 4 ilustra uma resposta ao impulso típica de um caminho de eco, obtida a partir da Recomendação ITU-T G.168, modelo #1 [17], na qual pode-se observar a concentração de valores diferentes de zero na região central da resposta. Os algoritmos adaptativos tradicionais (LMS e NLMS) não levam em consideração a esparsidade das respostas ao impulso da planta, realizando a adaptação de todos os elementos do vetor de coeficientes com o mesmo grau de importância. No entanto, estudos vêm mostrando que é possível explorar a característica de esparsidade da resposta ao impulso da planta para melhorar o desempenho do processo adaptativo.



Duas linhas de pesquisa têm sido desenvolvidas nesse sentido. Por um lado, verifica-se que priorizar os componentes mais significativos da planta em detrimento daqueles pouco significativos, atualizando cada elemento do vetor de coeficientes de forma proporcional a sua magnitude, pode aumentar de forma significativa a velocidade de convergência em sistemas esparsos. Essa é a hipótese que fundamenta os algoritmos da classe proporcional [16]. O algoritmo NLMS proporcional (PNLMS) [18] é o precursor dessa classe, sendo seguido por diversas outras estratégias que visam melhorar seu desempenho [19]-[25]. Em outra linha de pesquisa, constatando-se que alguns tipos de normas vetoriais se qualificam como uma métrica adequada na definição do grau de esparsidade de um vetor [15], utiliza-se tal informação em estratégias de adaptação para sistemas esparsos [26]. De forma geral, os algoritmos baseados em normas podem ser classificados em dois grupos, dependendo do tipo de abordagem escolhida para a incorporação da informação de norma do problema de otimização utilizado no processo de adaptação. No primeiro, a função representativa da norma é incluída diretamente na função custo a ser minimizada, alterando ou “penalizando” tal função com vistas a priorizar a esparsidade. Citando alguns representantes dessa linha, os algoritmos ZA-LMS (*zero attracting LMS*) e RZA-LMS (*reweighted zero attracting LMS*) [27] utilizam a norma  $\ell_1$  como penalização no processo adaptativo, criando uma tendência de “atração” para zero dos componentes do vetor. Os algoritmos que representam o segundo grupo, por outro lado, são baseados na estimação utilizando a teoria dos conjuntos [28] juntamente com o método de projeções, conforme discutido em [29]. Os algoritmos APL1 (*adaptive projection-based algorithm using  $\ell_1$  balls*) e APWL1 (*adaptive projection-based algorithm using weighted  $\ell_1$  balls*) [30], por exemplo, utili-

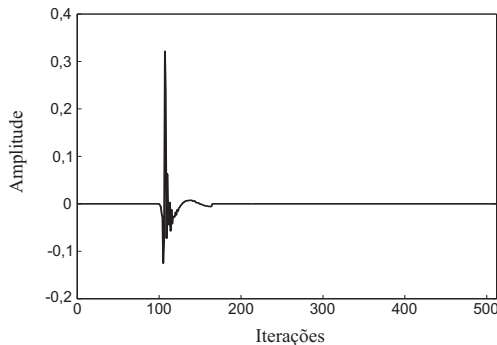


Figura 4: Resposta ao impulso típica de um caminho de eco em sistema de comunicação.

zam o método das projeções para projetar o vetor de coeficientes em uma determinada esfera de norma  $\ell_1$ , buscando uma solução que seja a mais esparsa possível. Algoritmos baseados em restrições de norma  $\ell_0$  também são encontrados na literatura [31], bem como algoritmos que combinam estratégias de algoritmos proporcionais e de restrição de norma [32], [33].

### 1.3 Objetivos do trabalho

Este trabalho de pesquisa tem como objetivo principal desenvolver novos algoritmos de filtragem adaptativa orientados à identificação de sistemas esparsos, buscando obter melhores desempenhos com baixa complexidade computacional e com reduzido número de parâmetros definidos pelo usuário. De forma semelhante aos algoritmos APL1 e APWL1, a proposta aqui apresentada é também fundamentada na teoria dos conjuntos e utiliza o método de projeções na atualização do vetor de coeficientes. No entanto, a restrição de erro *a posteriori* igual a zero (característica do algoritmo NLMS, como será visto no Capítulo 2) e a restrição que limita a solução a um determinado valor de norma são aplicadas simultaneamente ao processo de otimização, proporcionando uma solução que atenda a ambas as restrições a cada iteração. Tal abordagem difere da abordagem apresentada em [30], onde as restrições são tomadas separadamente, realizando as projeções de forma sequencial. A partir da estratégia adotada, é desenvolvida uma formulação geral que permita a utilização de métricas diferentes na definição do algoritmo, de acordo com o compromisso entre desempenho e complexidade desejado. Além da apresentação dos novos algoritmos, este trabalho visa também a realização de uma análise estocástica dos algoritmos propostos, buscando aprofundar o conhecimento acerca de seu desempenho, de suas condições de convergência e das limitações de seus parâmetros.

## 1.4 Organização do trabalho

Este trabalho está organizado como segue. O Capítulo 2 apresenta a fundamentação teórica sobre processos de estimação e filtragem adaptativa como também uma breve revisão sobre os principais algoritmos adaptativos para sistemas esparsos descritos na literatura. A Seção 2.1 apresenta métodos de otimização com e sem restrições bastante utilizados em estimação de sinais. Na Seção 2.2, técnicas de estimação aplicadas a processamento de sinais são discutidas. Considerando a utilização de normas vetoriais em diversos algoritmos voltados a sistemas esparsos, incluindo os algoritmos propostos neste trabalho, a Seção 2.3 apresenta uma discussão envolvendo as métricas vetoriais e esparsidade. Na Seção 2.4, uma revisão dos principais algoritmos direcionados à identificação de sistemas esparsos, abordados na literatura, encerra esse capítulo. O Capítulo 3 discute a teoria que fundamenta este trabalho de pesquisa, apresentando na Seção 3.1 uma formulação geral baseada na otimização com restrições de normas. Nas Seções 3.2 e 3.3, duas famílias de algoritmos são desenvolvidas a partir da formulação geral, dando origem aos algoritmos propostos nessa tese baseados na otimização com restrição de norma  $\ell_1$  e aos algoritmos com restrição de norma  $\ell_0$ . O Capítulo 4 mostra os resultados de simulação numérica visando à validação dos algoritmos propostos. No Capítulo 5, os algoritmos propostos são analisados através de modelos estocásticos. A Seção 5.1 revisita a modelagem estocástica do algoritmo NLMS, enquanto nas Seções 5.2, 5.3 e 5.4 são propostos modelos estocásticos para os novos algoritmos. Finalmente, no Capítulo 6, são apresentadas as conclusões e comentários finais do trabalho de pesquisa realizado, como também propostas para a continuação deste trabalho de pesquisa.



## Capítulo 2

### Fundamentação Teórica e Revisão Bibliográfica

Os algoritmos para filtragem adaptativa são propostos a partir de estratégias iterativas de estimação de sinais, considerando sistemas cujas características são desconhecidas *a priori*. O que pauta a definição da estratégia utilizada será sempre o desempenho que o algoritmo deve alcançar, considerando todos os aspectos de aplicação e construção que envolvem os sistemas de filtragem adaptativa. Nesse sentido, duas abordagens se destacam na definição das estratégias de estimação utilizadas no processo de adaptação. A primeira abordagem, mais tradicional, utiliza métodos que visam minimizar uma função que represente o erro entre o sinal estimado e o sinal desejado [4], enquanto a segunda baseia-se na teoria dos conjuntos [28], utilizando restrições para reduzir o conjunto de possíveis soluções para o estimador. Em ambas as abordagens, o processo iterativo é baseado em técnicas de otimização numérica, na qual se busca a melhor solução (ou solução ótima) dentre múltiplas ou infinitas possibilidades disponíveis. Dando apoio a tais abordagens de estimação, na seção seguinte, são apresentados métodos de otimização numérica utilizados em processos de estimação de sinais.

#### 2.1 Métodos de otimização numérica aplicados à estimação

Otimização numérica pode ser definida como o conjunto de técnicas aplicadas à determinação da melhor solução para um problema matemático que possui um conjunto de possíveis soluções, que normalmente não é unitário [34]. A otimização pode ser aplicada a sistemas físicos, econômicos, sociais ou quaisquer outros sistemas que contenham características a serem minimizadas ou maximizadas. Buscar a minimização de riscos em aplicações financeiras, a maximização da eficiência de uma fonte de energia, o menor índice de erros na transmissão de sinais, são todos exemplos de otimização. Para realizar a otimização de um processo, é necessário determinar um modelo que expresse adequadamente o sistema sob análise, bem como definir o objetivo a ser atingido com a otimização. Geralmente, o objetivo é representado por uma medida a ser minimizada ou maximizada em função das variáveis do sistema que são ajustadas no processo de otimização. Em muitos casos, a função que representa essa medida, denominada função objetivo ou função custo, está acompanhada de uma série de limites ou restrições impostas pela

natureza do sistema ou para orientar a solução do problema. De modo geral, o processo de otimização pode ser descrito como

$$\underset{\mathbf{x}}{\text{minimizar}} f(\mathbf{x}) \quad (2.1a)$$

$$\text{sujeito a } c_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, m \quad (2.1b)$$

isto é, minimizar a função custo  $f(\mathbf{x})$ , a partir de ajustes sobre o vetor  $\mathbf{x}$ , sujeito a  $m$  restrições  $c_i(\mathbf{x})$  [35], [36]. Para  $m = 0$ , o processo é denominado **otimização sem restrições**, enquanto para  $m > 0$ , o processo é denominado **otimização com restrições**.

### 2.1.1 Métodos de otimização sem restrições

Na otimização sem restrições, deseja-se determinar o mínimo de uma função custo  $f(\cdot) : \mathbb{R}^N \rightarrow \mathbb{R}$  sem que qualquer restrição seja imposta. Considerando uma função convexa<sup>1</sup> e continuamente diferenciável [35], [36], o processo de otimização

$$\underset{\mathbf{x}}{\text{minimizar}} f(\mathbf{x}) \quad (2.2)$$

possui uma única solução, em que

$$\nabla f(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} = \mathbf{x}^* \quad (2.3)$$

onde  $\nabla f(\mathbf{x})$  é o gradiente de  $f(\mathbf{x})$  em relação a  $\mathbf{x}$  e  $\mathbf{x}^*$  caracteriza o ponto ótimo que minimiza  $f(\mathbf{x})$ . Realizando uma aproximação iterativa na direção ao ponto de mínimo, a atualização do vetor a cada iteração pode ser representada pelo seguinte algoritmo:

$$\mathbf{x}(n+1) = \mathbf{x}(n) + \mu(n) \mathbf{d}(n) \quad (2.4)$$

onde  $\mu(n)$  é o passo de adaptação e  $\mathbf{d}(n)$ , um vetor determinando a direção da atualização no instante  $n$ . O objetivo do método é reduzir o valor da função custo a cada iteração, fazendo com que  $\mathbf{d}(n)$  seja direcionado de forma que

$$f[\mathbf{x}(n+1)] < f[\mathbf{x}(n)]. \quad (2.5)$$

---

<sup>1</sup>Uma função  $f(\cdot)$  é convexa se o seu domínio  $S$  é um conjunto convexo e se, para quaisquer dois pontos  $\mathbf{a}, \mathbf{b} \in S$ , a seguinte propriedade é satisfeita:  $f[\alpha \mathbf{a} + (1 - \alpha) \mathbf{b}] \leq \alpha f(\mathbf{a}) + (1 - \alpha)f(\mathbf{b})$ , para todo  $\alpha \in [0, 1]$ . Isso significa que uma reta ligando quaisquer dois pontos em uma função convexa não cortará o traçado da função.

Esses métodos são denominados métodos de descida (*descent methods*). Expandindo a função  $f(\mathbf{x})$  pela série de Taylor de primeira ordem no ponto  $\mathbf{x}(n)$ , pode-se demonstrar [36] que, para uma função convexa e continuamente diferenciável e considerando (2.5), a seguinte condição é sempre satisfeita:

$$\nabla f[\mathbf{x}(n)]^T \Delta \mathbf{x} < 0, \quad \Delta \mathbf{x} = \mathbf{x}(n+1) - \mathbf{x}(n). \quad (2.6)$$

Uma escolha que garante tal propriedade é  $\Delta \mathbf{x} = -\nabla f[\mathbf{x}(n)]$ . Assim, a partir de (2.4), o algoritmo pode ser escrito como

$$\mathbf{x}(n+1) = \mathbf{x}(n) - \mu(n) \nabla f[\mathbf{x}(n)]. \quad (2.7)$$

O algoritmo (2.7), que utiliza o gradiente como direção de descida, é denominado **algoritmo do gradiente**. Se  $f(\mathbf{x})$  é uma função quadrática, a direção contrária ao gradiente da função é a direção que proporciona maior variação de descida e, portanto, o algoritmo do gradiente é também o **algoritmo de descida mais íngreme** (*steepest descent algorithm*) [36].

Aliando-se o método do gradiente ao método de Newton-Raphson (método bastante utilizado para a determinação de raízes de uma função), é possível melhorar o desempenho do algoritmo que busca o valor ótimo, acrescentando informações de segunda ordem da função através de sua Hessiana  $\nabla' f[\mathbf{x}(n)]$  (matriz das derivadas segundas de  $f[\mathbf{x}(n)]$ ). Nesse caso, a equação de atualização obtida por esse método, denominado **método de Newton** [37], é dada por

$$\mathbf{x}(n+1) = \mathbf{x}(n) - \mu(n) \{ \nabla' f[\mathbf{x}(n)] \}^{-1} \nabla f[\mathbf{x}(n)] \quad (2.8)$$

onde o termo  $\{ \nabla' f[\mathbf{x}(n)] \}^{-1}$  modifica a direção de atualização levando em conta o grau de curvatura da função no ponto  $\mathbf{x}(n)$  em cada eixo.

### 2.1.2 Otimização com restrições baseada no método dos multiplicadores de Lagrange

Em muitos problemas de otimização, existe o interesse de se restringir o espaço de busca do valor ótimo, limitando o conjunto de possíveis soluções. Isto resulta em um problema de otimização com restrições que pode então ser formulado como

$$\underset{\mathbf{x}}{\text{minimizar}} \quad f(\mathbf{x}) \quad (2.9a)$$

$$\text{sujeito a } c_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, m \quad (2.9b)$$

sendo que as restrições  $c_i(\mathbf{x})$  podem ser desigualdades (menos restritivas) ou igualdades (mais restritivas). Uma das formas mais utilizadas para resolver o problema (2.9) é transformar o problema com restrições em um problema sem restrições equivalente [36], resolvendo-o, assim, através de métodos como o do gradiente. Tal transformação pode ser realizada pelo **método dos multiplicadores de Lagrange** [35], [36], conforme descrito a seguir.

Considerando, por exemplo, uma função custo  $f(\mathbf{x}) = x_0^2 + x_1^2$  com uma única restrição de igualdade, onde  $c(\mathbf{x}) = x_0 - 1 = 0$ , a partir de um ponto  $\mathbf{x}_1$  que não seja o ponto de mínimo e que atenda à restrição, define-se um vetor  $\mathbf{d}$  que aponte em uma direção de descida de  $f(\mathbf{x})$ , sem deixar de atender à restrição, como ilustrado na Figura 5. Para o ponto  $\mathbf{x}_1$ , considerando (2.6), devemos ter que

$$\nabla c(\mathbf{x}_1)^T \mathbf{d} = 0 \quad (\text{manutenção da restrição}) \quad \text{e} \quad (2.10a)$$

$$\nabla f(\mathbf{x}_1)^T \mathbf{d} < 0 \quad (\text{descida da função}). \quad (2.10b)$$

Sendo  $f(\mathbf{x})$  uma função convexa, para qualquer  $\mathbf{x}$  em que a condição (2.10a) é válida (a restrição de igualdade é mantida), também a condição (2.10b) será válida (ou seja, haverá uma direção de descida), exceto em um único ponto, aquele em que  $\nabla c(\mathbf{x})$  e  $\nabla f(\mathbf{x})$  são paralelos. Esse é o ponto de mínimo da função com restrição, onde  $\mathbf{x} = \mathbf{x}^*$  (veja Figura 5). Estendendo tal definição, pode-se demonstrar [35] que no ponto de mínimo de uma função custo convexa e continuamente diferenciável  $f(\mathbf{x})$ , com restrições convexas  $c_i(\mathbf{x}) \geq 0$  ( $i = 1, \dots, m$ ), o vetor gradiente da função custo é paralelo ao vetor soma dos gradientes das funções de restrição. Assim, existe um escalar  $\lambda_i$  para cada vetor gradiente das restrições que leva a

$$\nabla f(\mathbf{x}^*) = \sum_{i=1}^m \lambda_i \nabla c_i(\mathbf{x}^*). \quad (2.11)$$

Pode-se também reescrever (2.11) como

$$\nabla f(\mathbf{x}^*) - \sum_{i=1}^m \lambda_i \nabla c_i(\mathbf{x}^*) = \nabla \left[ f(\mathbf{x}^*) - \sum_{i=1}^m \lambda_i c_i(\mathbf{x}^*) \right] = \mathbf{0} \quad (2.12)$$

onde

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{i=1}^m \lambda_i c_i(\mathbf{x}) \quad (2.13)$$

é denominada **função de Lagrange**, que transforma um processo de otimização com restrições em um processo de otimização sem restrições.

A formulação consistente deste problema, no entanto, depende da de-



terminação dos valores escalares  $\lambda_i$ , denominados **multiplicadores de Lagrange**. A obtenção dos multiplicadores de Lagrange pode ser realizada através da **função dual de Lagrange** [35], [36] que, por definição, é o valor de mínimo (ínfimo) da função de Lagrange (2.13) em função do vetor dos multiplicadores  $\lambda_i$ . Assim,

$$q(\boldsymbol{\lambda}) := \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}). \quad (2.14)$$

Considerando que nenhum valor da função dual poderá ser maior do que a função de Lagrange, já que ela representa seu ínfimo, então a otimização da função dual

$$\underset{\boldsymbol{\lambda}}{\text{miximizar}} [q(\boldsymbol{\lambda})] \quad (2.15)$$

atenderá ao problema de otimização (2.9) e definirá o vetor  $\boldsymbol{\lambda}$  que caracteriza a função de Lagrange.

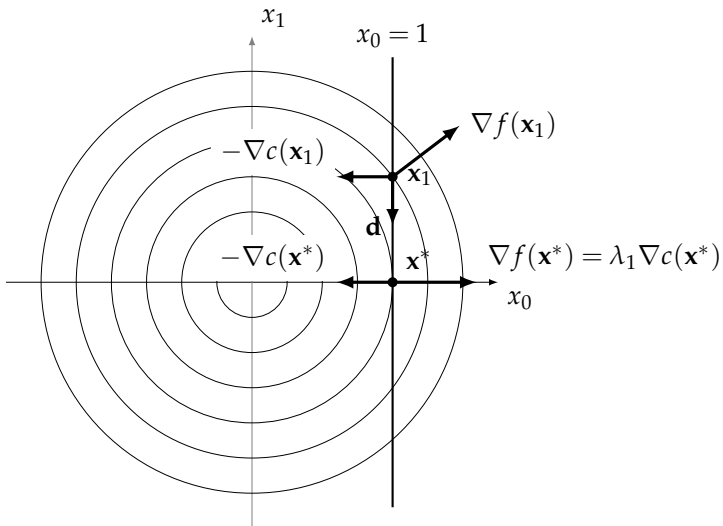


Figura 5: Representação de uma função com restrição, apresentando as direções dos vetores de gradiente e a direção de descida  $\mathbf{d}$ .

## 2.2 Estimação de sinais

Estimação é o processo de atribuição de valor a determinado parâmetro desconhecido a partir de observações relacionadas a esse parâmetro e/ou do conhecimento de informações prévias sobre o parâmetro que se quer estimar. Pode-se definir classicamente um estimador como

$$\hat{\theta} = f[x(0), x(1), \dots, x(N - 1)] \quad (2.16)$$

que determina o parâmetro estimado  $\hat{\theta}$  a partir de observações  $x(k)$  [38]. Existem várias técnicas de estimação com resultados muito eficientes, sendo a escolha do método dependente da aplicação, dos parâmetros a serem estimados, dos dados observados e das informações prévias disponíveis. No entanto, o melhor estimador escolhido nem sempre é realizável ou possui implementação muito complexa. Dessa forma, serão apresentadas a seguir duas abordagens distintas de implementação de estimadores muito utilizadas em processamento de sinais, mais especificamente em sistemas de filtragem adaptativa, que possuem grande aceitação por sua simplicidade e eficiência.

### 2.2.1 Estimação baseada na abordagem da minimização de uma função custo

O desempenho de um estimador pode ser medido utilizando-se algum critério de avaliação que dimensione o desvio entre o valor estimado e o valor desejado. Pode-se minimizar tal critério através da otimização de uma função que o caracteriza, denominada de função custo. Um critério bastante utilizado para a avaliação da eficácia de um estimador é a medida do erro quadrático médio (*mean-square error* - MSE) [37], [38], que determina o valor médio do quadrado do desvio entre o valor estimado e o valor desejado, definido pela seguinte função custo:

$$J(\hat{\theta}) = E[(\theta - \hat{\theta})^2]. \quad (2.17)$$

Considerando, no entanto, a dificuldade matemática e computacional de obtenção de estimadores ótimos através desse critério, é bastante comum a utilização de sistemas lineares de estimação. Dessa forma, o estimador dependerá apenas dos momentos de primeira e segunda ordens da função de densidade de probabilidade (PDF) da variável aleatória a ser estimada, incorporando, nesse caso, informações de média e variância do sinal de erro. Tal estratégia caracteriza, então, os estimadores lineares de mínimo erro quadrático médio [*linear minimum mean-square error (LMMSE) estimators*] [38].

Em processamento de sinais, considerando aplicações tipicamente estacionárias, os estimadores LMMSE são comumente denominados filtros de Wiener [4], [38]. Para sua determinação, considerando o problema geral de filtragem adaptativa ilustrado na Figura 1, no qual o sinal de erro é dado por (1.1), a função custo baseada no MSE é então

$$\begin{aligned} J &= E[e^2(n)] \\ &= E[d^2(n)] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w} \end{aligned} \quad (2.18)$$

onde  $\mathbf{p} = E[d(n)\mathbf{x}(n)]$  é o vetor de correlação cruzada entre o sinal desejado  $d(n)$  e o sinal de entrada  $\mathbf{x}(n)$  e  $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$  é a matriz de autocorrelação do sinal de entrada. Fazendo o gradiente de (2.18) igual a zero, conforme (2.3), o vetor de coeficientes que minimiza a função custo, definido como vetor de coeficientes ótimo (filtro de Wiener), é

$$\mathbf{w}^* = \mathbf{R}^{-1} \mathbf{p}. \quad (2.19)$$

Considerando que

$$\nabla J(n) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n) \quad (2.20)$$

pode-se, alternativamente, utilizar um método iterativo de otimização utilizando o gradiente (veja Seção 2.1.1) que, nesse caso, é também o método *steepest descent*, obtendo-se então o algoritmo de atualização dado por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu(n)[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]. \quad (2.21)$$

Recorrendo ao método de Newton e fazendo

$$\nabla' J(n) = 2\mathbf{R} \quad (2.22)$$

pode-se obter o vetor ótimo através da recursão

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \frac{1}{2} \mathbf{R}^{-1} [-2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n)] \\ &= \mathbf{R}^{-1} \mathbf{p} \\ &= \mathbf{w}^*. \end{aligned} \quad (2.23)$$

Observa-se que, para uma função  $J(n)$  quadrática, o método de Newton obtém o vetor ótimo em uma única iteração.

Geralmente, no entanto, as características estatísticas dos sinais envolvidos não são conhecidas *a priori*, inviabilizando a utilização de métodos baseados no erro quadrático médio. Como alternativa viável e de baixa complexidade, o algoritmo LMS, mencionado no Capítulo 1, tornou-se o mais po-

pular algoritmo baseado na abordagem da minimização de uma função custo. Pode-se interpretar sua formulação pela aplicação do método do gradiente estocástico [4], no qual, a cada iteração, os coeficientes são adaptados proporcionalmente a uma estimativa instantânea do gradiente da função custo baseada no MSE. Outra interpretação para o algoritmo LMS é a da minimização da função custo de erro quadrático instantâneo, isto é,

$$J(n) = e^2(n). \quad (2.24)$$

Minimizando iterativamente (2.24), utilizando o método do gradiente, o algoritmo LMS é dado por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}(n). \quad (2.25)$$

Contudo, a utilização de uma função custo instantânea, que se modifica a cada iteração, torna o processo de otimização muito ruidoso e lento, podendo tornar-se instável se o passo de adaptação  $\mu$  não for adequadamente escolhido. Por outro lado, conforme mencionado no Capítulo 1, o algoritmo NLMS se apresenta como uma solução para esse problema, minimizando a dependência do algoritmo à magnitude do vetor de entrada, observada em (2.25). Nesse sentido, o algoritmo NLMS pode ser interpretado como uma aplicação do método de Newton à função custo instantânea (2.24). Assim, ao invés de simplesmente apontar na direção de descida mais íngreme, como no caso do LMS, pode-se obter diretamente o erro mínimo instantâneo a cada iteração. Dessa forma, considerando (2.23) e acrescentando um termo de regularização  $\epsilon \mathbf{I}$  ( $0 < \epsilon \ll 1$ ) para evitar o mau condicionamento da matriz a ser invertida, a equação de atualização para o algoritmo NLMS pode então ser expressa como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) [\epsilon \mathbf{I} + \mathbf{x}(n) \mathbf{x}^T(n)]^{-1} \mathbf{x}(n). \quad (2.26)$$

Demonstra ainda [37] que (2.26) pode ser reescrita como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n)}{\epsilon + \mathbf{x}^T(n) \mathbf{x}(n)} \mathbf{x}(n) \quad (2.27)$$

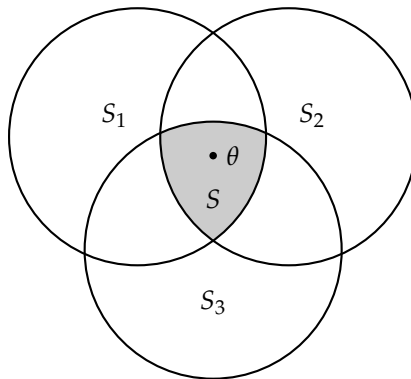
que é a expressão usual do algoritmo NLMS com termo de regularização.

## 2.2.2 Estimação baseada na abordagem da teoria de conjuntos

Com o objetivo de melhorar o desempenho dos algoritmos adaptativos, características específicas relacionadas à aplicação do sistema podem ser incluídas na forma de restrições impostas ao processo. Cada restrição apresentada delimita um conjunto de possíveis soluções  $S_i$  para o problema de estimação, sendo que a interseção de todos esses conjuntos definirá um subconjunto menor de soluções que atende a todas as restrições, dado por

$$S = \bigcap_{i=1}^m S_i \quad (2.28)$$

onde  $m$  é o número de restrições. O subconjunto  $S$  é denominado conjunto de soluções viáveis (*feasibility set*). Esse método de delimitação de soluções é denominado **estimação baseada na abordagem da teoria de conjuntos** (*set theoretic approach*) [28]. Se  $S \neq \emptyset$ , então a formulação do problema apresentada é **consistente**; se  $\theta \in S$ , a formulação é **realizável**; e se  $S = \theta$ , a formulação é **ideal** e apresenta uma única solução. A Figura 6 ilustra um caso no qual a formulação é consistente e realizável ( $\theta \in S$ ), porém não ideal.



$$S = S_1 \cap S_2 \cap S_3$$

Figura 6: Representação gráfica de estimação baseada na abordagem da teoria de conjuntos.

Um método clássico de estimação baseada na abordagem da teoria dos conjuntos é o método de projeções sobre conjuntos convexos (*projections onto convex sets* - POCS) [29]. A partir de um ponto qualquer no espaço vetorial, o método realiza a projeção desse ponto na superfície de soluções  $S_i$ , em seguida na superfície  $S_{i+1}$  e assim sucessivamente até atingir um ponto  $\hat{\theta} \in S$  que atenda todas as restrições. Esse método sequencial, ilustrado na Figura 7, foi proposto por Kaczmarz em 1937 [28], definindo o seguinte algoritmo:

$$\mathbf{w}_{n+1} = P_{i_n}(\mathbf{w}_n). \quad (2.29)$$

$P_{i_n}(\cdot)$  em (2.29) representa a função de projeção na superfície  $i$  na iteração  $n$ . Alternativamente, uma versão paralela do método de projeções, ilustrado na Figura 8, foi proposto por Cimmino em 1938 [28], sendo

$$\mathbf{w}_{n+1} = \frac{2}{m} \left[ \sum_{i=1}^m P_{i_n}(\mathbf{w}_n) - \mathbf{w}_n \right]. \quad (2.30)$$

Ambas as formulações podem ser ponderadas por fatores de relaxação, permitindo reescrever (2.29) como

$$\mathbf{w}_{n+1} = \mu_n P_{i_n}(\mathbf{w}_n) \quad (2.31)$$

e (2.30) como

$$\mathbf{w}_{n+1} = \frac{2\mu_n}{m} \left[ \sum_{i=1}^m P_{i_n}(\mathbf{w}_n) - \mathbf{w}_n \right]. \quad (2.32)$$

A determinação da projeção de um vetor em uma superfície convexa, resolvendo o problema (2.29), é obtida aplicando-se o princípio da menor distância [5], [29], dada por

$$d(\mathbf{w}_0, S) = \inf_{\mathbf{w}_1 \in S} \|\mathbf{w}_1 - \mathbf{w}_0\|_2 \quad (2.33)$$

onde  $\|\cdot\|_2$  é a norma euclidiana e  $\mathbf{w}_1$  é o ponto em  $S$  mais próximo de  $\mathbf{w}_0$ , ou seja, é a projeção de  $\mathbf{w}_0$  em  $S$ . A otimização da função convexa de norma  $\ell_2$  com restrição

$$\text{minimizar } \|\mathbf{w}_i - \mathbf{w}_0\|_2^2 \quad (2.34a)$$

$$\text{sujeito a } c(\mathbf{w}) \geq 0 \quad (2.34b)$$

determinará a projeção, sendo  $c(\mathbf{w})$  a função de restrição que gera o conjunto de soluções  $S$ .

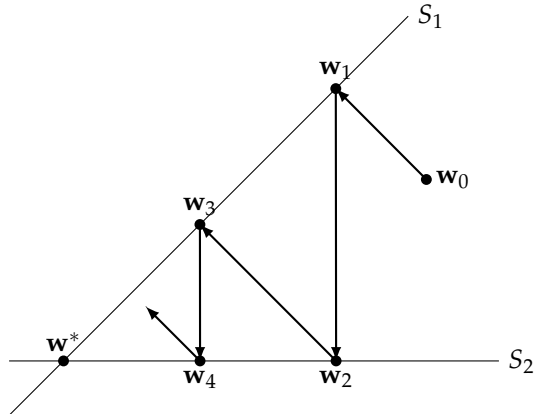


Figura 7: Princípio de operação do método sequencial de projeções sobre conjuntos convexos proposto por Kaczmarz [28].

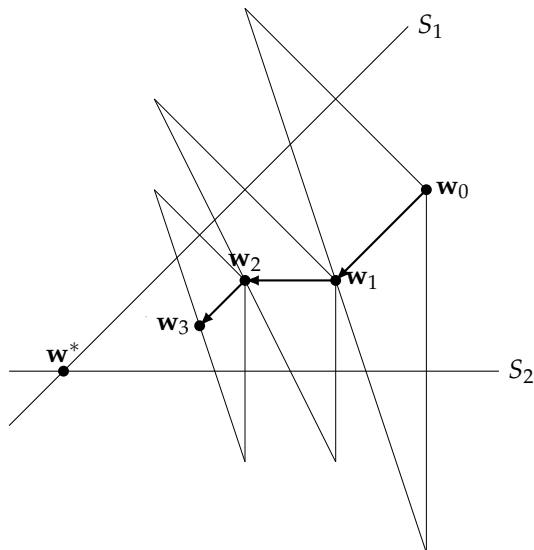


Figura 8: Princípio de operação do método paralelo de projeções sobre conjuntos convexos proposto por Cimmino [28].

### 2.2.3 O algoritmo NLMS como um problema de otimização com restrições

Restrições podem ser atualizadas à medida que o processo adaptativo evolui. Dessa forma, conjuntos diferentes de soluções  $S(n)$  são gerados a cada iteração para uma mesma função de restrição  $c[\mathbf{w}(n)]$ , obtendo-se infinitos conjuntos possíveis; se, no entanto, existir um conjunto de soluções viável  $S^*$ , gerado pela interseção de todos  $S(n)$ , é possível convergir para um ponto em  $S^*$  através de sucessivas projeções [29], como ilustrado na Figura 9. Em filtragem adaptativa, o algoritmo NLMS pode ser obtido a partir de um problema de otimização com restrições com solução obtida pelo método de projeções sucessivas [5]. Para esse algoritmo, a cada iteração, o estimador é adaptado de forma que o erro *a posteriori* seja igual a zero, isto é,

$$\varepsilon(n) = d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n) = 0 \quad (2.35)$$

que representa a restrição imposta ao processo de otimização. O algoritmo NLMS, então, é obtido a partir de

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (2.36a)$$

$$\text{sujeito a } d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n) = 0. \quad (2.36b)$$

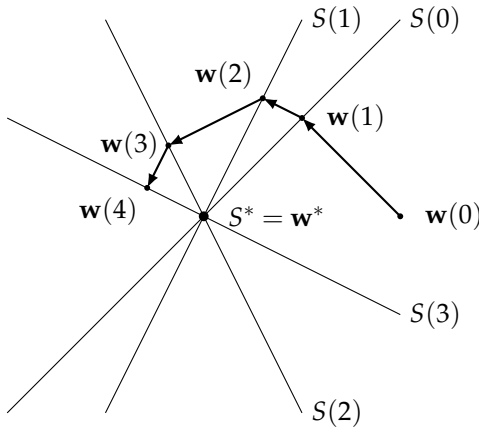


Figura 9: Processo iterativo de projeções sucessivas nas superfícies de soluções geradas por uma mesma restrição função de  $n$ .



Utilizando o método dos multiplicadores de Lagrange, transforma-se o problema com restrições em um problema sem restrições equivalente. A função de Lagrange é, então,

$$\mathcal{L}[\mathbf{w}(n+1)] = \frac{1}{2}[\mathbf{w}(n+1) - \mathbf{w}(n)]^T[\mathbf{w}(n+1) - \mathbf{w}(n)] + \lambda(n)[d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n)]. \quad (2.37)$$

Seu mínimo é determinado fazendo

$$\nabla \mathcal{L}[\mathbf{w}(n+1)] = \mathbf{w}(n+1) - \mathbf{w}(n) - \lambda(n)\mathbf{x}(n) = \mathbf{0} \quad (2.38)$$

obtendo-se a expressão recursiva

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda(n)\mathbf{x}(n). \quad (2.39)$$

O multiplicador de Lagrange  $\lambda(n)$  pode então ser obtido aplicando-se a função dual de Lagrange (2.15). Assim,

$$\lambda(n) = \frac{e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}. \quad (2.40)$$

Finalmente, a equação de atualização do algoritmo é dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n)}{\mathbf{x}^T(n)\mathbf{x}(n)}\mathbf{x}(n) \quad (2.41)$$

onde  $\mu$  é incluído para controlar o passo de adaptação.

É importante mencionar que o algoritmo NLMS pode ser definido como um caso especial do algoritmo de projeções afins [*affine projection algorithm* (APA)] [5], que utiliza, a cada iteração, um grupo de  $Q$  hiperplanos de erros *a posteriori* iguais a zero definidos pelo reuso de dados de entrada de iterações anteriores. O algoritmo APA atende ao seguinte problema de otimização:

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (2.42a)$$

$$\text{sujeito a } \mathbf{d}(n) - \mathbf{X}^T(n)\mathbf{w}(n+1) = 0 \quad (2.42b)$$

sendo  $\mathbf{d}(n) = [d(n) \ d(n-1) \ \dots \ d(n-Q+1)]^T$  e  $\mathbf{X}(n) = [\mathbf{x}(n) \ \mathbf{x}(n-1) \ \dots \ \mathbf{x}(n-Q+1)]^T$ . O algoritmo NLMS pode ser obtido a partir do algoritmo APA fazendo  $Q = 1$ .

### 2.3 Métrica vetorial e esparsidade

Existem diversas maneiras de medir a esparsidade de um vetor [15]. Considerando a definição mais estrita de esparsidade, relacionada à quantidade de coeficientes exatamente iguais a zero, a norma  $\ell_0$  do vetor, dada por

$$\|\mathbf{w}\|_0 = \sum_{k=0}^{N-1} f(w_k) \quad (2.43)$$

com

$$f(w_k) = \begin{cases} 1, & w_k \neq 0 \\ 0, & w_k = 0 \end{cases} \quad (2.44)$$

apresenta-se como a forma mais natural para efetuar tal medição [15], [23], [39]. No entanto, devido principalmente à presença de ruído e ao comportamento não contínuo da função de norma  $\ell_0$ , essa métrica não é eficaz na maioria das aplicações em processamento de sinais. Assim, a esparsidade de uma planta passa a ser definida pela concentração de energia em uma pequena região de sua resposta ao impulso. Então, a medida da esparsidade deve considerar outras métricas vetoriais em sua determinação. Considerando a definição geral de norma vetorial [39], na qual a norma  $\ell_p$  de um vetor  $\mathbf{w} \in \mathbb{R}^N$  é dada por

$$\|\mathbf{w}\|_p = \left[ \sum_{k=0}^{N-1} |w_k|^p \right]^{1/p}, \quad p \geq 1 \quad (2.45)$$

duas importantes métricas são muito úteis na avaliação da esparsidade de um vetor. A norma  $\ell_2$  (norma euclidiana), dada por

$$\|\mathbf{w}(n)\|_2 = \sqrt{\sum_{k=0}^{N-1} |w_k(n)|^2} \quad (2.46)$$

que representa a medida de comprimento de um vetor (menor distância entre dois pontos) e a norma  $\ell_1$ , dada por

$$\|\mathbf{w}(n)\|_1 = \sum_{k=0}^{N-1} |w_k(n)| \quad (2.47)$$

identificada pela soma das projeções do vetor nos eixos ortogonais. Pode-se demonstrar [40] que

$$\|\mathbf{w}\|_1 \geq \|\mathbf{w}\|_2 \quad (2.48)$$

para qualquer  $\mathbf{w}$ . A relação entre essas normas pode definir uma medida da esparsidade, levando em conta os seguintes aspectos:

- i) se o vetor está exatamente sobre um dos eixos ortogonais, apresentando assim seu maior grau de esparsidade, então  $\|\mathbf{w}\|_1 = \|\mathbf{w}\|_2$ ;
- ii) se todos os coeficientes do vetor tem exatamente a mesma magnitude, o vetor apresenta a menor esparsidade possível e  $\|\mathbf{w}\|_1 = \sqrt{N}\|\mathbf{w}\|_2$  (maior diferença entre as duas normas).

Então, uma expressão bastante utilizada para quantificar o grau de esparsidade de um vetor é dada por [14], [10]

$$\mathcal{S}(\mathbf{w}) = \frac{N}{N - \sqrt{N}} \left( 1 - \frac{\|\mathbf{w}\|_1}{\sqrt{N}\|\mathbf{w}\|_2} \right) \quad (2.49)$$

com  $0 \leq \mathcal{S}(\mathbf{w}) \leq 1$ .

Isoladamente, apenas a norma  $\ell_0$  determina o grau de esparsidade de um vetor, mesmo assim, apenas para a definição de esparsidade mais estrita. Por outro lado, associada a outras informações, a norma  $\ell_1$  se mostra muito eficaz na determinação de esparsidade. Se dois vetores diferentes possuem o mesmo comprimento (mesma norma  $\ell_2$ ), por exemplo, é mais esparsa aquele que apresentar a menor norma  $\ell_1$ , conforme (2.49). Processos de otimização em sistemas esparsos utilizando norma  $\ell_1$  como parâmetro de medida e adaptação têm sido foco em muitos estudos [41], [42]. Mais recentemente, uma teoria denominada *compressive sensing* [39], [43], [44], definida basicamente como a aquisição de um sinal esparsos de forma comprimida, tem explorado de forma extensiva os algoritmos baseados na norma  $\ell_1$ .

## 2.4 Revisão sobre algoritmos adaptativos para sistemas esparsos

Os algoritmos para filtragem adaptativa convencionais, tais como o LMS e o NLMS, não consideram em seu processo de adaptação qualquer informação que identifique o grau de esparsidade do sistema que está sendo adaptado. Verificando que essa característica pode ser explorada para melhorar o desempenho dos algoritmos durante a adaptação, diversas estratégias vêm sendo extensivamente discutidas na literatura.

### 2.4.1 Algoritmos de adaptação proporcional

Os algoritmos com adaptação proporcional têm como característica fundamental a ponderação individual dos elementos do vetor de coeficientes que está sendo adaptado. Um dos primeiros algoritmos propostos com tal finalidade é o algoritmo NLMS proporcional, PNLMS (*proportionate NLMS*), introduzido por Duttweiler em 2000 [18]. Utilizando ganhos individuais, o algoritmo PNLMS atualiza cada coeficiente do filtro adaptativo de forma proporcional à sua própria magnitude, isto é, aplicando maior peso na atualização dos elementos que têm maior magnitude. Essa redistribuição de pesos, transferindo “energia” dos coeficientes de menor magnitude para aqueles de maior magnitude é proporcionada pela inserção, ao algoritmo NLMS, de uma matriz de ganhos  $\mathbf{G}(n)$ , sendo a atualização do vetor de coeficientes dada por [9]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n)e(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} \mathbf{x}(n) \quad (2.50)$$

onde  $\mu$  é o parâmetro de controle de passo. A matriz de ganhos é dada por

$$\mathbf{G}(n) = \text{diag}[g_1(n), g_2(n), \dots, g_N(n)] \quad (2.51)$$

com

$$g_i(n) = \frac{\gamma_i(n)}{\sum_{k=0}^{N-1} \gamma_k(n)} \quad (2.52)$$

sendo  $\gamma_i(n)$  uma função de proporcionalidade definida para cada coeficiente. Como (2.52) é normalizado por  $\sum_{k=0}^{N-1} \gamma_k(n)$ , a soma de todos os ganhos é igual a 1. Para evitar que a atualização de algum elemento do vetor de coeficientes seja “congelada” quando esse elemento é igual a zero [24], define-se um limiar mínimo de ganho a cada iteração, denominado fator de ativação. Portanto,

$$\gamma_{\min}(n) = \varrho \max[\delta_p, |w_0(n)|, |w_1(n)|, \dots, |w_{N-1}(n)|] \quad (2.53)$$

onde  $\varrho$  é um parâmetro de proporcionalidade e  $\delta_p$  um parâmetro de inicialização. Assim, a função de proporcionalidade de cada coeficiente é dada por

$$\gamma_i(n) = \max[\gamma_{\min}(n), |w_i(n)|]. \quad (2.54)$$

O algoritmo PNLMS é bem eficiente na parte inicial da convergência, mas tem seu desempenho deteriorado à medida que o filtro converge para a solução desejada [24]. Além disso, o algoritmo apresenta desempenho ruim quando aplicado a sistemas com baixa esparsidade (plantas dispersi-

vas), com resultados piores do que os do algoritmo NLMS. Para reduzir tais deficiências, algoritmos alternativos têm sido propostos. O algoritmo PNLMS++ [19], por exemplo, se caracteriza por comutar entre os algoritmos NLMS e PNLMS em esquemas de comutação que dependem do tipo de aplicação. Dessa forma, o algoritmo PNLMS++ se vale dos benefícios de cada um dos algoritmos onde eles são mais eficientes.

O algoritmo PNLMS melhorado (*improved* PNLMS - IPNLMS) [20], por sua vez, mescla os algoritmos NLMS e PNLMS através de um parâmetro de controle  $\alpha$ , sendo o novo ganho para cada coeficiente dado por

$$g_i(n) = \frac{1 - \alpha}{2N} + (1 + \alpha) \frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1}. \quad (2.55)$$

Dessa forma, quando  $\alpha = 1$ , os ganhos  $k_i(n)$  mantêm apenas a parcela proporcional, tornando o algoritmo semelhante ao PNLMS. Já quando  $\alpha = -1$ , os ganhos passam a ser uniformes, fazendo com que o algoritmo se assemelhe ao algoritmo NLMS. Valores intermediários de  $\alpha$  determinam uma mistura dos dois algoritmos, sendo típicos os valores de  $\alpha$  iguais a 0 ou  $-0,5$ . O algoritmo IPNLMS tem demonstrado melhor desempenho do que o NLMS e o PNLMS padrão em diversas condições de esparsidade.

De forma geral, as matrizes de ganho que caracterizam os algoritmos PNLMS e seus derivados aqui citados são obtidas de forma *ad-hoc*, geralmente baseadas em critérios intuitivos ou experimentais. Procurando aproximar a abordagem proporcional de critérios mais clássicos de otimização, é proposta em [45] uma estratégia baseada no método de descida mais íngreme (*steepest descent*), denominado algoritmo PNLMS com lei- $\mu$  ( $\mu$ -law PNLMS - MPNLMS). Para este algoritmo, a função de proporcionalidade definida para cada coeficiente é dada por

$$\gamma_i(n) = \max\{\gamma_{\min}(n), F[|w_i(n)|]\} \quad (2.56)$$

com

$$\gamma_{\min}(n) = \varrho \max\{\delta_p, F[|w_0(n)|], F[|w_1(n)|], \dots, F[|w_{N-1}(n)|]\} \quad (2.57)$$

e

$$F[|w_i(n)|] = \frac{\ln[1 + |w_i(n)|/\epsilon]}{\ln(1 + 1/\epsilon)} \quad (2.58)$$

onde  $\epsilon$  é um número pequeno positivo definido pelo usuário que delimita uma faixa de proximidade dos coeficientes ao seu valor ótimo após um certo número de iterações. Considerando (2.58), verifica-se que a função de proporcionalidade do algoritmo MPNLMS assemelha-se ao clássico critério de

codificação utilizado em telecomunicações definido pela lei- $\mu$  [17], o que sugere o nome do algoritmo. Apesar de demonstrar desempenho superior a outros algoritmos da classe proporcional [9], o algoritmo MPNLMS apresenta complexidade computacional bastante elevada, o que pode desqualificá-lo como opção em diversas aplicações. Dessa forma, uma simplificação por segmentos de reta é proposta em [45], de forma a definir um novo algoritmo, agora denominado SPNLMS (*segment PNLMS*), mais competitivo. Para o algoritmo SPNLMS

$$F[|w_i(n)|] = \begin{cases} 600|w_i(n)|, & |w_i(n)| < 0,005 \\ 3, & \text{demais.} \end{cases} \quad (2.59)$$

Diversos outros algoritmos de adaptação proporcional têm sido propostos e discutidos na literatura da área, visando aprimoramentos na forma de cálculo dos pesos e ganhos individuais, podendo-se citar, o algoritmo IPNLMS melhorado (*improved IPNLMS - IIPNLMS*) [22], o algoritmo IPNLMS- $\ell_0$  [23], com função de ganho regida pela norma  $\ell_0$ , o algoritmo com fatores de ativação individuais, denominado IAF-PNLMS (*individual-activation-factor PNLMS algorithm*) [24] e o algoritmo com fator de proporcionalidade  $\rho(n)$  em função do grau de esparsidade, SC-PNLMS (*sparseness-controlled PNLMS*) [25], dentre outros, citando apenas os algoritmos baseados no NLMS<sup>2</sup>. Sendo essa uma área de pesquisa bastante rica e vasta, uma série de variações dos algoritmos proporcionais continua sendo discutida na literatura, buscando-se melhorar seu desempenho e/ou reduzir sua complexidade computacional, como pode-se verificar em [46]-[49].

#### 2.4.2 Algoritmos baseados na minimização de uma função custo com penalização de norma

Como discutido na Seção 2.3, podem-se utilizar determinadas métricas vetoriais para a determinação do grau de esparsidade de um vetor. A norma  $\ell_1$ , em especial, vem sendo utilizada em muitas aplicações que manipulam vetores esparsos. Baseado no método do gradiente de minimização de uma função custo, Chen *et al* [27] desenvolveram o algoritmo ZA-LMS (*zero attracting LMS*) em que a norma  $\ell_1$  do vetor de coeficientes é incorporada à função custo de erro quadrático instantâneo como uma forma de “penalização”, originando uma nova função a ser minimizada. Assim, a função custo

---

<sup>2</sup>Uma série de outros algoritmos proporcionais são baseados ainda nos algoritmos de projeções afins (APA) e de mínimos quadrados recursivo [*recursive least-square (RLS) algorithms*].

a ser minimizada pelo algoritmo ZA-LMS será

$$L(n) = \frac{1}{2}e^2(n) + \gamma \|\mathbf{w}(n)\|_1. \quad (2.60)$$

A penalização de norma  $\ell_1$  enfatiza a função custo nas proximidades de  $\mathbf{w} = \mathbf{0}$  (onde  $\|\mathbf{w}(n)\|_1 = 0$ ), determinando uma “atração” para zero no processo de descida, justificando a denominação do algoritmo. A atualização do vetor de coeficientes é então dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) - \rho \operatorname{sgn}[\mathbf{w}(n)] \quad (2.61)$$

sendo  $\operatorname{sgn}[\mathbf{w}(n)]$  o vetor de sinais de  $\mathbf{w}(n)$ ,  $\mu$  o passo de adaptação e  $\rho$  um parâmetro que controla o grau de atração da norma  $\ell_1$ . Esse algoritmo apresenta melhor desempenho do que o algoritmo LMS para adaptação de sistemas esparsos; no entanto, mostra-se bastante sensível ao valor do parâmetro  $\rho$ , podendo levar à polarização da solução. É também apresentado em [27] um algoritmo alternativo ao algoritmo ZA-LMS, denominado ZA-LMS ponderado (*reweighted zero-attracting LMS* - RZA-LMS). O algoritmo RZA-LMS minimiza a seguinte função custo:

$$L(n) = \frac{1}{2}e^2(n) + \gamma' \sum_{k=0}^{N-1} \log[1 + |w_k(n)|/\varepsilon'] \quad (2.62)$$

onde  $\gamma'$  e  $\varepsilon'$  são parâmetros de controle. A aplicação da norma  $\ell_1$  em (2.60) pondera a atualização (2.61) de maneira uniforme. Já a função custo (2.62) gera um algoritmo com um termo de penalização inversamente proporcional à magnitude de cada componente. Assim,

$$w_k(n+1) = w_k(n) + \mu e(n)x(n-k) - \rho \frac{\operatorname{sgn}[w_k(n)]}{1 + \varepsilon|w_k(n)|} \quad (2.63)$$

com  $\rho = \frac{\mu\gamma'}{\varepsilon'}$  e  $\varepsilon = \frac{1}{\varepsilon'}$ . Tal ponderação melhora o desempenho do algoritmo, mas lhe adiciona um significativo custo computacional em função da inclusão de uma operação de divisão. Os autores citam ainda a aplicação da penalização também ao algoritmo NLMS (ZA-NLMS e RZA-NLMS), buscando as vantagens que esse algoritmo pode oferecer à abordagem. Seguindo essa mesma estratégia, Gu *et al* [31] apresentam o algoritmo  $\ell_0$ -LMS, que inclui uma penalização de norma  $\ell_0$  à função custo utilizando a seguinte aproximação:

$$f[w_i(n)] = 1 - e^{-\beta|w_i(n)|} \quad (2.64)$$

com  $\beta \geq 1$ . Assim, a função custo a ser minimizada agora é

$$L(n) = \frac{1}{2}e^2(n) + \gamma \sum_{k=0}^{N-1} [1 - e^{-\beta|w_k(n+1)|}] \quad (2.65)$$

sendo a atualização de cada coeficiente dada por

$$w_k(n+1) = w_k(n) + \mu e(n)x(n-k) - \kappa\beta \operatorname{sgn}[w_k(n)]e^{-\beta|w_k(n+1)|}. \quad (2.66)$$

Alternativamente, é proposta uma aproximação à norma  $\ell_0$  utilizando a série de Taylor de primeira ordem para a função exponencial<sup>3</sup>. Uma versão do algoritmo baseada no algoritmo NLMS, denominada  $\ell_0$ -NLMS, é também discutida em [31]. A equação de atualização para esse algoritmo é obtida a partir da normalização do vetor de entrada. Assim,

$$w_k(n+1) = w_k(n) + \mu \frac{e(n)x(n-k)}{\mathbf{x}^T(n)\mathbf{x}(n)} - \kappa\beta \operatorname{sgn}[w_k(n)]e^{-\beta|w_k(n+1)|}. \quad (2.67)$$

Pesquisas recentes têm apresentado variações dos algoritmos com penalização de norma, conforme pode-se constatar em [50]-[53].

### 2.4.3 Algoritmos baseados nas projeções em esferas de normas

Baseados na abordagem da teoria dos conjuntos, os algoritmos pertencentes a esse grupo aplicam a norma vetorial como uma nova restrição que limita o conjunto de soluções. Utilizando tal estratégia, Kopsinis *et al* [30] apresentam um algoritmo baseado em projeções, denominado APL1 (*adaptive projection-based algorithm using  $\ell_1$  balls*), no qual a função de norma  $\ell_1$  gera um volume de restrição fixo onde o vetor de coeficientes é projetado a cada iteração. A abordagem utilizada desenvolve-se em duas etapas. A primeira é baseada na técnica de projeções paralelas de Cimmino, conforme (2.32), reutilizando superfícies de erro *a posteriori* de iterações anteriores. Tal projeção considera uma certa faixa de tolerância para cada superfície, sendo estas denominadas *hyperslabs*. Cada *hyperslab* que compõe o conjunto de soluções desta técnica é caracterizado por

$$S_{\bar{\epsilon}(k)}(n) = \{\mathbf{w}(n+1) \in \mathbb{R}^N : |d(n-k) - \mathbf{w}^T(n+1)\mathbf{x}(n-k)| \leq \phi\} \quad (2.68)$$

---

<sup>3</sup>A aproximação da norma  $\ell_0$  pela série de Taylor é detalhada na Seção 3.3.



sendo  $k = 0, 1, \dots, Q - 1$ , com  $Q$  representando o número de *hyperslabs* e  $\phi$  definindo a espessura de cada *hyperslab*. Tal técnica é apresentada em [54] como alternativa ao algoritmo de projeções afins (APA), com menor custo computacional e menor sensibilidade ao ruído. A segunda etapa, por sua vez, caracteriza-se por uma segunda projeção do vetor de coeficientes, agora na esfera delimitada pela norma considerada (*norm ball*), definida por

$$S_{\ell_p}(n) = \{\mathbf{w}(n+1) \in \mathbb{R}^N : \|\mathbf{w}(n+1)\|_p \leq \delta_{\ell_p}\} \quad (2.69)$$

com  $\|\mathbf{w}(n+1)\|_p$  representando a norma  $\ell_p$  de  $\mathbf{w}(n+1)$  e  $\delta_{\ell_p}$  o limite de norma definido pelo usuário. Essa estratégia de projeções sucessivas define a seguinte equação de atualização:

$$\mathbf{w}(n+1) = P_{S_{\ell_1}} \left[ \mathbf{w}(n) + \mu(n) \left\{ \sum_{k=0}^{Q-1} q_k P_{S_{\bar{\varepsilon}(k)}} [\mathbf{w}(n)] - \mathbf{w}(n) \right\} \right] \quad (2.70)$$

com  $P_{S_{\bar{\varepsilon}(k)}}(\cdot)$  caracterizando um mapeamento de projeção no  $k$ -ésimo *hyperslab*  $S_{\bar{\varepsilon}(k)}(n)$  (com peso  $0 < q_k \leq 1$  definido pelo usuário, sendo  $\sum_{k=0}^{Q-1} q_k = 1$ ) e  $P_{S_{\ell_1}}(\cdot)$ , o subsequente mapeamento de projeção em  $S_{\ell_1}$ . Uma detalhada descrição desses mapeamentos, bem como da escolha de  $\mu(n)$  é dada em [30]. No exemplo ilustrado pela Figura 10, observa-se a seqüência de projeções resultante de uma iteração do algoritmo (2.70), considerando  $Q = 1$ .

Uma variação do algoritmo APL1, denominado algoritmo APL1 com norma ponderada (*adaptive projection-based algorithm using weighted  $\ell_1$  balls* - APWL1), é também introduzido em [30]. O algoritmo APWL1 é baseado na norma  $\ell_1$  ponderada descrita por Candès em [55], expressa por

$$\sum_{i=1}^m p_i |w_i| \quad (2.71)$$

onde  $p_i$  é um peso individual dado ao elemento  $w_i$  do vetor de coeficientes. No algoritmo APWL1, os pesos individuais são proporcionais ao inverso do valor dos elementos do vetor. Assim,

$$p_i(n+1) = \frac{1}{|w_i(n)|}. \quad (2.72)$$

A ponderação (2.71), ilustrada pelo diamante em traço contínuo na Figura 11, propicia um melhor desempenho do algoritmo; no entanto, uma maior complexidade computacional é obtida devido à operação de divisão considerada em (2.72).

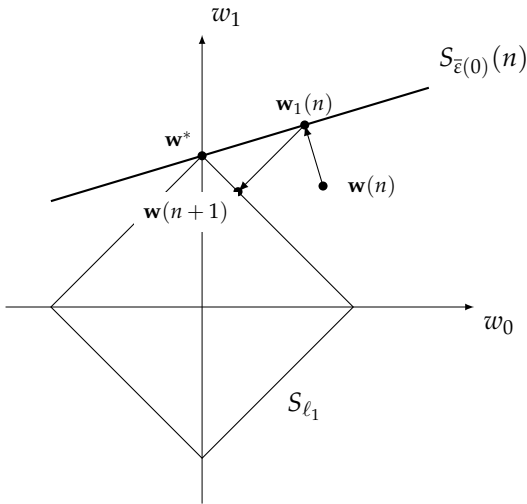


Figura 10: Princípio de operação do algoritmo APL1.

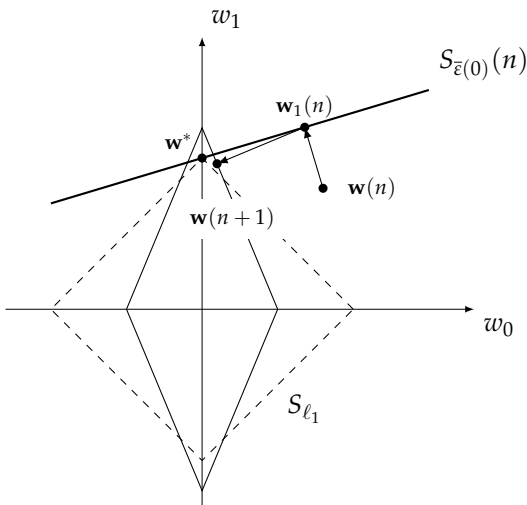


Figura 11: Princípio de operação do algoritmo APWL1, apresentando modificação da esfera de norma  $\ell_1$  através de ponderação (2.71).

## 2.5 Conclusões

Neste capítulo foram discutidos alguns tópicos importantes ao alicerçamento da proposta apresentada neste trabalho de pesquisa. As teorias de otimização e estimação que fundamentam a formulação dos algoritmos aqui descritos são apresentadas, sendo parte essencial ao desenvolvimento das novas propostas, que serão abordadas no Capítulo 3. Também foram discutidos alguns algoritmos orientados a sistemas esparsos oriundos da literatura, considerando três abordagens distintas, a saber, os algoritmos de adaptação proporcional, os algoritmos baseados na minimização de uma função custo com penalização de norma e os algoritmos baseados nas projeções em esferas de normas.



## Capítulo 3

# Algoritmos Adaptativos com Restrição de Norma Propostos

No capítulo anterior, foram discutidos algoritmos adaptativos que exploram a esparsidade do sistema com a finalidade de melhorar seu desempenho. A formulação proposta neste trabalho de pesquisa também é orientada à identificação de sistemas esparsos, desenvolvendo-se a partir da estimação baseada na teoria de conjuntos, usando otimização com restrição de norma vetorial, de forma semelhante aos algoritmos descritos na Seção 2.4.3. Em contraste com os algoritmos apresentados naquela seção, a estratégia desenvolvida aqui propõe a aplicação simultânea de restrições no processo de otimização, de forma a obter um conjunto de soluções pertencente à interseção entre as superfícies delimitadas por tais restrições a cada iteração. A nova proposta também difere daqueles algoritmos no que diz respeito ao reúso de informações prévias e à utilização de *hyper-slabs*. Neste caso, apenas o erro *a posteriori* atual é considerado ( $Q = 1$ ), considerando ainda parâmetro  $\phi$  igual a zero. Considerando tais características, a proposta discutida neste trabalho posiciona-se como algoritmos baseados nas projeções em esferas de normas, porém, aproxima-se também da formulação dos algoritmos baseados no NLMS, buscando-se, dessa forma, um compromisso desempenho versus complexidade computacional compatível com os principais algoritmos direcionados a sistemas esparsos. Uma formulação geral que permite a aplicação de normas vetoriais diversas como restrição para o processo de estimação é apresentado na Seção 3.1, sendo os algoritmos resultantes denominados **algoritmos com restrição de norma** (*norm constrained algorithms*), referenciados aqui por algoritmos  $\ell_p$ NC, onde o índice  $p$  indica o tipo de norma considerado. Utilizando essa formulação geral, dois grupos de algoritmos com restrição de norma são obtidos e apresentados nas Seções 3.2 e 3.3, a saber, os algoritmos com restrição de norma  $\ell_1$  ( $\ell_1$ NC) e com restrição de norma  $\ell_0$  ( $\ell_0$ NC). Versões melhoradas desses algoritmos também são elaboradas, com o objetivo de torná-los independentes da escolha de parâmetros de restrição de norma pelo usuário.

### 3.1 Formulação geral baseada em otimização com restrição de norma

A filtragem adaptativa baseada em estimação utilizando a teoria de conjuntos pode ser concebida aplicando-se o método dos multiplicadores de Lagrange para otimização com múltiplas restrições, conforme tratado na Seção 2.1.2 e aplicado na Seção 2.2.2. A partir de tais conceitos, uma formulação geral que determina o conjunto de algoritmos baseados em otimização com restrição de norma é detalhado a seguir.

#### 3.1.1 Problema de otimização

Neste método, buscar-se-á, a cada iteração, uma solução que atenda simultaneamente às seguintes restrições: a) o erro instantâneo *a posteriori* igual a zero, dado por

$$\varepsilon(n) = d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n) = 0 \quad (3.1)$$

que caracteriza a restrição aplicada ao algoritmo NLMS, e b) a norma do vetor de coeficientes limitada a um determinado valor máximo, isto é,

$$\|\mathbf{w}(n+1)\|_p \leq \delta_{\ell_p} \quad (3.2)$$

onde  $\delta_{\ell_p}$  é um parâmetro de limitação de norma determinado previamente. A restrição (3.1) gera um hiperplano de soluções possíveis dado por

$$S_\varepsilon(n) = [\mathbf{w}(n+1) \in \mathbb{R}^N : d(n) - \mathbf{w}(n+1)^T \mathbf{x}(n) = 0] \quad (3.3)$$

enquanto a restrição (3.2) delimita o semi-espço

$$S_{\ell_p}(n) = [\mathbf{w}(n+1) \in \mathbb{R}^N : \|\mathbf{w}(n+1)\|_p \leq \delta_{\ell_p}]. \quad (3.4)$$

Na prática, tal restrição implica uma decisão a cada iteração, considerando se o vetor está ou não dentro do semi-espço delimitado em (3.4). Se o vetor já se encontra dentro de tal semi-espço, a restrição está inativa. Por outro lado, pode-se forçar para que o algoritmo sempre busque a interseção entre o hiperplano  $S_\varepsilon(n)$  e a superfície fronteira de  $S_{\ell_p}(n)$ , onde  $\|\mathbf{w}(n+1)\|_p = \delta_{\ell_p}$ , considerando que tal limitação é o valor de norma que melhor representa a medida da solução procurada. Sendo assim, o problema de otimização (como

definido na Seção 2.2.2), considerando essas duas restrições, é descrito como

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (3.5a)$$

$$\text{sujeito a } \begin{cases} d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n) = 0 \\ \|\mathbf{w}(n+1)\|_p = \delta_{\ell_p}. \end{cases} \quad (3.5b)$$

Para que o problema de otimização seja convexo, no entanto, as restrições de igualdade devem ser afins [36], o que geralmente não é verdadeiro em relação à restrição de norma. Além disso, o problema muitas vezes não é viável em aplicações práticas, devido a escolhas inadequadas de  $\delta_{\ell_p}$ , à presença de ruídos e à necessidade de conhecimento de informações que ainda não estão disponíveis. Por conseguinte, uma certa relaxação de (3.5) torna-se necessária visando à obtenção de algoritmos adaptativos práticos. Nesse contexto, o seguinte problema de otimização, com relaxação da restrição de norma, é aqui considerado:

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (3.6a)$$

$$\text{sujeito a } \begin{cases} d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n) = 0 \\ f_{\ell_p}[\mathbf{w}(n+1)] = \delta_{\ell_p} \end{cases} \quad (3.6b)$$

onde  $f_{\ell_p}[\mathbf{w}(n+1)] \cong \|\mathbf{w}(n+1)\|_p$  é uma função afim que aproxima a norma  $\ell_p$ . Se a aproximação for escolhida de forma que  $f_{\ell_p}[\mathbf{w}(n+1)] \rightarrow \|\mathbf{w}(n+1)\|_p$  em regime permanente, pode-se dizer que a solução obtida tenderá para a superfície de norma  $\ell_p$  escolhida.

A solução de (3.6) pode ser obtida de forma mais simples pela transformação do processo de otimização com restrições em otimização sem restrições, utilizando-se o método dos multiplicadores de Lagrange (veja Seção 2.1.2). Assim, a função de Lagrange para (3.6) é então

$$\begin{aligned} \mathcal{L}[\mathbf{w}(n+1)] &= \frac{1}{2} [\mathbf{w}(n+1) - \mathbf{w}(n)]^T [\mathbf{w}(n+1) - \mathbf{w}(n)] \\ &\quad + \lambda_1(n) [d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n)] \\ &\quad + \lambda_p(n) \{f_{\ell_p}[\mathbf{w}(n+1)] - \delta_{\ell_p}\} \end{aligned} \quad (3.7)$$

onde  $\lambda_1(n)$  e  $\lambda_{\ell_p}(n)$  são os multiplicadores de Lagrange correspondentes às restrições (3.1) e à linearização de (3.2), respectivamente.

### 3.1.2 Atualização do vetor de coeficientes

Sendo a função de Lagrange (3.7) uma função convexa e continuamente diferenciável, determina-se o ponto de mínimo a partir de seu gradiente, dado por

$$\nabla \mathcal{L}[\mathbf{w}(n+1)] = \mathbf{w}(n+1) - \mathbf{w}(n) - \lambda_1(n) \mathbf{x}(n) + \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n+1) \quad (3.8)$$

onde  $\mathbf{g}_{\ell_p}(n+1) = \nabla f_{\ell_p}[\mathbf{w}(n+1)]$  é o gradiente da função que representa a norma  $\ell_p$  em relação a  $\mathbf{w}(n+1)$ . Fazendo (3.8) igual a zero, obtém-se o vetor atualizado

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda_1(n) \mathbf{x}(n) - \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n+1). \quad (3.9)$$

Para determinar o multiplicador de Lagrange correspondente à restrição de erro *a posteriori* igual a zero, deve-se substituir (3.9) em (3.1), obtendo-se

$$d(n) - \mathbf{w}^T(n) \mathbf{x}(n) - \lambda_1(n) \mathbf{x}^T(n) \mathbf{x}(n) + \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}^T(n+1) \mathbf{x}(n) = 0 \quad (3.10)$$

ou

$$e(n) - \lambda_1(n) \mathbf{x}^T(n) \mathbf{x}(n) + \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}^T(n+1) \mathbf{x}(n) = 0. \quad (3.11)$$

A partir de (3.11), pode-se então determinar  $\lambda_1(n)$  em função de  $\lambda_{\ell_p}(n)$ . Assim,

$$\lambda_1(n) = \alpha_1(n) + \alpha_{\ell_p}(n) \lambda_{\ell_p}(n) \quad (3.12)$$

com

$$\alpha_1(n) = \frac{e(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (3.13)$$

e

$$\alpha_{\ell_p}(n) = \frac{\mathbf{g}_{\ell_p}^T(n+1) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}. \quad (3.14)$$

Considerando (3.12), pode-se reescrever a equação de adaptação (3.9) em função de  $\lambda_{\ell_p}(n)$ , obtendo-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_p}(n) \lambda_{\ell_p}(n) \mathbf{x}(n) - \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n+1). \quad (3.15)$$



O multiplicador de Lagrange  $\lambda_{\ell_p}(n)$ , por sua vez, é determinado substituindo-se (3.15) em  $f_{\ell_p}[\mathbf{w}(n+1)] = \delta_{\ell_p}$  de forma a obter

$$f_{\ell_p}\{\mathbf{w}(n) + [\alpha_1(n) + \alpha_{\ell_p}(n)\lambda_{\ell_p}(n)]\mathbf{x}(n) - \lambda_{\ell_p}(n)\mathbf{g}_{\ell_p}(n+1)\} - \delta_{\ell_p} = 0. \quad (3.16)$$

A partir de (3.16),  $\lambda_{\ell_p}(n)$  é calculado após a substituição da expressão apropriada para  $\mathbf{g}_{\ell_p}(n+1)$ , definida especificamente para cada tipo de norma a ser utilizado.

A partir das expressões obtidas nesta seção, podemos definir a formulação geral para derivar os algoritmos com restrição de norma baseando-se nos seguintes passos:

- i) determinar a função gradiente  $\mathbf{g}_{\ell_p}(n+1)$  para a norma considerada;
- ii) substituir este resultado em (3.14) para determinar  $\alpha_{\ell_p}(n)$ ;
- iii) obter  $\lambda_{\ell_p}(n)$  para a norma considerada utilizando (3.16);
- iv) substituir a expressão obtida para  $\alpha_{\ell_p}(n)$  e  $\lambda_{\ell_p}(n)$  em (3.15).

Essa formulação determina o padrão geral dos algoritmos propostos neste trabalho baseados em normas com aplicação simultânea das restrições. Formulações específicas com normas  $\ell_1$  e  $\ell_0$  são discutidas nas seções que seguem.

### 3.2 Algoritmos com restrição de norma $\ell_1$

Considerando a importância da norma  $\ell_1$  na determinação e medição da esparsidade de vetores (veja Seção 2.3), a aplicação dessa métrica tem recebido destaque na concepção de algoritmos adaptativos para sistemas esparsos, como discutido nas Seções 2.4.2 e 2.4.3. Dessa forma, a aplicação da norma  $\ell_1$  à formulação geral da Seção 3.1 é proposta neste trabalho, denominando-se essa formulação **algoritmos com restrição de norma  $\ell_1$**  ( *$\ell_1$ -norm constrained algorithms*) -  $\ell_1\mathbf{NC}$ .

Partindo da formulação geral descrita na seção anterior, pode-se descrever o processo de otimização com restrição de norma  $\ell_1$  como

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (3.17a)$$

$$\text{sujeito a } \begin{cases} d(n) - \mathbf{w}^T(n+1)\mathbf{x}(n) = 0 \\ f_{\ell_1}[\mathbf{w}(n+1)] = \delta_{\ell_1} \end{cases} \quad (3.17b)$$

onde  $f_{\ell_1}[\mathbf{w}(n+1)] \cong \|\mathbf{w}(n+1)\|_1$  é uma função afim que aproxima a norma  $\ell_1$  do vetor de coeficientes. A partir da definição do problema (3.17), pode-se escrever a função de Lagrange correspondente. Assim,

$$\begin{aligned} \mathcal{L}[\mathbf{w}(n+1)] &= \frac{1}{2} [\mathbf{w}(n+1) - \mathbf{w}(n)]^T [\mathbf{w}(n+1) - \mathbf{w}(n)] \\ &\quad + \lambda_1(n) [d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n)] \\ &\quad + \lambda_{\ell_1}(n) \{f_{\ell_1}[\mathbf{w}(n+1)] - \delta_{\ell_1}\} \end{aligned} \quad (3.18)$$

sendo  $\lambda_1(n)$  e  $\lambda_{\ell_1}(n)$  os multiplicadores de Lagrange calculados para o algoritmo  $\ell_1$ NC. A norma  $\ell_1$  do vetor de coeficientes, dada por

$$\|\mathbf{w}(n)\|_1 = \sum_{k=0}^{N-1} |w_k(n)| \quad (3.19)$$

pode ser escrita na forma vetorial como

$$\|\mathbf{w}(n)\|_1 = \mathbf{s}^T(n) \mathbf{w}(n) \quad (3.20)$$

onde  $\mathbf{s}(n) = [s_0(n) \ s_1(n) \ \cdots \ s_{N-1}(n)]^T$  é o vetor de sinais de  $\mathbf{w}(n)$ , sendo seu  $k$ -ésimo elemento definido como

$$s_k(n) = \text{sgn}[w_k(n)] = \begin{cases} 1, & w_k(n) > 0 \\ 0, & w_k(n) = 0 \\ -1, & w_k(n) < 0. \end{cases} \quad (3.21)$$

Agora, considerando (3.20), pode-se escrever a norma  $\ell_1$  de  $\mathbf{w}(n+1)$  como  $\|\mathbf{w}(n+1)\|_1 = \mathbf{s}^T(n+1) \mathbf{w}(n+1)$ , que é uma função convexa, porém, não afim. Para tornar o problema de otimização convexo e viável na prática, é adotada a seguinte aproximação da norma  $\ell_1$ :

$$\mathbf{s}^T(n+1) \mathbf{w}(n+1) \cong \mathbf{s}^T(n) \mathbf{w}(n+1) = f_{\ell_1}[\mathbf{w}(n+1)] \quad (3.22)$$

que representa a segunda restrição afim em (3.17b). A Figura 12 ilustra o impacto de tal modificação para o caso de um filtro com dois coeficientes cujos valores *a priori* são positivos [isto é, com  $\mathbf{w}(n)$  correspondendo a um ponto no primeiro quadrante do plano  $w_0 \times w_1$ ]. Pode-se observar dessa figura que, no quadrante correspondente ao vetor  $\mathbf{w}(n)$ ,  $S_{\ell_1}(n)$  (limite de norma  $\ell_1$  com valor  $\delta_{\ell_1}$  representado pela linha tracejada) corresponde ao conjunto de soluções modificado  $S'_{\ell_1}(n)$  (linha sólida cinza) que satisfaz a restrição afim  $\mathbf{s}^T(n) \mathbf{w}(n+1) = \delta_{\ell_1}$ . Como consequência, apenas nas iterações em que

$\mathbf{w}(n+1)$  corresponde a um ponto em um quadrante diferente do quadrante em que se encontra  $\mathbf{w}(n)$  (veja Figura 12),  $\mathbf{w}(n+1)$  não terá uma norma igual a  $\delta_{\ell_1}$ . Mesmo nesta condição, o erro *a posteriori* ainda será igual a zero, uma vez que  $\mathbf{w}(n+1)$  será sempre pertencente a  $S_\varepsilon(n)$ . Assim, considerando (3.22), o gradiente da função  $f_{\ell_1}[\mathbf{w}(n+1)]$  com respeito a  $\mathbf{w}(n+1)$  é

$$\mathbf{g}_{\ell_1}(n+1) = \mathbf{s}(n). \quad (3.23)$$

Dessa forma, a equação de atualização do algoritmo  $\ell_1$ NC será representada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda_1(n) \mathbf{x}(n) - \lambda_{\ell_1}(n) \mathbf{s}(n). \quad (3.24)$$

Partindo da formulação geral (3.12), o multiplicador de Lagrange  $\lambda_1(n)$  pode ser escrito como

$$\lambda_1(n) = \alpha_1(n) + \alpha_{\ell_1}(n) \lambda_{\ell_1}(n) \quad (3.25)$$

com

$$\alpha_1(n) = \frac{e(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (3.26)$$

e

$$\alpha_{\ell_1}(n) = \frac{\mathbf{s}^T(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}. \quad (3.27)$$

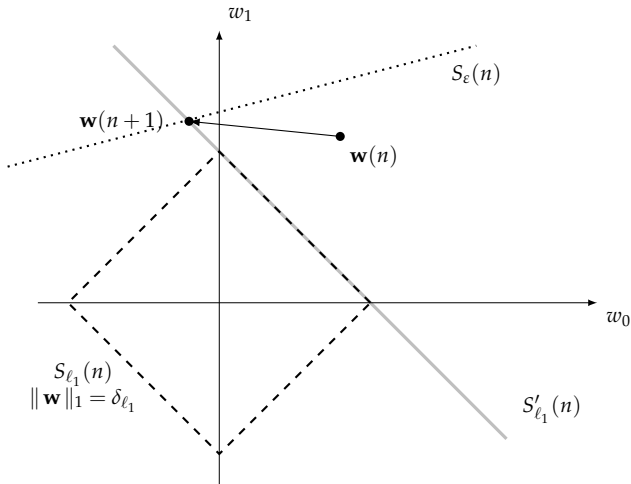


Figura 12: Projeção na interseção entre as superfícies  $S_\varepsilon(n)$  e  $S'_{\ell_1}(n)$ .

Da mesma forma, a atualização do vetor pode ser reescrita em função apenas de  $\lambda_{\ell_1}(n)$ ; portanto,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_1}(n) \lambda_{\ell_1}(n) \mathbf{x}(n) - \lambda_{\ell_1}(n) \mathbf{s}(n). \quad (3.28)$$

Substituindo-se (3.28) em (3.22), então tem-se

$$f_{\ell_1}[\mathbf{w}(n+1)] = \mathbf{s}^T(n) [\mathbf{w}(n) + \alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_1}(n) \lambda_{\ell_1}(n) \mathbf{x}(n) - \lambda_{\ell_1}(n) \mathbf{s}(n)]. \quad (3.29)$$

Assim, fazendo  $f_{\ell_1}[\mathbf{w}(n+1)] = \delta_{\ell_1}$ , o multiplicador  $\lambda_{\ell_1}(n)$  pode ser determinando, obtendo-se

$$\lambda_{\ell_1}(n) = \frac{\mathbf{s}^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n) - \delta_{\ell_1}}{\mathbf{s}^T(n) \mathbf{s}(n) - \alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)}. \quad (3.30)$$

Em (3.30), o termo  $\mathbf{s}^T(n) \mathbf{s}(n)$  leva a uma soma de valores unitários, podendo-se aproximá-lo por  $N$ , que representa a dimensão do vetor<sup>1</sup>.

O algoritmo desenvolvido nesta seção propõe a projeção do vetor de coeficientes no instante  $n$  na exata interseção entre as superfícies  $S_\varepsilon(n)$  e  $S'_{\ell_1}(n)$ , desprezando-se, nesse caso, a presença de ruídos adicionais. Na presença de ruídos, no entanto, pode ser necessário um certo grau de relaxação na amplitude da projeção (*relaxed projection* [29]) de forma a evitar possíveis instabilidades do algoritmo. Para tal, é incluído ao algoritmo um parâmetro de controle de passo, redefinindo a equação de adaptação como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \{ [\alpha_1(n) + \alpha_{\ell_1}(n) \lambda_{\ell_1}(n)] \mathbf{x}(n) - \lambda_{\ell_1}(n) \mathbf{s}(n) \}. \quad (3.31)$$

Um controle ainda maior pode ser acrescentado ao algoritmo, individualizando-se os passos de adaptação a cada parcela da equação de adaptação. Dessa forma, pode-se relativizar as ações de cada restrição, conforme se fizer necessário. Assim,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_1}(n) \lambda_{\ell_1}(n)] \mathbf{x}(n) - \mu_3 \lambda_{\ell_1}(n) \mathbf{s}(n). \quad (3.32)$$

onde os parâmetros  $\mu_1$ ,  $\mu_2$  e  $\mu_3$  representam os passos de adaptação para cada etapa. Um resumo do algoritmo  $\ell_1$ NC é apresentado na Tabela 1.

A implementação do algoritmo proposto, conforme o resumo da Ta-

---

<sup>1</sup>Mesmo considerando um vetor esparso, devido a ruídos inerentes ao sistema, raramente haverá elementos com valores exatamente iguais a zero, tornando essa aproximação verdadeira na maioria das situações.

bela 1, depende da escolha do parâmetro de limitação de norma  $\delta_{\ell_1}$ . Idealmente, esse parâmetro deve ser definido próximo ao valor da norma  $\ell_1$  do vetor de coeficiente que representa o sistema a ser identificado  $\mathbf{h}$  para se obter o melhor desempenho em regime permanente. O problema dessa abordagem é que  $\mathbf{h}$  é geralmente desconhecido ou pode variar ao longo do tempo. Como consequência, o algoritmo, muitas vezes, apresenta desempenho limitado devido a valores indevidamente escolhidos de  $\delta_{\ell_1}$  (veja Exemplo 3.1 na Seção 3.2.2). Um problema semelhante é encontrado em implementações do algoritmo APL1, que também requer a definição de um parâmetro que depende do conhecimento de  $\|\mathbf{h}\|_1$  [30].

Tabela 1: Sumário do algoritmo  $\ell_1$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\alpha_{\ell_1}(n) = \mathbf{s}^T(n) \mathbf{x}(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\lambda_{\ell_1}(n) = \frac{\mathbf{s}^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n) - \delta_{\ell_1}}{N - \alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_1}(n) \lambda_{\ell_1}(n)] \mathbf{x}(n) - \mu_3 \lambda_{\ell_1}(n) \mathbf{s}(n)$$


---

### 3.2.1 Algoritmo $\ell_1$ NC com parâmetro de limitação de norma variável

De forma a contornar a dificuldade de escolher um valor apropriado para  $\delta_{\ell_1}$ , propomos o uso de um parâmetro de limitação de norma variável  $\delta_{v\ell_1}(n)$  cujo valor é atualizado a cada iteração do algoritmo. Para tanto, a abordagem aqui proposta considera, como referência para a limitação de norma, a aproximação da norma  $\ell_1$  de  $\mathbf{w}_N(n+1)$ , obtido pelo algoritmo NLMS com  $\mu = 1$  [veja (1.5)], definida como

$$f_{\ell_1}[\mathbf{w}_N(n+1)] = \mathbf{s}^T(n) \mathbf{w}_N(n+1) \quad (3.33)$$

mantendo o vetor resultante sobre a superfície  $S_\varepsilon(n)$ . Assim, a ideia aqui é definir  $\delta_{v\ell_1}(n)$  com o objetivo de obter um  $\mathbf{w}(n+1)$  pertencente a  $S_\varepsilon(n)$ , porém, com  $f_{\ell_1}[\mathbf{w}(n+1)] < f_{\ell_1}[\mathbf{w}_N(n+1)]$ , sendo

$$\delta_{v\ell_1}(n) = f_{\ell_1}[\mathbf{w}_N(n+1)] - \rho(n). \quad (3.34)$$

Em (3.34),  $\rho(n)$  é definido como um fator de redução, com valor positivo, dado por

$$\rho(n) = \text{abs}\{f_{\ell_1}[\mathbf{w}_N(n+1)] - \|\mathbf{w}(n)\|_1\}. \quad (3.35)$$

É importante observar que  $\mathbf{w}_N(n+1) \cong \mathbf{w}(n) \rightarrow \mathbf{h}$ , à medida que a adaptação aproxima-se do regime permanente. Como resultado, o valor de  $\rho(n)$  tende a aproximar-se de zero [ $\rho(n) \cong 0$ ] e, de (3.34), verifica-se que  $\delta_{v\ell_1}(n) = f_{\ell_1}[\mathbf{w}_N(n+1)] \rightarrow \|\mathbf{h}\|_1$ . Portanto, verifica-se que  $\delta_{v\ell_1}(n)$  tenderá para a norma  $\ell_1$  da planta em regime permanente, que é a escolha ideal para o parâmetro ligado a norma do algoritmo  $\ell_1\text{NC}$ . Também vale a pena mencionar que, antes de chegar a  $\|\mathbf{h}\|_1$ ,  $\delta_{v\ell_1}(n)$  pode assumir valores que estão longe de um valor ideal, como, por exemplo, zero ou até mesmo valores negativos. Contudo, mesmo nesses casos, o problema de otimização irá ainda ser viável (devido ao uso da restrição relaxada de norma) e o vetor  $\mathbf{w}(n+1)$  continuará pertencendo ao hiperplano de erro *a posteriori* igual a zero  $S_\varepsilon(n)$ . Então, como o algoritmo se aproxima do estado estacionário,  $\mathbf{w}(n+1)$  tende a pertencer a ambas as superfícies,  $S_\varepsilon(n)$  e  $S_{\ell_1}(n)$ , visto que  $\delta_{v\ell_1}(n) \rightarrow \|\mathbf{h}\|_1$ .

Agora, substituindo (3.20) e (3.33) em (3.35), o fator de redução  $\rho(n)$  pode ser reescrito como

$$\begin{aligned} \rho(n) &= \text{abs}[\mathbf{s}^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n) - \mathbf{s}^T(n) \mathbf{w}(n)] \\ &= \text{abs}[\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)]. \end{aligned} \quad (3.36)$$

Assim, (3.34) pode também ser reescrita como

$$\delta_{v\ell_1}(n) = \mathbf{s}^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n) - \text{abs}[\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)]. \quad (3.37)$$

Então, substituindo  $\delta_{\ell_1}$  por (3.37) em (3.30), o multiplicador de Lagrange modificado  $\lambda_{v\ell_1}(n)$  é obtido, sendo

$$\lambda_{v\ell_1}(n) = \frac{\text{abs}[\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)]}{\mathbf{s}^T(n) \mathbf{s}(n) - \alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)}. \quad (3.38)$$

Usando (3.38) ao invés de  $\lambda_{\ell_1}(n)$  em (3.31), obtemos um novo algoritmo, o qual é denominado **algoritmo com restrição de norma  $\ell_1$  variável** (*variable  $\ell_1$ -norm constrained algorithm*) - **V- $\ell_1\text{NC}$** , que é independente da es-

colha, pelo usuário, de um valor apropriado para o parâmetro de limitação de norma. Além de tal vantagem, o algoritmo V- $\ell_1$ NC ainda apresenta um custo computacional menor do que o algoritmo  $\ell_1$ NC, o que se pode verificar comparando (3.30) com (3.38). O desempenho do algoritmo V- $\ell_1$ NC, quando comparado com o do algoritmo  $\ell_1$ NC com  $\delta_{\ell_1}$  adequadamente escolhido, também é promissor, como apresentado no Exemplo 3.2 (Seção 3.2.2). Um sumário do algoritmo V- $\ell_1$ NC é mostrado na Tabela 2.

Tabela 2: Sumário do algoritmo V- $\ell_1$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\alpha_{\ell_1}(n) = \mathbf{s}^T(n) \mathbf{x}(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\lambda_{\nu\ell_1}(n) = \frac{\text{abs}[\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)]}{N - \alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_1}(n) \lambda_{\nu\ell_1}(n)] \mathbf{x}(n) - \mu_3 \lambda_{\nu\ell_1}(n) \mathbf{s}(n)$$


---

### 3.2.2 Exemplos

Nesta seção, são apresentados exemplos que ilustram a discussão teórica sobre os algoritmos propostos baseados na norma  $\ell_1$ . Para tal, o desempenho dos algoritmos é avaliado pelos valores médios de desalinhamento normalizado (1.3) obtidos por simulações de Monte Carlo (100 realizações independentes) considerando um problema de identificação de sistemas com as características especificadas na Tabela 3.

**Exemplo 3.1.** Neste exemplo, verifica-se o impacto da escolha de um valor fixo para  $\delta_{\ell_1}$  no desempenho do algoritmo  $\ell_1$ NC em regime permanente ( $n \rightarrow \infty$ ). Um conjunto de simulações, para diferentes valores de  $\delta_{\ell_1}$ , é apresentado na Figura 13. Observa-se que o melhor desempenho do algo-

Tabela 3: Sumário do problema de identificação de sistemas utilizado nos exemplos

---

|  |
|--|
| Planta a ser identificada $\mathbf{h}_1$   |
| comprimento do vetor: 100 coeficientes   |
| coeficientes diferentes de zero: $\{ 0,1 \quad 1,0 \quad -0,5 \quad 0,1 \}$        |
| posições dos coeficientes diferentes de zero: $\{ 1 \quad 30 \quad 35 \quad 38 \}$ |
| grau de esparsidade: $\mathcal{S}(\mathbf{h}_1) = 0,9435$ (veja Seção 2.3)         |
| norma $\ell_1$ : $\ \mathbf{h}_1\ _1 = 1,7$  |
| norma $\ell_0$ : $\ \mathbf{h}_1\ _0 = 4$  |
| Sinal de entrada $x(n)$ : gaussiano branco com variância $\sigma_x^2 = 1$          |
| Ruído de medição $r(n)$ : gaussiano branco com variância $\sigma_r^2 = 10^{-3}$    |

---

ritmo ocorre com  $\delta_{\ell_1}$  próximo a  $\|\mathbf{h}_1\|_1$ , com valor de  $\delta_{\ell_1} = 1,78$ , apresentando rápida degradação quando os valores adotados estão mais afastados. A diferença entre o valor de  $\|\mathbf{h}_1\|_1$  e o melhor  $\delta_{\ell_1}$  ocorre em função do ruído de adaptação, já que, em regime permanente, existirão valores residuais correspondentes aos coeficientes não significativos, os quais serão acrescidos positivamente à norma  $\ell_1$  do vetor. Pode-se verificar que o valor do melhor  $\delta_{\ell_1}$  tende para  $\|\mathbf{h}_1\|_1$  à medida que o ruído de adaptação é reduzido com a redução do passo de adaptação  $\mu$  (veja Figura 14).

**Exemplo 3.2.** Uma comparação dos desempenhos dos algoritmos  $\ell_1$ NC, com  $\delta_{\ell_1}$  fixo, e V- $\ell_1$ NC é discutida nesse exemplo. Todos os algoritmos envolvidos na simulação utilizam passo unitário,  $\mu = 1$ , sendo que o algoritmo  $\ell_1$ NC foi ajustado com  $\delta_{\ell_1} = 1,78$  (melhor desempenho verificado no exemplo anterior para  $\mu = 1$ ). Os resultados das simulações são mostrados na Figura 15, onde pode-se verificar que o algoritmo com restrição variável (V- $\ell_1$ NC) apresenta resultados satisfatórios em relação aos demais algoritmos da comparação. Observa-se ainda que sua velocidade de convergência, para o caso em questão, é maior do que a do algoritmo original ( $\ell_1$ NC).



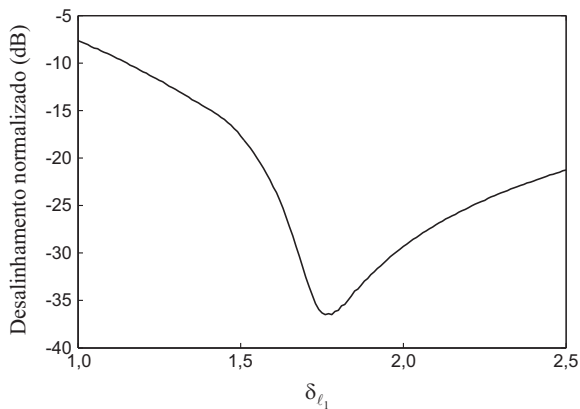


Figura 13: Exemplo 3.1. Desempenho em regime permanente do algoritmo  $\ell_1$ NC para  $\mu = 1$  em função de  $\delta_{\ell_1}$ .

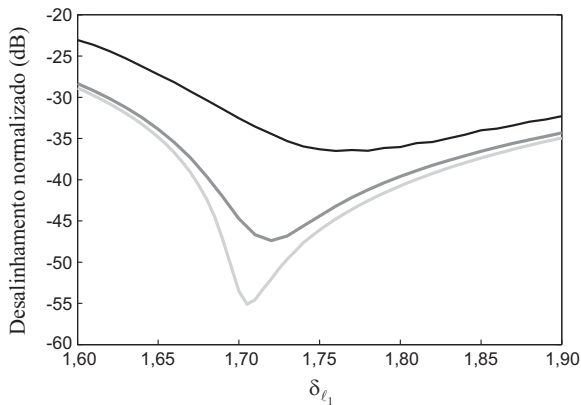


Figura 14: Exemplo 3.1. Desempenho em regime permanente do algoritmo  $\ell_1$ NC para  $\mu = 1$  (linha escura),  $\mu = 0,1$  (linha intermediária) e  $\mu = 0,01$  (linha clara) em função de  $\delta_{\ell_1}$ .

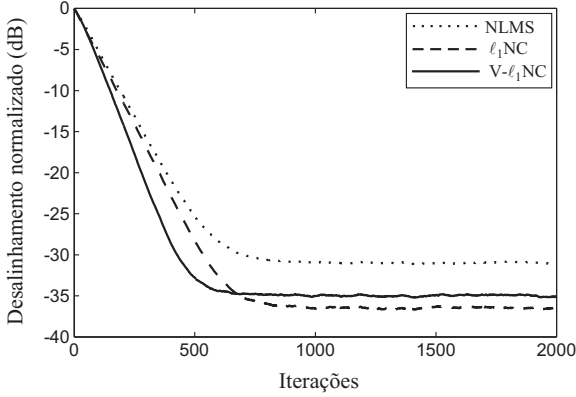


Figura 15: Exemplo 3.2. Comparação de desempenho entre os algoritmos NLMS,  $\ell_1$ NC e V- $\ell_1$ NC.

### 3.3 Algoritmos com restrição de norma $\ell_0$

Conforme discutido na Seção 2.3, a norma  $\ell_0$ , dada por

$$\|\mathbf{w}(n)\|_0 = \sum_{k=0}^{N-1} f[w_k(n)] \quad (3.39)$$

com

$$f[w_k(n)] = \begin{cases} 1, & w_k(n) \neq 0 \\ 0, & w_k(n) = 0 \end{cases} \quad (3.40)$$

pode ser considerada a métrica vetorial que melhor determina a esparsidade de um vetor, levando em conta a definição mais estrita de esparsidade, dada pela simples contagem do número de coeficientes diferentes de zero de um vetor [15]. No entanto, sendo (3.40) uma função descontínua e não convexa, em sistemas que requerem processos de otimização, geralmente são utilizadas aproximações que guardam características da norma  $\ell_0$  e, ao mesmo tempo, permitem sua aplicação prática. Uma aproximação usual para (3.40), citada na literatura [23], [31], [56] e utilizada em alguns algoritmos apresentados na Seção 2.4, utiliza a função exponencial para representar cada elemento do vetor na composição de sua norma  $\ell_0$ , tornando-a uma função convexa e contínua. Tal aproximação é dada por

$$f_C[w_k(n)] = 1 - e^{-\beta|w_k(n)|}. \quad (3.41)$$

Dessa forma, a norma  $\ell_0$  aproximada por essa função contínua é expressa por

$$\|\mathbf{w}(n)\|_{0_C} = \sum_{k=0}^{N-1} \left[ 1 - e^{-\beta|w_k(n)|} \right]. \quad (3.42)$$

A Figura 16 ilustra o comportamento de (3.41) considerando diferentes valores de  $\beta$ . Para qualquer valor de  $\beta$ , a função converge assintoticamente para  $f_C(w_k) = 1$  quando  $|w_k| \rightarrow \infty$ . Para  $\beta \rightarrow \infty$ , a função é igual a zero apenas para  $w_k = 0$ , sendo igual a 1 para os demais valores, fazendo com que (3.42) resulte em igualdade. A Figura 17 ilustra as curvas de nível para (3.42) igual a 1 ( $\mathbf{w} \in \mathbb{R}^2$ ) para diferentes valores de  $\beta$ .

A função exponencial possui, no entanto, alguns inconvenientes em relação à complexidade computacional do algoritmo [31]. Para viabilizar seu uso na prática, uma solução possível é a utilização de tabelas de pesquisa (*look-up tables*) [23] para a obtenção de valores, o que, no entanto, implica grande exigência de memória. A solução adotada neste trabalho, portanto, é a aproximação da exponencial utilizando a expansão em série de Taylor [31], [57], limitada à parte positiva de seu primeiro termo. Dessa forma,

$$e^{-\beta|w_k(n)|} \cong z_k(n) = \begin{cases} 1 - \beta|w_k(n)|, & |w_k(n)| \leq \frac{1}{\beta} \\ 0, & \text{demais.} \end{cases} \quad (3.43)$$

Assim, (3.41) pode ser aproximada por uma função linearizada, dada por

$$f_L[w_k(n)] = 1 - z_k(n) \quad (3.44)$$

sendo a norma  $\ell_0$  agora aproximada por

$$\|\mathbf{w}(n)\|_{0_L} = \sum_{k=0}^{N-1} 1 - z_k(n) = N - \mathbf{s}_z^T(n) \mathbf{s}_z(n) + \beta \mathbf{s}_z^T(n) \mathbf{w}(n) \quad (3.45)$$

onde  $\mathbf{s}_z(n) = [s_{z0}(n) \ s_{z1}(n) \ \dots \ s_{z(N-1)}(n)]^T$  é o vetor cujos elementos são dados por

$$s_{zk}(n) = \begin{cases} \text{sgn}[w_k(n)], & |w_k(n)| \leq \frac{1}{\beta} \\ 0, & \text{demais.} \end{cases} \quad (3.46)$$

A Figura 18 ilustra o comportamento de (3.44) para diferentes valores de  $\beta$ . Em contraste com a aproximação anterior, tal função se caracteriza pela linearidade quando  $|w_k| \leq 1/\beta$ , o que reduz significativamente sua complexidade matemática. No entanto, a não suavidade apresentada em  $|w_k| = 1/\beta$  implica

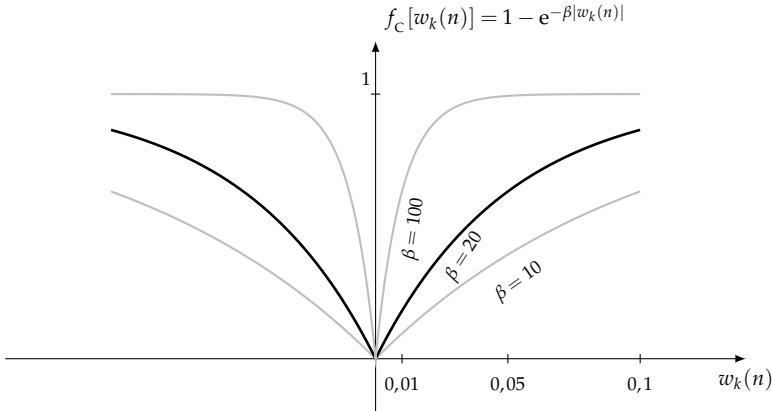


Figura 16: Comportamento da função (3.41), que representa cada elemento do vetor na aproximação da norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), aproximação para  $\beta = 20$ .

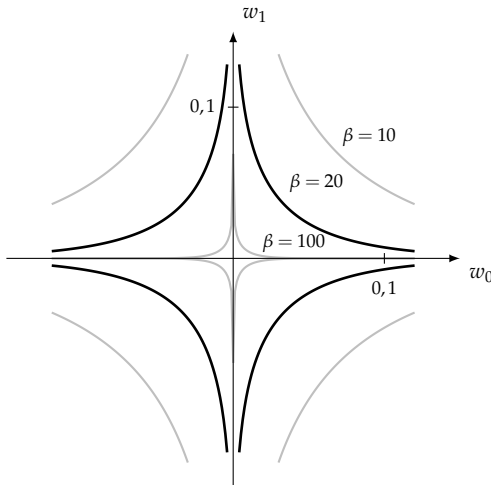


Figura 17: Curvas de nível com valor igual a 1 da função (3.42), que aproxima a norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), curva de nível para  $\beta = 20$ .

a necessidade de tomada de decisão, como verificado em (3.43). A Figura 19 ilustra as curvas de nível para (3.45) igual a 1, considerando diferentes valores de  $\beta$  ( $\mathbf{w} \in \mathbb{R}^2$ ).

Considerando, no entanto, que (3.45) não é uma função afim, pois depende de (3.44), que é uma função claramente não afim (veja Figura 18), deve-se obter uma relaxação da restrição de norma  $\ell_0$  para viabilizar o algoritmo. Reescrevendo (3.45) para o vetor de coeficientes *a posteriori* como

$$\|\mathbf{w}(n+1)\|_{0_L} = N - \mathbf{s}_z^T(n+1) \mathbf{s}_z(n+1) + \beta \mathbf{s}_z^T(n+1) \mathbf{w}(n+1) \quad (3.47)$$

então, analogamente ao caso do algoritmo  $\ell_1$ NC, substituímos  $\mathbf{s}_z(n+1)$  por  $\mathbf{s}_z(n)$  em (3.47), resultando na seguinte aproximação afim para a restrição de norma  $\ell_0$ :

$$\begin{aligned} f_{\ell_0}[\mathbf{w}(n+1)] &= N - \mathbf{s}_z^T(n) \mathbf{s}_z(n) + \beta \mathbf{s}_z^T(n) \mathbf{w}(n+1) \\ &= N - N_z(n) + \beta \mathbf{s}_z^T(n) \mathbf{w}(n+1) \end{aligned} \quad (3.48)$$

sendo  $N_z(n) = \mathbf{s}_z^T(n) \mathbf{s}_z(n)$  uma variável escalar no instante  $n$  que representa o número de coeficientes de  $\mathbf{w}(n)$  com valor menor do que  $1/\beta$ .

Considerando as aproximações para norma  $\ell_0$  definidas nesta seção, podemos, então, obter o algoritmo denominado **algoritmo com restrição de norma  $\ell_0$**  ( *$\ell_0$ -norm constrained algorithm*) -  $\ell_0$ NC, caracterizado pelo processo de otimização com restrição de norma  $\ell_0$  dado por

$$\text{minimizar } \frac{1}{2} \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_2^2 \quad (3.49a)$$

$$\text{sujeito a } \begin{cases} d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n) = 0 \\ f_{\ell_0}[\mathbf{w}(n+1)] = \delta_{\ell_0}. \end{cases} \quad (3.49b)$$

Assim, com o objetivo de desenvolver um algoritmo adaptativo com base na formulação geral apresentada na Seção 3.1, deve-se obter o gradiente de (3.48) em relação a  $\mathbf{w}(n+1)$ , que resulta em

$$\mathbf{g}_{\ell_0}(n+1) = \beta \mathbf{s}_z(n). \quad (3.50)$$

Dessa forma, a partir de (3.9), a equação de atualização do vetor é reescrita como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \lambda_1(n) \mathbf{x}(n) - \beta \lambda_{\ell_0}(n) \mathbf{s}_z(n). \quad (3.51)$$

Agora, considerando (3.12), o multiplicador de Lagrange  $\lambda_1(n)$  pode ser obtido por

$$\lambda_1(n) = \alpha_1(n) + \alpha_{\ell_0}(n) \lambda_{\ell_0}(n) \quad (3.52)$$

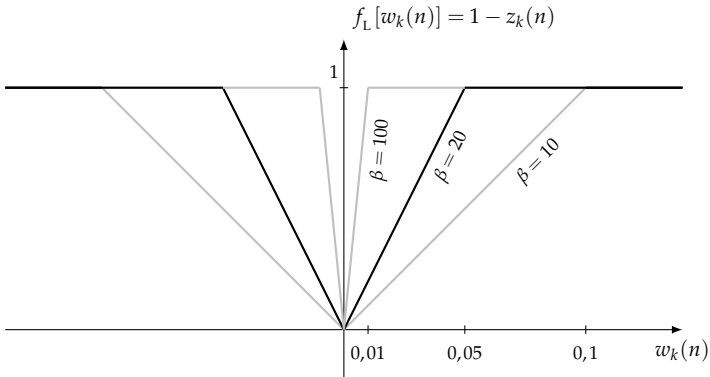


Figura 18: Comportamento da função (3.44), que representa cada elemento do vetor na aproximação da norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), aproximação para  $\beta = 20$ .

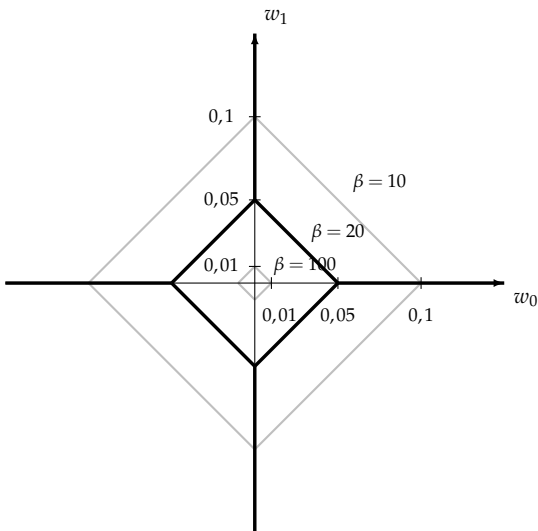


Figura 19: Curvas de nível com valor igual a 1 da função (3.45), que aproxima a norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), curva de nível para  $\beta = 20$ .

com

$$\alpha_1(n) = \frac{e(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (3.53)$$

e

$$\alpha_{\ell_0}(n) = \beta \frac{\mathbf{s}_z^T(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}. \quad (3.54)$$

Substituindo (3.52) em (3.51), obtem-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_0}(n) \lambda_{\ell_0}(n) \mathbf{x}(n) - \beta \lambda_{\ell_0}(n) \mathbf{s}_z(n). \quad (3.55)$$

Para determinar  $\lambda_{\ell_0}(n)$ , substituímos (3.55) em (3.48), igualando a equação resultante a  $\delta_{\ell_0}$ , obtendo a expressão

$$\lambda_{\ell_0}(n) = \frac{\beta \mathbf{s}_z^T(n) \mathbf{w}(n) + \beta \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n) + N - N_z(n) - \delta_{\ell_0}}{\beta^2 N_z(n) - \beta \alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)} \quad (3.56)$$

que pode ser reescrita como

$$\lambda_{\ell_0}(n) = \frac{1}{\beta} \left\{ \frac{\mathbf{s}_z^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n) + [N - N_z(n) - \delta_{\ell_0}]/\beta}{N_z(n) - \alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)/\beta} \right\}. \quad (3.57)$$

Incluindo-se os parâmetros de controle de passo à equação de atualização, então,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_0}(n) \lambda_{\ell_0}(n)] \mathbf{x}(n) - \mu_3 \beta \lambda_{\ell_0}(n) \mathbf{s}_z(n). \quad (3.58)$$

No entanto, redefinindo (3.54) e (3.57) como

$$\alpha_{\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (3.59)$$

e

$$\lambda_{\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n) + [N - N_z(n) - \delta_{\ell_0}]/\beta}{N_z(n) - \alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)/\beta} \quad (3.60)$$

respectivamente, pode-se obter o mesmo resultado de (3.58) com

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_0}(n) \lambda_{\ell_0}(n)] \mathbf{x}(n) - \mu_3 \lambda_{\ell_0}(n) \mathbf{s}_z(n) \quad (3.61)$$

tornando a formulação mais simples e aproximando-a da formulação do algoritmo  $\ell_1$ NC. A Tabela 4 apresenta um sumário do algoritmo  $\ell_0$ NC.

Tabela 4: Sumário do algoritmo  $\ell_0$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

Para  $k$  de 0 até  $N - 1$

$$z_k(n) = \max[0, (1 - \beta |w_k(n)|)]$$

$$s_{zk}(n) = s_k(n) \operatorname{sgn}[z_k(n)]$$

$$\alpha_{\ell_0}(n) = \mathbf{s}_z^T(n) \mathbf{x}(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\lambda_{\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n) + [N - N_z(n) - \delta_{\ell_0}] / \beta}{N_z(n) - \alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{\ell_0}(n) \lambda_{\ell_0}(n)] \mathbf{x}(n) - \mu_3 \lambda_{\ell_0}(n) \mathbf{s}_z(n)$$


---

### 3.3.1 Considerações sobre o comportamento do algoritmo $\ell_0$ NC

É fácil verificar em (3.58), através da presença do vetor  $\mathbf{s}_z(n)$  [cujos elementos estão definidos em (3.46)], que a atualização dos coeficientes com valores absolutos maiores do que  $1/\beta$ , definidos aqui como coeficientes significativos, é diferente daqueles com valores absolutos menores do que  $1/\beta$ , denominados coeficientes não-significativos. Dessa forma, pode-se considerar o comportamento do algoritmo  $\ell_0$ NC, com relação ao parâmetro  $\beta$ , levando-se em conta as seguintes situações de convergência:

- i) Se todos os coeficientes do filtro têm valores absolutos maiores que  $1/\beta$ ,  $\mathbf{s}_z(n)$  é um vetor de zeros e, portanto, (3.58) torna-se igual à equação de atualização do algoritmo NLMS dada por (1.5).
- ii) Se, por outro lado, todos os coeficientes do filtro têm valores absolutos menores ou iguais a  $1/\beta$ ,  $\mathbf{s}_z(n)$  e  $\mathbf{s}_z^T(n) \mathbf{s}_z(n)$  são iguais a  $\mathbf{s}(n)$  e  $N$ , respectivamente, e o processo adaptativo utilizando o algoritmo  $\ell_0$ NC é



idêntico ao do  $\ell_1$ NC com  $\delta_{\ell_1} = \delta_{\ell_0} / \beta$  [veja (3.28), (3.27), e (3.30)].

- iii) Para um vetor de coeficientes com apenas alguns valores absolutos maiores do que  $1/\beta$ , situação desejada na implementação do algoritmo, um comportamento intermediário entre os casos anteriores é obtido. Nesse caso, especificamente em relação aos coeficientes não-significativos [aqueles com  $|w_k(n)| \leq 1/\beta$ ], uma análise do algoritmo (resumido na Tabela 4) revela que a atualização desses coeficientes é equivalente à dos coeficientes do algoritmo  $\ell_1$ NC com  $\delta_{\ell_1} = [\delta_{\ell_0} - N + N_z(n)]/\beta$ , observando-se uma relação direta entre o parâmetro  $\delta_{\ell_0}$  e o número de elementos do vetor considerados não-significativos a cada iteração.

Tais situações nos mostram que o algoritmo  $\ell_0$ NC apresenta características que o aproximam tanto do algoritmo NLMS quanto do algoritmo  $\ell_1$ NC, sendo a escolha de  $\beta$ , bem como de  $\delta_{\ell_0}$ , importantes para a obtenção de seu melhor desempenho. No caso do parâmetro de restrição, além de uma possível escolha incorreta de  $\delta_{\ell_0}$ , sua manutenção como um parâmetro fixo também pode comprometer o funcionamento do algoritmo  $\ell_0$ NC. Quando a norma  $\ell_0$  do vetor de coeficientes no instante  $n$  é muito maior que a restrição especificada na configuração do algoritmo, isto é,  $N - N_z(n) \gg \delta_{\ell_0}$  (situação comum nas iterações iniciais do processo de adaptação), pode ocorrer aumento exagerado dos valores de  $\lambda_{\ell_0}(n)$  [veja (3.60)], produzindo grandes variações nos coeficientes, o que resultaria em desalinhamentos elevados. Esse efeito, que depende das especificações de  $\delta_{\ell_0}$  e  $\beta$ , é discutido nos Exemplos 3.3 e 3.4 (Seção 3.3.5). Buscando minimizar tais problemas inerentes a  $\delta_{\ell_0}$  com valor fixo, a utilização de um parâmetro de restrição de norma  $\ell_0$  variável é discutida na seção seguinte.

### 3.3.2 Algoritmo $\ell_0$ NC com parâmetro de limitação de norma variável

A partir das discussões sobre o comportamento do algoritmo  $\ell_0$ NC, verifica-se que, se um parâmetro de limitação de norma variável definido por

$$\delta_{v\ell_0}(n) = N - \mathbf{s}_z^T(n) \mathbf{s}_z(n) = N - N_z(n) \quad (3.62)$$

for utilizado em substituição a  $\delta_{\ell_0}$  em (3.57), os coeficientes com valores absolutos menores ou iguais a  $1/\beta$  (não-significativos) serão forçados para zero. Pode-se demonstrar tal relação, considerando a restrição  $f_{\ell_0}[\mathbf{w}(n+1)] = \delta_{v\ell_0}(n)$  e a expressão da aproximação afim para a restrição de norma  $\ell_0$  (3.48). Assim,

$$N - N_z(n) + \beta \mathbf{s}_z^T(n) \mathbf{w}(n+1) = N - N_z(n) \quad (3.63)$$

e

$$\beta \mathbf{s}_z^T(n) \mathbf{w}(n+1) = 0. \quad (3.64)$$

Observa-se ainda que tal restrição variável elimina a possibilidade de instabilidade gerada por  $N - N_z \gg \delta_{\ell_0}$ , já que a restrição acompanha a norma  $\ell_0$  do vetor de coeficientes no instante  $n$ . Essa estratégia de escolha de uma restrição de norma variável define o algoritmo denominado **algoritmo com restrição de norma  $\ell_0$  variável** (*variable  $\ell_0$ -norm constrained algorithm*) - **V- $\ell_0$ NC**. Então, substituindo  $\delta_{\ell_0}$  em (3.57) por (3.62) e fazendo as devidas simplificações, o algoritmo V- $\ell_0$ NC é obtido. Assim,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \{ [\alpha_1(n) + \alpha_{v\ell_0}(n) \lambda_{v\ell_0}(n)] \mathbf{x}(n) - \lambda_{v\ell_0}(n) \mathbf{s}_z(n) \} \quad (3.65)$$

ou, utilizando passos individuais,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{v\ell_0}(n) \lambda_{v\ell_0}(n)] \mathbf{x}(n) - \mu_3 \lambda_{v\ell_0}(n) \mathbf{s}_z(n) \quad (3.66)$$

com  $\alpha_1(n)$  dado por (3.53),

$$\alpha_{v\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (3.67)$$

e

$$\lambda_{v\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n)}{N_z(n) - \alpha_{v\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)}. \quad (3.68)$$

Vale a pena ressaltar que o uso de  $\delta_{v\ell_0}(n)$  ao invés de  $\delta_{\ell_0}$  não implica qualquer alteração sobre a restrição (3.6a) do problema de otimização. Assim, o vetor de coeficientes obtido usando o algoritmo V- $\ell_0$ NC, para  $\mu = 1$ , leva também a um erro *a posteriori* igual a zero, mantendo a consistência do processo de adaptação. Os Exemplos 3.5 e 3.6 (Seção 3.3.5) ilustram o comportamento do algoritmo V- $\ell_0$ NC em função da variação de  $\beta$  e seu desempenho em comparação com o algoritmo  $\ell_0$ NC (com  $\delta_{\ell_0}$  fixo). Um sumário do algoritmo V- $\ell_0$ NC é apresentado na Tabela 5.

### 3.3.3 Algoritmo $\ell_0$ NC com aproximação suavizada

Os algoritmos com restrição de norma  $\ell_0$  analisados até aqui baseiam-se em uma aproximação da função exponencial obtida por expansão em série de Taylor com polinômio de primeira ordem, sendo a função afim que representa a restrição de norma para estes algoritmos dada por (3.48). Sendo o gradiente de (3.48) um valor constante para todos os elementos de  $\mathbf{w}(n)$  com

valor menor que  $1/\beta$ , não há distinção entre a atuação do algoritmo sobre coeficientes próximos a zero e próximos a  $1/\beta$ . Tal condição pode causar efeitos indesejáveis sobre a adaptação se coeficientes considerados significativos que estiverem próximos ao limiar forem levados para zero. Uma alternativa a essa configuração é a aproximação da função exponencial definida em (3.42) através da expansão por um polinômio de segunda ordem de Taylor. Tal polinômio, no entanto, deve ser configurado de forma a apresentar valor igual a zero quando  $|w_k(n)| = 1/\beta$ , para qualquer  $k$ , para garantir a continuidade da função. Dessa forma, uma nova função de aproximação para a norma  $\ell_0$  é obtida, sendo a função correspondente a cada elemento do vetor definida por

$$f_s[w_k(n)] = 1 - z'_k(n) \quad (3.69)$$

Tabela 5: Sumário do algoritmo V- $\ell_0$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

Para  $k$  de 0 até  $N - 1$

$$z_k(n) = \max[0, (1 - \beta|w_k(n)|)]$$

$$z_{sk}(n) = s_k(n) \operatorname{sgn}[z_k(n)]$$

$$\alpha_{v\ell_0}(n) = \mathbf{s}_z^T(n) \mathbf{x}(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\lambda_{v\ell_0}(n) = \frac{\mathbf{s}_z^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n)}{N_z(n) - \alpha_{v\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{v\ell_0}(n) \lambda_{v\ell_0}(n)] \mathbf{x}(n) - \mu_3 \lambda_{v\ell_0}(n) \mathbf{s}_z(n)$$


---

sendo

$$e^{-\beta|w_k(n)|} \cong z'_k(n) = \begin{cases} 1 - 2\beta|w_k(n)| + \beta^2|w_k(n)|^2, & |w_k(n)| \leq \frac{1}{\beta} \\ 0, & \text{demais.} \end{cases} \quad (3.70)$$

Considerando (3.43) e (3.70), pode-se verificar que  $z'_k(n) = z_k^2(n)$ . Assim, a aproximação da norma  $\ell_0$  para o algoritmo com aproximação suavizada é descrita por

$$\begin{aligned} \|\mathbf{w}(n)\|_{0_S} &= \sum_{k=0}^{N-1} 1 - z'_k(n) \\ &= N - \mathbf{1}^T \mathbf{z}'(n) \end{aligned} \quad (3.71)$$

sendo

$$\mathbf{1}^T \mathbf{z}'(n) = \mathbf{s}_z^T(n) \mathbf{s}_z(n) - 2\beta \mathbf{s}_z^T(n) \mathbf{w}(n) + \beta^2 \mathbf{w}^T(n) \mathbf{S}_z(n) \mathbf{w}(n) \quad (3.72)$$

com  $\mathbf{S}_z(n)$  representando uma matriz diagonal com os elementos de  $\mathbf{s}_z(n)$  compondo sua diagonal. Assim,

$$\begin{aligned} \|\mathbf{w}(n+1)\|_{0_S} &= N - \mathbf{s}_z^T(n+1) \mathbf{s}_z(n+1) + 2\beta \mathbf{s}_z^T(n+1) \mathbf{w}(n+1) - \\ &\quad \beta^2 \mathbf{w}^T(n+1) \mathbf{S}_z(n+1) \mathbf{w}(n+1). \end{aligned} \quad (3.73)$$

As Figuras 20 e 21 ilustram o efeito dessa nova aproximação. É possível observar a suavização das curvas, tanto no comportamento da função que representa cada elemento do vetor na aproximação da norma  $\ell_0$  quanto nas curvas de nível para  $\|\mathbf{w}\|_{0_S} = 1$ . Tal comportamento confere a esses algoritmos a denominação de **algoritmos com restrição de norma  $\ell_0$  suavizada** (*smoothed  $\ell_0$ -norm constrained algorithm*) - **S- $\ell_0$ NC**.

Essa nova aproximação, por sua característica não linear, determina um processo de otimização não convexo, sendo necessária uma série de aproximações que viabilizem a implantação do algoritmo. Inicialmente, uma aproximação de (3.73) é realizada substituindo-se  $\mathbf{s}_z(n+1)$  por  $\mathbf{s}_z(n)$ , de forma semelhante à aproximação utilizada nos demais algoritmos  $\ell_0$ NC, obtendo-se a seguinte função quadrática:

$$\begin{aligned} f_{s\ell_0}[\mathbf{w}(n+1)] &= N - \mathbf{s}_z^T(n) \mathbf{s}_z(n) + 2\beta \mathbf{s}_z^T(n) \mathbf{w}(n+1) - \\ &\quad \beta^2 \mathbf{w}^T(n+1) \mathbf{S}_z(n) \mathbf{w}(n+1). \end{aligned} \quad (3.74)$$

O gradiente de (3.74) com respeito a  $\mathbf{w}(n+1)$  será então

$$\mathbf{g}_{s\ell_0}(n+1) = 2\beta \mathbf{s}_z(n) - 2\beta^2 \mathbf{w}(n+1). \quad (3.75)$$

Como (3.75) depende ainda de  $\mathbf{w}(n+1)$ , uma nova aproximação deve ser aplicada, agora substituindo  $\mathbf{w}(n+1)$  por  $\mathbf{w}(n)$  apenas nessa expressão. Assim,

$$\begin{aligned} \mathbf{g}_{s\ell_0}(n+1) &\cong 2\beta \mathbf{s}_z(n) - 2\beta^2 \mathbf{w}(n) \\ &\cong 2\beta \mathbf{s}_z(n) [\mathbf{1} - \beta \mathbf{S}_z(n) \mathbf{w}(n)] \\ &\cong 2\beta \mathbf{S}_z(n) \mathbf{z}(n). \end{aligned} \quad (3.76)$$

Definindo um vetor  $\mathbf{z}_s(n) = [z_{s0}(n) \ z_{s1}(n) \ \dots \ z_{s(N-1)}(n)]^T$ , sendo  $z_{sk}(n) = s_k(n)z_k(n)$ , então

$$\mathbf{g}_{s\ell_0}(n+1) \cong 2\beta \mathbf{z}_s(n). \quad (3.77)$$

Verifica-se em (3.77) que o valor da derivada para cada elemento do vetor  $\mathbf{w}(n+1)$  depende de  $z_{sk}(n) = s_k(n)z_k(n)$  e, portanto, será menor se  $w_k(n)$  for maior, como pode-se verificar através da representação na Figura 22. Nessa Figura, são apresentadas duas situações distintas, sendo  $w_1(n)$  um determinado componente do vetor de coeficientes do filtro que se encontra próximo a zero no instante  $n$  e  $w_2(n)$  outro componente que se encontra próximo ao limiar  $1/\beta$  no instante  $n$ . Observa-se que as inclinações das retas tangentes nos dois pontos da curva da função  $f_s[w_k(n)]$  são diferentes para cada elemento do vetor, representando diferentes derivadas. Sendo  $z_1(n) > z_2(n)$ , verifica-se, a partir de (3.77), que a derivada de  $w_1(n)$  é maior do que a de  $w_2(n)$ . Nesse sentido, as chances de coeficientes significativos próximos ao limiar serem atraídos para zero são reduzidas nessa proposta.

A derivação do algoritmo S- $\ell_0$ NC segue o mesmo padrão do algoritmo  $\ell_0$ NC. Assim, considerando as expressões para o algoritmo  $\ell_0$ NC (3.54), (3.57) e (3.58), além da expressão para o gradiente do algoritmo S- $\ell_0$ NC (3.77), obtêm-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\alpha_1(n) + \alpha_{s\ell_0}(n)\lambda_{s\ell_0}(n)\mathbf{x}(n)] - \lambda_{s\ell_0}(n)\mathbf{z}_s(n) \quad (3.78)$$

com o multiplicador de Lagrange da restrição de norma  $\ell_0$  dado por

$$\lambda_{s\ell_0}(n) = \frac{\mathbf{z}_s^T(n)\mathbf{w}(n) + \alpha_1(n)\mathbf{z}_s^T(n)\mathbf{x}(n) + [N - N_z(n) - \delta_{\ell_0}]/2\beta}{N_z(n) - \alpha_{s\ell_0}(n)\mathbf{z}_s^T(n)\mathbf{x}(n)} \quad (3.79)$$

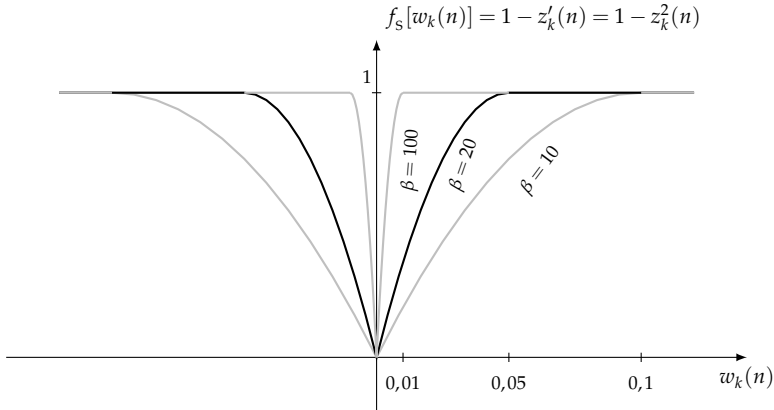


Figura 20: Comportamento da função (3.69), que representa cada elemento do vetor na aproximação da norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), aproximação para  $\beta = 20$ .

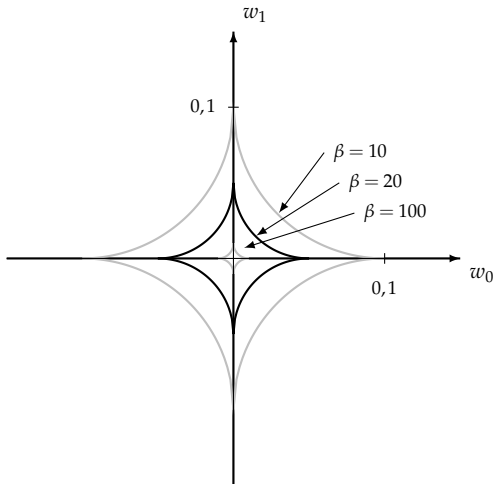


Figura 21: Curvas de nível com valor igual a 1 da função (3.73), que aproxima a norma  $\ell_0$ , para diferentes valores de  $\beta$ . Em destaque (linha escura), curva de nível para  $\beta = 20$ .

e

$$\alpha_{s\ell_0}(n) = \frac{\mathbf{z}_s^T(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}. \quad (3.80)$$

Da mesma forma que o algoritmo V- $\ell_0$ NC, uma versão com restrição variável é aqui desenvolvida. A equação de atualização do algoritmo VS- $\ell_0$ NC, já incluindo-se os parâmetros de controle de passos individuais, é obtida por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{s\ell_0}(n) \lambda_{vs\ell_0}(n) \mathbf{x}(n)] - \mu_3 \lambda_{vs\ell_0}(n) \mathbf{z}_s(n) \quad (3.81)$$

com o multiplicador da restrição de norma  $\ell_0$  dado agora por

$$\lambda_{vs\ell_0}(n) = \frac{\mathbf{z}_s^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{z}_s^T(n) \mathbf{x}(n)}{N_z(n) - \alpha_{s\ell_0}(n) \mathbf{z}_s^T(n) \mathbf{x}(n)}. \quad (3.82)$$

A suavização da aproximação da norma  $\ell_0$  que define este algoritmo, observada nas Figuras 20 e 21, tem como principal característica o abrandamento do limite  $1/\beta$ , permitindo que a adaptação de coeficientes significativos próximos a esse limite não sejam absorvidos com tanta facilidade para o rol de coeficientes não-significativos durante o processo de adaptação. Isso implica a possibilidade de ampliação desse limite, quando comparado com o algoritmo V- $\ell_0$ NC, como pode ser observado no Exemplo 3.7 (Seção 3.3.5). A Tabela 6 apresenta um resumo do algoritmo com aproximação suavizada e restrição variável VS- $\ell_0$ NC.

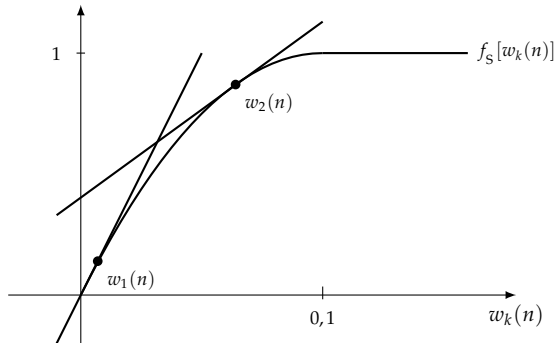


Figura 22: Representação das derivadas de (3.69) em dois pontos distintos.

Tabela 6: Sumário do algoritmo VS- $\ell_0$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

Para  $k$  de 0 até  $N - 1$

$$z_k(n) = \max[0, (1 - \beta |w_k(n)|)]$$

$$z_{sk}(n) = s_k(n) z_k(n)$$

$$\alpha_{s\ell_0}(n) = \mathbf{z}_s^T(n) \mathbf{x}(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\lambda_{vs\ell_0}(n) = \frac{\mathbf{z}_s^T(n) \mathbf{w}(n) + \alpha_1(n) \mathbf{z}_s^T(n) \mathbf{x}(n)}{N_z(n) - \alpha_{s\ell_0}(n) \mathbf{z}_s^T(n) \mathbf{x}(n)}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + [\mu_1 \alpha_1(n) + \mu_2 \alpha_{s\ell_0}(n) \lambda_{vs\ell_0}(n) \mathbf{x}(n)] - \mu_3 \lambda_{vs\ell_0}(n) \mathbf{z}_s(n)$$


---

### 3.3.4 Algoritmo com restrição de norma $\ell_0$ com coeficientes iguais a zero

Todos os algoritmos com restrição de norma apresentados até aqui neste capítulo são baseados na aplicação simultânea das restrições linear e de norma, realizando a projeção na interseção entre as superfícies de soluções, a cada iteração. Por sua vez, os algoritmos descritos na Seção 2.4.3, que também se baseiam em restrições de normas, caracterizam-se por projeções sequenciais sobre as superfícies, projetando inicialmente o vetor de coeficientes no hiperplano ou *hyperslab* de erro zero e, na sequência, essa projeção em uma superfície de norma  $\ell_1$  definida pelo usuário.

Nesta seção, estamos propondo um algoritmo que aplica a filosofia das projeções sequenciais, porém, utilizando a norma  $\ell_0$  como restrição. Com o intuito de manter um baixo custo computacional, também não serão aplicadas aqui as estruturas que oneram os algoritmos da família APL1 [30], como o reúso de informações e os *hyperslabs*. Pode-se verificar ainda, na sequência de formulações a seguir, que não será necessária a utilização de qualquer mecanismo de manutenção do hiperquadrante no processo de adaptação, como



nos algoritmos propostos em [30]. Dessa forma, o algoritmo proposto aqui pode ser representado pela seguinte expressão geral:

$$\mathbf{w}(n+1) = P_{S_{\ell_0}(k)} \left[ \mathbf{w}(n) + \mu \left\{ P_{S_e(k)} [\mathbf{w}(n)] - \mathbf{w}(n) \right\} \right] \quad (3.83)$$

onde  $P_{S_e(k)}$  é a projeção do vetor de coeficientes no hiperplano de erro *a posteriori* igual a zero no instante  $k$  e  $P_{S_{\ell_0}(k)}(\cdot)$  é a projeção do vetor resultante da projeção anterior na superfície de norma  $\ell_0$  no instante  $k$ . Considerando a seguinte função de aproximação para a norma  $\ell_0$ , dada por

$$\begin{aligned} \|\mathbf{w}(n)\|_{0_Z} &= \sum_{k=0}^{N-1} 1 - \bar{z}_k(n) \\ &= N - \mathbf{1}^T \bar{\mathbf{z}}(n) \end{aligned} \quad (3.84)$$

sendo

$$\bar{z}_k(n) = \begin{cases} 1, & |w_k(n)| \leq \frac{1}{\beta} \\ 0, & \text{demais} \end{cases} \quad (3.85)$$

a equação de atualização do algoritmo é obtida, sendo escrita como

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mu \alpha_1(n) \mathbf{x}(n) - \Lambda_{z\ell_0}(n) \bar{\mathbf{s}}_z(n) \\ &= \mathbf{w}_{\text{NLMS}}(n) - \Lambda_{z\ell_0}(n) \bar{\mathbf{s}}_z(n). \end{aligned} \quad (3.86)$$

Em (3.86), o termo  $\mathbf{w}_{\text{NLMS}}(n) = \mathbf{w}(n) + \mu \alpha_1(n) \mathbf{x}(n)$  representa a primeira projeção, dada pela atualização do vetor de coeficientes utilizando o algoritmo NLMS. Já o termo  $\Lambda_{z\ell_0}(n) \bar{\mathbf{s}}_z(n)$  em (3.86), representa a projeção sequencial definida pela restrição de norma  $\ell_0$ , sendo o  $k$ -ésimo elemento de  $\bar{\mathbf{s}}_z(n)$  é definido por

$$\bar{s}_{zk}(n) = \bar{s}_k(n) \bar{z}_k(n) \quad (3.87)$$

onde  $\bar{s}_k(n)$  é o sinal de  $w_{\text{NLMS}k}(n)$ , que é o  $k$ -ésimo coeficiente do vetor  $\mathbf{w}_{\text{NLMS}}(n)$ , e

$$\Lambda_{z\ell_0}(n) = \begin{bmatrix} |w_{\text{NLMS}0}(n)| & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & |w_{\text{NLMS}(N-1)}(n)| \end{bmatrix}. \quad (3.88)$$

A partir de (3.83) - (3.88), verifica-se que a projeção do vetor de coeficientes na superfície de norma  $\ell_0$ , neste caso, pode ser efetivada simplesmente levando para zero os elementos do vetor de coeficientes após atualização pelo

algoritmo NLMS que possuem valor menor ou igual a  $1/\beta$ . Considerando tal característica, o algoritmo aqui proposto é denominado **algoritmo com restrição de norma  $\ell_0$  com coeficientes iguais a zero** (*zeroed-coefficients  $\ell_0$ -norm constrained algorithm*) - **Z- $\ell_0$ NC**. O princípio de operação do algoritmo proposto é ilustrado na Figura 23 para  $\mathbf{w} \in \mathbb{R}^2$ . Nesse exemplo, apenas um coeficiente do vetor ótimo  $\mathbf{w}^*$  é diferente de zero, sendo a superfície de norma  $\ell_0$  coincidente com o eixo  $w_1$ .

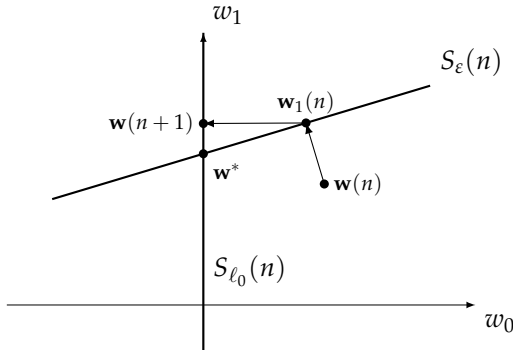


Figura 23: Princípio de operação do algoritmo Z- $\ell_0$ NC.

Na prática, o algoritmo Z- $\ell_0$ NC executa os seguintes passos:

- Projeta o vetor de coeficientes atual no hiperplano de erro *a posteriori* igual a zero, como o algoritmo NLMS.
- Verifica quais coeficientes do novo vetor possuem valores menores ou iguais a  $1/\beta$ , tornando-os iguais a zero.

Em termos de complexidade computacional, o algoritmo Z- $\ell_0$ NC acrescenta apenas  $N$  comparações (operação de baixo custo) ao algoritmo NLMS, sendo, então, um dos algoritmos desenvolvidos para sistemas esparsos com menor custo computacional, como pode-se verificar no sumário apresentado na Tabela 7. No sentido de reduzir ainda mais o custo computacional do algoritmo, pode-se prever a paralisação da adaptação de um coeficiente não-significativo a partir de um determinado número de iterações em que seu valor foi levado para zero. Assim, ao final da adaptação, apenas um pequeno número de coeficientes (definidos como significativos) continuam sendo adaptados. O Exemplo 3.8 (Seção 3.3.5) mostra o comportamento da adaptação de um coeficiente não-significativo utilizando o algoritmo Z- $\ell_0$ NC em comparação com o algoritmo V- $\ell_0$ NC.

Tabela 7: Sumário do algoritmo Z- $\ell_0$ NC

---

Inicialização

$$\mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

Fazer para  $n \geq 0$

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n)$$

$$\alpha_1(n) = e(n) / [\mathbf{x}^T(n) \mathbf{x}(n)]$$

$$\mathbf{w}_{\text{NLMS}}(n) = \mathbf{w}(n) + \mu \alpha_1(n) \mathbf{x}(n)$$

Para  $k$  de 0 até  $N - 1$

$$\text{Se } w_{\text{NLMS}k}(n) > 1/\beta, \text{ então, } w_k(n+1) = w_{\text{NLMS}k}(n)$$

$$\text{Se não, } w_k(n+1) = 0$$


---

### 3.3.5 Exemplos

Nesta seção, são apresentados exemplos que ilustram a discussão teórica sobre os algoritmos propostos baseados na norma  $\ell_0$ . Como no caso dos algoritmos  $\ell_1$ NC, são realizadas aqui simulações de Monte Carlo considerando o problema de identificação de sistemas da Tabela 3 (veja Seção 3.2.2).

**Exemplo 3.3.** Neste exemplo, é realizado um conjunto de simulações para o algoritmo  $\ell_0$ NC (com  $\mu = 1$  e  $\beta = 13$ ), considerando diferentes valores de  $\delta_{\ell_0}$ . Os resultados são mostrados na Figura 24. Verifica-se que, como no caso do algoritmo  $\ell_1$ NC com relação ao parâmetro  $\delta_{\ell_1}$ , existe uma grande sensibilidade deste algoritmo ao parâmetro  $\delta_{\ell_0}$ . No entanto, nesse caso, constata-se a atuação de dois fatores distintos no desempenho do algoritmo. Por um lado, na região determinada por a), o desempenho é muito ruim em função do grande desalinhamento provocado pelo fator  $N - N_z(n) \gg \delta_{\ell_0}$ , que ocorre desde o início da adaptação e não é recuperado ao longo do processo, determinando que coeficientes considerados significativos sejam atraídos para zero. Já na segunda região, determinada por b), o algoritmo se comporta como seria de se esperar, apresentando resultados piores à medida que o valor do parâmetro de restrição  $\delta_{\ell_0}$  se afasta de  $\|\mathbf{h}_1\|_0 = 4$ . Observa-se nesse exemplo (veja Figura 24) que a desestabilização do algoritmo [região identificada por a)] ocorre antes mesmo de se atingir o valor ideal da restrição de norma ( $\delta_{\ell_0} = 4$ ), obtendo-se o melhor resultado para  $\delta_{\ell_0} = 4,4$ .

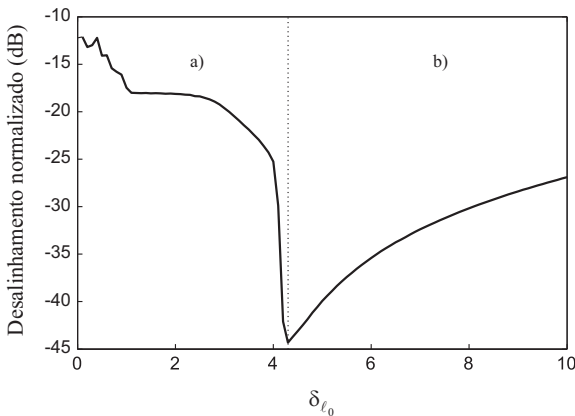


Figura 24: Exemplo 3.3. Desempenho em regime permanente do algoritmo  $\ell_0$ NC em função de  $\delta_{\ell_0}$ .

**Exemplo 3.4.** No exemplo anterior, o algoritmo  $\ell_0$ NC foi configurado com valor de  $\beta = 13$ , sendo essa escolha pautada em duas características, a saber. O limite  $1/\beta$  deve estar abaixo dos menores coeficientes significativos da planta, sob pena de serem suprimidos no filtro adaptado. No caso do exemplo, o menor coeficiente é igual a 0,1, portanto,  $\beta$  deve ser maior do que 10. Por outro lado, o valor de  $\beta$  não pode ser muito elevado a ponto de manter poucos elementos abaixo do limite  $1/\beta$  durante o processo de adaptação, determinando um valor de  $N_z(n)$  muito pequeno e, consequentemente,  $N - N_z \gg \delta_{\ell_0}$ , provocando grande desalinhamento entre o filtro adaptado e a planta. Nesse exemplo, o algoritmo  $\ell_0$ NC é configurado com  $\delta_{\ell_0}$  fixo igual a 4,4 (melhor valor observado no exemplo anterior), porém com valores de  $\beta$  variando de 1 até 50 (veja Figura 25). Observa-se, nessas simulações, que o algoritmo apresenta resultados satisfatórios dentro de uma faixa de valores de aproximadamente  $12 \leq \beta \leq 22$ . Abaixo dessa faixa, os coeficientes significativos de menor valor são identificados pelo algoritmo como coeficientes não-significativos. Já acima dessa faixa, o algoritmo entra na região de instabilidade em função do baixo valor de  $N_z$  ( $N - N_z \gg \delta_{\ell_0}$ ), resultando em desalinhamentos muito elevados.

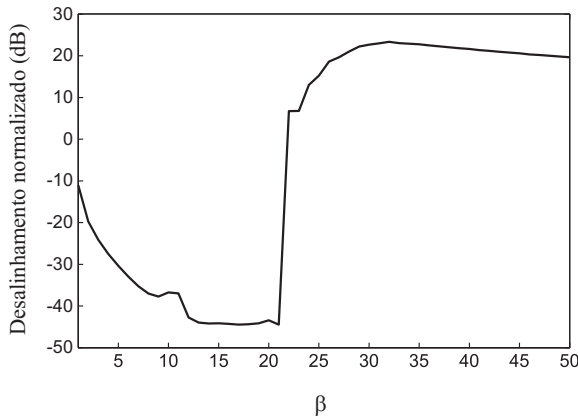


Figura 25: Exemplo 3.4. Desempenho em regime permanente do algoritmo  $\ell_0$ NC em função de  $\beta$ .

**Exemplo 3.5.** Em função da utilização da restrição variável (3.62), a escolha de  $\beta$  para o algoritmo V- $\ell_0$ NC depende de questões distintas daquelas estabelecidas para o algoritmo  $\ell_0$ NC. Nesse caso, não se corre o risco de se ter um  $\delta_{\ell_1}$  equivalente negativo, já que os coeficientes não significativos sempre são atraídos para zero. Por outro lado, devido a essa mesma atração para zero, é importante guardar maior distância entre o limite  $1/\beta$  e os coeficientes significativos. Nesse exemplo, o algoritmo V- $\ell_0$ NC é avaliado para diversos valores de  $\beta$ , com os resultados mostrados na Figura 26. Observa-se que três faixas diferentes de valores de  $\beta$ , com diferentes características de desempenho, se destacam. Na primeira, indicada por a), o desempenho do algoritmo é prejudicado consideravelmente devido ao fato de o limite  $1/\beta$  ser demasiado grande e, por conseguinte, alguns dos coeficientes diferentes de zero do filtro serem considerados não significativa e forçado a zero. A segunda faixa, que é indicada por b), corresponde aos valores dentro da faixa de, aproximadamente,  $70 \leq \beta \leq 550$  para o qual um desempenho superior do algoritmo é obtido (considerando uma margem de 3 dB em relação ao valor mínimo de desalinhamento normalizado obtido). Tal comportamento ocorre em função da adequada identificação dos coeficientes significativos e não significativos. Finalmente, no terceiro intervalo, indicado por c), o limite definido por  $1/\beta$  é tão pequeno que os elementos do filtro usados para modelar os coeficientes com valor zero da planta passam a ser tratados como significativos, até o ponto em que o algoritmo não mais se diferencie do NLMS.

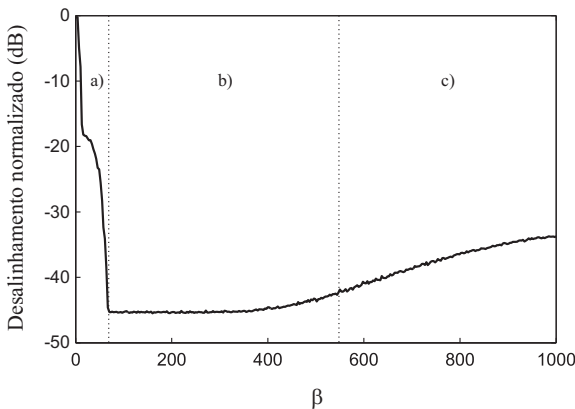


Figura 26: Exemplo 3.5. Desempenho em regime permanente do algoritmo V- $\ell_0$ NC em função de  $\beta$ .

**Exemplo 3.6.** Este exemplo apresenta uma comparação entre os desempenhos dos algoritmos  $\ell_0$ NC ( $\delta_{\ell_0} = 4,4$  e  $\beta = 13$ ) e V- $\ell_0$ NC ( $\beta = 120$ ), tendo o algoritmo NLMS como referência (veja Figura 27). Todos os algoritmos utilizam  $\mu = 1$ . O desempenho do algoritmo V- $\ell_0$ NC, de modo geral, é ainda melhor do que o desempenho do algoritmo  $\ell_0$ NC, com  $\delta_{\ell_0}$  fixo, considerando o exemplo em questão. Isso ocorre, basicamente, porque no algoritmo com restrição fixa de norma  $\ell_0$  a disparidade entre as normas  $\ell_0$  do vetor de coeficientes e  $\delta_{\ell_0}$  é muito grande no início do processo de adaptação, forçando o algoritmo a reduzir a amplitude dos coeficientes, incluindo os significativos que ainda têm valor abaixo de  $1/\beta$ . No algoritmo V- $\ell_0$ NC, como  $\delta_{\ell_0}(n)$  é também adaptado a cada iteração, as diferenças são bem menores, mesmo no início do processo, permitindo a adaptação mais adequada dos coeficientes significativos.

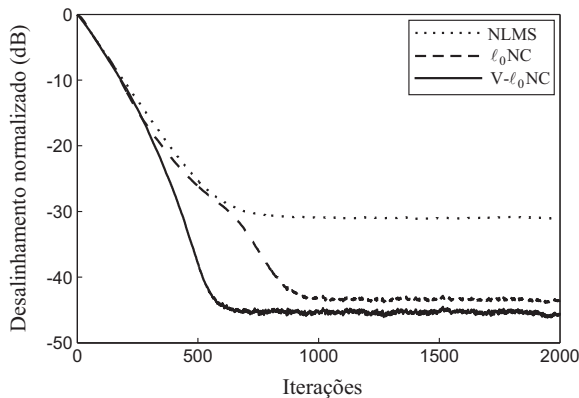


Figura 27: Exemplo 3.6. Comparação de desempenho entre os algoritmos NLMS,  $\ell_0$ NC e V- $\ell_0$ NC.

**Exemplo 3.7.** Neste exemplo, os desempenhos dos algoritmos  $V\text{-}\ell_0\text{NC}$  e  $VS\text{-}\ell_0\text{NC}$  são avaliados em regime permanente para um conjunto de simulações com valores diferentes de  $\beta$  (veja Figura 28). Observa-se claramente o efeito da suavização da restrição de norma proposta pelo algoritmo  $VS\text{-}\ell_0\text{NC}$ . A faixa de valores de  $\beta$  para a qual o desempenho do algoritmo  $VS\text{-}\ell_0\text{NC}$  é considerado superior inicia com  $\beta = 26$ , muito antes do algoritmo  $V\text{-}\ell_0\text{NC}$ , que tem desempenho melhor a partir de  $\beta = 70$ . Por outro lado, mais rapidamente os resultados do algoritmo  $VS\text{-}\ell_0\text{NC}$  se aproximam do NLMS com valores de  $\beta$  maiores. Tal característica também é consequência da suavização, visto que os elementos de menor valor, correspondentes aos coeficientes não significativos, são atraídos para zero com menor ênfase.

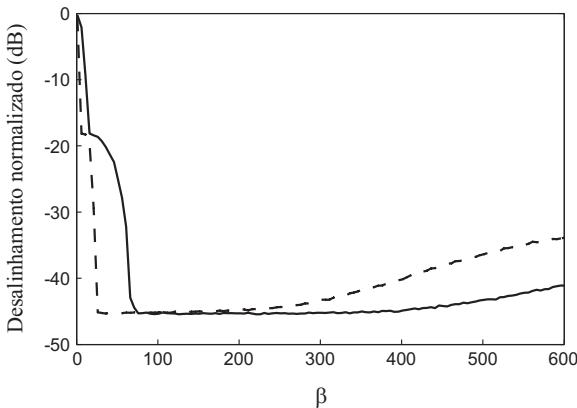


Figura 28: Exemplo 3.7. Desempenho em regime permanente dos algoritmos  $V\text{-}\ell_0\text{NC}$  (linha contínua) e  $VS\text{-}\ell_0\text{NC}$  (linha tracejada) em função de  $\beta$ .



**Exemplo 3.8.** O algoritmo  $Z\text{-}\ell_0\text{NC}$  caracteriza-se, fundamentalmente, pela redução para zero dos coeficientes que estão abaixo do limite  $1/\beta$ . Nesse exemplo, tal característica é avaliada em comparação com os algoritmos da família  $\ell_0\text{NC}$ . Na Figura 29 são apresentados os resultados de simulações dos algoritmos  $V\text{-}\ell_0\text{NC}$ ,  $VS\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$ . Observa-se claramente o melhor desempenho do algoritmo  $Z\text{-}\ell_0\text{NC}$  em regime permanente, considerando que a planta utilizada tem características favoráveis a esse algoritmo, com grande quantidade de coeficientes não significativos iguais a zero. A diferença de comportamento da adaptação dos coeficientes não significativos entre os algoritmos  $V\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$  pode ser observada ao acompanhar a adaptação de um único coeficiente não significativo (veja Figura 30, para o algoritmo  $V\text{-}\ell_0\text{NC}$ , e Figura 31, para o algoritmo  $Z\text{-}\ell_0\text{NC}$ ). Aqui, a adaptação do coeficiente da posição 90 é observada em uma única realização, com os dois algoritmos atuando simultaneamente. Observa-se que, em regime permanente, um pequeno ruído de adaptação é notado no coeficiente adaptado pelo algoritmo  $V\text{-}\ell_0\text{NC}$ , enquanto o coeficiente adaptado pelo algoritmo  $Z\text{-}\ell_0\text{NC}$  permanece igual a zero após a convergência.

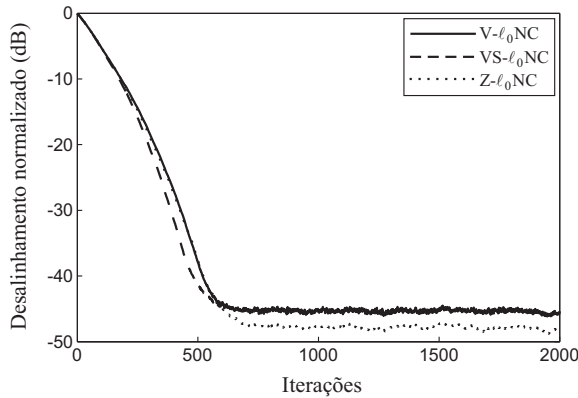


Figura 29: Exemplo 3.8. Comparação de desempenho entre os algoritmos  $V\text{-}\ell_0\text{NC}$ ,  $VS\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$ .

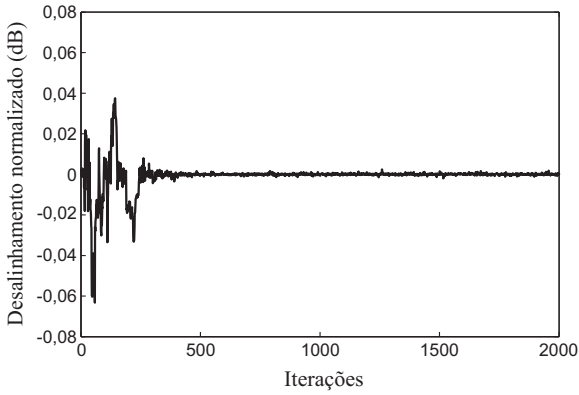


Figura 30: Exemplo 3.8. Evolução da adaptação do coeficiente não significativo da posição 90 para o algoritmo V- $\ell_0$ NC.

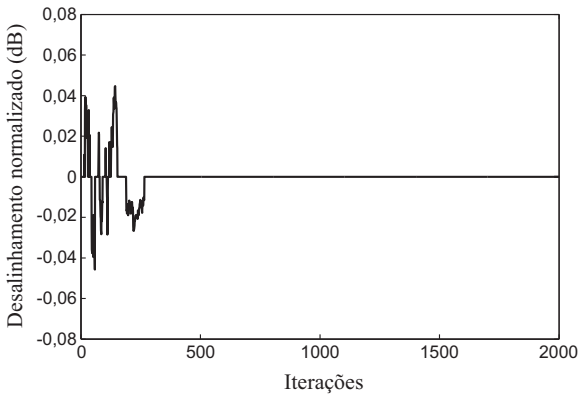


Figura 31: Exemplo 3.8. Evolução da adaptação de um coeficiente não significativo da posição 90 para o algoritmo Z- $\ell_0$ NC.

### 3.4 Considerações sobre os algoritmos propostos

Os algoritmos baseados em restrição de norma discutidos neste capítulo apresentam uma estrutura geral definida pela equação de adaptação (3.9), dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_p}(n) \lambda_{\ell_p}(n) \mathbf{x}(n) - \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n+1) \quad (3.89)$$

com  $\mathbf{g}_{\ell_p}(n+1) = \nabla[\|\mathbf{w}(n+1)\|_p]$ , conforme discutido na Seção 3.1. Observam-se três termos de atualização em (3.89). O primeiro termo corresponde à projeção ortogonal do vetor de coeficientes no hiperplano  $S_\varepsilon(n)$ , como no algoritmo NLMS. O segundo, estende ou contrai essa projeção de forma a manter o erro *a posteriori* igual a zero após a aplicação do último termo. O último termo de (3.89), por sua vez, atualiza o vetor na direção contrária ao gradiente da norma, gerando efeitos diferentes para cada grupo de algoritmos:

- Algoritmos  $\ell_1$ NC e V- $\ell_1$ NC. Para esses algoritmos o gradiente  $\mathbf{g}_{\ell_p}(n+1) \cong \mathbf{s}(n)$  impõe uma variação uniforme a todos os elementos do vetor, independentemente da magnitude dos coeficientes (como ilustrado na Figura 32 para  $\mathbf{w} \in \mathbb{R}^2$ ).
- Algoritmos  $\ell_0$ NC e V- $\ell_0$ NC. O gradiente  $\mathbf{g}_{\ell_p}(n+1) \cong \mathbf{s}_z(n)$  determina uma variação também de mesma magnitude, porém, atualizando apenas os coeficientes com módulos de valores menores do que  $1/\beta$  (não-significativos).
- Algoritmos S- $\ell_0$ NC e VS- $\ell_0$ NC. Nesse caso, o gradiente  $\mathbf{g}_{\ell_p}(n+1) \cong 2\beta \mathbf{z}_s(n)$  determina uma variação também apenas dos coeficientes considerados não-significativos, porém, essa variação não é uniforme, sendo os elementos atualizados de forma proporcional a  $1 - \beta|w_k|$ .
- Algoritmo Z- $\ell_0$ NC. Este algoritmo tem uma proposta diferenciada em relação aos demais aqui representados, seguindo a abordagem das projeções sequenciais. Nesse caso, os elementos referentes aos coeficientes não-significativos (menores do que  $1/\beta$ ) são todos levados para zero.

A partir dessas considerações, verifica-se que os algoritmos baseados em norma  $\ell_0$  produzem efeitos diferenciados na atualização dos coeficientes significativos e não-significativos, de forma análoga aos algoritmos proporcionais, como o algoritmo PNLMS e seus derivados. No entanto, tais algoritmos enfatizam a atualização dos coeficientes significativos em detrimento

daqueles de menor valor. Assim, proporcionam velocidades de convergência maiores inicialmente, porém, demorando a atingir níveis satisfatórios em regime permanente em função da baixa velocidade de convergência decorrente do grande número de coeficientes pouco significativos. Por outro lado, os algoritmos baseados em norma  $\ell_0$  aqui propostos tendem a reduzir rapidamente os coeficientes não-significativos sem prejudicar de maneira importante a convergência dos significativos, proporcionando velocidades de convergência e níveis de aproximação ao valor ótimo em regime permanente normalmente melhores que os proporcionais. Os algoritmos propostos baseados em norma  $\ell_1$ , por sua vez, por não distinguirem os coeficientes significativos e não-significativos na adaptação, poderão apresentar desempenho não tão elevado. Como vantagem, no entanto, os algoritmos  $\ell_1$ NC apresentam complexidade computacional menor (veja Tabela 8, aliada à independência de parâmetros a serem definidos pelo usuário.

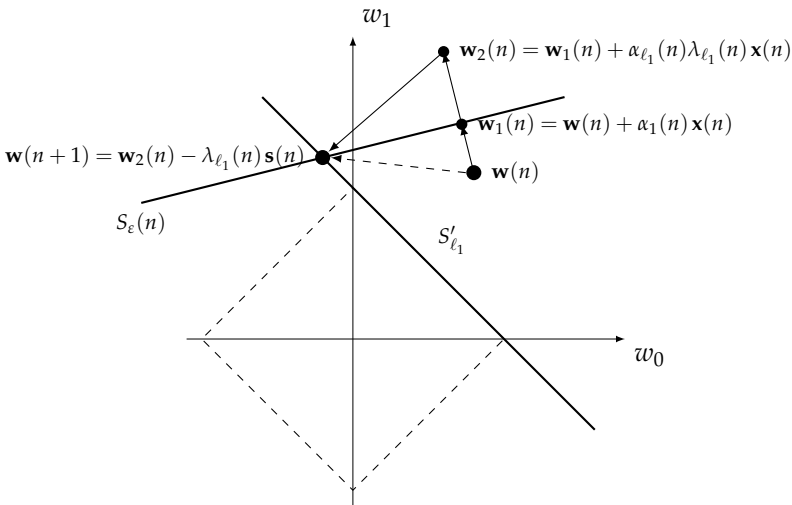


Figura 32: Etapas do processo de adaptação do algoritmo  $\ell_1$ NC.

### 3.4.1 Sobre a complexidade computacional dos algoritmos propostos

A complexidade computacional de um algoritmo é um quesito muito importante de avaliação de seu desempenho. Um menor número de operações e a menor complexidade de cada operação envolvida implica em me-

nor exigência de requisitos computacionais e, portanto, maior velocidade de execução. Os algoritmos propostos neste trabalho apresentam características de complexidade computacional que os colocam em posição de vantagem em comparação com diversos algoritmos desenvolvidos para a identificação de sistemas esparsos (veja Tabela 8). Mesmo apresentando expressões mais extensas que as de outros algoritmos, como se pode observar nas Tabelas 2, 5 e 6, os algoritmos aqui propostos possuem carga computacional reduzida pelo fato de apresentarem diversas operações em que vetores de sinais estão presentes, definindo operações de simples troca de sinal. Sua complexidade computacional, dessa forma, é semelhante a de algoritmos para sistemas esparsos com formulação mais simples, como o PNLMS, e menor do que a de outros algoritmos consagrados na literatura, como IPNLMS, MPNLMS, RZA-NLMS e APWL1. Nesses dois últimos casos, a complexidade computacional é ainda maior devido ao elevado número de divisões realizadas a cada iteração. Além disso, os algoritmos da família  $\ell_0$ NC podem dispensar o cálculo de diversos elementos do vetor que envolvem multiplicação por zero. No caso do algoritmo Z- $\ell_0$ NC, além de possuir complexidade semelhante ao do algoritmo NLMS, ainda pode ter a carga computacional reduzida ao longo do processo adaptativo, à medida que os coeficientes não significativos tornam-se iguais a zero e não precisam mais ser adaptados.

Tabela 8: Complexidade computacional de alguns algoritmos para sistemas esparsos, em ordem de menor para maior número de multiplicações

| Algoritmo       | Multipl. | Somas           | Div.    | Compar. | Mod/Sgn  | Mem. |
|-----------------|----------|-----------------|---------|---------|----------|------|
| NLMS            | $3N + 1$ | $3N$            | 1       | 0       | 0        | $N$  |
| Z- $\ell_0$ NC  | $3N + 1$ | $3N$            | 1       | $N$     | 0        | $N$  |
| V- $\ell_1$ NC  | $3N + 6$ | $5N$            | 3       | 0       | $2N + 1$ | $2N$ |
| ZA-NLMS         | $4N + 1$ | $4N$            | 1       | 0       | $N$      | $2N$ |
| PNLMS           | $4N + 3$ | $4N - 1$        | 1       | $2N$    | $N$      | $N$  |
| $\ell_0$ -NLMS  | $4N + 4$ | $4N$            | 1       | $N$     | $N$      | $2N$ |
| APL1            | $4N + 2$ | $5N + 1$        | 2       | 0       | $N$      | $3N$ |
| V- $\ell_0$ NC  | $4N + 6$ | $5N + 2N_z + 2$ | 3       | $N$     | $3N_z$   | $3N$ |
| VS- $\ell_0$ NC | $4N + 6$ | $5N + 2N_z + 2$ | 3       | $N$     | $3N_z$   | $3N$ |
| IPNLMS          | $5N + 3$ | $4N + 4$        | 2       | 0       | $N$      | $2N$ |
| MPNLMS          | $5N + 3$ | $4N - 1$        | 1       | $3N$    | $N$      | $N$  |
| RZA-NLMS        | $5N + 1$ | $5N$            | $N + 1$ | 0       | $N$      | $3N$ |
| APWL1           | $5N + 2$ | $6N + 1$        | $N + 2$ | 0       | $N$      | $3N$ |

### 3.4.2 Sobre a convergência dos algoritmos $\ell_p$ NC

Considerando (3.89) que, em situação de ausência de ruído, leva a projeção do vetor de coeficientes para um hiperplano de erro *a posteriori* igual a zero, pode-se escrever

$$\varepsilon(n) = d(n) - \mathbf{x}^T(n)[\mathbf{w}(n) + \Delta_w] = 0 \quad (3.90)$$

com  $\Delta_w$  representando a expressão dentro dos colchetes do lado direito de (3.89). De (1.1) e (3.90), pode-se inferir que  $e(n) = \mathbf{x}^T(n)\Delta_w$  e assim, para  $\mu = 2$ , tem-se

$$\begin{aligned} \varepsilon(n) &= d(n) - \mathbf{x}^T(n)[\mathbf{w}(n) + 2\Delta_w] \\ &= d(n) - \mathbf{x}^T(n)[\mathbf{w}(n) + \Delta_w] - \mathbf{x}^T(n)\Delta_w \\ &= -e(n) \end{aligned} \quad (3.91)$$

o que implica  $\varepsilon^2(n) = e^2(n)$ . Além disso, para  $\mu = 0$ ,  $\mathbf{w}(n+1) = \mathbf{w}(n)$  e conseqüentemente,  $\varepsilon^2(n) = e^2(n)$ . Portanto, considerando que  $\varepsilon^2(n)$  é uma função convexa de  $\mathbf{w}(n+1)$ , pode-se verificar que  $\varepsilon^2(n) \leq e^2(n)$  para  $0 \leq \mu \leq 2$ . Essa característica é típica dos algoritmos baseados no NLMS e, portanto, observa-se que os algoritmos propostos têm semelhanças importantes com o NLMS em termos de convergência e estabilidade. Uma diferença importante entre os algoritmos propostos e o NLMS, no entanto, é que este último apresenta convergência monotônica, ou seja,  $[\mathbf{w}(n+1) - \mathbf{w}_o] < [\mathbf{w}(n) - \mathbf{w}_o]$  para todo  $n$  (considerando a ausência de ruído), enquanto os algoritmos propostos podem não ter esse tipo de comportamento em todas as iterações. Para ilustrar essa característica, vamos considerar a modelagem de um sistema desconhecido cujo vetor de coeficientes é representado por  $\mathbf{w}_o$ . Nesse caso, o vetor *a posteriori*  $\mathbf{w}(n+1)$  obtidos pelos algoritmos propostos é geralmente mais próximo de  $\mathbf{w}_o$  do que  $\mathbf{w}(n)$  devido ao fato que  $\mathbf{w}(n+1)$  é obtido por meio de uma projeção ortogonal de  $\mathbf{w}(n)$  em um conjunto convexo que geralmente contém  $\mathbf{w}_o$ . No entanto, devido às aproximações de norma consideradas, o conjunto solução convexa pode não conter  $\mathbf{w}_o$  em algumas das iterações do algoritmo e, portanto, a distância euclidiana pode não ser reduzida. Como verificado através de simulações numéricas, essas poucas iterações do algoritmo (em que a distância euclidiana para  $\mathbf{w}_o$  não é reduzida), em geral, não ameaçam o processo de convergência do algoritmo.

### 3.5 Conclusões

Este capítulo apresentou novas propostas de algoritmos orientados a sistemas esparsos. Os algoritmos aqui propostos são baseados em restrições de norma vetorial, sendo apresentadas versões baseadas em norma  $\ell_1$  e norma  $\ell_0$ . Uma formulação geral apresenta diretrizes para a aplicação das normas  $\ell_1$  e  $\ell_0$  utilizadas, além de outras métricas de restrição que possam vir a ser estabelecidas. A formulação proposta é baseada na otimização com múltiplas restrições, aplicando-se o método dos multiplicadores de Lagrange para a determinação dos algoritmos. O uso simultâneo das restrições linear e de norma determinada pelo método é um diferencial da proposta com relação a outras propostas baseadas em normas já publicadas, procurando manter o vetor de soluções sobre o hiperplano de erro *a posteriori* igual a zero a cada iteração. Além disso, em contraste com os algoritmos baseados nas projeções de norma apresentados no Capítulo 2, os algoritmos aqui propostos se aproximam dos algoritmos baseados no NLMS em termos de complexidade computacional, além de apresentarem menor número de parâmetros a serem definidos pelo usuário. Alternativamente, um novo algoritmo baseado na formulação de projeções sequenciais foi apresentado, tendo a norma  $\ell_0$  como restrição. Como característica principal, esse algoritmo apresenta custo computacional muito próximo ao do algoritmo NLMS, com desempenho semelhante aos demais algoritmos de norma  $\ell_0$ . No próximo capítulo são apresentadas simulações realizadas para avaliar os novos algoritmos em situações distintas, comparando-os com outros algoritmos projetados para sistemas esparsos. No Capítulo 5 são propostos modelos estocásticos para os algoritmos aqui desenvolvidos, buscando-se analisar em detalhes seu comportamento analítico.





## Capítulo 4

# Avaliação de Desempenho dos Algoritmos Propostos

Neste capítulo, resultados de simulações são apresentados com o objetivo de avaliar o desempenho dos algoritmos propostos em comparação com outros algoritmos adaptativos voltados para a identificação de sistemas esparsos, como os descritos na Seção 2.4. Cenários distintos são utilizados nas simulações, visando obter uma ampla avaliação dos algoritmos.

### 4.1 Definição dos cenários para as simulações

As simulações são realizadas considerando a estrutura adaptativa para identificação de sistemas ilustrada na Figura 2 (veja Capítulo 1) e utilizando as seguintes plantas, de acordo com a aplicação desejada:

- Planta esparsa artificial ( $\mathbf{h}_1$ ). A mesma planta utilizada nos exemplos do Capítulo 3 [24], possui resposta ao impulso com 100 coeficientes, apresentando coeficientes significativos com valores  $\{0,1 \ 1,0 \ -0,5 \ 0,1\}$  localizados nas posições  $\{1 \ 30 \ 35 \ 38\}$  do vetor. Os demais coeficientes possuem valores iguais a zero. O grau de esparsidade dessa planta [dado por (2.49)] é  $\mathcal{S}(\mathbf{h}_1) = 0,9435$ .
- Planta esparsa artificial com todos os coeficientes diferentes de zero ( $\mathbf{h}_2$ ). Semelhante à planta  $\mathbf{h}_1$ , possui os mesmos coeficientes significativos daquela planta, porém, nesse caso, os coeficientes considerados como não-significativos possuem valores gerados aleatoriamente, com média zero e variância igual a  $1 \times 10^{-5}$ . O grau de esparsidade dessa planta é  $\mathcal{S}(\mathbf{h}_2) = 0,9219$ .
- Planta quase-esparsa artificial ( $\mathbf{h}_3$ ). Também possui vetor de coeficientes de comprimento igual a 100, porém, com 20 coeficientes diferentes de zero localizados entre as posições 30 e 49. Os valores dos coeficientes diferentes de zero são  $\{-0,13 \ 0,23 \ 0,11 \ -0,47 \ 0,18 \ -0,16 \ -0,42 \ -0,09 \ -0,13 \ -0,15 \ 0,12 \ 0,09 \ -0,07 \ 0,52 \ -0,09 \ 0,18 \ -0,22 \ 0,12 \ 0,08 \ -0,25\}$ , resultando em um grau de esparsidade de  $\mathcal{S}(\mathbf{h}_3) = 0,6989$ .

- Planta esparsa realística ( $\mathbf{h}_4$ ). Nesse caso, a resposta ao impulso (planta) é representativa de um caminho de eco padronizada pela *International Telecommunications Union*, definido pela recomendação ITU-T G.168 [17] (modelo #1, 64 coeficientes), com acréscimo de 100 coeficientes de valor zero no início da resposta ao impulso e outros 348 coeficientes no final, perfazendo 512 coeficientes, como ilustrado pela Figura 33. O grau de esparsidade dessa planta é  $\mathcal{S}(\mathbf{h}_4) = 0,9281$ . Esse modelo de planta vem sendo largamente utilizado na literatura para avaliação de algoritmos dedicados a sistemas esparsos [9], [16], [21]-[23].
- Planta quase-esparsa realística ( $\mathbf{h}_5$ ). Uma planta com grau de esparsidade médio [ $\mathcal{S}(\mathbf{h}_5) = 0,7253$ ] também representativa de um caminho de eco padronizada pela *International Telecommunications Union*, definido pela recomendação ITU-T G.168 (modelo #4, 128 coeficientes), com acréscimo de 100 coeficientes de valor zero no início da resposta ao impulso e outros 284 coeficientes no final, perfazendo 512 coeficientes, como ilustrado pela Figura 34.

O sinal de entrada  $x(n)$  utilizado nas simulações é obtido a partir de um processo AR(2) [38] com variância  $\sigma_x^2 = 1$ , dado por

$$x(n) = b_1 x(n-1) + b_2 x(n-2) + v(n) \quad (4.1)$$

onde  $v(n)$  é um ruído gaussiano branco. Dois tipos de sinais de entrada são utilizados nas simulações, a saber:

- Sinal de entrada gaussiano branco, com variância igual a 1, sendo  $b_1 = 0$ ,  $b_2 = 0$ .
- Sinal de entrada gaussiano com alta dispersão de autovalores da matriz de autocorrelação de entrada<sup>1</sup> (correlacionado), com  $\sigma_v^2 = 0,29$ ; nesse caso  $b_1 = 0,80$ ,  $b_2 = -0,80$ , para  $\chi(\mathbf{R}) = 211$ .

Um ruído branco de medição  $r(n)$ , com variância  $\sigma_r^2 = 10^{-3}$  (caracterizando uma razão sinal-ruído de 30 dB, entre o sinal de entrada e o ruído de medição), é adicionado à saída do sistema a ser identificado.

Conforme especificado no Capítulo 1, a avaliação do desempenho dos algoritmos neste trabalho de pesquisa é realizada através do desalinhamento

---

<sup>1</sup>A dispersão é calculada por  $\chi(\mathbf{R}) = \frac{\lambda_{max}}{\lambda_{min}}$ , onde  $\lambda_{max}$  é o maior autovalor e  $\lambda_{min}$  é o menor autovalor da matriz de autocorrelação de entrada.

normalizado, dado por

$$\kappa(n) = 10 \log_{10} \frac{\|\mathbf{w}_o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}_o\|_2^2} \quad (4.2)$$

onde  $\mathbf{w}(n)$  é o vetor de coeficientes do filtro no instante  $n$  obtido através da média de simulações de Monte Carlo com 100 realizações independentes.

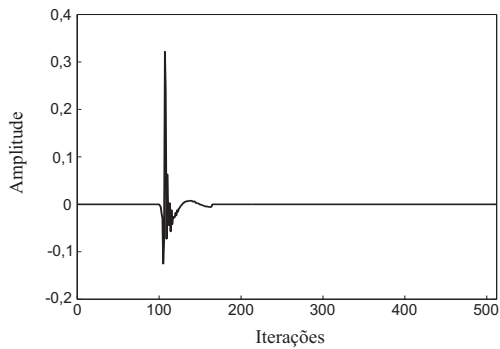


Figura 33: Resposta ao impulso representativa de um caminho de eco definida pela Recomendação ITU-T G.168, modelo #1, com 512 coeficientes.

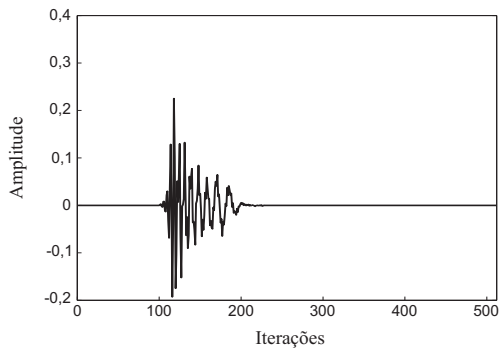


Figura 34: Resposta ao impulso representativa de um caminho de eco definida pela Recomendação ITU-T G.168, modelo #4, com 512 coeficientes.

## 4.2 Resultados de simulações

Nesta seção, são apresentados os resultados de simulações comparando o desempenho dos algoritmos propostos com de outros algoritmos concorrentes na identificação de sistemas esparsos, considerando tanto a classe proporcional quanto os algoritmos baseados em normas vetoriais (veja Seção 2.4). Os parâmetros dos algoritmos concorrentes a serem definidos pelo projetista são aqui ajustados com o objetivo de alcançar um desalinhamento normalizado em regime permanente com mesmo nível ao dos algoritmos propostos neste trabalho. Nesse contexto, alguns dos parâmetros, tais como  $\varrho$  e  $\delta_p$  dos algoritmos da classe proporcional [9], são fixados com seus valores típicos, enquanto outros parâmetros são ajustados através de um procedimento de busca através de repetidas simulações de Monte Carlo. No caso dos algoritmos APL1 e APWL1 [30], o número de *hyperlabs* é definido igual a 1 ( $Q = 1$ ), com espessura  $\phi = 0$ . Além disso, para todos os algoritmos, são utilizados parâmetros de regularização [4] com valores iguais a  $10^{-4}$ .

### 4.2.1 Algoritmos baseados na norma $\ell_1$

Nos exemplos a seguir, o algoritmo V- $\ell_1$ NC é avaliado em comparação com os algoritmos ZA-NLMS [27] e APL1 [30], ambos baseados na norma  $\ell_1$ , considerando plantas com características distintas. Para todos os exemplos, o parâmetro de passo do algoritmo V- $\ell_1$ NC é escolhido com valor unitário ( $\mu = \mu_1 = \mu_2 = \mu_3 = 1$ ), enquanto os parâmetros dos demais algoritmos são obtidos por busca extensiva.

**Exemplo 4.1.** Nesse exemplo, os algoritmos NLMS (apresentado como referência), V- $\ell_1$ NC, ZA-NLMS e APL1 são avaliados por simulações de Monte Carlo, utilizando a planta  $\mathbf{h}_1$  (com  $\|\mathbf{h}_1\|_1 = 1,7$ ), tendo como entrada um sinal gaussiano branco. Para essa simulação os algoritmos são configurados de acordo com a Tabela 9, sendo os resultados de simulação apresentados na Figura 35. Observa-se um melhor desempenho obtido pelo algoritmo V- $\ell_1$ NC quando comparado com outros algoritmos baseados na norma  $\ell_1$ . Deve-se destacar aqui que, no caso do algoritmo APL1, utilizou-se a mesma configuração já adotada para o algoritmo  $\ell_1$ NC (com  $\delta_{\ell_1}$  fixo), obtendo resultado semelhante a este último algoritmo (veja Exemplo 3.2). É importante lembrar que o ajuste do valor exato de  $\delta$  depende do conhecimento prévio da norma  $\ell_1$  da planta.

Tabela 9: Exemplo 4.1. Parâmetros de configuração dos algoritmos, considerando a planta  $\mathbf{h}_1$

| Algoritmo      | $\mu$ | $\delta$ | $\rho$             | $\varepsilon$ |
|----------------|-------|----------|--------------------|---------------|
| NLMS           | 0,55  |          |                    |               |
| ZA-NLMS        | 0,8   |          | $1 \times 10^{-5}$ | 10            |
| APL1           | 1     | 1,78     |                    |               |
| V- $\ell_1$ NC | 1     |          |                    |               |

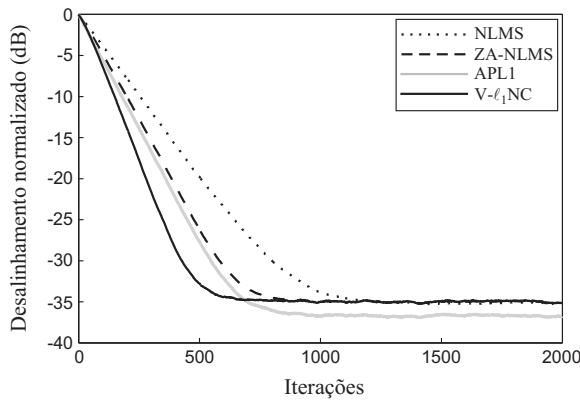


Figura 35: Exemplo 4.1. Curvas de desalinamento normalizado considerando a planta  $\mathbf{h}_1$ .

**Exemplo 4.2.** Nessa simulação, utilizando a planta esparsa realística  $\mathbf{h}_4$  (sendo  $\|\mathbf{h}_4\|_1 = 1,46$ ) e sinal de entrada gaussiano branco, os algoritmos são configurados conforme a Tabela 10. Os resultados dessa simulação são apresentados na Figura 36. Observa-se que os algoritmos baseados na norma  $\ell_1$  continuam com desempenho melhor do que o NLMS, sendo o desempenho do algoritmo V- $\ell_1$ NC ainda melhor do que o dos demais algoritmos.

**Exemplo 4.3.** Considera-se agora nesse exemplo uma simulação utilizando a planta quase-esparsa  $\mathbf{h}_5$  (com  $\|\mathbf{h}_5\|_1 = 4,03$ ), sendo os algoritmos configurados conforme a Tabela 11. Observa-se dos resultados obtidos (veja Figura 37) que os algoritmos V- $\ell_1$ NC, ZA-NLMS e APL1 apresentam desempenhos muito próximos (com pequena vantagem do algoritmo V- $\ell_1$ NC), porém, todos ainda melhores do que o algoritmo NLMS.

Tabela 10: Exemplo 4.2. Parâmetros de configuração dos algoritmos, considerando a planta  $\mathbf{h}_4$

| Algoritmo      | $\mu$ | $\delta$ | $\rho$             | $\varepsilon$ |
|----------------|-------|----------|--------------------|---------------|
| NLMS           | 0,65  |          | $1 \times 10^{-6}$ | 20            |
| ZA-NLMS        | 0,85  |          |                    |               |
| APL1           | 1     | 1,6      |                    |               |
| V- $\ell_1$ NC | 1     |          |                    |               |

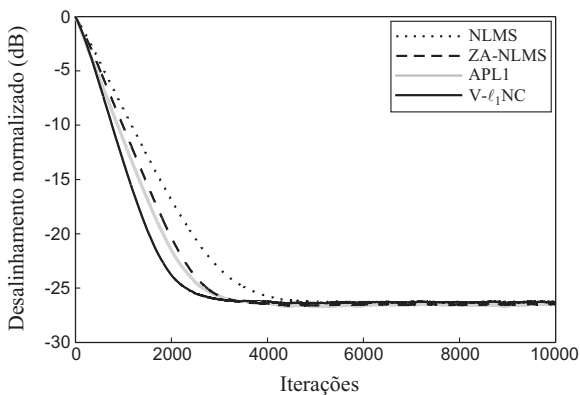


Figura 36: Exemplo 4.2. Curvas de desalinhamento normalizado considerando a planta  $\mathbf{h}_4$ .

Tabela 11: Exemplo 4.3. Parâmetros de configuração dos algoritmos, considerando a planta  $\mathbf{h}_5$

| Algoritmo      | $\mu$ | $\delta$ | $\rho$             | $\varepsilon$ |
|----------------|-------|----------|--------------------|---------------|
| NLMS           | 0,75  |          | $1 \times 10^{-6}$ | 20            |
| ZA-NLMS        | 0,9   |          |                    |               |
| APL1           | 1     | 4,2      |                    |               |
| V- $\ell_1$ NC | 1     |          |                    |               |

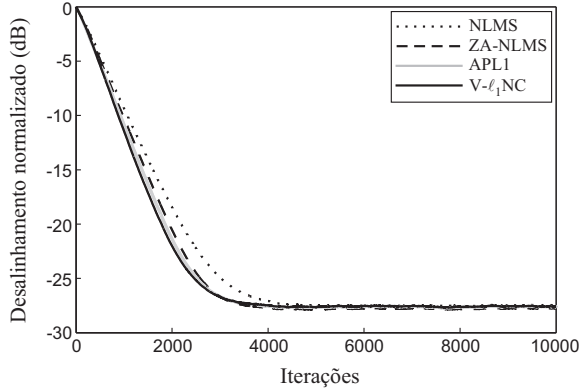


Figura 37: Exemplo 4.3. Curvas de desalinhamento normalizado considerando a planta  $\mathbf{h}_5$ .

## 4.2.2 Variação abrupta do grau de esparsidade

No capítulo anterior, foram discutidos os efeitos negativos dos parâmetros de restrição de norma fixos no desempenho dos algoritmos  $\ell_1$ NC e  $\ell_0$ NC (veja Exemplos 3.1 e 3.3). Tais efeitos motivaram o desenvolvimento de algoritmos com restrição de norma variável, a saber: os algoritmos V- $\ell_1$ NC e V- $\ell_0$ NC. No exemplo a seguir, o comportamento desses algoritmos é avaliado considerando alterações do grau de esparsidade da planta a ser identificada, comparando seu desempenho com outros algoritmos concorrentes.

**Exemplo 4.4.** Nesse exemplo, os algoritmos com restrição de norma variável propostos são comparados com outros algoritmos para sistemas esparsos, destacando-se os algoritmos APL1 e APWL1 (que possuem parâmetros de restrição de norma fixos), considerando uma variação abrupta no grau de esparsidade da planta a ser identificada. São utilizadas nesse exemplo as plantas  $\mathbf{h}_1$ , na primeira parte da adaptação, e  $\mathbf{h}_3$ , na segunda parte da adaptação (simulando uma perturbação do sistema). Os parâmetros de configuração dos algoritmos utilizados são relacionados na Tabela 12, ressaltando o uso de parâmetros de passo de adaptação iguais para os algoritmos  $\ell_p$ NC, isto é,  $\mu = \mu_1 = \mu_2 = \mu_3$ . Nessa avaliação, os algoritmos são ajustados para obter o mesmo desalinhamento em regime permanente daquele obtido pelo algoritmo V- $\ell_0$ NC com passo unitário ( $\mu = 1$ ), considerando a planta  $\mathbf{h}_1$ . Os resultados de simulação são ilustrados nas Figuras 38 e 39. Na Figura 38, são mostrados os algoritmos SPNLMS [45],  $\ell_0$ -NLMS [31], APL1 [30], APWL1 [30], V- $\ell_1$ NC e V- $\ell_0$ NC, com a perturbação (substituição da planta  $\mathbf{h}_1$  pela planta

$\mathbf{h}_3$ ) ocorrendo na iteração  $n = 6000$ , enquanto na Figura 39, são apresentados os resultados para os algoritmos  $V\text{-}\ell_0\text{NC}$ ,  $VS\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$ , com a perturbação ocorrendo na iteração  $n = 1000$ . Observa-se na Figura 38, que os algoritmos baseados na norma  $\ell_0$  (incluindo-se os algoritmos  $\ell_0\text{-NLMS}$  e  $\text{APWL1}$ ) exibem melhor desempenho do que os demais para a planta  $\mathbf{h}_1$ . No entanto, com a aplicação da perturbação, os resultados obtidos para cada algoritmo variam bastante, sendo que os algoritmos  $V\text{-}\ell_1\text{NC}$  e  $V\text{-}\ell_0\text{NC}$  apresentam, após a perturbação, um desempenho em regime permanente ligeiramente inferior ao apresentado antes da perturbação. Por outro lado, como esperado, os desempenhos dos algoritmos com restrição de norma fixa ( $\text{APL1}$  e  $\text{APWL1}$ ) são extremamente prejudicados com a aplicação da perturbação, considerando que as normas  $\ell_1$  e  $\ell_0$  da planta  $\mathbf{h}_3$  são diferentes dos parâmetros de restrição de norma  $\delta$  previamente ajustados. Comparando agora apenas os algoritmos baseados na norma  $\ell_0$ , na Figura 39, pode-se observar desempenhos aproximados. O algoritmo  $Z\text{-}\ell_0\text{NC}$ , que possui o melhor resultado em regime permanente na primeira parte da adaptação é, no entanto, o algoritmo que mais sofreu o efeito da perturbação.

Tabela 12: Exemplo 4.4. Parâmetros de configuração dos algoritmos

| Algoritmo                   | $\mu$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ | $\kappa$           |
|-----------------------------|-------|---------|----------|--------|------------|--------------------|
| SPNLMS                      | 0,09  |         |          | 0,01   | 0,01       |                    |
| $\ell_0\text{-NLMS}$        | 1     | 40      |          |        |            | $8 \times 10^{-6}$ |
| APL1                        | 0,13  |         | 1,74     |        |            |                    |
| APWL1                       | 1     |         | 15       |        |            |                    |
| $V\text{-}\ell_1\text{NC}$  | 0,15  |         |          |        |            |                    |
| $V\text{-}\ell_0\text{NC}$  | 1     | 120     |          |        |            |                    |
| $VS\text{-}\ell_0\text{NC}$ | 1     | 55      |          |        |            |                    |
| $Z\text{-}\ell_0\text{NC}$  | 1     | 120     |          |        |            |                    |



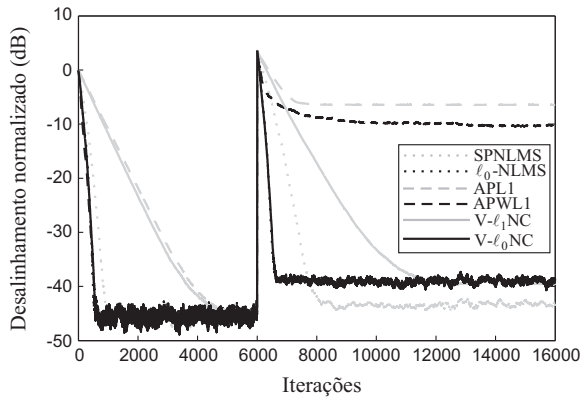


Figura 38: Exemplo 4.4. Curvas de desalinhamento normalizado (perturbação na iteração  $n = 6000$ ).

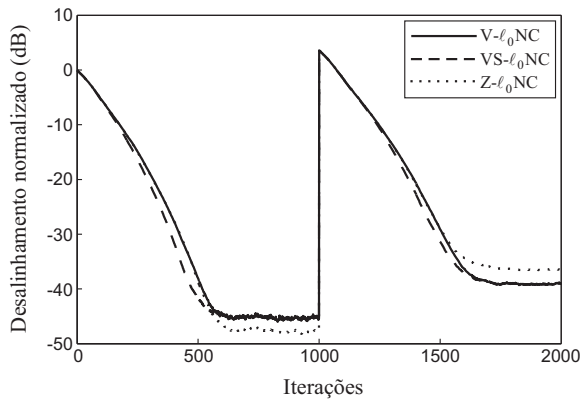


Figura 39: Exemplo 4.4. Curvas de desalinhamento normalizado (perturbação na iteração  $n = 1000$ ).

### 4.2.3 Identificação de plantas com ruído agregado

Conforme discutido na Seção 1.2, uma planta esparsa caracteriza-se pela concentração da energia em uma pequena quantidade de elementos de sua resposta ao impulso, sendo a maior parte dos coeficientes nulos ou com amplitude desprezível. Os algoritmos propostos baseados na norma  $\ell_0$  utilizam tal característica a seu favor, forçando para zero os componentes não significativos. A fim de verificar o impacto dessa estratégia no desempenho do algoritmo, o Exemplo 4.5 analisa seu comportamento, em comparação com outros algoritmos dedicados à identificação de sistemas esparsos, utilizando plantas com pequenas variações nos valores dos coeficientes não significativos.

**Exemplo 4.5.** Nesse exemplo, os algoritmos da família  $\ell_0$ NC são avaliados utilizando-se duas plantas semelhantes,  $\mathbf{h}_1$  e  $\mathbf{h}_2$ , diferenciadas pelo fato de que a primeira possui todos os coeficientes não significativos iguais a zero, enquanto na segunda é adicionado um pequeno nível de ruído aos coeficientes não significativos (veja Seção 4.1). Os desempenhos dos algoritmos V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC é comparado ao do algoritmo NLMS (como referência aos demais) e ao de outros algoritmos dedicados a sistemas esparsos, a saber: os algoritmos V- $\ell_1$ NC e APL1 [30] (representando os algoritmos baseados em norma  $\ell_1$ ), APWL1 [30] e SPNLMS [45]. Para a simulação considerando a planta  $\mathbf{h}_1$ , todos os algoritmos são configurados com passos de adaptação unitários, sendo os demais parâmetros ajustados por busca extensiva (veja Tabela 13). Os resultados de simulação realizada com a planta  $\mathbf{h}_1$  são apresentados na Figura 40. Observa-se que os algoritmos baseados na norma  $\ell_0$ , assim como o algoritmo APWL1 (que, por sua característica de ponderação dos coeficientes, aproxima-se dos algoritmos baseados na norma  $\ell_0$  [30]), possuem desempenho em regime permanente muito superiores às dos demais algoritmos [veja Figura 40(a)], podendo-se destacar, dos algoritmos de norma  $\ell_0$  propostos, o algoritmo Z- $\ell_0$ NC [veja Figura 40(b)].

Na simulação utilizando a planta  $\mathbf{h}_2$ , os mesmos algoritmos considerados na simulação anterior são utilizados, mantendo as configurações da Tabela 13. Os resultados dessa simulação são ilustrados na Figura 41, onde observa-se a clara degradação do desempenho dos algoritmos baseados na norma  $\ell_0$ , ao forçar para zero os componentes do filtro adaptativo referentes aos coeficientes não significativos da planta (que agora não são mais iguais a zero). No entanto, não apenas tais algoritmos, mas também os algoritmos SPNLMS e APL1 são penalizados nessa situação. Por outro lado, é importante ressaltar que o algoritmo V- $\ell_1$ NC mantém seu desempenho com muito pouca variação, sendo este último o algoritmo que apresentou a menor sen-

sibilidade à variação dos valores dos coeficientes não significativos dentre os algoritmos avaliados.

Tabela 13: Exemplo 4.5. Parâmetros de configuração dos algoritmos

| Algoritmo       | $\mu$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ |
|-----------------|-------|---------|----------|--------|------------|
| NLMS            | 1     |         |          |        |            |
| SPNLMS          | 1     |         |          | 0,01   | 0,01       |
| APL1            | 1     |         | 1,78     |        |            |
| APWL1           | 1     |         | 8        |        |            |
| V- $\ell_1$ NC  | 1     |         |          |        |            |
| V- $\ell_0$ NC  | 1     | 120     |          |        |            |
| VS- $\ell_0$ NC | 1     | 55      |          |        |            |
| Z- $\ell_0$ NC  | 1     | 120     |          |        |            |

#### 4.2.4 Identificação de plantas realísticas

Os algoritmos baseados na norma  $\ell_0$  propostos neste trabalho (V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC) possuem características importantes que os destacam quando comparados ao algoritmo V- $\ell_1$ NC e outros algoritmos desenvolvidos com a finalidade de identificar sistemas esparsos, como observado nos resultados apresentados nos exemplos anteriores. Nos exemplos a seguir, esses algoritmos são agora avaliados na identificação de plantas mais próximas de aplicações práticas reais (utilizando as plantas realísticas  $\mathbf{h}_4$  e  $\mathbf{h}_5$ ), comparando-os a outros algoritmos da literatura, a saber: NLMS (como referência geral de comparação), IPNLMS [20] e SPNLMS [45] (representando os algoritmos proporcionais),  $\ell_0$ -NLMS [31] (baseado na minimização de uma função custo com penalização de norma) e APWL1 [30] (representando os algoritmo baseado nas projeções em esferas de normas).

**Exemplo 4.6.** Nesse exemplo, os algoritmos  $\ell_0$ NC (V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC), bem como os algoritmos NLMS, IPNLMS, SPNLMS,  $\ell_0$ -NLMS e APWL1, são avaliados para a identificação de uma planta esparsa realística  $\mathbf{h}_4$ , com grau de esparsidade  $\mathcal{S}(\mathbf{h}_4) = 0,9281$ , considerando sinais de entrada gaussianos branco e colorido (veja Seção 4.1). Os parâmetros de cada algoritmo são ajustados, tendo como referência o algoritmo V- $\ell_0$ NC com passos de adaptação unitários (veja Tabela 14, para sinal de entrada gaussiano branco, e Tabela 15, para sinal de entrada gaussiano colorido). Aqui, apenas o algoritmo VS- $\ell_0$ NC utiliza passos de adaptação distintos ( $\mu_1 \neq \mu_2 = \mu_3$ ), o que lhe confere melhor desempenho. Os resultados de simulação são mostrados nas Figuras 42 (para sinal de entrada gaussiano branco) e 43 (para sinal de

entrada gaussiano colorido). Em ambas as situações, todos os algoritmos testados obtiveram resultados melhores do que o algoritmo NLMS. De modo geral, o algoritmo SPNLMS é o que apresenta melhores resultados, em especial na condição de sinal de entrada gaussiano branco. Excepcionalmente, o algoritmo IPNLMS apresenta resultado melhor do que os demais na simulação utilizando a planta  $\mathbf{h}_4$ , com sinal de entrada gaussiano colorido [veja Figura 43(a)], desempenho que não se repete nas demais simulações. Os algoritmos propostos baseados na norma  $\ell_0$  (V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC) apresentam desempenhos muito bons em ambas as situações, demonstrando maior robustez quanto ao tipo de sinal de entrada utilizado do que os algoritmos IPNLMS e SPNLMS. O algoritmo VS- $\ell_0$ NC é o que exibe o melhor desempenho dentre os algoritmos baseados na norma  $\ell_0$ , tanto com sinal de entrada gaussiano branco quanto com sinal de entrada gaussiano colorido [veja Figuras 42(b) e 43(b)].

Tabela 14: Exemplo 4.6. Parâmetros de configuração dos algoritmos para a planta  $\mathbf{h}_4$  e sinal de entrada gaussiano branco

| Algoritmo       | $\mu/\mu_1$ | $\mu_2$ | $\mu_3$ | $\alpha$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ | $\kappa$  |
|-----------------|-------------|---------|---------|----------|---------|----------|--------|------------|-----------|
| NLMS            | 0,25        |         |         |          |         |          |        |            |           |
| IPNLMS          | 0,25        |         |         | 0        |         |          | 0,01   | 0,01       |           |
| SPNLMS          | 0,25        |         |         |          |         |          | 0,01   | 0,01       |           |
| $\ell_0$ -NLMS  | 0,3         | 0,3     |         |          | 5       |          |        |            | $10^{-7}$ |
| APWL1           | 0,75        |         |         |          |         | 200      |        |            |           |
| V- $\ell_0$ NC  | 1           | 1       | 1       |          | 2500    |          |        |            |           |
| VS- $\ell_0$ NC | 1           | 0,1     | 0,1     |          | 500     |          |        |            |           |
| Z- $\ell_0$ NC  | 1           | 1       | 1       |          | 2500    |          |        |            |           |

Tabela 15: Exemplo 4.6. Parâmetros de configuração dos algoritmos para a planta  $\mathbf{h}_4$  e sinal de entrada gaussiano colorido

| Algoritmo       | $\mu/\mu_1$ | $\mu_2$ | $\mu_3$ | $\alpha$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ | $\kappa$  |
|-----------------|-------------|---------|---------|----------|---------|----------|--------|------------|-----------|
| NLMS            | 0,4         |         |         |          |         |          |        |            |           |
| IPNLMS          | 0,4         |         |         | 0        |         |          | 0,01   | 0,01       |           |
| SPNLMS          | 0,09        |         |         |          |         |          | 0,01   | 0,01       |           |
| $\ell_0$ -NLMS  | 0,7         | 0,7     |         |          | 40      |          |        |            | $10^{-8}$ |
| APWL1           | 0,8         |         |         |          |         | 300      |        |            |           |
| V- $\ell_0$ NC  | 1           | 1       | 1       |          | 3000    |          |        |            |           |
| VS- $\ell_0$ NC | 1           | 0,1     | 0,1     |          | 1000    |          |        |            |           |
| Z- $\ell_0$ NC  | 1           | 1       | 1       |          | 3000    |          |        |            |           |

**Exemplo 4.7.** Os mesmos algoritmos considerados no exemplo anterior são aqui avaliados na identificação de sistema utilizando uma planta quase-espar-sa realística  $\mathbf{h}_5$ , com grau de esparsidade  $\mathcal{S}(\mathbf{h}_5) = 0,7253$ , com sinal de entrada gaussiano branco (veja Figura 44) e sinal de entrada gaussiano colorido (veja Figura 45). Novamente, os parâmetros de cada algoritmo são ajusta-dos tendo como referência o algoritmo V- $\ell_0$ NC com passos de adaptação unitários (veja Tabela 16, para sinal de entrada gaussiano branco, e Tabela 17, para sinal de entrada gaussiano colorido). Para esse tipo de planta, observa-se que os algoritmos SPNLMS e IPNLMS já não se destacam dos demais algo-ritmos, como no caso da planta  $\mathbf{h}_4$ . Aqui, o algoritmo SPNLMS ainda exibe resultado um pouco melhor do que os demais algoritmos no caso de sinal de entrada gaussiano branco [veja Figura 44(a)], mas já é superado pelo algo-ritmo V- $\ell_0$ NC no caso de sinal de entrada gaussiano colorido [veja Figura 45(a)]. Os algoritmos propostos baseados na norma  $\ell_0$  (V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC), por outro lado, mantêm um desempenho destacado dos demais, especialmente no caso que utilizam sinais de entrada gaussianos coloridos. Explica-se tal comportamento pelo fato de que o último termo da equação de atualização dos algoritmos propostos (3.66) tem uma forte atuação na con-vergência para zero dos coeficientes não significativos e dependerem muito pouco do sinal de entrada. Nas Figuras 44(b) e 45(b), observa-se que os re-sultados obtidos pelos algoritmos V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC são bastante próximos entre si. No entanto, verifica-se que o algoritmo VS- $\ell_0$ NC nova-mente é o que apresenta resultados melhores do que os dos demais algoritmos  $\ell_0$  propostos.

Tabela 16: Exemplo 4.7. Parâmetros de configuração dos algoritmos para a planta  $\mathbf{h}_5$  e sinal de entrada gaussiano branco

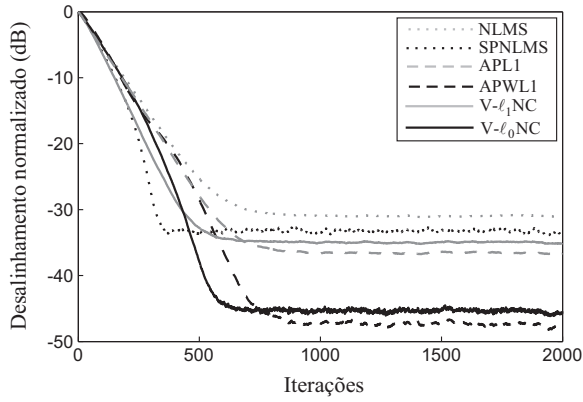
| Algoritmo       | $\mu/\mu_1$ | $\mu_2$ | $\mu_3$ | $\alpha$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ | $\kappa$  |
|-----------------|-------------|---------|---------|----------|---------|----------|--------|------------|-----------|
| NLMS            | 0,3         |         |         |          |         |          |        |            |           |
| IPNLMS          | 0,3         |         |         | 0        |         |          | 0,01   | 0,01       |           |
| SPNLMS          | 0,3         |         |         |          |         |          | 0,01   | 0,01       |           |
| $\ell_0$ -NLMS  | 0,55        | 0,55    |         |          | 100     |          |        |            | $10^{-8}$ |
| APWL1           | 0,8         |         |         |          |         | 256      |        |            |           |
| V- $\ell_0$ NC  | 1           | 1       | 1       |          | 2300    |          |        |            |           |
| VS- $\ell_0$ NC | 1           | 0,1     | 0,1     |          | 500     |          |        |            |           |
| Z- $\ell_0$ NC  | 1           | 1       | 1       |          | 2300    |          |        |            |           |

Tabela 17: Exemplo 4.7. Parâmetros de configuração dos algoritmos para a planta  $h_5$  e sinal de entrada gaussiano colorido

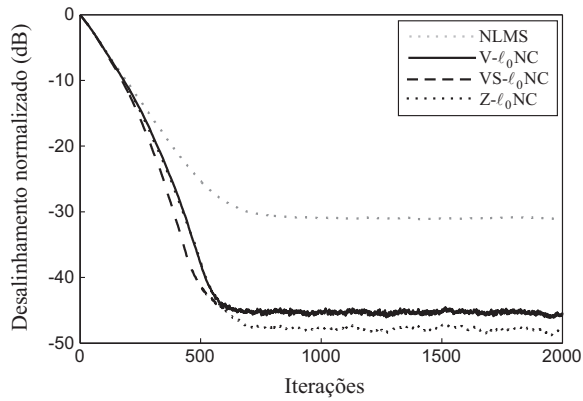
| Algoritmo       | $\mu/\mu_1$ | $\mu_2$ | $\mu_3$ | $\alpha$ | $\beta$ | $\delta$ | $\rho$ | $\delta_p$ | $\kappa$  |
|-----------------|-------------|---------|---------|----------|---------|----------|--------|------------|-----------|
| NLMS            | 0,45        |         |         |          |         |          |        |            |           |
| IPNLMS          | 0,5         |         |         | 0        |         |          | 0,01   | 0,01       |           |
| SPNLMS          | 0,2         |         |         |          |         |          | 0,01   | 0,01       |           |
| $\ell_0$ -NLMS  | 0,6         | 0,6     |         |          | 60      |          |        |            | $10^{-8}$ |
| APWL1           | 0,8         |         |         |          |         | 300      |        |            |           |
| V- $\ell_0$ NC  | 1           | 1       | 1       |          | 3000    |          |        |            |           |
| VS- $\ell_0$ NC | 1           | 0,1     | 0,1     |          | 1000    |          |        |            |           |
| Z- $\ell_0$ NC  | 1           | 1       | 1       |          | 4000    |          |        |            |           |

### 4.3 Conclusões

Este capítulo apresentou simulações realizadas com a finalidade de avaliar os algoritmos propostos, comparando-os com outros algoritmos orientados à identificação de sistemas esparsos. O desempenho e a robustez dos algoritmos foram avaliados considerando cenários diversos, obtendo-se resultados que mostraram a validade e a competitividade dos algoritmos propostos. Tais resultados são discutidos com mais detalhes no Capítulo 6. No próximo capítulo, modelos estocásticos dos algoritmos propostos são discutidos e avaliados.



(a)



(b)

Figura 40: Exemplo 4.5. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_1$ .

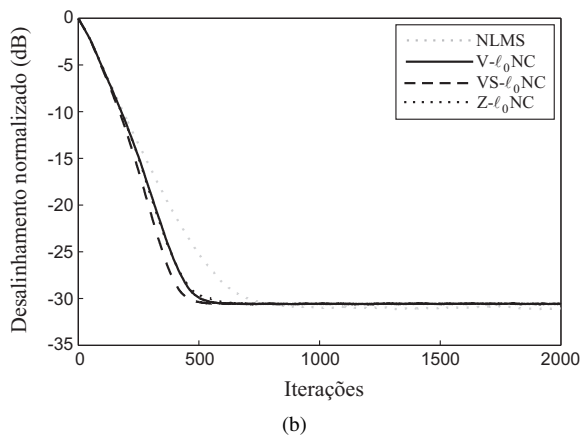
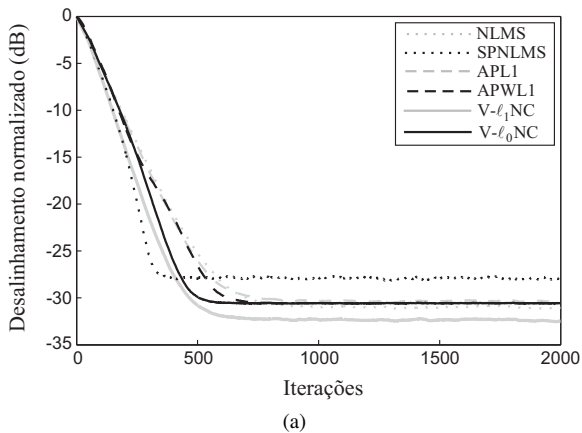
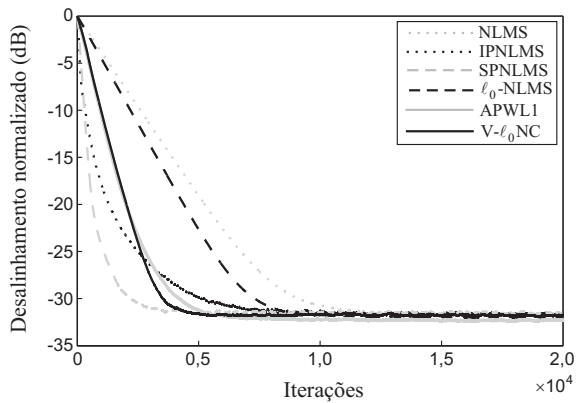
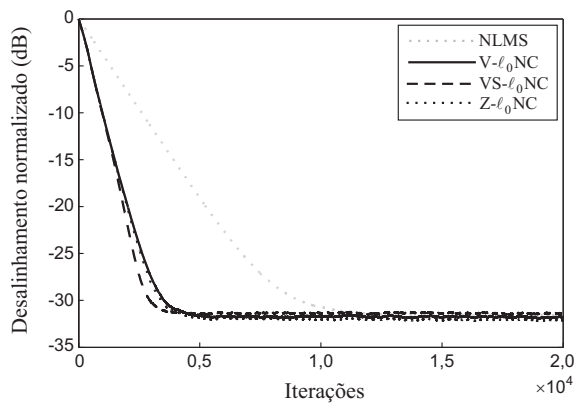


Figura 41: Exemplo 4.5. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_2$ .





(a)



(b)

Figura 42: Exemplo 4.6. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_4$  e sinal de entrada gaussiano branco.

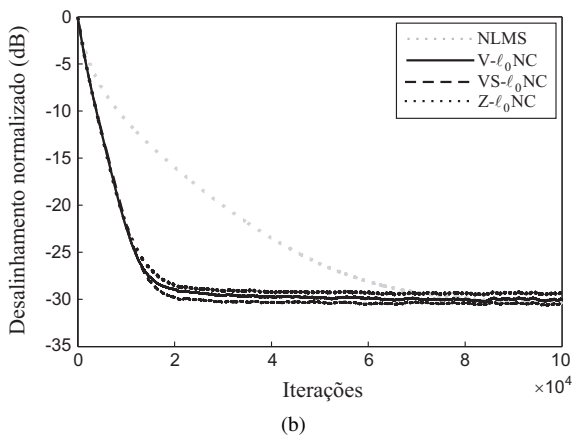
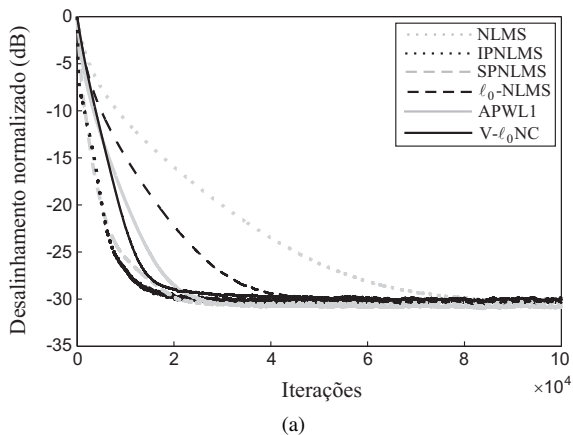
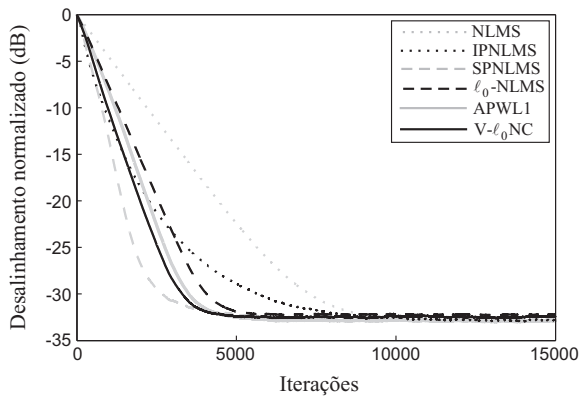
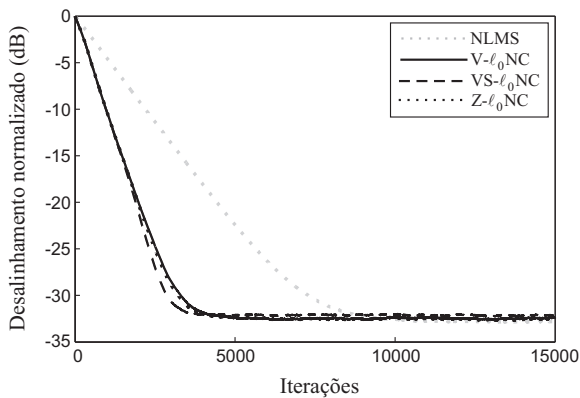


Figura 43: Exemplo 4.6. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_4$  e sinal de entrada gaussiano colorido.

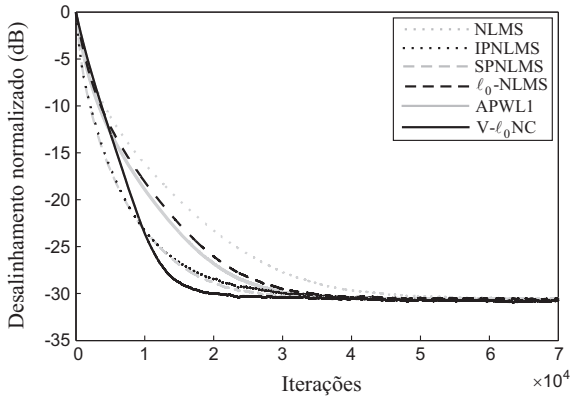


(a)

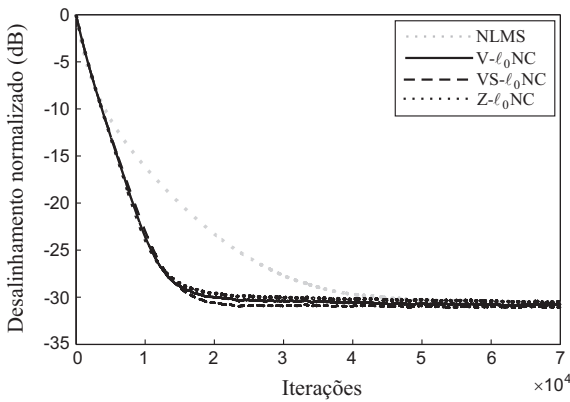


(b)

Figura 44: Exemplo 4.7. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_5$  e sinal de entrada gaussiano branco.



(a)



(b)

Figura 45: Exemplo 4.7. Curvas de desalinhamento normalizado, considerando a planta  $\mathbf{h}_5$  e sinal de entrada gaussiano colorido.

## Capítulo 5

# Modelagem Estocástica dos Algoritmos Propostos

Neste capítulo, são propostos modelos estocásticos para os algoritmos baseados em projeções com restrição de norma desenvolvidos no presente trabalho de pesquisa. Tais modelos visam prever o desempenho dos algoritmos frente as mais diversas condições de operação. A partir dos modelos obtidos, é possível determinar diretrizes para ajuste dos parâmetros dos algoritmos de forma analítica, reduzindo a necessidade da realização de extensivas simulações de Monte Carlo (MC) durante a etapa de projeto.

### 5.1 Modelagem estocástica do algoritmo NLMS

Os algoritmos com restrição de norma até então descritos guardam uma grande proximidade com o algoritmo NLMS, o qual também pode ser interpretado como um algoritmo baseado em restrições [5]. Especificamente no NLMS existe uma única restrição (erro *a posteriori* igual a zero), também presente nos algoritmos da família  $\ell_p$ NC. Dessa forma, é evidente que os modelos estocásticos dos algoritmos propostos possam ser desenvolvidos a partir dos resultados obtidos na modelagem do algoritmo NLMS [58]-[60]. Nesse contexto, a modelagem estocástica do algoritmo NLMS é brevemente revisitada a seguir, assumindo o caso particular de sinal de entrada gaussiano real e não-correlacionado, bem como um problema de identificação de sistemas (veja Figura 2 no Capítulo 1).

#### 5.1.1 Comportamento médio do vetor de coeficientes

O comportamento médio dos coeficientes do algoritmo NLMS pode ser determinado tomando o valor esperado de ambos os lados de (1.5), isto é

$$E[\mathbf{w}(n+1)] = E[\mathbf{w}(n)] + \mu E \left[ \frac{e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right] \quad (5.1)$$

com

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= [\mathbf{w}_0 - \mathbf{w}(n)]^T \mathbf{x}(n) + r(n) \end{aligned} \quad (5.2)$$

em que  $r(n)$  denota o ruído de medição com média zero e variância  $\sigma_r^2$ , independente de  $\mathbf{x}(n)$ .

Considerando a teoria da independência [4], isto é, assumindo passo de adaptação pequeno (adaptação lenta), pode-se estabelecer que o vetor de coeficientes  $\mathbf{w}(n)$  é estatisticamente independente do sinal de entrada  $\mathbf{x}(n)$ . Assim, a partir de (5.2) e considerando que  $E[r(n)] = 0$ , (5.1) pode ser reescrita como

$$E[\mathbf{w}(n+1)] \cong (\mathbf{I} - \mu \mathbf{R}_1) E[\mathbf{w}(n)] + \mu \mathbf{R}_1 \mathbf{w}_0 \quad (5.3)$$

onde  $\mathbf{I}$  denota a matriz identidade e

$$\mathbf{R}_1 = E \left[ \frac{\mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right]. \quad (5.4)$$

Aqui, a determinação da matriz de autocorrelação normalizada  $\mathbf{R}_1$  segue o desenvolvimento apresentado em [60]. Em tal trabalho, mostra-se que  $\mathbf{R}_1$  pode ser decomposta como

$$\mathbf{R}_1 = \mathbf{Q} \mathbf{H} \mathbf{Q}^T \quad (5.5)$$

onde  $\mathbf{Q}$  é a matriz de autovetores obtida através da autodecomposição de  $\mathbf{R} = E[\mathbf{x}(n) \mathbf{x}^T(n)]$  [4] e  $\mathbf{H}$  uma matriz diagonal de dimensão  $N \times N$  cuja determinação depende das características do sinal de entrada. Particularmente, para o caso de sinal de entrada real e não-correlacionado, tem-se que

$$\mathbf{H} = \frac{\mathbf{I}}{N} \quad (5.6)$$

e, portanto,

$$\mathbf{R}_1 = \frac{\mathbf{I}}{N}. \quad (5.7)$$

### 5.1.2 Curva de aprendizagem

O desempenho de algoritmos adaptativos é usualmente avaliado através da curva de aprendizagem, a qual define a evolução do algoritmo ao longo do

processo de adaptação para um dado critério de interesse, a saber [4]:

- o erro quadrático médio (*mean-square error - MSE*), definido como

$$J(n) = E[e^2(n)] \quad (5.8)$$

- e o desvio quadrático médio (*mean-square deviation - MSD*), definido como

$$D(n) = E[\|\mathbf{v}(n)\|_2^2] \quad (5.9)$$

onde

$$\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_o \quad (5.10)$$

representa o vetor de erro nos coeficientes. É comum ainda a utilização de uma versão normalizada do desvio quadrático médio (MSD normalizado), obtida como

$$D_N(n) = 10 \log_{10} \frac{D(n)}{\|\mathbf{w}_o\|_2^2}. \quad (5.11)$$

### 5.1.2.1 Curva de MSE

De (5.2) e (5.10), tem-se que o MSE pode ser determinado como

$$\begin{aligned} J(n) &= E[e^2(n)] \\ &= J_{\min} + J_{\text{ex}}(n) \end{aligned} \quad (5.12)$$

onde

$$\begin{aligned} J_{\min} &= E[r^2(n)] \\ &= \sigma_r^2 \end{aligned} \quad (5.13)$$

representa o MSE mínimo atingível em regime permanente, quando  $\mathbf{w}(n) = \mathbf{w}_o$ , e

$$J_{\text{ex}}(n) = E[\mathbf{v}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{v}(n)] \quad (5.14)$$

denota o MSE em excesso. Note que  $J_{\min}$  depende apenas do cenário de operação considerado, enquanto  $J_{\text{ex}}(n)$  depende do algoritmo responsável pela adaptação. A partir da definição da matriz de correlação do vetor de erro nos coeficientes

$$\mathbf{K}(n) = E[\mathbf{v}(n) \mathbf{v}^T(n)] \quad (5.15)$$

tem-se que (5.14) pode ser reescrita como

$$\begin{aligned} J_{\text{ex}}(n) &= \text{tr}[\mathbf{R}\mathbf{K}(n)] \\ &= \text{tr}[\mathbf{\Lambda}\mathbf{Q}^T\mathbf{K}(n)\mathbf{Q}] \end{aligned} \quad (5.16)$$

com  $\mathbf{\Lambda}$  representando a matriz de autovalores de  $\mathbf{R}$ . Finalmente, definindo

$$\begin{aligned} \mathbf{K}'(n) &= \text{E}[\mathbf{v}'(n)\mathbf{v}'^T(n)] \\ &= [\mathbf{Q}^T\mathbf{K}(n)\mathbf{Q}] \end{aligned} \quad (5.17)$$

sendo  $\mathbf{v}'(n) = \mathbf{Q}^T\mathbf{v}(n)$ , o vetor de erro nos coeficientes nas coordenadas definidas pelos autovetores de  $\mathbf{R}$ , (5.16) reduz-se a

$$\begin{aligned} J_{\text{ex}}(n) &= \text{tr}[\mathbf{\Lambda}\mathbf{K}'(n)] \\ &= \boldsymbol{\lambda}^T\mathbf{k}'(n) \\ &= \sum_{k=0}^{N-1} \lambda_k \text{E}[v_k'^2(n)] \end{aligned} \quad (5.18)$$

onde  $\boldsymbol{\lambda}$  denota um vetor contendo os autovalores de  $\mathbf{R}$  e  $\mathbf{k}'(n)$ , um vetor com os elementos da diagonal de  $\mathbf{K}'(n)$ . Então, para o caso particular de sinal de entrada não-correlacionado ( $\lambda_k = \sigma_x^2 \forall k$ ), (5.18) pode ser simplificada para

$$J_{\text{ex}}(n) = \sigma_x^2 \sum_{k=0}^{N-1} \text{E}[v_k'^2(n)]. \quad (5.19)$$

### 5.1.2.2 Curva de MSD

Por sua vez, é possível mostrar que o MSD pode ser expresso como

$$\begin{aligned} D(n) &= \text{tr}[\mathbf{K}(n)] \\ &= \text{tr}[\mathbf{K}'(n)] \\ &= \mathbf{1}^T\mathbf{k}'(n) \\ &= \sum_{k=0}^{N-1} \text{E}[v_k'^2(n)] \end{aligned} \quad (5.20)$$



onde  $\mathbf{1}$  é um vetor de dimensão  $N \times 1$  com todos os elementos iguais a 1. Assim, a partir de (5.18) e (5.20), é possível inferir que

$$\frac{J_{\text{ex}}(n)}{\lambda_{\min}} \geq D(n) \geq \frac{J_{\text{ex}}(n)}{\lambda_{\max}} \quad \forall n \quad (5.21)$$

sendo que, para o caso de sinal de entrada não-correlacionado, considerando agora (5.19), a relação entre o MSE em excesso e o MSD é dada por

$$J_{\text{ex}}(n) = \sigma_x^2 D(n). \quad (5.22)$$

Note que, subtraindo  $\mathbf{w}_o$  de ambos os lados de (1.5) e determinando o valor esperado de  $\mathbf{v}^T(n+1)\mathbf{v}(n+1)$ , obtém-se a seguinte expressão recursiva descrevendo o MSD para o algoritmo NLMS:

$$D(n+1) \cong \left[ 1 - \frac{\mu(2-\mu)}{N} \right] D(n) + \frac{\mu^2}{(N-2)} \frac{\sigma_r^2}{\sigma_x^2}. \quad (5.23)$$

### 5.1.3 Matriz de correlação do vetor de erro nos coeficientes

Uma expressão recursiva descrevendo a evolução da matriz de correlação do vetor erro nos coeficientes pode ser obtida subtraindo  $\mathbf{w}_o$  de ambos os lados de (1.5) e determinando o valor esperado de  $\mathbf{v}(n+1)\mathbf{v}^T(n+1)$ , chegando-se então a seguinte expressão:

$$\mathbf{K}(n+1) = \mathbf{K}(n) - \mu \mathbf{K}(n) \mathbf{R}_1 - \mu \mathbf{R}_1 \mathbf{K}(n) + \mu^2 \mathbf{R}_2(n) + \mu^2 \mathbf{R}_3 \sigma_r^2 \quad (5.24)$$

onde

$$\mathbf{R}_2(n) = \text{E} \left\{ \frac{\mathbf{x}(n) \mathbf{x}^T(n) \mathbf{v}(n) \mathbf{v}^T(n) \mathbf{x}(n) \mathbf{x}^T(n)}{[\mathbf{x}^T(n) \mathbf{x}(n)]^2} \right\} \quad (5.25)$$

e

$$\mathbf{R}_3 = \text{E} \left\{ \frac{\mathbf{x}(n) \mathbf{x}^T(n)}{[\mathbf{x}^T(n) \mathbf{x}(n)]^2} \right\}. \quad (5.26)$$

Então, pré- e pós-multiplicando (5.24) por  $\mathbf{Q}^T$  e  $\mathbf{Q}$ , respectivamente, e tomando os elementos da diagonal principal da expressão resultante, pode-se definir uma equação recursiva para  $\mathbf{k}'(n)$  como [60]

$$\mathbf{k}'(n+1) = \mathbf{B} \mathbf{k}'(n) + \mathbf{g} \mathbf{m} \quad (5.27)$$

com

$$\mathbf{B} = \mathbf{I} - 2\mu \mathbf{H} + \mu^2 (2\mathbf{T} + \mathbf{P}) \quad (5.28)$$

e

$$g = \mu^2 J_{min} \quad (5.29)$$

sendo  $\mathbf{T}$  e  $\mathbf{P}$ , matrizes de dimensão  $N \times N$ , e  $\mathbf{m}$ , um vetor de dimensão  $N \times 1$ , determinados em função das características do sinal de entrada. Particularmente para o caso de sinal de entrada não-correlacionado,  $\mathbf{T}$ ,  $\mathbf{P}$  e  $\mathbf{m}$  podem ser expressos como [60]

$$\mathbf{T} = \frac{\mathbf{I}}{N(N+2)} \quad (5.30)$$

$$\mathbf{P} = \frac{\mathbf{1}\mathbf{1}^T}{N(N+2)} \quad (5.31)$$

e

$$\mathbf{m} = \frac{\mathbf{1}}{N(N-2)\sigma_x^2}. \quad (5.32)$$

#### 5.1.4 Análise em regime permanente

Para estudar o desempenho do algoritmo em regime permanente, faz-se necessária a determinação de  $\mathbf{k}'(n)$  para  $n \rightarrow \infty$ . Então, assumindo que o algoritmo converge, isto é,

$$\lim_{n \rightarrow \infty} \mathbf{k}'(n+1) = \lim_{n \rightarrow \infty} \mathbf{k}'(n) = \mathbf{k}'(\infty) \quad (5.33)$$

a partir de (5.27), obtém-se

$$\mathbf{k}'(\infty) = g(\mathbf{I} - \mathbf{B})^{-1} \mathbf{m}. \quad (5.34)$$

Ainda, considerando (5.28), (5.29), (5.30), (5.31) e (5.32), tem-se que

$$\mathbf{B} = \left[ \mathbf{I} - \frac{2\mu}{N} \mathbf{I} + \frac{\mu^2}{N(N+2)} (2\mathbf{I} + \mathbf{1}\mathbf{1}^T) \right] \quad (5.35)$$

e, conseqüentemente,

$$\mathbf{k}'(\infty) = \left[ \frac{\mu}{(2-\mu)} \frac{1}{(N-2)} \frac{J_{min}}{\sigma_x^2} \right] \mathbf{1}. \quad (5.36)$$

Finalmente, substituindo (5.36) em (5.18) e escrevendo  $\boldsymbol{\lambda} = \sigma_x^2 \mathbf{1}$ , o MSE em excesso em regime permanente pode ser expresso por [60]

$$J_{\text{ex}}(\infty) = \frac{\mu}{(2 - \mu)} \frac{N}{(N - 2)} J_{\text{min}}. \quad (5.37)$$

Analogamente, substituindo (5.36) em (5.20), o MSD em regime permanente é obtido como [60]

$$D(\infty) = \frac{\mu}{(2 - \mu)} \frac{N}{(N - 2)} \frac{J_{\text{min}}}{\sigma_x^2}. \quad (5.38)$$

Note que (5.38) pode ser determinado diretamente de (5.23) fazendo  $\lim_{n \rightarrow \infty} D(n + 1) = \lim_{n \rightarrow \infty} D(n) = D(\infty)$ .

## 5.2 Modelagem estocástica do algoritmo $\ell_p$ NC

Nesta seção, é proposto um modelo estocástico geral para os algoritmos da família  $\ell_p$ NC, levando em consideração a modelagem do algoritmo NLMS revisitada na seção anterior. Modelos específicos dos algoritmos  $\ell_1$ NC e  $\ell_0$ NC são derivados, a partir de tal formulação geral, no decorrer das próximas seções.

### 5.2.1 Comportamento médio do vetor de coeficientes

Acrescentando-se um passo de adaptação à (3.15), obtemos a equação de adaptação relaxada do algoritmo  $\ell_p$ NC, dada por

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu[\alpha_1(n) \mathbf{x}(n) + \alpha_{\ell_p}(n) \lambda_{\ell_p}(n) \mathbf{x}(n) - \lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n + 1)]. \quad (5.39)$$

Tem-se, então, que o comportamento médio do vetor de coeficientes para essa família de algoritmos pode ser expressa como

$$\begin{aligned} \mathbb{E}[\mathbf{w}(n + 1)] &= (\mathbf{I} - \mu \mathbf{R}_1) \mathbb{E}[\mathbf{w}(n)] + \mu \mathbf{R}_1 \mathbf{w}_o + \\ &\quad \mu \mathbb{E}[\alpha_{\ell_p}(n) \lambda_{\ell_p}(n) \mathbf{x}(n)] - \mu \mathbb{E}[\lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n + 1)]. \end{aligned} \quad (5.40)$$

Substituindo a expressão de  $\alpha_{\ell_p}(n)$  [dada por (3.14)] em (5.40) e aproximando

$$\mathbb{E}[\lambda_{\ell_p}(n) \mathbf{g}_{\ell_p}(n + 1)] \cong \mathbb{E}[\lambda_{\ell_p}(n)] \mathbb{E}[\mathbf{g}_{\ell_p}(n + 1)] \quad (5.41)$$

e

$$E \left[ \frac{\mathbf{x}(n) \mathbf{x}^T(n) \mathbf{g}_{\ell_p}(n+1)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right] \cong \mathbf{R}_1 E[\mathbf{g}_{\ell_p}(n+1)] \quad (5.42)$$

então, (5.40) pode ser reescrita como

$$E[\mathbf{w}(n+1)] \cong (\mathbf{I} - \mu \mathbf{R}_1) E[\mathbf{w}(n)] + \mu \mathbf{R}_1 \mathbf{w}_o - \mu (\mathbf{I} - \mathbf{R}_1) E[\lambda_{\ell_p}(n)] E[\mathbf{g}_{\ell_p}(n+1)]. \quad (5.43)$$

Note, a partir de (5.3), que os dois primeiros termos a direita em (5.43) representam o comportamento médio do vetor de coeficientes referente ao algoritmo NLMS. Assim, substituindo esses termos por  $E[\mathbf{w}_{\text{NLMS}}(n+1)]$ , pode-se reescrever (5.43) como

$$E[\mathbf{w}(n+1)] \cong E[\mathbf{w}_{\text{NLMS}}(n+1)] - \mu (\mathbf{I} - \mathbf{R}_1) E[\lambda_{\ell_p}(n)] E[\mathbf{g}_{\ell_p}(n+1)]. \quad (5.44)$$

Logo, para  $\mathbf{R}_1 = \mathbf{I}/N$ , (5.44) é reescrito como

$$E[\mathbf{w}(n+1)] \cong E[\mathbf{w}_{\text{NLMS}}(n+1)] - \frac{\mu(N-1)}{N} E[\lambda_{\ell_p}(n)] E[\mathbf{g}_{\ell_p}(n+1)] \quad (5.45)$$

que, para  $N$  grande, pode ser simplificada para

$$E[\mathbf{w}(n+1)] \cong E[\mathbf{w}_{\text{NLMS}}(n+1)] - \mu E[\lambda_{\ell_p}(n)] E[\mathbf{g}_{\ell_p}(n+1)]. \quad (5.46)$$

### 5.2.2 Matriz de correlação do vetor de erro nos coeficientes

A partir de (5.39) e considerando (3.13) e (3.14), uma expressão para obtenção do vetor de erro nos coeficientes para os algoritmos  $\ell_p$ NC pode ser expressa como

$$\mathbf{v}(n+1) = \mathbf{X}_1(n) \mathbf{v}(n) - \lambda_{\ell_p}(n) \mathbf{X}_2(n) \mathbf{g}_{\ell_p}(n+1) + \mu \frac{r(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \quad (5.47)$$

onde

$$\mathbf{X}_1(n) = \left[ \mathbf{I} - \mu \frac{\mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right] \quad (5.48)$$

e

$$\mathbf{X}_2(n) = \mu \left[ \mathbf{I} - \frac{\mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right]. \quad (5.49)$$

Então, determinando  $\mathbf{v}(n+1)\mathbf{v}^T(n+1)$ , tomando o valor esperado de ambos os lados e considerando a aproximação definida em (5.41), obtém-se

$$\begin{aligned} \mathbf{K}(n+1) &\cong \mathbb{E}[\mathbf{X}_1(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{X}_1^T(n)] - & (5.50) \\ &\quad \mathbb{E}[\lambda_{\ell_p}(n)]\mathbb{E}[\mathbf{X}_1(n)\mathbf{v}(n)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{X}_2^T(n)] - \\ &\quad \mathbb{E}[\lambda_{\ell_p}(n)]\mathbb{E}[\mathbf{X}_2(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{v}^T(n)\mathbf{X}_1^T(n)] + \\ &\quad \mathbb{E}[\lambda_{\ell_p}^2(n)]\mathbb{E}[\mathbf{X}_2(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{X}_2^T(n)] + \\ &\quad \mu^2\mathbf{R}_3\sigma_r^2 \end{aligned}$$

com

$$\mathbb{E}[\mathbf{X}_1(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{X}_1^T(n)] \cong \mathbf{K}(n) - \mu\mathbf{K}(n)\mathbf{R}_1 - \mu\mathbf{R}_1\mathbf{K}(n) + \mu^2\mathbf{R}_2(n) \quad (5.51)$$

$$\begin{aligned} \mathbb{E}[\mathbf{X}_1(n)\mathbf{v}(n)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{X}_2^T(n)] &\cong \mu\mathbf{K}_{vs}(n) - \mu\mathbf{K}_{vs}(n)\mathbf{R}_1 - \\ &\quad \mu^2\mathbf{R}_1\mathbf{K}_{vs}(n) + \mu^2\mathbf{R}_4(n) \end{aligned} \quad (5.52)$$

$$\begin{aligned} \mathbb{E}[\mathbf{X}_2(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{v}^T(n)\mathbf{X}_1^T(n)] &\cong \mu\mathbf{K}_{sv}(n) - \mu\mathbf{K}_{sv}(n)\mathbf{R}_1 - \\ &\quad \mu^2\mathbf{R}_1\mathbf{K}_{sv}(n) + \mu^2\mathbf{R}_5(n) \end{aligned} \quad (5.53)$$

e

$$\begin{aligned} \mathbb{E}[\mathbf{X}_2(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{X}_2^T(n)] &\cong \mu^2\mathbf{K}_{ss}(n) - \mu^2\mathbf{K}_{ss}(n)\mathbf{R}_1 - \\ &\quad \mu^2\mathbf{R}_1\mathbf{K}_{ss}(n) + \mu^2\mathbf{R}_6(n) \end{aligned} \quad (5.54)$$

onde

$$\mathbf{R}_4(n) = \mathbb{E} \left\{ \frac{\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{v}(n)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{x}(n)\mathbf{x}^T(n)}{[\mathbf{x}^T(n)\mathbf{x}(n)]^2} \right\} \quad (5.55)$$

$$\mathbf{R}_5(n) = \mathbb{E} \left\{ \frac{\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{v}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)}{[\mathbf{x}^T(n)\mathbf{x}(n)]^2} \right\} \quad (5.56)$$

$$\mathbf{R}_6(n) = \mathbb{E} \left\{ \frac{\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{g}_{\ell_p}(n+1)\mathbf{g}_{\ell_p}^T(n+1)\mathbf{x}(n)\mathbf{x}^T(n)}{[\mathbf{x}^T(n)\mathbf{x}(n)]^2} \right\} \quad (5.57)$$

$$\mathbf{K}_{\text{vs}}(n) = \text{E}[\mathbf{v}(n) \mathbf{g}_{\ell_p}^{\text{T}}(n+1)] \quad (5.58)$$

$$\mathbf{K}_{\text{sv}}(n) = \text{E}[\mathbf{g}_{\ell_p}(n+1) \mathbf{v}^{\text{T}}(n)] \quad (5.59)$$

e

$$\mathbf{K}_{\text{ss}}(n) = \text{E}[\mathbf{g}_{\ell_p}(n+1) \mathbf{g}_{\ell_p}^{\text{T}}(n+1)]. \quad (5.60)$$

Finalmente, uma expressão recursiva para  $\mathbf{k}'(n)$  é determinada pré- e pós-multiplicando (5.50) por  $\mathbf{Q}^{\text{T}}$  e  $\mathbf{Q}$ , respectivamente, e tomando os elementos da diagonal principal da expressão resultante, isto é,

$$\mathbf{k}'(n+1) = \mathbf{B} \mathbf{k}'(n) - 2\text{E}[\lambda_{\ell_p}(n)] \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \text{E}[\lambda_{\ell_p}^2(n)] \mathbf{B}_{\text{ss}} \mathbf{k}'_{\text{ss}}(n) + g \mathbf{m} \quad (5.61)$$

onde

$$\mathbf{B}_{\text{sv}} = \mu[\mathbf{I} - (1 + \mu) \mathbf{H} + \mu(2\mathbf{T} + \mathbf{P})] \quad (5.62)$$

$$\mathbf{B}_{\text{ss}} = \mu^2[\mathbf{I} - 2\mathbf{H} + (2\mathbf{T} + \mathbf{P})] \quad (5.63)$$

$$\begin{aligned} \mathbf{k}'_{\text{sv}}(n) &= \text{diag}[\mathbf{Q}^{\text{T}} \mathbf{K}_{\text{sv}}(n) \mathbf{Q}] \\ &= \text{diag}[\mathbf{Q}^{\text{T}} \mathbf{K}_{\text{vs}}(n) \mathbf{Q}] \\ &= \text{diag}\{\text{E}[\mathbf{g}'_{\ell_p}(n+1) \mathbf{v}'^{\text{T}}(n)]\} \end{aligned} \quad (5.64)$$

e

$$\begin{aligned} \mathbf{k}'_{\text{ss}}(n) &= \text{diag}[\mathbf{Q}^{\text{T}} \mathbf{K}_{\text{ss}}(n) \mathbf{Q}] \\ &= \text{diag}\{\text{E}[\mathbf{g}'_{\ell_p}(n+1) \mathbf{g}'_{\ell_p}^{\text{T}}(n+1)]\} \end{aligned} \quad (5.65)$$

sendo  $\mathbf{g}'_{\ell_p}(n+1) = \mathbf{Q}^{\text{T}} \mathbf{g}_{\ell_p}(n+1)$  o vetor gradiente nas coordenadas definidas pelos autovetores de  $\mathbf{R}$ . Note que, para o caso particular abordado (sinal de entrada não-correlacionado) tem-se que  $\mathbf{Q} = \mathbf{I}$ , podendo-se reescrever (5.64) e (5.65) como

$$\mathbf{k}'_{\text{sv}}(n) = \text{diag}\{\text{E}[\mathbf{g}_{\ell_p}(n+1) \mathbf{v}^{\text{T}}(n)]\} \quad (5.66)$$

e

$$\mathbf{k}'_{\text{ss}}(n) = \text{diag}\{\text{E}[\mathbf{g}_{\ell_p}(n+1) \mathbf{g}_{\ell_p}^{\text{T}}(n+1)]\}. \quad (5.67)$$

Ainda, comparando (5.27) com (5.61) tem-se

$$\mathbf{k}'(n+1) = \mathbf{k}'_{\text{NLMS}}(n+1) - 2\text{E}[\lambda_{\ell_p}(n)] \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \text{E}[\lambda_{\ell_p}^2(n)] \mathbf{B}_{\text{ss}} \mathbf{k}'_{\text{ss}}(n) \quad (5.68)$$

onde  $\mathbf{k}'_{\text{NLMS}}(n+1)$  representa a equação recursiva para  $\mathbf{k}'(n)$  do algoritmo NLMS.

### 5.2.3 Curva de aprendizagem

É possível determinar a curva de aprendizagem para os algoritmos  $\ell_p$ NC a partir da matriz de correlação do vetor de erro nos coeficientes. Para sinal de entrada não-correlacionado, tal relação é equivalente a utilizar uma expressão iterativa para o MSD (e consequentemente o MSE em excesso), considerando o valor esperado de  $\mathbf{v}^T(n+1)\mathbf{v}(n+1)$ . Dessa forma, a partir de (5.47) e levando em conta que  $E[r(n)] = 0$ , obtém-se

$$\begin{aligned} D(n+1) \cong & E[\mathbf{v}^T(n)\mathbf{v}(n)] - 2\mu E[\mathbf{v}^T(n)\mathbf{R}_1\mathbf{v}(n)] + \\ & \mu^2 E[\mathbf{v}^T(n)\mathbf{R}_1\mathbf{v}(n)] - 2\mu E[\lambda_{\ell_p}(n)] E[\mathbf{v}^T(n)\mathbf{g}_{\ell_p}(n+1)] + \\ 2\mu E[\lambda_{\ell_p}(n)] E[\mathbf{v}^T(n)\mathbf{R}_1\mathbf{g}_{\ell_p}(n)] & + \mu^2 E[\lambda_{\ell_p}^2(n)] E[\mathbf{g}_{\ell_p}^T(n+1)\mathbf{g}_{\ell_p}(n+1)] - \\ \mu^2 E[\lambda_{\ell_p}^2(n)] E[\mathbf{v}^T(n)\mathbf{R}_1\mathbf{g}_{\ell_p}(n)] & + \mu^2 E\left[\frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)}\right] E[r^2(n)]. \quad (5.69) \end{aligned}$$

Considerando (5.66), (5.67) e  $\mathbf{R}_1 = \mathbf{I}/N$ , (5.69) pode ser escrita como

$$\begin{aligned} D(n+1) \cong & \left[1 - \frac{\mu(2-\mu)}{N}\right] D(n) - 2\mu \frac{N-1}{N} E[\lambda_{\ell_p}(n)] \|\mathbf{k}'_{\text{sv}}(n)\|_1 + \\ & \mu^2 \frac{N-1}{N} E[\lambda_{\ell_p}^2(n)] \|\mathbf{k}'_{\text{ss}}(n)\|_1 + \frac{\mu^2}{(N-2)} \frac{\sigma_r^2}{\sigma_x^2}. \quad (5.70) \end{aligned}$$

Por fim, considerando o MSD para o algoritmo NLMS em (5.23) e assumindo  $N$  grande, (5.70) pode ser expressa por

$$D(n+1) \cong D_{\text{NLMS}}(n) - 2\mu E[\lambda_{\ell_p}(n)] \|\mathbf{k}'_{\text{sv}}(n)\|_1 + \mu^2 E[\lambda_{\ell_p}^2(n)] \|\mathbf{k}'_{\text{ss}}(n)\|_1. \quad (5.71)$$

### 5.2.4 Análise em regime permanente

O MSD dos algoritmos  $\ell_p$ NC em regime permanente pode ser obtido fazendo  $\lim_{n \rightarrow \infty} D(n+1) = \lim_{n \rightarrow \infty} D(n) = D(\infty)$  em (5.70). Assim,

$$D(\infty) \cong \frac{N}{\mu(2-\mu)} \left\{ -2\mu E[\lambda_{\ell_p}(\infty)] \| \mathbf{k}'_{sv}(\infty) \|_1 + \mu^2 E[\lambda_{\ell_p}^2(\infty)] \| \mathbf{k}'_{ss}(\infty) \|_1 + \frac{\mu^2}{(N-2)} \frac{\sigma_r^2}{\sigma_x^2} \right\}. \quad (5.72)$$

Ainda, considerando (5.38), (5.72) pode ser simplificada para

$$D(\infty) \cong D_{\text{NLMS}}(\infty) - \frac{N}{(2-\mu)} \left\{ 2E[\lambda_{\ell_p}(\infty)] \| \mathbf{k}'_{sv}(\infty) \|_1 - \mu E[\lambda_{\ell_p}^2(\infty)] \| \mathbf{k}'_{ss}(\infty) \|_1 \right\}. \quad (5.73)$$

Observa-se na modelagem geral apresentada para os algoritmos  $\ell_p$ NC que é possível determinar os modelos específicos para cada algoritmo de sua família obtendo-se as expressões correspondentes para  $E[\lambda_{\ell_p}(n)]$ ,  $E[\mathbf{g}_{\ell_p}(n)]$ ,  $\mathbf{k}'_{sv}(n)$  e  $\mathbf{k}'_{ss}(n)$ . Dessa forma, nas seções seguintes, os modelos para os algoritmos baseados na norma  $\ell_1$  e na norma  $\ell_0$  são determinados.

## 5.3 Modelagem estocástica dos algoritmos $\ell_1$ NC

A partir da modelagem geral definida na seção anterior, são discutidos aqui os aspectos específicos relacionados ao modelo do algoritmo  $\ell_1$ NC.

### 5.3.1 Considerações sobre $E[\mathbf{g}_{\ell_1}(n)]$ , $\mathbf{k}'_{sv}(n)$ e $\mathbf{k}'_{ss}(n)$ .

Lembrando que  $\mathbf{g}_{\ell_1}(n+1) = \mathbf{s}(n)$  [veja (3.23)], pode-se escrever  $E[\mathbf{g}_{\ell_1}(n)]$ ,  $\mathbf{k}'_{sv}(n)$  e  $\mathbf{k}'_{ss}(n)$  para o algoritmo  $\ell_1$ NC como

$$\begin{aligned} E[\mathbf{g}_{\ell_1}(n)] &= E[\mathbf{s}(n)] \\ &\cong \text{sgn}\{E[\mathbf{w}(n)]\} \end{aligned} \quad (5.74)$$

com



$$\begin{aligned} \text{sgn}\{E[\mathbf{w}(n)]\} &= (\text{sgn}\{E[w_0(n)]\}, \text{sgn}\{E[w_1(n)]\}, \dots, \text{sgn}\{E[w_{N-1}(n)]\})^T, \\ \mathbf{k}'_{\text{ss}}(n) &= \text{diag}\{E[\mathbf{s}(n) \mathbf{s}^T(n)]\} \\ &= \mathbf{1} \end{aligned} \quad (5.75)$$

e

$$\mathbf{k}'_{\text{sv}}(n) = \text{diag}\{E[\mathbf{s}(n) \mathbf{v}^T(n)]\}. \quad (5.76)$$

A partir de (5.75), pode-se ainda determinar que

$$\begin{aligned} \|\mathbf{k}'_{\text{ss}}(n)\|_1 &= E[\mathbf{s}^T(n) \mathbf{s}(n)] \\ &= N. \end{aligned} \quad (5.77)$$

Entretanto, a determinação dos elementos de  $\mathbf{k}'_{\text{sv}}(n)$  requer algumas simplificações a saber:

- Assume-se que o processo de adaptação é iniciado com  $\mathbf{w}(0) = \mathbf{0}$ . Logo, para os coeficientes significativos,  $E[|w_i(n)|] \leq |w_{o,i}|$  quando  $n > 0$ . Por outro lado, para os coeficientes não-significativos ou nulos,  $E[|w_i(n)|] \geq |w_{o,i}|$  quando  $n > 0$ .
- Considera-se ainda que, para  $N$  grande, os vetores  $\mathbf{w}(n)$  e  $\mathbf{w}_o$  mantêm-se no mesmo hiperquadrante durante o processo de adaptação.

Assim,  $s_i(n)w_i(n) = |w_i(n)|$  e  $s_i(n)w_{o,i}(n) = |w_{o,i}|$ , o que implica

$$\begin{aligned} \mathbf{k}'_{\text{sv}}(n) &= \begin{bmatrix} E[|w_0(n)|] - |w_{o,0}| \\ E[|w_1(n)|] - |w_{o,1}| \\ \vdots \\ E[|w_{(N-1)}(n)|] - |w_{o,(N-1)}| \end{bmatrix} \\ &= \begin{bmatrix} s_{v,0}(n) E[|v_0(n)|] \\ s_{v,1}(n) E[|v_1(n)|] \\ \vdots \\ s_{v,(N-1)}(n) E[|v_{(N-1)}(n)|] \end{bmatrix} \end{aligned} \quad (5.78)$$

com  $s_{v,i}(n) = -1$  para  $E[|w_i(n)|] < |w_{o,i}|$  (coeficientes significativos) e  $s_{v,i}(n) = 1$  para  $E[|w_i(n)|] > |w_{o,i}|$  (coeficientes não-significativos). Logo, utilizando a aproximação

$$\begin{aligned} E[|v_i(n)|] &\cong \sqrt{E[v_i^2(n)]} \\ &\cong \sqrt{k'_i(n)} \end{aligned} \quad (5.79)$$

em que  $k'_i(n)$  representa o  $i$ -ésimo elemento do vetor  $\mathbf{k}'(n)$ ,

$$\mathbf{k}'_{sv}(n) \cong \begin{bmatrix} s_{v,0}(n) \sqrt{k'_{0}(n)} \\ s_{v,1}(n) \sqrt{k'_{1}(n)} \\ \vdots \\ s_{v,(N-1)}(n) \sqrt{k'_{(N-1)}(n)} \end{bmatrix}. \quad (5.80)$$

Considerando os comportamentos distintos para os coeficientes significativos e não-significativos observados em (5.80), pode-se definir dois subvetores de  $\mathbf{k}'_{sv}(n)$  referentes a esses dois subconjuntos de coeficientes. Assim,

$$\sqrt{k'_i(n)} = \begin{cases} u_{s,k}(n), & s_{v,i}(n) = -1 \\ u_{z,j}(n), & s_{v,i}(n) = 1 \end{cases} \quad (5.81)$$

sendo  $u_{s,k}(n)$  o  $k$ -ésimo elemento de  $\mathbf{u}_s(n) = [u_{s,0}(n) \ u_{s,1}(n) \ \dots \ u_{s,(N-N_z-1)}(n)]^T$ , referente aos coeficientes significativos, e  $u_{z,j}(n)$  o  $j$ -ésimo elemento de  $\mathbf{u}_z(n) = [u_{z,0}(n) \ u_{z,1}(n) \ \dots \ u_{z,(N_z-1)}(n)]^T$ , referente aos coeficientes não-significativos. Assim, pode-se escrever

$$\begin{aligned} \|\mathbf{k}'_{sv}(n)\|_1 &= E[\mathbf{s}^T(n) \mathbf{v}(n)] \\ &\cong \sum_{k=0}^{N-1} u_{z,k}(n) - \sum_{k=0}^{N-1} u_{s,k}(n) \\ &\cong \|\mathbf{u}_z(n)\|_1 - \|\mathbf{u}_s(n)\|_1 \\ &\cong \Delta_{\mathbf{u}}(n). \end{aligned} \quad (5.82)$$

### 5.3.2 Valor esperado de $\lambda_{\ell_1}(n)$

A partir de (3.30), tomando o valor esperado de ambos os lados, a seguinte aproximação pode ser considerada:

$$E[\lambda_{\ell_1}(n)] \cong \frac{E[\mathbf{s}^T(n) \mathbf{w}(n)] + E[\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)] - \delta_{\ell_1}}{E[\mathbf{s}^T(n) \mathbf{s}(n)] - E[\alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)]}. \quad (5.83)$$

Logo, considerando (5.77), o denominador de (5.83) pode ser simplificado para

$$\begin{aligned} E[\mathbf{s}^T(n) \mathbf{s}(n)] - E[\alpha_{\ell_1}(n) \mathbf{s}^T(n) \mathbf{x}(n)] &\cong N - E\left[\frac{\mathbf{s}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{s}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}\right] \\ &\cong N - \frac{E[\mathbf{s}^T(n) \mathbf{s}(n)]}{N} \\ &\cong N - 1. \end{aligned} \quad (5.84)$$

Assim, considerando (3.26) e

$$E[\mathbf{s}^T(n) \mathbf{w}(n)] = E[\mathbf{s}^T(n) \mathbf{v}(n)] + \|\mathbf{w}_o\|_1 \quad (5.85)$$

então, (5.83) pode ser reescrita como

$$\begin{aligned} E[\lambda_{\ell_1}(n)] &\cong \frac{1}{N-1} \left\{ E[\mathbf{s}^T(n) \mathbf{v}(n)] - E\left[\frac{\mathbf{s}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{v}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}\right] + \|\mathbf{w}_o\|_1 - \delta_{\ell_1} \right\} \\ &\cong \frac{1}{N-1} \left\{ \frac{N-1}{N} E[\mathbf{s}^T(n) \mathbf{v}(n)] + \|\mathbf{w}_o\|_1 - \delta_{\ell_1} \right\} \\ &\cong \frac{1}{N} E[\mathbf{s}^T(n) \mathbf{v}(n)] + \frac{1}{N-1} (\|\mathbf{w}_o\|_1 - \delta_{\ell_1}). \end{aligned} \quad (5.86)$$

Considerando ainda (5.82) e  $N$  grande, então pode-se agora reescrever (5.86) como

$$E[\lambda_{\ell_1}(n)] \cong \frac{1}{N} [\Delta_{\mathbf{u}}(n) + \|\mathbf{w}_o\|_1 - \delta_{\ell_1}]. \quad (5.87)$$

Finalmente, fazendo

$$\Delta_{\ell_1}(n) = \Delta_{\mathbf{u}}(n) + \|\mathbf{w}_o\|_1 - \delta_{\ell_1} \quad (5.88)$$

tem-se que

$$E[\lambda_{\ell_1}(n)] \cong \frac{\Delta_{\ell_1}(n)}{N}. \quad (5.89)$$

### 5.3.3 Valor esperado de $\lambda_{v\ell_1}(n)$

Para o algoritmo  $V\text{-}\ell_1\text{NC}$ , o valor esperado de  $\lambda_{v\ell_1}(n)$  pode ser determinado a partir de (3.38), levando em conta o desenvolvimento do valor

esperado de  $\lambda_{\ell_1}(n)$  na seção anterior. Assim, tem-se que

$$\begin{aligned} E[\lambda_{\nu\ell_1}(n)] &\cong \frac{E[|\alpha_1(n) \mathbf{s}^T(n) \mathbf{x}(n)|]}{N-1} & (5.90) \\ &\cong \frac{1}{N-1} E \left[ \left| \frac{\mathbf{s}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{v}(n)}{\mathbf{x}^T(n) \mathbf{x}(n)} \right| \right] \\ &\cong \frac{E[|\mathbf{s}^T(n) \mathbf{v}(n)|]}{N(N-1)}. \end{aligned}$$

Agora, aproximando  $E[|\mathbf{s}^T(n) \mathbf{v}(n)|] \cong |E[\mathbf{s}^T(n) \mathbf{v}(n)]|$  e assumindo  $N$  grande, obtém-se

$$E[\lambda_{\nu\ell_1}(n)] \cong \frac{|\Delta_{\mathbf{u}}(n)|}{N^2}. \quad (5.91)$$

### 5.3.4 Sumário do modelo do algoritmo $\ell_1$ NC

A partir da modelagem geral desenvolvida para  $\ell_p$ NC (veja Seção 5.2), considerando (5.74), (5.76), (5.77), (5.80), (5.82) e (5.89), pode-se obter as expressões do modelo do algoritmo  $\ell_1$ NC, conforme sumário a seguir:

- Matriz de correlação do vetor de erros nos coeficientes

$$\mathbf{k}'(n+1) = \mathbf{k}'_{\text{NLMS}}(n+1) - \frac{2\Delta_{\ell_1}(n)}{N} \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \frac{\Delta_{\ell_1}^2(n)}{N^2} \mathbf{B}_{\text{ss}} \mathbf{1} \quad (5.92)$$

- Comportamento médio do vetor de coeficientes

$$E[\mathbf{w}(n+1)] \cong E[\mathbf{w}_{\text{NLMS}}(n+1)] - \mu \frac{\Delta_{\ell_1}(n)}{N} E[\mathbf{s}(n)] \quad (5.93)$$

- Curva de MSD

$$D(n+1) \cong D_{\text{NLMS}}(n+1) - \frac{\mu}{N} \left[ 2\Delta_{\ell_1}(n) \Delta_{\mathbf{u}}(n) - \mu \Delta_{\ell_1}^2(n) \right] \quad (5.94)$$

- MSD em regime permanente

$$D(\infty) \cong D_{\text{NLMS}}(\infty) - \frac{1}{(2-\mu)} \left[ 2\Delta_{\ell_1}(\infty) \Delta_{\mathbf{u}}(\infty) - \mu \Delta_{\ell_1}^2(\infty) \right] \quad (5.95)$$

### 5.3.5 Sumário do modelo do algoritmo V- $\ell_1$ NC

Da mesma forma, a partir da modelagem geral desenvolvida para  $\ell_p$ NC (veja Seção 5.2), considerando (5.74), (5.76), (5.75), (5.82), (5.77) e (5.91), pode-se obter as expressões do modelo do algoritmo V- $\ell_1$ NC, conforme sumário a seguir:

- Matriz de correlação do vetor de erros nos coeficientes

$$\mathbf{k}'(n+1) = \mathbf{k}'_{\text{NLMS}}(n) - \frac{2|\Delta_{\mathbf{u}}(n)|}{N^2} \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \frac{\Delta_{\mathbf{u}}^2(n)}{N^4} \mathbf{B}_{\text{ss}} \mathbf{1} \quad (5.96)$$

- Comportamento médio do vetor de coeficientes

$$\mathbb{E}[\mathbf{w}(n+1)] \cong \mathbb{E}[\mathbf{w}_{\text{NLMS}}(n)] - \mu \frac{|\Delta_{\mathbf{u}}(n)|}{N^2} \mathbb{E}[\mathbf{s}(n)] \quad (5.97)$$

- Curva de MSD

$$D(n+1) \cong D_{\text{NLMS}}(n) - \frac{\mu}{N^2} \left[ 2|\Delta_{\mathbf{u}}(n)|\Delta_{\mathbf{u}}(n) - \frac{\mu}{N-1} \Delta_{\mathbf{u}}^2(n) \right] \quad (5.98)$$

- MSD em regime permanente

$$D(\infty) \cong D_{\text{NLMS}}(\infty) - \frac{1}{N(2-\mu)} \left[ 2|\Delta_{\mathbf{u}}(\infty)|\Delta_{\mathbf{u}}(\infty) - \frac{\mu}{N(N-1)} \Delta_{\mathbf{u}}^2(\infty) \right] \quad (5.99)$$

### 5.3.6 Exemplos

Nesta seção, são apresentados exemplos visando avaliar os modelos propostos para os algoritmos baseados em norma  $\ell_1$ . Os exemplos aqui discutidos, da mesma forma que no capítulo anterior, consideram um problema de identificação de sistema com as características especificadas na Tabela 18. Para verificar a precisão dos modelos propostos, são realizadas comparações desses modelos com os resultados obtidos por simulações de Monte Carlo (100 realizações independentes).

Tabela 18: Sumário do problema de identificação de sistemas utilizado nos exemplos

---

Planta a ser identificada  $\mathbf{h}_1$

comprimentos do vetor: 100 coeficientes

coeficientes diferentes de zero:  $\{ 0,1 \quad 1,0 \quad -0,5 \quad 0,1 \}$

posições dos coeficientes diferentes de zero:  $\{ 1 \quad 30 \quad 35 \quad 38 \}$

grau de esparsidade:  $\mathcal{S}(\mathbf{h}_1) = 0,9435$

norma  $\ell_1$ :  $\|\mathbf{h}_1\|_1 = 1,7$

norma  $\ell_0$ :  $\|\mathbf{h}_1\|_0 = 4$

Sinal de entrada  $x(n)$ : gaussiano branco com variância  $\sigma_x^2 = 1$

Ruído de medição  $r(n)$ : gaussiano branco com variância  $\sigma_r^2 = 10^{-3}$

---

**Exemplo 5.1.** Nesse exemplo, o modelo do algoritmo  $\ell_1$ NC (com  $\mu = 1$ ) é avaliado para diferentes escolhas de restrição de norma  $\ell_1$ , considerando seu MSD em regime permanente para  $\delta_{\ell_1}$  variando entre 1 e 2,5. Verifica-se, através da comparação com simulações de Monte Carlo apresentada na Figura 46, uma excelente predição do modelo proposto para toda a faixa de valores de  $\delta_{\ell_1}$  considerada. Sobretudo, observa-se que a identificação do melhor valor de  $\delta_{\ell_1}$  está bastante próximo do valor obtido por simulações.

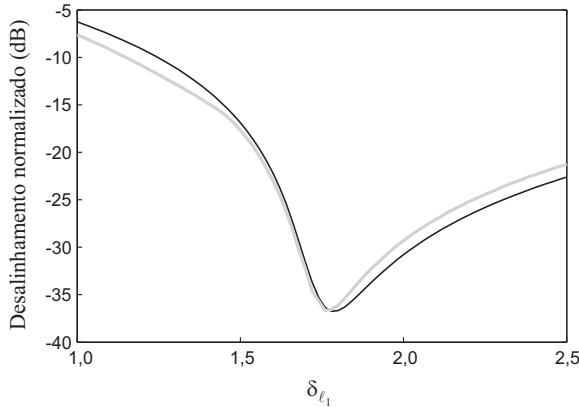


Figura 46: Exemplo 5.1. MSD em regime permanente do algoritmo  $\ell_1$ NC, comparando o modelo (linha escura) com simulações de Monte Carlo (linha clara).

**Exemplo 5.2.** Nesse exemplo, os modelos dos algoritmos NLMS [60],  $\ell_1$ NC e V- $\ell_1$ NC são avaliados conjuntamente, considerando  $\mu = 1$  para todos os algoritmos, sendo que o algoritmo  $\ell_1$ NC utiliza  $\delta_{l_1} = 1,78$  (melhor valor de  $\delta_{l_1}$  para esse cenário, conforme verificado no Capítulo 3). A Figura 47 ilustra o comportamento médio dos coeficientes para os algoritmos estudados, focando apenas o comportamento referente aos coeficientes significativos. A partir de tal figura, observa-se uma excelente precisão dos modelos propostos na predição do comportamento dos coeficientes no cenário considerado. O desempenho dos algoritmos é também verificado através das curva de MSD. Os resultados obtidos são ilustrados na Figura 48, da qual constata-se uma precisão muito boa na predição do desalinhamento em regime permanente de todos os algoritmos avaliados. Todavia, é identificado um descasamento na predição em regime transitório, que é comum na modelagem utilizando  $\mu = 1$ , considerando que os modelos são obtidos pressupondo passo de adaptação lento.

Uma avaliação dos valores esperados de  $\lambda_{\ell_1}(n)$  e  $\lambda_{v\ell_1}(n)$  é também realizada, como ilustrado na Figura 49. Observa-se nos resultados uma boa concordância entre as curvas, principalmente quando a adaptação atinge o regime permanente.

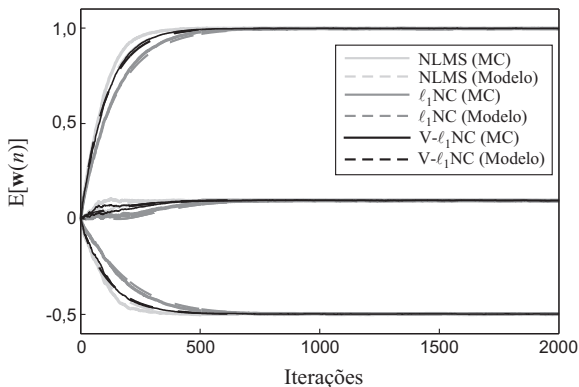


Figura 47: Exemplo 5.2. Comportamento médio dos coeficientes significativos para os algoritmos NLMS,  $\ell_1$ NC e V- $\ell_1$ NC.

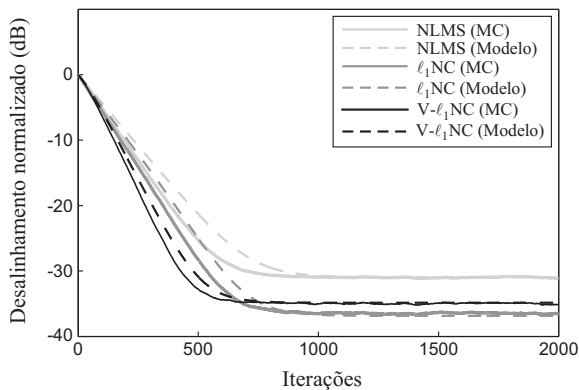


Figura 48: Exemplo 5.2. MSD dos algoritmos NLMS,  $\ell_1$ NC e V- $\ell_1$ NC.



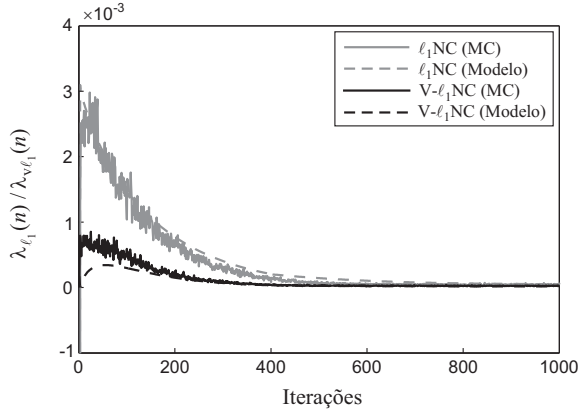


Figura 49: Exemplo 5.2. Comportamento médio dos parâmetros  $\lambda_{\ell_1}(n)$  e  $\lambda_{v\ell_1}(n)$ .

## 5.4 Modelagem estocástica do algoritmo $\ell_0$ NC

Nesta seção, seguindo as mesmas orientações utilizadas para a modelagem do algoritmo  $\ell_1$ NC, será discutido o comportamento estocástico do algoritmo  $\ell_0$ NC para entrada não-correlacionada.

### 5.4.1 Considerações sobre $E[\mathbf{g}_{\ell_0}(n+1)]$ , $\mathbf{k}'_{ss}(n)$ e $\mathbf{k}'_{sv}(n)$

Sendo  $\mathbf{g}_{\ell_0}(n+1) = \beta \mathbf{s}_z(n)$  [veja (3.50)], pode-se escrever seu valor esperado como

$$E[\mathbf{g}_{\ell_0}(n+1)] = \beta E[\mathbf{s}_z(n)] \quad (5.100)$$

$$= \beta \mathbf{s}_{z_0} \quad (5.101)$$

onde  $\mathbf{s}_{z_0} = E[\mathbf{s}_z(n)]$  é o vetor cujos elementos são dados por

$$s_{z_0,k} = E[s_{z,k}(n)] \cong \begin{cases} 1, & w_{o,k} = 0 \text{ (não-significativos)} \\ 0, & \text{demais (significativos)} \end{cases} \quad (5.102)$$

com  $w_{o,k}$  sendo o  $k$ -ésimo elemento da resposta ao impulso  $\mathbf{w}_o$  do sistema a ser identificada. Já os vetores  $\mathbf{k}'_{sv}(n)$  e  $\mathbf{k}'_{ss}(n)$  podem ser escritos como

$$\begin{aligned}\mathbf{k}'_{ss}(n) &= \text{diag}\{E[\mathbf{s}_z(n) \mathbf{s}_z^T(n)]\} \\ &= \mathbf{s}_{z0}\end{aligned}\quad (5.103)$$

e

$$\mathbf{k}'_{sv}(n) = \text{diag}\{E[\mathbf{s}_z(n) \mathbf{v}^T(n)]\}. \quad (5.104)$$

A partir de (5.102), observa-se que ambos os vetores,  $\mathbf{k}'_{sv}(n)$  e  $\mathbf{k}'_{ss}(n)$ , possuem elementos iguais a zero nas posições referentes aos coeficientes significativos. Assim, pode-se escrever

$$\begin{aligned}\|\mathbf{k}'_{ss}(n)\|_1 &= E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)] \\ &\cong N_z\end{aligned}\quad (5.105)$$

onde  $N_z$  é o número de coeficientes não-significativos de  $\mathbf{w}_o$ , bem como

$$\begin{aligned}\|\mathbf{k}'_{sv}(n)\|_1 &= E[\mathbf{s}_z^T(n) \mathbf{v}(n)] \\ &\cong \|\mathbf{u}_z(n)\|_1.\end{aligned}\quad (5.106)$$

#### 5.4.2 Valor esperado de $\lambda_{\ell_0}(n)$

A partir de (3.60), pode-se obter o valor esperado de  $\lambda_{\ell_0}(n)$ . Assim,

$$E[\lambda_{\ell_0}(n)] \cong \frac{E[\mathbf{s}_z^T(n) \mathbf{w}(n)] + E[\alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n)] + \{N - E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)] - \delta_{\ell_0}\} / \beta}{E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)] - E[\alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)]}. \quad (5.107)$$

Considerando (3.59) e (5.105), o denominador de (5.107) pode ser aproximado por

$$\begin{aligned}E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)] - E[\alpha_{\ell_0}(n) \mathbf{s}_z^T(n) \mathbf{x}(n)] &\cong N_z - E\left[\frac{\mathbf{s}_z^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{s}_z(n)}{\mathbf{x}^T(n) \mathbf{x}(n)}\right] \\ &\cong N_z - \frac{E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)]}{N} \\ &\cong \frac{N_z(N-1)}{N}.\end{aligned}\quad (5.108)$$

Assim, sabendo que  $E[\mathbf{s}_z^T(n) \mathbf{w}_o(n)] = 0$  e, portanto,  $E[\mathbf{s}_z^T(n) \mathbf{w}(n)] = E[\mathbf{s}_z^T(n) \mathbf{v}(n)]$ , então (5.107) pode ser reescrito como

$$\begin{aligned} E[\lambda_{\ell_0}(n)] &\cong \frac{N}{N_z(N-1)} \left\{ E[\mathbf{s}_z^T(n) \mathbf{v}(n)] + E[\alpha_1(n) \mathbf{s}_z^T(n) \mathbf{x}(n)] + \right. \\ &\quad \left. (N - N_z - \delta_{\ell_0}) / \beta \right\} \\ &\cong \frac{N}{N_z(N-1)} \left\{ \frac{N-1}{N} \|\mathbf{u}_z(n)\|_1 + (N - N_z - \delta_{\ell_0}) / \beta \right\} \\ &\cong \frac{\|\mathbf{u}_z(n)\|_1}{N_z} + \frac{N(N - N_z - \delta_{\ell_0})}{\beta N_z(N-1)} \end{aligned} \quad (5.109)$$

que, para  $N$  grande, pode ainda ser simplificada para

$$E[\lambda_{\ell_0}(n)] \cong \frac{1}{N_z} \left[ \|\mathbf{u}_z(n)\|_1 + \frac{N - N_z - \delta_{\ell_0}}{\beta} \right]. \quad (5.110)$$

Ainda, fazendo

$$\Delta_{\ell_0}(n) = \|\mathbf{u}_z(n)\|_1 + \frac{N - N_z - \delta_{\ell_0}}{\beta} \quad (5.111)$$

tem-se

$$E[\lambda_{\ell_0}(n)] \cong \frac{\Delta_{\ell_0}(n)}{N_z}. \quad (5.112)$$

### 5.4.3 Sumário do modelo do algoritmo $\ell_0$ NC

A partir da modelagem geral desenvolvida para  $\ell_p$ NC (veja Seção 5.2), considerando (5.100), (5.103), (5.104), (5.105), (5.106) e (5.112), pode-se obter as expressões do modelo do algoritmo  $\ell_0$ NC, conforme sumário a seguir:

- Matriz de correlação do vetor de erros nos coeficientes

$$\mathbf{k}'(n+1) = \mathbf{k}'_{\text{NLMS}}(n+1) - \frac{2\Delta_{\ell_0}(n)}{N_z} \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \frac{\Delta_{\ell_0}^2(n)}{N_z^2} \mathbf{B}_{\text{ss}} \mathbf{s}_z \quad (5.113)$$

- Comportamento médio do vetor de coeficientes

$$E[\mathbf{w}(n+1)] \cong E[\mathbf{w}_{\text{NLMS}}(n+1)] \quad (5.114)$$

- Curva de MSD

$$D(n+1) \cong D_{\text{NLMS}}(n+1) - \frac{\mu}{N_z} \left[ 2\Delta_{\ell_0}(n) \|\mathbf{u}_z(n)\|_1 - \mu\Delta_{\ell_0}^2(n) \right] \quad (5.115)$$

- MSD em regime permanente

$$D(\infty) \cong D_{\text{NLMS}}(\infty) - \frac{1}{2-\mu} \frac{N-1}{N_z} \left[ 2\Delta_{\ell_0}(\infty) \|\mathbf{u}_z(\infty)\|_1 - \mu\Delta_{\ell_0}^2(\infty) \right] \quad (5.116)$$

#### 5.4.4 Valor esperado de $\lambda_{v\ell_0}(n)$

Para o algoritmo V- $\ell_0$ NC, substituindo  $\delta_{\ell_0}$  por  $\delta_{v\ell_0}$  em (5.111) e fazendo  $\delta_{v\ell_0} = N - N_z$ , ontém-se

$$\Delta_{v\ell_0}(n) = \|\mathbf{u}_z(n)\|_1 \quad (5.117)$$

e

$$E[\lambda_{v\ell_0}(n)] \cong \frac{\|\mathbf{u}_z(n)\|_1}{N_z}. \quad (5.118)$$

#### 5.4.5 Análise em regime permanente para V- $\ell_0$ NC

Considerando o processo adaptativo inicializado com todos os coeficientes iguais a zero, isto é,  $[\mathbf{w}(0) = \mathbf{0}]$ , pode-se verificar facilmente que todos os elementos do vetor  $\mathbf{u}_z(n)$  são iguais ao longo da convergência. Assim,  $u_{zi}(n) = u_z(n) \forall i$ , o que permite definir o MSD referente aos coeficientes não-significativos como

$$\begin{aligned} D_z(n) &= \sum_{i=0}^{N_z-1} k'_{z,i}(n) \\ &= N_z k'_z(n) \\ &= N_z u_z^2(n). \end{aligned} \quad (5.119)$$

Sendo  $\|\mathbf{u}_z(n)\|_1^2 = [N_z u_z(n)]^2$ , então,

$$D_z(n) = \frac{\|\mathbf{u}_z(n)\|_1^2}{N_z}. \quad (5.120)$$

A partir de (5.116), (5.117) e (5.120), pode-se escrever que

$$\begin{aligned}
 D(\infty) &\cong D_{\text{NLMS}}(\infty) - \frac{1}{2-\mu} \frac{N-1}{N_z} \left[ 2\|\mathbf{u}_z(\infty)\|_1^2 - \mu\|\mathbf{u}_z(\infty)\|_1^2 \right] \\
 &\cong D_{\text{NLMS}}(\infty) - \frac{N-1}{N_z} \|\mathbf{u}_z(\infty)\|_1^2 \\
 &\cong D_{\text{NLMS}}(\infty) - (N-1)D_z(\infty).
 \end{aligned} \tag{5.121}$$

Agora, sabendo que o comportamento médio dos coeficientes significativos para os algoritmos  $\ell_0\text{NC}$  é semelhante ao comportamento médio dos mesmos coeficientes para o algoritmo NLMS [veja (5.114)], então, pode-se escrever

$$D_s(\infty) = \frac{N-N_z}{N} D_{\text{NLMS}}(\infty) \tag{5.122}$$

onde  $D_s(\infty)$  corresponde ao MSD em regime permanente do algoritmo  $V\text{-}\ell_0\text{NC}$  considerando apenas os coeficientes significativos. Assim, levando em conta (5.121) e (5.122), e sabendo que  $D(\infty) = D_s(\infty) + D_z(\infty)$ , tem-se

$$D_{\text{NLMS}}(\infty) - (N-1)D_z(\infty) = \frac{N-N_z}{N} D_{\text{NLMS}}(\infty) + D_z(\infty). \tag{5.123}$$

Dessa forma,

$$D_z(\infty) = \frac{N_z}{N^2} D_{\text{NLMS}}(\infty) \tag{5.124}$$

e, portanto,

$$\begin{aligned}
 D(\infty) &\cong \frac{N-N_z}{N} D_{\text{NLMS}}(\infty) + \frac{N_z}{N^2} D_{\text{NLMS}}(\infty) \\
 &\cong \left[ 1 - \frac{N_z}{N} + \frac{N_z}{N^2} \right] D_{\text{NLMS}}(\infty).
 \end{aligned} \tag{5.125}$$

Assim, conhecendo o número aproximado de coeficientes da planta que possuem valor igual ou muito próximo a zero, é possível determinar analiticamente a diferença entre os desempenhos dos algoritmos  $V\text{-}\ell_0\text{NC}$  e NLMS em regime permanente. As curvas de aprendizagem (MSD) dos algoritmos  $\ell_0\text{NC}$  e  $V\text{-}\ell_0\text{NC}$  são comparadas às simulações de Monte Carlo no Exemplo 5.4 (Seção 5.4.8). Já o desempenho do algoritmo  $\ell_0\text{NC}$  em regime permanente, para vários valores de  $\delta_{\ell_0}$ , é avaliado no Exemplo 5.3 (Seção 5.4.8).

### 5.4.6 Sumário do modelo do algoritmo V- $\ell_0$ NC

Agora, a partir da modelagem obtida para  $\ell_0$ NC e considerando (5.118) e (5.125), pode-se obter as expressões de modelo do algoritmo V- $\ell_0$ NC, conforme sumário a seguir:

- Matriz de correlação do vetor de erros nos coeficientes

$$\mathbf{k}'(n+1) = \mathbf{k}'_{\text{NLMS}}(n+1) - \frac{2\|\mathbf{u}_z(n)\|_1}{N_z} \mathbf{B}_{\text{sv}} \mathbf{k}'_{\text{sv}}(n) + \frac{\|\mathbf{u}_z(n)\|_1^2}{N_z^2} \mathbf{B}_{\text{ss}} \mathbf{s}_{z_0} \quad (5.126)$$

- Comportamento médio do vetor de coeficientes

$$\mathbb{E}[\mathbf{w}(n+1)] \cong \mathbb{E}[\mathbf{w}_{\text{NLMS}}(n+1)] \quad (5.127)$$

- Curva de MSD

$$D(n+1) \cong D_{\text{NLMS}}(n+1) - \frac{\mu(2-\mu)}{N_z} \|\mathbf{u}_z(n)\|_1^2 \quad (5.128)$$

- MSD em regime permanente

$$D(\infty) \cong \left[ 1 - \frac{N_z}{N} + \frac{N_z}{N^2} \right] D_{\text{NLMS}}(\infty) \quad (5.129)$$

### 5.4.7 Especificidades do modelo para o algoritmo VS- $\ell_0$ NC

Comparando-se os algoritmos V- $\ell_0$ NC e VS- $\ell_0$ NC (veja Tabelas 5 e 6, Capítulo 3), observa-se que a diferença básica entre seus equacionamentos está na substituição do termo  $\mathbf{s}_z(n)$  (algoritmo V- $\ell_0$ NC) por  $\mathbf{z}_s(n)$  (algoritmo VS- $\ell_0$ NC) nas equações de obtenção de parâmetros e de adaptação do vetor de coeficientes. Sendo  $z_k(n) = \max[0, (1 - \beta|w_k(n)|)]$  e  $z_{sk}(n) = s_k(n)z_k(n)$ , então

$$\begin{aligned} \mathbf{z}_s(n) &= \mathbf{s}_z(n) - \beta \mathbf{\Lambda}_{\text{vs}\ell_0}(n) \mathbf{s}_z(n) \\ &= \mathbf{s}_z(n) - \beta \mathbf{w}_z(n) \end{aligned} \quad (5.130)$$

onde

$$\mathbf{\Lambda}_{\text{vs}\ell_0}(n) = \begin{bmatrix} |w_0(n)| & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & |w_{(N-1)}(n)| \end{bmatrix}. \quad (5.131)$$

O vetor  $\mathbf{w}_z(n) = \mathbf{\Lambda}_{\text{vs}\ell_0}(n) \mathbf{s}_z(n)$  em (5.130) é definido aqui como um vetor equivalente a  $\mathbf{w}(n)$ , porém, com todos os elementos que representam os coeficientes significativos iguais a zero.

Considerando que  $E[\mathbf{s}_z^T(n) \mathbf{w}_0] = 0$ , então  $\mathbf{z}_s^T(n) \mathbf{w}(n) = \mathbf{z}_s^T(n) \mathbf{v}(n)$ , assim, a partir de (5.130), pode-se obter os seguintes valores esperados a serem utilizados na modelagem do algoritmo VS- $\ell_0$ NC:

$$\begin{aligned} E[\mathbf{z}_s(n)] &= E[\mathbf{s}_z(n)] - \beta E[\mathbf{w}_z(n)] \\ &= \mathbf{0} \end{aligned} \quad (5.132)$$

$$\begin{aligned} E[\mathbf{z}_s^T(n) \mathbf{v}(n)] &= E[\mathbf{s}_z^T(n) \mathbf{v}(n)] - \beta E[\mathbf{u}_s^T(n) \mathbf{u}_z(n)] \\ &= \|\mathbf{u}_z(n)\| - \beta \mathbf{k}'_z(n) \end{aligned} \quad (5.133)$$

e

$$\begin{aligned} E[\mathbf{z}_s^T(n) \mathbf{z}_s(n)] &= E[\mathbf{s}_z^T(n) \mathbf{s}_z(n)] - 2\beta E[\mathbf{s}_z^T(n) \mathbf{v}(n)] + \beta^2 E[\mathbf{u}_s^T(n) \mathbf{u}_z(n)] \\ &= N_z - 2\beta \|\mathbf{u}_z(n)\| + \beta^2 \mathbf{k}'_z(n). \end{aligned} \quad (5.134)$$

### 5.4.8 Exemplos

Nesta seção, são apresentados exemplos que mostram a validade dos modelos propostos para os algoritmos baseados em norma  $\ell_0$ . Como no caso dos algoritmos  $\ell_1$ NC, os exemplos aqui discutidos consideram um problema de identificação de sistema cujas características são especificadas na Tabela 18 (veja Seção 5.3.6). Os modelos são comparados com resultados obtidos por simulações de Monte Carlo (para 100 realizações independentes).

**Exemplo 5.3.** Nesse exemplo, o modelo do algoritmo  $\ell_0$ NC (com  $\mu = 1$ ) é avaliado para diferentes escolhas de restrição de norma  $\ell_0$ , considerando seu MSD em regime permanente para  $\delta_{\ell_0}$  variando entre 0 e 10. Verifica-se, através da comparação com simulações de Monte Carlo apresentada na Figura 50, uma predição bastante consistente do modelo a partir do valor de  $\delta_{\ell_0} \cong 4,3$ . Por outro lado, para valores de  $\delta_{\ell_0}$  menores do que esse limite, o desempenho do algoritmo é significativamente prejudicado em função da incapacidade de distinguir coeficientes significativos de menor valor. Como o modelo não leva em conta o processo de detecção de coeficientes significativos ou não significativos, sua curva resultante não acompanha o resultado das simulações. A partir do modelo, observa-se ainda que o valor ótimo de  $\delta_{\ell_0}$  (nesse caso igual a 4) corresponde à norma  $\ell_0$  da planta utilizada.

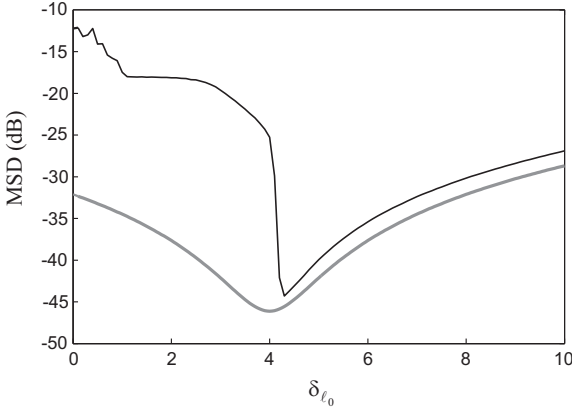


Figura 50: Exemplo 5.3. MSD em regime permanente do algoritmo  $\ell_0$ NC, comparando o modelo (linha escura) com simulações de Monte Carlo (linha clara).

**Exemplo 5.4.** Aqui, os modelos dos algoritmos NLMS [60],  $\ell_0$ NC e V- $\ell_0$ NC são mostrados conjuntamente, considerando  $\mu = 1$  para todos os algoritmos, com  $\delta_{\ell_0} = 4, 3$  e  $\beta = 15$  para  $\ell_0$ NC e  $\beta = 120$  para V- $\ell_0$ NC. A Figura 51 ilustra os resultados referentes ao comportamento médio dos coeficientes (considerando apenas os coeficientes significativos), onde observa-se um casamento perfeito de todos os modelos avaliados. No entanto, existe uma diferença significativa entre o modelo e o resultado de simulação para o algoritmo  $\ell_0$ NC, em especial durante o regime transitório. Essa diferença ocorre porque, no início do processo de convergência, a norma  $\ell_0$  do vetor de coeficientes do filtro que está sendo adaptado ainda está distante do valor de  $\delta_{\ell_0}$  definido pelo projetista, em função da identificação ainda imperfeita dos coeficientes significativos e não significativos, o que dificulta a convergência inicial dos coeficientes. Tal comportamento é observado também na curva de MSD, conforme ilustrado na Figura 52. Observando ainda os resultados nesta última figura, verifica-se que, no caso do algoritmo V- $\ell_0$ NC, existe uma predição satisfatória de seu desempenho, quando compara-se a curva de MSD do modelo com a simulação de Monte Carlo.

Os valores esperados de  $\lambda_{\ell_0}(n)$  e  $\lambda_{v\ell_0}(n)$  também são avaliados neste exemplo, com os resultados ilustrados na Figura 53. Observa-se aqui uma concordância muito boa entre os resultados obtidos do modelo e as simulações para  $\lambda_{\ell_0}(n)$ ; no entanto, o mesmo resultado não é observado no caso de  $\lambda_{v\ell_0}(n)$ , principalmente no início do processo de adaptação. Essa diferença está diretamente relacionada à dificuldade de modelar o comportamento dos



coeficientes nos instantes iniciais da convergência, quando os coeficientes não são ainda adequadamente identificados como significativos ou não significativos.

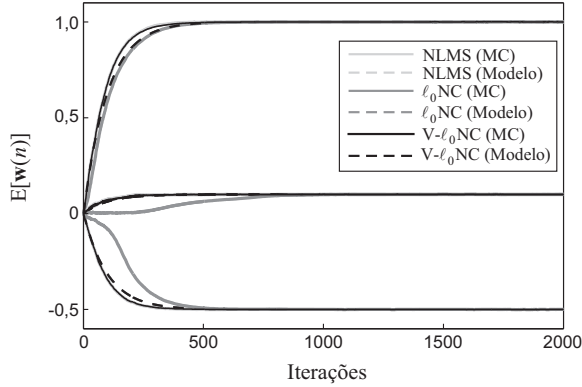


Figura 51: Exemplo 5.4. Comportamento médio dos coeficientes significativos para os algoritmos NLMS,  $\ell_0$ NC e V- $\ell_0$ NC.

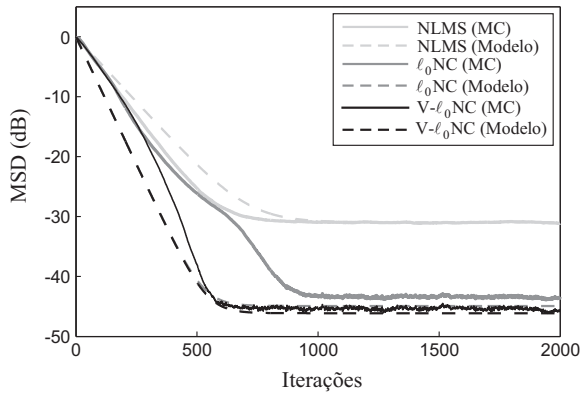


Figura 52: Exemplo 5.4. MSD dos algoritmos NLMS,  $\ell_0$ NC e V- $\ell_0$ NC.

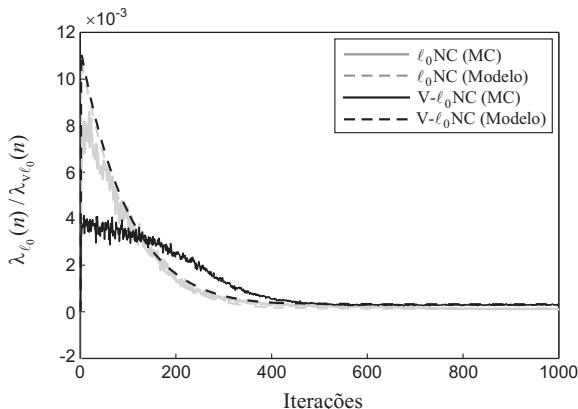


Figura 53: Exemplo 5.4. Valores esperados de  $\lambda_{\ell_0}(n)$  e  $\lambda_{V, \ell_0}(n)$ .

## 5.5 Conclusões

Neste capítulo, foram desenvolvidos modelos estocásticos para os algoritmos baseados em restrição de norma propostos neste trabalho. Os modelos discutidos utilizam a base teórica apresentada para o algoritmo NLMS [60], sendo desenvolvida uma formulação geral para os algoritmos  $\ell_p$ NC. Modelos específicos para os algoritmos baseados nas normas  $\ell_1$  e  $\ell_0$  também foram apresentados e avaliados. Através de exemplos de simulação, é possível verificar que os modelos obtidos são adequados para o estudo analítico dos algoritmos propostos, destacando-se a modelagem do comportamento dos algoritmos em regime permanente. No próximo capítulo, os resultados obtidos para os algoritmos e os modelos propostos são discutidos com mais detalhes e as conclusões finais sobre este trabalho de pesquisa são apresentadas.

## Capítulo 6

### Comentários e Conclusões Finais

Diversas aplicações de filtragem adaptativa possuem plantas com resposta ao impulso esparsa. Os algoritmos adaptativos tradicionais (LMS e NLMS), por sua vez, não levam em consideração a esparsidade dos sistemas a serem identificados, realizando a atualização dos elementos do vetor de coeficientes do filtro sem fazer qualquer distinção entre coeficientes significativos e não-significativos. Dessa forma, uma linha de pesquisa importante vem sendo desenvolvida buscando explorar a informação de esparsidade dos sistemas desconhecidos em benefício do processo adaptativo. Por um lado, os algoritmos da classe proporcional priorizam na adaptação os componentes mais significativos do vetor de coeficientes, atualizando cada elemento de forma proporcional à sua magnitude. Em outra linha de pesquisa, são utilizadas normas vetoriais que identificam o grau de esparsidade do vetor de coeficientes do filtro (em especial  $\ell_1$  e  $\ell_0$ ) em estratégias de adaptação para sistemas esparsos. Neste trabalho, foram apresentados estudos sobre novas estruturas de filtragem adaptativa voltadas à identificação de sistemas esparsos. A proposta é baseada em restrição de norma vetorial, sendo desenvolvida uma formulação geral que permite a utilização de métricas vetoriais diferentes na definição do algoritmo, de acordo com o compromisso entre desempenho e complexidade desejado. A metodologia proposta, em contraste com outras abordagens apresentadas na literatura (tais como os algoritmos APL1 e APWL1), incorpora a restrição de norma vetorial simultaneamente à restrição linear utilizada no algoritmo NLMS. Duas famílias de algoritmos baseados em uma formulação geral foram propostas neste trabalho de pesquisa: os algoritmos com restrição de norma  $\ell_1$  ( $\ell_1$ NC) e os algoritmos com restrição de norma  $\ell_0$  ( $\ell_0$ NC). Simulações numéricas avaliando o desempenho dos algoritmos propostos e comparando-os com outros algoritmos apresentados na literatura foram realizadas. Modelos estocásticos para esses algoritmos também foram desenvolvidos e avaliados. Os principais resultados obtidos nas simulações e na modelagem são discutidos neste capítulo.

## 6.1 Sumário

Este documento de tese de doutorado foi iniciado com uma introdução sobre filtragem adaptativa e filtragem adaptativa para sistemas esparsos, discutindo brevemente sobre as principais linhas de pesquisa voltadas à identificação de sistemas esparsos.

O Capítulo 2 apresentou a fundamentação teórica que dá suporte à este trabalho de pesquisa, discutindo os processos de estimação e de filtragem adaptativa, como também fazendo uma breve revisão sobre os principais algoritmos adaptativos, descritos na literatura, dedicados para a identificação de sistemas esparsos. Na Seção 2.1, foram destacados alguns aspectos importantes sobre o uso de multiplicadores de Lagrange para solução de problemas com restrições e, na Seção 2.2, foram apresentadas abordagens distintas de estimação de sinais, destacando-se as abordagens de estimação pela minimização de uma função custo e pela teoria de conjuntos. A Seção 2.3 discutiu alguns aspectos sobre métricas vetoriais e esparsidade, considerando que normas vetoriais são utilizadas em diversos algoritmos para sistemas esparsos, incluindo os algoritmos propostos neste trabalho. Na Seção 2.4, foram apresentados os principais algoritmos para sistemas esparsos, podendo-se classificá-los em três grupos, de acordo com a estratégia de adaptação: a) os algoritmos de adaptação proporcional, como os algoritmos PNLMS [18], IPNLMS [20] e SPNLMS [45]; b) os algoritmos com penalização de norma baseados na minimização de uma função custo, podendo-se citar os algoritmos ZA-NLMS [27], RZA-NLMS [27] e  $\ell_0$ -NLMS [31]; e c) os algoritmos baseados nas projeções em esferas de normas, tais como os algoritmos APL1 [30] e APWL1 [30].

O Capítulo 3 focou sobre a teoria que fundamenta os algoritmos derivados neste trabalho de pesquisa, apresentando na Seção 3.1 uma formulação geral baseada na otimização com restrição de norma. Na Seção 3.2, foi desenvolvida a formulação dos algoritmos  $\ell_1$ NC, tendo sido discutida uma versão V- $\ell_1$ NC onde a restrição de norma é variável, não exigindo o ajuste desse parâmetro na fase de projeto. Na Seção 3.3, a formulação dos algoritmos  $\ell_0$ NC foi estabelecida. Derivadas dessa formulação, foram também desenvolvidas versões com parâmetro de restrição de norma variável, a saber: os algoritmos V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC, sendo este último baseado em projeções sequenciais nas superfícies de erro *a posteriori* igual a zero e norma  $\ell_0$ .

O Capítulo 4 apresentou os resultados de simulação numérica dos algoritmos propostos, comparando seu desempenho com outros algoritmos dedicados à identificação de sistemas esparsos. As simulações foram realizadas em cenários distintos, incluindo plantas artificiais (com projeto específico para ressaltar características de interesse nos testes) e plantas realísticas (mais

próximas das aplicações práticas), bem como utilizando sinal de entrada gaussiano não-correlacionado (branco) e sinal de entrada gaussiano correlacionado (colorido).

No Capítulo 5, foram discutidos modelos estocásticos para os algoritmos propostos, objetivando o estudo do comportamento médio dos algoritmos através de uma formulação analítica, podendo-se prever melhores configurações de projeto. Na Seção 5.1, foi discutida a modelagem do algoritmo NLMS, que fundamenta a análise dos algoritmos propostos. Na Seção 5.2, é proposto um modelo estocástico geral, que servirá de base à modelagem dos algoritmos baseados nas normas  $\ell_1$  e  $\ell_0$ . A Seção 5.3 tratou da modelagem específica dos algoritmos  $\ell_1$ NC, enquanto na Seção 5.4 foram desenvolvidos os modelos para os algoritmos  $\ell_0$ NC.

## 6.2 Discussão dos Resultados

Os resultados de simulações, apresentados no Capítulo 4, mostraram que os algoritmos propostos neste trabalho são bastante eficientes, destacando-se de outros algoritmos concorrentes em diversos aspectos. Na Seção 4.2.1, onde o algoritmo V- $\ell_1$ NC é comparado com outros algoritmos baseados na norma  $\ell_1$ , observou-se uma clara vantagem desse algoritmo em relação aos demais, principalmente quando o grau de esparsidade da planta é mais elevado. O algoritmo V- $\ell_1$ NC mostrou também bastante robustez à mudança de esparsidade (veja Seção 4.2.2) pelo fato de possuir um parâmetro variável de restrição de norma, o que não ocorre com os algoritmos APL1 e APWL1. Já na avaliação dos algoritmos em um cenário em que os coeficientes não-significativos da planta são diferentes de zero (veja Seção 4.2.3), observou-se que o algoritmo V- $\ell_1$ NC é o que apresenta melhor desempenho, sendo pouco afetado com a adição de ruído à planta. Adicionando-se a essas qualidades, a baixa complexidade computacional (principalmente em comparação com o algoritmo APL1) e a facilidade de projeto (sendo necessário o ajuste apenas do passo de adaptação), o algoritmo V- $\ell_1$ NC mostrou ser interessante em aplicações práticas para plantas exibindo alto grau de esparsidade em que a simplicidade de implantação é uma exigência.

Os algoritmos baseados na norma  $\ell_0$  (V- $\ell_0$ NC, VS- $\ell_0$ NC e Z- $\ell_0$ NC), por sua vez, apresentaram melhor desempenho em comparação com o algoritmo V- $\ell_1$ NC, como se pode observar nas Seções 4.2.2 [veja Figura 38] e 4.2.3 [veja Figura 40(a)]. No entanto, esses algoritmos são mais sensíveis às variações dos valores de coeficientes não-significativos da planta a ser identificada, já que possuem a característica de forçar os coeficientes não-significativos do filtro adaptativo para zero (veja Seção 4.2.3). Por outro lado,

comparando-os com outros algoritmos voltados para sistemas esparsos, os algoritmos  $V\text{-}\ell_0\text{NC}$ ,  $VS\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$  exibiram resultados muito significativos, tanto no que se refere ao desempenho na adaptação quanto à sua sensibilidade a mudanças de esparsidade. Na Seção 4.2.2, pode-se verificar sua grande vantagem em relação aos algoritmos APL1 e APWL1 no que se refere à sensibilidade a mudança de esparsidade, como é ilustrado na Figura 38, sendo seu desempenho semelhante ao de outros algoritmos, como o  $\ell_0\text{-NLMS}$ . Na comparação utilizando plantas mais realísticas (veja Seção 4.2.4), os algoritmos propostos baseados na norma  $\ell_0$  também mostraram resultados bastante significativos. No caso do Exemplo 4.7, utilizando planta de menor grau de esparsidade e sinal de entrada gaussiano colorido (veja Figura 45), os algoritmos  $V\text{-}\ell_0\text{NC}$ ,  $VS\text{-}\ell_0\text{NC}$  e  $Z\text{-}\ell_0\text{NC}$  se destacaram dos demais, com resultados muito superiores.

Sendo assim, os algoritmos adaptativos baseados em projeções com restrição de norma, apresentados neste trabalho, mostraram-se viáveis e bastante eficientes, podendo-se destacar características específicas para cada algoritmo. O algoritmo  $V\text{-}\ell_1\text{NC}$  possui custo computacional bastante próximo ao do algoritmo NLMS, é de fácil ajuste na etapa de projeto e pouco sensível à mudança de esparsidade e de variações dos valores de coeficientes não significativos, sendo bastante interessante em aplicações que exigem robustez e simplicidade. Já o algoritmo  $V\text{-}\ell_0\text{NC}$  apresentou destacado desempenho em todas as avaliações, sempre utilizando passos de adaptação iguais e unitários ( $\mu_1 = \mu_2 = \mu_3 = 1$ ), mostrando-se um algoritmo de fácil ajuste e de desempenho superior a muitos algoritmos dedicados a sistemas esparsos, em especial quando utilizado sinal de entrada correlacionado. O algoritmo  $VS\text{-}\ell_0\text{NC}$ , por sua vez, mesmo utilizando passos de adaptação diferentes ( $\mu_1 \neq \mu_2 \neq \mu_3$ ) em alguns casos, exibiu resultados ligeiramente melhores do que os demais algoritmos baseados em norma  $\ell_0$  para quase todos os exemplos apresentados. Por fim, o algoritmo  $Z\text{-}\ell_0\text{NC}$ , que mostrou resultados semelhantes aos demais algoritmos da família  $\ell_0\text{NC}$ , destacou-se por apresentar o menor custo computacional de todos os algoritmos dedicados a sistemas esparsos aqui estudados.

Resultados muito bons também foram verificados na modelagem dos algoritmos propostos, desenvolvida no Capítulo 5, destacando-se a estreita relação entre os algoritmos propostos e o algoritmo NLMS, observada nas equações de modelo obtidas. Como se pôde constatar através dos exemplos apresentados na Seção 5.3.6, os modelos desenvolvidos para os algoritmos  $\ell_1\text{NC}$  conseguem prever muito bem o comportamento desses algoritmos. Avaliado para a variação de  $\delta_{\ell_1}$  (veja Exemplo 5.1), o modelo do algoritmo com parâmetro de restrição de norma fixo estimou de forma muito precisa seu desempenho em regime permanente, identificando claramente o valor ótimo

a ser obtido para esse parâmetro. Para os algoritmos  $\ell_0$ NC, o modelo apenas não foi capaz de prever precisamente o comportamento dos algoritmos nas iterações iniciais do processo adaptativo, pois, nessa etapa, ainda não é possível identificar quais coeficientes do filtro serão significativos e quais serão não-significativos (veja Exemplos da Seção 5.4.8). Pelo mesmo motivo, a modelagem do comportamento do algoritmo  $\ell_0$ NC em regime permanente não é muito precisa para toda a faixa de valores de  $\delta_{\ell_0}$  (veja Exemplo 5.3). Por outro lado, a análise em regime permanente do algoritmo V- $\ell_0$ NC mostrou-se bastante precisa (veja Exemplo 5.4) e de fácil obtenção, através de uma expressão analítica simples e elegante (5.129) derivada do modelo do algoritmo NLMS.

### 6.3 Trabalhos publicados

Os resultados obtidos neste trabalho de pesquisa deram origem a 2 artigos científicos, sendo que outros 2 estão em fase de desenvolvimento. Os artigos publicados são:

- E. Beck, E. L. O. Batista, and R. Seara, “Norm-constrained adaptive algorithms for sparse system identification based on projections onto intersections of hyperplanes,” *Signal Processing*, vol. 118, pp. 259–271, Jan. 2016.
- E. Beck, E. L. O. Batista e R. Seara, “Um novo algoritmo adaptativo baseado em projeções com restrição de norma  $\ell_0$ ,” in *Anais do XXXIII Simpósio Brasileiro de Telecomunicações (SBRT)*, Juiz de Fora, MG, Brasil, Set. 2015, pp. 648-652.

### 6.4 Sugestões para trabalhos futuros

Para a continuidade do presente trabalho, são propostos os seguintes tópicos de pesquisa complementares:

- Formular estratégias de aprimoramento de projeto dos algoritmos  $\ell_0$ NC em que o parâmetro de limiar  $\beta$  seja obtido em função de informações conhecidas sobre o sistema a ser identificado ou através de métodos adaptativos.
- Aprimorar a modelagem dos algoritmos  $\ell_0$ NC de forma a melhorar a predição do comportamento dos coeficientes não-significativos em todo o processo de adaptação.
- Desenvolver novas formulações de algoritmos com restrição de norma, utilizando outras métricas de esparsidade, especialmente com norma  $\ell_p$  dentro do intervalo  $0 < p < 1$ , tendo como base a formulação geral descrita na Seção 3.1.



## Referências Bibliográficas

- [1] J. Taylor, “Engineering the information age,” *Engineering Management Journal*, vol. 8, no. 6, pp. 277–287, Dec. 1998.
- [2] G. Frantz, “Signal core: a short history of the digital signal processor,” *IEEE Solid-State Circuits Magazine*, pp. 16–20, Jun. 2012.
- [3] T. Takahashi, *Sociedade da Informação no Brasil: Livro Verde*. Brasília, Brasil: Ministério da Ciência e Tecnologia, 2000.
- [4] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [5] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. New York: Springer, 2008.
- [6] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*. New York: John Wiley & Sons, 1998.
- [7] B. Widrow and S. Stearns, *Adaptive Signal Processing*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [8] S. C. Douglas, “Introduction to Adaptive Filters,” in *Digital Signal Processing Handbook*. Boca Raton: LLC, CRC Press, 1999.
- [9] C. Paleologu, J. Benesty, and S. Ciochina, *Sparse Adaptive Filters for Echo Cancellation*. San Rafael, CA: Morgan & Claypool, 2010.
- [10] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin, Germany: Springer, 2006.
- [11] B. Widrow, “Thinking about thinking: the discovery of the LMS algorithm,” *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 100–103, Jan. 2005.
- [12] J. Nagumo and A. Noda, “A learning method for system identification,” *IEEE Transactions on Automatic Control*, vol. 4, no. 3, pp. 282–287, Jun. 1967.
- [13] A. E. Albert and L. A. Gardner Jr., *Stochastic Approximation and Non-Linear Regression*. Cambridge, MA: MIT Press, 1967.

- [14] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, Dec. 2004.
- [15] N. Hurley and S. Rickard, “Comparing measures of sparsity,” *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, Oct. 2009.
- [16] P. Naylor, J. Cui, and M. Brookes, “Adaptive algorithms for sparse echo cancellation,” *Signal Processing*, vol. 86, no. 6, pp. 1182–1192, Jun. 2006.
- [17] ITU-T, *Recommendation G. 711, General Aspects of Digital Transmission Systems*. Geneva, Switzerland: Int. Telecomm. Union, 1993.
- [18] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sep. 2000.
- [19] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in *Proc. Thirty-Second Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1998, pp. 394–398.
- [20] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, Orlando, FL, May 2002, pp. 1881–1884.
- [21] H. Deng and M. Doroslovacki, “Proportionate adaptive algorithms for network echo cancellation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1794–1803, May 2006.
- [22] J. Cui, P. A. Naylor, and D. T. Brown, “An improved IPNLMS algorithm for echo cancellation in packet-switched networks,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, Montreal, Canada, May 2004, pp. 141–144.
- [23] C. Paleologu, J. Benesty, and S. Ciochina, “An improved proportionate NLMS algorithm based on the  $\ell_0$  norm,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, TX, Mar. 2010, pp. 309–312.
- [24] F. C. Souza, O. J. Tobias, R. Seara, and D. R. Morgan, “A PNLMS algorithm with individual activation factors,” *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2036–2047, Apr. 2010.

- [25] P. Loganathan, A. W. H. Khong, and P. A. Naylor, "A class of sparseness-controlled algorithms for echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1591–1601, Nov. 2009.
- [26] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Processing*, vol. 91, no. 8, pp. 1910–1919, Aug. 2011.
- [27] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3125–3128.
- [28] P. L. Combettes, "The foundations of set theoretic estimation," *Proceedings of the IEEE*, vol. 81, no. 2, pp. 182–208, Feb. 1993.
- [29] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, Jan. 2011.
- [30] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted  $\ell_1$  balls," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 936–952, Mar. 2011.
- [31] Y. Gu, J. Jin, and S. Mei, " $\ell_0$  norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, Sep. 2009.
- [32] R. L. Das and M. Chakraborty, "A zero attracting proportionate normalized least mean square algorithm," in *Proc. Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Hollywood, CA, Dec. 2012, pp. 1–4.
- [33] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.
- [34] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley-Interscience, 2000.
- [35] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY: Springer Science, 2006.
- [36] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ Pr, 2004.

- [37] A. H. Sayed, *Adaptive Filters*. New Jersey: John Wiley & Sons, 2008.
- [38] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. New Jersey: Prentice-Hall, 1993.
- [39] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [40] C. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: SIAM, 2000.
- [41] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, p. 129, Dec. 2001.
- [42] J. A. Cadzow, "Minimum  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norm approximate solutions to an overdetermined system of linear equations," *Digital Signal Processing*, vol. 12, no. 4, pp. 524–560, Oct. 2002.
- [43] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
- [44] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [45] M. Doroslovacki, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 181–184, Mar. 2005.
- [46] Y. Zhang, C. Wang, and N. Li, "A fast converging algorithm for system with highly sparse impulse response," in *Proc. International Joint Conference on Computational Sciences and Optimization*, Harbin and Wudalianchi, China, Jun. 2012, pp. 831–834.
- [47] F. C. Souza, R. Seara, and D. R. Morgan, "An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 3301–3307, Jun. 2012.
- [48] G. Hirano and T. Shimamura, "A modified IPNLMS algorithm using system sparseness," in *Proc. International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, New Taipei City, Taiwan, Nov. 2012, pp. 876–879.

- [49] J. Yang and G. E. Sobelman, "A gradient-controlled proportionate technique for acoustic echo cancellation," in *Proc. 2013 Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2013, pp. 1941–1945.
- [50] M. S. Salman, M. N. S. Jahromi, A. Hocanin, and O. Kukrer, "A zero-attracting variable step-size LMS algorithm for sparse system identification," in *Proc. International Symposium on Telecommunications (BIH-TEL)*, Sarajevo, Bosnia and Herzegovina, Oct. 2012, pp. 1–4.
- [51] B. K. Das and M. Chakraborty, "Sparse adaptive filtering by an adaptive convex combination of the LMS and the ZA-LMS algorithms." *IEEE Transactions on Circuits and Systems on Circuits and Systems*, vol. 61, no. 5, pp. 1499–1507, May 2014.
- [52] R. de Lamare and R. Sampaio-Neto, "Sparsity-aware adaptive algorithms based on alternating optimization and shrinkage," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 225–229, Feb. 2014.
- [53] Z. Bo, Z. Lingyun, and C. Chang, "A new LMS algorithm based on sparsity and  $\ell_p$ -norm constraint," in *Proc. International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Hunan, China, Jan. 2014, pp. 25–28.
- [54] I. Yamada, K. Slavakis, and K. Yamada, "An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques," *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1091–1101, May 2002.
- [55] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted  $\ell_1$  minimization," *The Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.
- [56] P. Bradley and O. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. Machine Learning Proceedings of the Fifteenth International Conference (ICML)*, Madison, WI, Jul. 1998, pp. 82–90.
- [57] G. Su, J. Jin, Y. Gu, and J. Wang, "Performance analysis of  $\ell_0$  norm constraint least mean square algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2223–2235, May 2012.
- [58] N. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 793–806, Aug. 1986.

- [59] M. H. Costa and J. Bermudez, “An improved model for the normalized LMS algorithm with Gaussian inputs and large number of coefficients,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, Orlando, FL, May 2002, pp. 1385–1388.
- [60] E. V. Kuhn and R. Seara, “Modelo estocástico do algoritmo NLMS para sinais de entrada gaussianos complexos,” in *Anais do XXX Simpósio Brasileiro de Telecomunicações (SBrT)*, Brasília, DF, Brasil, Set. 2012, pp. 1–5.