

Fábio Luis Perez

**CONTRIBUIÇÕES SOBRE ALGORITMOS ADAPTATIVOS  
LMS NORMALIZADOS PROPORCIONAIS**

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, SC  
2015

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

Perez, Fábio Luis

Contribuições sobre Algoritmos Adaptativos LMS  
Normalizados Proporcionais / Fábio Luis Perez ;  
orientador, Rui Seara - Florianópolis, SC, 2015.  
224 p.

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

Inclui referências

1. Engenharia Elétrica. 2. Engenharia Elétrica.  
3. Processamento de Sinais. 4. Filtragem  
Adaptativa. 5. Algoritmos Adaptativos  
Proporcionais. I. Seara, Rui. II. Universidade  
Federal de Santa Catarina. Programa de  
PósGraduação em Engenharia Elétrica. III. Título.

# CONTRIBUIÇÕES SOBRE ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS PROPORCIONAIS

Fábio Luis Perez

Esta Tese foi julgada adequada para 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, 21 de maio de 2015.

---

Prof. Carlos Galup Montoro, Dr.

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

---

Prof. Rui Seara, Dr. - UFSC  
Orientador

---

Prof. João Marcos Travassos Romano, Dr. - UNICAMP

---

Prof. Luís Geraldo Pedroso Meloni, Dr. - UNICAMP

---

Prof. Eduardo Luiz Ortiz Batista, Dr. - UFSC

---

Prof. Sidnei Noceti Filho, Dr. - UFSC



Dedico este trabalho de pesquisa à minha esposa Fabiana, minha filha Vanessa e meu filho Fábio, pelo apoio e compreensão durante esse período.



## AGRADECIMENTOS

Gostaria de deixar aqui meus sinceros agradecimentos a todos que contribuíram de alguma forma para a realização deste trabalho, em especial:

Ao meu orientador Prof. Rui Seara que, com sua paciência, me orientou com dedicação e entusiasmo durante a realização deste trabalho de tese.

A toda minha família, pela compreensão e apoio nos difíceis momentos dessa grandiosa jornada.

Aos companheiros do programa de pós-graduação, pela amizade e apoio em todos os momentos necessários.

Ao pessoal do LINSE, em especial ao Elton Luiz Fontão, cuja participação (na edição figuras, revisões e formatação de artigos) foi fundamental para a boa finalização desse trabalho de pesquisa.

À Universidade Regional de Blumenau (FURB), pelo suporte financeiro que viabilizou esta formação, em especial aos colegas do Departamento de Engenharia Elétrica e de Telecomunicações, pelo apoio e amizade em todas as horas.

Aos professores membros da banca examinadora, por suas contribuições para o aprimoramento deste trabalho.

A todos que de alguma forma contribuíram para a conclusão deste trabalho de tese.

A Deus por me amparar nos momentos difíceis e pela graça de ter me permitido concluir este trabalho.



## RESUMO

Este trabalho de pesquisa apresenta uma nova política de distribuição de ganho para algoritmos tipo proporcional baseada na convergência individual dos coeficientes. Para isso, uma taxa de variação suavizada e normalizada da magnitude do coeficiente é concebida para avaliação de convergência individual dos coeficientes. A nova abordagem visa melhorar a distribuição de ganho durante o processo adaptativo. Para tal, ganhos associados a coeficientes ativos que estão na vizinhança de seus valores ótimos são reduzidos e redistribuídos a outros coeficientes visando, assim, acelerar a velocidade de convergência global do algoritmo. A partir da nova política de distribuição de ganho, três novas versões de algoritmos tipo proporcional são derivadas. Além disso, uma nova versão do algoritmo adaptativo proporcional ao desvio quadrático médio dos coeficientes ( *$z^2$ -proportionate*) é apresentada. Este último algoritmo combina uma distribuição de ganho proporcional com ganho uniforme. Tal estratégia é dependente do conhecimento do nível de potência do ruído de medição presente no sistema que, na prática, não está sempre disponível. Assim, para contornar essa dependência, um novo procedimento de distribuição de ganho baseado na autocorrelação do sinal de erro é apresentado e discutido. O novo algoritmo supera o algoritmo original em termos de velocidade de convergência e resposta a perturbações na planta. Por fim, uma nova política de distribuição de ganho para algoritmos tipo proporcional para operação em ambientes com elevada esparsidade é proposta. A nova política utiliza uma função de amplificação do ganho de coeficientes ativos visando aumentar sua velocidade de convergência. A partir da nova política, dois novos algoritmos para operação com plantas cujas respostas ao impulso exibem elevada esparsidade são introduzidos. Resultados de simulação corroboram a eficácia dos algoritmos propostos.

**Palavras-Chave:** Algoritmo LMS normalizado proporcional (PNLMS), autocorrelação do sinal de erro, filtragem adaptativa, ganho proporcional ao desvio quadrático médio dos coeficientes, identificação de sistemas, resposta ao impulso esparsa.



## ABSTRACT

This research work presents a new gain distribution policy for proportionate-type algorithms based on individual-coefficient convergence. To this end, a normalized and smoothed variation rate of the individual-coefficient magnitude is derived in order to assess the individual-coefficient convergence. The new approach aims to enhance the gain distribution during the adaptation process. Thereby, gains of the active coefficients that are close to their optimum values are reduced and redistributed to other coefficients, increasing the convergence speed of the algorithm. By using this policy, three new versions of proportionate algorithms have been conceived. Moreover, an alternative version of the mean-square weight deviation-proportionate gain algorithm ( $z^2$ -proportionate) is introduced. This latter algorithm applies a rule combining the mean-square weight deviation-proportionate gain and a uniform gain to obtain the whole algorithm gain distribution. Such a rule is strongly dependent on the knowledge of the measurement noise variance, requiring therefore its estimate. Thereby, a novel approach aiming to circumvent such a dependence, based on error autocorrelation, is presented and discussed. Lastly, a new proportionate gain distribution strategy for operating with plants exhibiting high sparseness is proposed. The new policy uses an amplification function of the gain assigned to active coefficients in order to increase their convergence rate. Thereby, two proportionate algorithms have been developed. Through numerical simulation results, the effectiveness of the proposed algorithms is verified.

**Keywords:** Proportionate normalized least-mean-square (PNLMS) algorithm, error autocorrelation, adaptive filtering, mean-square weight deviation-proportionate gain, system identification, sparse impulse response.



## LISTA DE FIGURAS

- 1.1 Classes básicas de aplicações da filtragem adaptativa. (a) Classe I: identificação. (b) Classe II: modelagem inversa. (c) Classe III: predição. (d) Classe IV: cancelamento de interferência..... 36
- 1.2 Diagrama de blocos de um problema geral de filtragem adaptativa..... 37
- 1.3 Tipos de resposta ao impulso de planta. (a) Resposta não esparsa (ou dispersiva). (b) Resposta esparsa..... 40
- 3.1 Resposta ao impulso  $\mathbf{w}_A^o$  obtida a partir do Modelo 1 de caminho de eco da norma ITU-T Recomendação G.168 com comprimento  $N = 128$  e esparsidade igual a 0,78..... 69
- 3.2 Curvas de desalinhamento normalizado do algoritmo PNLMS e NLMS para sinal de entrada branco com variância unitária usando  $\beta = 0,1$ ,  $\xi = 10^{-4}$ ,  $\rho = 0,01$  e  $\delta = 0,01$ . ..... 69
- 3.3 Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\max} = w_7(n)$ ]. (a) Ganho atribuído. (b) Evolução do coeficiente..... 71
- 3.4 Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\text{int}} = w_5(n)$ ]. (a) Ganho atribuído. (b) Evolução do coeficiente..... 72
- 3.5 Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$ . (a), (c) e (e) Ganho atribuído aos coeficientes inativos. (b), (d) e (f) Evolução do coeficiente..... 75
- 3.6 Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\max} = w_7(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ . ... 80

3.7 Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{int} = w_5(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ . .....81

3.8 Comportamento do coeficiente inativo da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_i = w_{13}(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ . .....82

4.1 Exemplo 4.1. Curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS usando  $\Delta$  igual a  $N/4$ ,  $N/2$  e  $N$  com  $\varepsilon = 5 \times 10^{-2}$ . (a) SNR = 20 dB. (b) SNR = 40 dB. ....94

4.2 Exemplo 4.1. Curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS para  $\Delta = N$  e diferentes valores do limiar  $\varepsilon$ . (a) SNR = 20 dB. (b) SNR = 40 dB. ....95

4.3 Exemplo 4.1. Comportamento da taxa de variação suavizada e normalizada (NSVR) para quatro coeficientes ativos com magnitudes distintas, considerando SNR igual a 20 e 40 dB. ....97

4.4 Exemplo 4.2. Curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . ....99

4.5 Exemplo 4.2. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos PNLMS e NSVR-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. ....100

4.6 Exemplo 4.2. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos PNLMS e NSVR-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. ....101

- 4.7 Exemplo 4.2. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos PNLMS e NSVR-PNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente..... 103
- 4.8 Exemplo 4.3. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . Quando  $n = 10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita. .... 104
- 4.9 Exemplo 4.4. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . Inicialmente a planta variante no tempo  $\mathbf{w}^o$  é igual a  $\mathbf{w}_A^o$  com grau de esparsidade 0,78. Para  $n = 10000$ , o grau de esparsidade da planta é modificado para 0,32. .... 105
- 4.10 Exemplo 4.5. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . A planta  $\mathbf{w}_A^o$  [com esparsidade  $S(\mathbf{w}_A^o) = 0,78$ ] é trocada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^o$  [com esparsidade  $S(\mathbf{w}_B^o) = 0,42$ ]. .... 106
- 4.11 Exemplo 4.6. Curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando um sinal de entrada AR(2) com variância  $\sigma_x^2$  unitária e dispersão  $\chi = 10$  usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$ , em  $n = 20000$ , é deslocada para direita em 12 posições. .... 108
- 4.12 Exemplo 4.7. Curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ . .... 113

- 4.13 Exemplo 4.7. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. ....114
- 4.14 Exemplo 4.7. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. ....115
- 4.15 Exemplo 4.7. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente. ....117
- 4.16 Exemplo 4.8. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ . Quando  $n = 10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita. ....118
- 4.17 Exemplo 4.9. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ . A planta  $\mathbf{w}^o$  é variante no tempo e inicialmente igual  $\mathbf{w}_A^o$  [ $S(\mathbf{w}_A^o) = 0,78$ ]. Para  $n = 10000$ , o grau de esparsidade da planta é alterado para 0,32. ....119
- 4.18 Exemplo 4.10. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$  [com esparsidade  $S(\mathbf{w}_A^o) = 0,78$ ] é trocada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^o$  [com esparsidade  $S(\mathbf{w}_B^o) = 0,42$ ]. ....120

- 4.19 Exemplo 4.11. Curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando um sinal de entrada AR(2) com variância  $\sigma_x^2$  unitária e dispersão  $\chi=10$  usando  $\Delta=N$  e  $\varepsilon=1\times 10^{-3}$ . A planta  $\mathbf{w}_A^o$ , em  $n=15000$ , é deslocada para a direita em 12 posições. .... 121
- 4.20 Exemplo 4.12. Curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta=N$  e  $\varepsilon=5\times 10^{-2}$ . .... 127
- 4.21 Exemplo 4.12. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. .... 128
- 4.22 Exemplo 4.12. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente. .... 129
- 4.23 Exemplo 4.12. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente. .... 131
- 4.24 Exemplo 4.13. Desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta=N$  e  $\varepsilon=5\times 10^{-2}$ . Quando  $n=10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita. .... 132

- 4.25 Exemplo 4.14. Desalinhamento normalizado do algoritmo IAF-PNLMS e NSVR-IAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ . A planta  $\mathbf{w}^o$  é variante no tempo inicialmente igual  $\mathbf{w}_A^o$  com grau de esparsidade 0,78. Para  $n = 10000$ , o grau de esparsidade da planta é 0,32. ....133
- 4.26 Exemplo 4.15. Desalinhamento normalizado dos algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 10 \times 10^{-2}$ . A resposta ao impulso da planta  $\mathbf{w}_A^o$  é mudada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^o$ . ....134
- 4.27 Exemplo 4.16. Desalinhamento normalizado para os algoritmos IAF-PNLMS, EAIFF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada correlacionado com variância  $\sigma_x^2$  unitária e  $\chi = 10$  usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$  é deslocada de 12 amostras para a direita em  $n = 15000$ . ....135
- 4.28 Curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.....138
- 4.29 Curvas de desalinhamento normalizado do algoritmo NSVR-IPNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.....138
- 4.30 Curvas de desalinhamento normalizado do algoritmo NSVR-IAF-PNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.....139

5.1	Curvas de desalinhamento normalizado do algoritmo PDQMC considerando um sinal de entrada branco com variância $\sigma_x^2 = 1$ e variância do ruído de medição $\sigma_v^2 = 17$ dB. ....	146
5.2	Curvas de desalinhamento normalizado do algoritmo PDQMC considerando três valores do parâmetro $\omega$ para sinal de entrada branco com variância $\sigma_x^2 = 1$ e variância do ruído de medição $\sigma_v^2$ é perfeita, isto é, $\hat{\sigma}_v^2 \equiv \sigma_v^2$ . ....	147
5.3	Princípio da distribuição de ganho do novo algoritmo resultante da inter-relação de $m(n)$ com $T_{lo}$ e $T_{hi}$ . ....	152
5.4	Exemplo 5.1. Curvas de desalinhamento normalizado do algoritmo proposto usando $T_{lo}$ igual a $1,2 \times 10^{-2}$ (linha tracejada) e $0,6 \times 10^{-2}$ (linha cinza) com $T_{hi} = 3,5 \times 10^{-2}$ . A resposta ao impulso da planta $\mathbf{w}_A^o$ é trocada para $\mathbf{w}_B^o$ em $n = 5000$ . (a) SNR = 10 dB. (b) SNR = 40 dB. ....	155
5.5	Exemplo 5.1. Curvas de desalinhamento normalizado do algoritmo proposto usando igual a $3 \times 10^{-2}$ (linha tracejada) e $4,5 \times 10^{-2}$ (linha cinza) com $T_{lo} = 0,9 \times 10^{-2}$ . A resposta ao impulso da planta é chaveada de $\mathbf{w}_A^o$ para $\mathbf{w}_B^o$ em $n = 5000$ . (a) SNR = 10 dB. (b) SNR = 40 dB. ....	156
5.6	Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando sinal de entrada branco com variância $\sigma_x^2 = 1$ e uma planta esparsa. Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata. (a) SNR = 10 dB. (b) SNR = 40 dB. ....	157

- 5.7 Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando casos de subestimação da variância do ruído de medição. (a) SNR = 10 dB. (b) SNR = 40 dB. ....158
- 5.8 Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto, considerando casos de superestimação da variância do ruído de medição. (a) SNR = 10 dB. (b) SNR = 40 dB. ....159
- 5.9 Exemplo 5.3. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando sinal de entrada branco com variância  $\sigma_x^2 = 1$  e uma planta com média esparsidade. Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata. (a) SNR = 10 dB. (b) SNR = 40 dB. ....161
- 5.10 Exemplo 5.4. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando um sinal de entrada branco com variância  $A$  resposta ao impulso da planta  $w_B^o$  é deslocada de 12 amostras para a direita em  $n = 6000$ . Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata. ....162
- 5.11 Exemplo 5.5. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto usando sinal de entrada branco com variância  $\sigma_x^2 = 1$  considerando o caso  $\sigma_x^2 = 1$ . de ruído de medição não estacionário dado por (5.13). (a) Estimativa imprecisa. (b) Estimativa perfeita. ....163
- 5.12 Exemplo 5.5. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto usando sinal de entrada branco com variância  $\sigma_x^2 = 1$  considerando um caso de ruído de medição não estacionário dado por (5.14). (a) Estimativa imprecisa. (b) Estimativa perfeita. ....165

5.13	Exemplo 5.6. Curvas de desalinhamento normalizado dos algoritmos CWF, Sub-ótimo 1, Sub-ótimo 2 e do algoritmo proposto para dados de entrada coloridos considerando $\hat{\sigma}_v^2 \equiv \sigma_v^2$ . (a) SNR = 10dB. (b) SNR = 40dB. ....	166
5.14	Exemplo 5.6. Curvas de desalinhamento normalizado dos algoritmos CWF, Sub-ótimo 1, Sub-ótimo 2 e do algoritmo proposto para dados de entrada coloridos, considerando SNR = 10 dB e erros de estimativa na variância do ruído de medição. (a) Caso subestimado (erro de -3 dB). (b) Caso superestimado (erro de +3 dB). ....	167
6.1	Curvas de desalinhamento normalizado dos algoritmos NLMS e IAF-PNLMS para resposta ao impulso de alta esparsidade, considerando resposta ao impulso da planta $\mathbf{w}_C^o$ , dados de entrada branco com variância unitária e SNR = 30 dB. ....	171
6.2	Curvas de desalinhamento normalizado de coeficientes ativos e inativos considerando resposta ao impulso da planta $\mathbf{w}_C^o$ , dados de entrada branco com variância unitária e SNR = 30 dB. (a) Algoritmo NLMS. (b) Algoritmo IAF-PNLMS. ....	175
6.3	Exemplo 6.1. Curvas de desalinhamento normalizado do algoritmo LAF-PNLMS usando $K_a$ igual a 8 e 32 com $K_c = 14$ para sinal de entrada branco com variância $\sigma_x^2$ unitária. ....	182
6.4	Exemplo 6.1. Curvas de desalinhamento normalizado do algoritmo LAF-PNLMS usando $K_c$ igual a 8 e 32 com $K_a = 10$ para sinal de entrada branco com variância $\sigma_x^2$ unitária. ....	182
6.5	Exemplo 6.2. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância $\sigma_x^2$ unitária. ....	183

- 6.6 Exemplo 6.3. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 25000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita. ....185
- 6.7 Exemplo 6.4. Desalinhamento normalizado dos algoritmos PDQMC, MPNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 15000$ , o vetor de coeficientes da planta .. é deslocado de 12 amostras para a direita.....186
- 6.8 Exemplo 6.5. Curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e LAF-PNLMS para sinal de entrada correladiconado e SNR = 40 dB. ....187
- 6.9 Exemplo 6.6. Curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e LAF-PNLMS para sinal de entrada correlacionado e SNR = 40 dB. O vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado em  $n = 100000$  de 12 amostras para a direita.....188
- 6.10 Exemplo 6.7. Curvas de desalinhamento normalizado do algoritmo PAF-PNLMS usando  $K_m$  igual a 2, 3 e 4 com  $K_c = 18$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.....193
- 6.11 Exemplo 6.7. Curvas de desalinhamento normalizado do algoritmo PAF-PNLMS usando  $K_c$  igual a 8 e 32 com  $K_m = 2$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.....193
- 6.12 Exemplo 6.8. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária. ....194

- 6.13 Exemplo 6.9. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 25000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita. .... 196
- 6.14 Exemplo 6.10. Desalinhamento normalizado dos algoritmos PDQMC, MPNLMS e PAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 15000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita. .... 197
- 6.15 Exemplo 6.11. Curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e PAF-PNLMS para sinal de entrada correlacionado e  $\text{SNR} = 40$  dB. .... 198
- 6.16 Exemplo 6.12. Curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e PAF-PNLMS para sinal de entrada correlacionado e  $\text{SNR} = 40$  dB. Em  $n = 100000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita. .... 199



## LISTA DE TABELAS

2.1. ALGORITMO PNLMS.....	47
2.2. ALGORITMO IPNLMS .....	51
2.3. ALGORITMO IIPNLMS .....	52
2.4. ALGORITMO MPNLMS .....	54
2.5. ALGORITMO SC-PNLMS .....	56
2.6. ALGORITMO IAF-PNLMS .....	58
2.7. ALGORITMO PDQMC.....	62
3.1. CARGA COMPUTACIONAL E MEMÓRIA REQUERIDA PARA A DETERMINAÇÃO DA NSVR DA MAGNITUDE DOS COEFICIENTES DO FILTRO ADAPTATIVO.....	78
4.1. ALGORITMO NSVR-PNLMS.....	89
4.2. DISTÂNCIA $D_i$ PARA SNR IGUAL A 20, 30 E 40 dB USANDO $\varepsilon = 5 \times 10^{-2}$ .....	98
4.3. VALORES DE $\tau_i$ PARA SNR IGUAL A 20, 30 E 40 dB USANDO $\varepsilon = 5 \times 10^{-2}$ .....	98
4.4. ALGORITMO NSVR-IPNLMS .....	110
4.5. ALGORITMO NSVR-IAF-PNLMS .....	123
4.6. COMPARAÇÃO DA COMPLEXIDADE COMPUTACIONAL PARA $\Delta = N$ .....	135
5.1. COMPARAÇÃO DE COMPLEXIDADE COMPUTACIONAL .....	152
6.1. ALGORITMO LAF-PNLMS.....	179
6.2. VALOR MÁXIMO E TOTAL DE GANHO DISTRIBUÍDO SOBRE 25000 ITERAÇÕES.....	183
6.3. ALGORITMO PAF-PNLMS .....	190
6.4. VALOR MÁXIMO E TOTAL DE GANHO DISTRIBUÍDO SOBRE 25000 ITERAÇÕES.....	195
6.5. COMPLEXIDADE COMPUTACIONAL PARA $\Delta = N$ .....	200



## LISTA DE ABREVIATURAS E SIGLAS

LMS – *Least-mean-square*

NLMS – *Normalized LMS* (LMS normalizado)

PNLMS – *Proportionate NLMS* (NLMS proporcional)

IPNLMS – *Improved PNLMS* (PNLMS melhorado)

IIPNLMS – *Improved IPNLMS* (IPNLMS melhorado)

SC-PNLMS – *Sparseness controlled PNLMS* (PNLMS controlado por esparsidade)

SC-IPNLMS – *Sparseness controlled IPNLMS* (IPNLMS controlado por esparsidade)

MPNLMS –  $\mu$ -*law PNLMS* (PNLMS com lei  $\mu$ )

AMPNLMS – *Adaptive MPNLMS* (MPNLMS adaptativo)

IAF-PNLMS – *Individual-activation-factor PNLMS* (PNLMS com fatores de ativação individual)

EIAF-PNLMS – *Enhanced IAF-PNLMS* (IAF-PNLMS melhorado)

$l_0$ -LMS – LMS com restrição de norma zero

ZA-LMS – *Zero-attracting LMS* (LMS com atração para zero)

RZA-LMS – *Reweighted zero-attracting LMS* (LMS com atração para zero ponderada).

PDQMC – Proporcional ao desvio quadrático médio dos coeficientes

NSVR – *Normalized smoothed variation rate* (Taxa de variação suavizada e normalizada)

NSVR-PNLMS – PNLMS baseado na NSVR

AR – *Autoregressive* (Auto-regressivo)

NSVR-IPNLMS – IPNLMS baseado na NSVR

NSVR-IAF-PNLMS – IAF-PNLMS baseado na NSVR

CWF – *Colored water-filling* (*water-filling* colorido)

LAF-PNLMS – *Linear amplification function PNLMS* (PNLMS com função de amplificação linear)

PAF-PNLMS – *Proportionate amplification function PNLMS* (PNLMS com função de amplificação proporcional)



## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>33</b>
1.1 Filtragem Adaptativa .....	33
1.2 Categorias Básicas de Aplicação da Filtragem Adaptativa.....	35
1.3 Algoritmos LMS e NLMS .....	37
1.4 Filtragem Adaptativa Proporcional.....	38
1.4.1 Plantas Esparsas.....	39
1.4.2 Algoritmos Tipo Proporcional .....	40
1.5 Objetivos do Trabalho.....	43
1.6 Trabalhos Publicados .....	43
1.7 Organização do Trabalho.....	44
1.8 Conclusões .....	44
<b>2. ALGORITMOS DA CLASSE PROPORCIONAL.....</b>	<b>45</b>
2.1 Introdução .....	45
2.2 Expressões Gerais dos Algoritmos da Classe Proporcional	45
2.3 Algoritmo PNLMS .....	46
2.4 Algoritmo PNLMS++ .....	48
2.5 Algoritmo IPNLMS .....	49
2.6 Algoritmo IIPNLMS .....	51
2.7 Algoritmo MPNLMS .....	52
2.8 Algoritmo SC-PNLMS .....	54
2.9 Algoritmo IAF-PNLMS.....	55
2.10 Algoritmo proporcional ao desvio quadrático médio dos coeficientes (PDQMC) .....	59
2.11 Conclusões .....	63
<b>3. ANÁLISE DA DISTRIBUIÇÃO DE GANHO E COMPORTAMENTO DOS COEFICIENTES DO ALGORITMO PNLMS.....</b>	<b>65</b>
3.1 Introdução .....	65
3.2 Ganhos de Adaptação do Algoritmo PNLMS.....	65
3.3 Impacto da Distribuição de ganho e Convergência dos Coeficientes .....	68
3.4 Taxa de Variação da Magnitude do Coeficiente do Filtro Adaptativo.....	75
3.4.1 Taxa de Variação Suavizada e Normalizada (NSVR) da Magnitude do Coeficiente.....	76
3.4.2 Complexidade Computacional e Memória Requerida ....	77

3.4.3 Eficácia do Método de Estimação da Taxa de Variação da Magnitude dos Coeficientes .....	79
3.5 Conclusões.....	82
<b>4. NOVA POLÍTICA DE DISTRIBUIÇÃO DE GANHO PARA ALGORITMOS TIPO PROPORCIONAL BASEADA NA CONVERGÊNCIA DO COEFICIENTE .....</b>	<b>85</b>
4.1 Introdução.....	85
4.2 Política de Distribuição de Ganho .....	86
4.3 Algoritmo PNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente.....	87
4.3.1 Algoritmo NSVR-PNLMS .....	87
4.3.2 Análise da Distribuição de Ganho do Algoritmo NSVR-PNLMS.....	90
4.4 Resultados de Simulação do Algoritmo NSVR-PNLMS .....	92
4.4.1 Exemplo 4.1.....	93
4.4.2 Exemplo 4.2.....	98
4.4.3 Exemplo 4.3.....	103
4.4.4 Exemplo 4.4.....	104
4.4.5 Exemplo 4.5.....	105
4.4.6 Exemplo 4.6.....	106
4.5 Algoritmo IPNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente.....	107
4.5.1 Algoritmo NSVR-IPNLMS .....	108
4.5.2 Análise da Distribuição de Ganho do Algoritmo NSVR-IPNLMS .....	109
4.6 Resultados de Simulação do Algoritmo NSVR-IPNLMS .....	112
4.6.1 Exemplo 4.7.....	112
4.6.2 Exemplo 4.8.....	117
4.6.3 Exemplo 4.9.....	118
4.6.4 Exemplo 4.10.....	119
4.6.5 Exemplo 4.11 .....	120
4.7 Algoritmo IAF-PNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente .....	121
4.7.1 Algoritmo NSVR-IAF-PNLMS.....	122
4.7.2 Análise da Distribuição de Ganho do Algoritmo NSVR-IAF-PNLMS .....	124
4.8 Resultados de Simulação do Algoritmo NSVR-IAF-PNLMS..	126
4.8.1 Exemplo 4.12.....	126
4.8.2 Exemplo 4.13.....	131
4.8.3 Exemplo 4.14.....	131

4.8.4 Exemplo 4.15.....	132
4.8.5 Exemplo 4.16.....	133
4.9 Complexidade Computacional.....	134
4.9.1 Redução da Complexidade Computacional.....	136
4.10 Discussões Sobre a Nova Política de Ganho Proporcional....	139
4.11 Conclusões.....	141
<b>5. ALGORITMO PROPORCIONAL AO DESVIO QUADRÁTICO MÉDIO DOS COEFICIENTES BASEADO NA AUTOCORRELAÇÃO DO ERRO.....</b>	<b>143</b>
5.1 Introdução.....	143
5.2 Impacto da Estimativa da Variância do Ruído de Medição e do Parâmetro $\omega$ sobre o Comportamento do Algoritmo PDQMC.....	144
5.3 Abordagem Proposta.....	147
5.3.1 Autocorrelação do Erro.....	148
5.3.2 Algoritmo Proposto.....	148
5.3.3 Influência da Variância do Ruído de Medição sobre $m(n)$ .....	150
5.3.4 Complexidade Computacional.....	151
5.4 Distribuição de ganho e Escolha dos Valores de $T_{lo}$ e $T_{hi}$ do Algoritmo Proposto.....	152
5.5 Resultados de Simulação.....	153
5.5.1 Exemplo 5.1.....	153
5.5.2 Exemplo 5.2.....	154
5.5.3 Exemplo 5.3.....	160
5.5.4 Exemplo 5.4.....	160
5.5.5 Exemplo 5.5.....	162
5.5.6 Exemplo 5.6.....	164
5.6 Conclusões.....	168
<b>6. NOVA ABORDAGEM PARA ALGORITMOS TIPO PROPORCIONAL PARA OPERAÇÃO EM AMBIENTES COM ELEVADA ESPARSIDADE.....</b>	<b>169</b>
6.1 Introdução.....	169
6.2 Análise do Comportamento de Algoritmos Adaptativos em Plantas com Elevada Esparsidade.....	169
6.3 Abordagem Proposta.....	176
6.4 Algoritmo Adaptativo Proporcional Baseado em Função de Amplificação de Ganho Linear.....	177
6.4.1 Algoritmo LAF-PNLMS.....	177
6.4.2 Distribuição de Ganho do Algoritmo LAF-PNLMS.....	178

6.5 Resultados de Simulação do Algoritmo LAF-PNLMS .....	181
6.5.1 Exemplo 6.1.....	181
6.5.2 Exemplo 6.2.....	182
6.5.3 Exemplo 6.3.....	184
6.5.4 Exemplo 6.4.....	185
6.5.5 Exemplo 6.5.....	186
6.5.6 Exemplo 6.6.....	187
<b>6.6 Algoritmo Adaptativo Proporcional Baseado em Função de Amplificação de Ganho Proporcional.....</b>	<b>188</b>
6.6.1 Algoritmo PAF-PNLMS.....	188
6.6.2 Distribuição de Ganho do Algoritmo PAF-PNLMS.....	191
6.7 Resultados de Simulação do Algoritmo PAF-PNLMS.....	192
6.7.1 Exemplo 6.7.....	192
6.7.2 Exemplo 6.8.....	194
6.7.3 Exemplo 6.9.....	196
6.7.4 Exemplo 6.10.....	196
6.7.5 Exemplo 6.11.....	197
6.7.6 Exemplo 6.12.....	198
6.8 Complexidade Computacional.....	199
6.9 Conclusões.....	201
<b>7. COMENTÁRIOS E CONCLUSÕES FINAIS .....</b>	<b>203</b>
<b>Apêndice I. Análise Estocástica da Taxa de Variação Suavizada e Normalizada da Magnitude do Coeficiente Adaptativo .....</b>	<b>207</b>
<b>Apêndice II. Análise Estocástica da Autocorrelação Quadrática do Erro Normalizado.....</b>	<b>209</b>
<b>Apêndice III. Impacto da Alocação de Ganhos a Coeficientes Ativos e Inativos no Erro em Excesso em Regime Permanente.....</b>	<b>213</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>217</b>

## INTRODUÇÃO

### 1.1 Filtragem Adaptativa

Nas últimas décadas, a área de processamento digital de sinais tem registrado significativos progressos motivados, principalmente, por um expressivo avanço tecnológico relacionado ao desenvolvimento de processadores digitais de sinais (DSP) cada vez mais rápidos, eficientes e de baixo custo. Os filtros digitais são um exemplo de aplicação de processamento digital de sinais, os quais são capazes de transformar uma sequência de números em outra e, com isso, modelar adequadamente o comportamento de um sistema real [1], [2]. Filtros digitais invariantes no tempo possuem estrutura e parâmetros fixos para representar uma função de transferência racional e são amplamente utilizados para remoção de ruídos, processamento de som e imagem, sistemas de comunicação, dentre outras aplicações [3]. Considerando conhecidas as características do sinal a ser filtrado (sinal de entrada), a estrutura a ser utilizada e as especificações da filtragem requerida, é possível realizar o projeto de um filtro ótimo satisfazendo algum dado critério de desempenho. Contudo, quando o conhecimento das características do sinal de entrada não está totalmente disponível ou, ainda, em casos de sistemas variantes no tempo, o projeto de um filtro digital fixo (convencional) ótimo torna-se inviável. A utilização de filtros adaptativos pode ser uma solução interessante nesse contexto, uma vez que os filtros adaptativos são variantes no tempo e seus parâmetros alterados através de um algoritmo recursivo visando atender um determinado critério de desempenho. Assim, o uso da filtragem adaptativa tem se mostrado eficiente em uma vasta gama de aplicações, como em sistemas para cancelamento de eco, equalização de canais, controle de ruído ativo, conformação de feixes, dentre outras [1], [2], [4]–[6]. Por exemplo, no controle da intensidade de ruídos acústicos considerando ruídos de baixa frequência e casos em que as condições de operação se modificam continuamente (resultando em alterações nos componentes de frequência e nos níveis de perturbações indesejáveis), a utilização de filtros adaptativos tem demonstrado muito bons resultados [7]–[14]. Outra aplicação desafiadora ao uso de filtragem adaptativa é na

conformação de feixe em arranjo de antenas para comunicações móveis celulares. Em tais sistemas, o crescente número de usuários e a necessidade de maiores taxas de transmissão vêm exigindo o desenvolvimento de novas técnicas que visam aumentar a quantidade/qualidade dos canais de comunicação [15]-[17]. Para tal, arranjos adaptivos de antenas (também chamados de antenas inteligentes) têm se mostrado uma importante ferramenta de melhoria para os sistemas de comunicações móveis [17]-[20].

De fundamental importância para os atuais sistemas de telecomunicação, a filtragem adaptativa em sistemas de cancelamento de eco também tem sido alvo de muitos trabalhos de pesquisa na área de processamento de sinais. Em geral, há dois tipos de ecos em sistemas de comunicação, isto é, eco acústico e eco de linha. O eco acústico é ocasionado pela reflexão do som a partir de múltiplos caminhos e superfícies, tais como paredes, tetos e pisos. Por outro lado, o eco de linha é originado nas híbridas (dispositivos utilizados em telefonia para realizar a transição entre circuitos a quatro fios para circuitos a dois fios e vice-versa) e causado pelo retorno do sinal transmitido em função do descasamento de impedância entre os circuitos a dois e quatro fios [6]. O princípio básico de um sistema de cancelamento de eco é obter, em sua saída, a partir de algoritmos de filtragem adaptativa, uma réplica “em tempo real” do sinal de eco, que deve ser subtraída do sinal contaminado pelo eco. Contudo, algumas características específicas encontradas em aplicações de cancelamento de eco, tais como caminhos de eco variantes no tempo e respostas com comprimento excessivamente grande, representam um grande desafio para qualquer algoritmo adaptativo [21]. Em geral, a fim de tratar com respostas ao impulso de grande comprimento e variantes no tempo, o algoritmo “ideal” deve ter uma alta velocidade de convergência com baixo erro de desajuste associado a uma adequada capacidade de rastreamento; além do mais, sua complexidade computacional deve ser moderada, fornecendo tanto um desempenho apropriado quanto um baixo custo para implementação em tempo real. Nesse contexto, os algoritmos *least-mean-square* (LMS) [4] e o LMS normalizado (NLMS - *normalized LMS*) [5] vêm sendo utilizados em aplicações de cancelamento de eco principalmente em função da baixa complexidade computacional associada à sua muito boa estabilidade. Por outro lado, ainda existe necessidade de melhorar o desempenho de tais algoritmos para esse tipo de aplicação, mais especificamente de aumentar a velocidade de convergência e a habilidade de rastreamento, uma vez que o desempenho de tais algoritmos é deficiente para casos de filtros adaptativos de grande comprimento. Tal restrição pode ser contornada

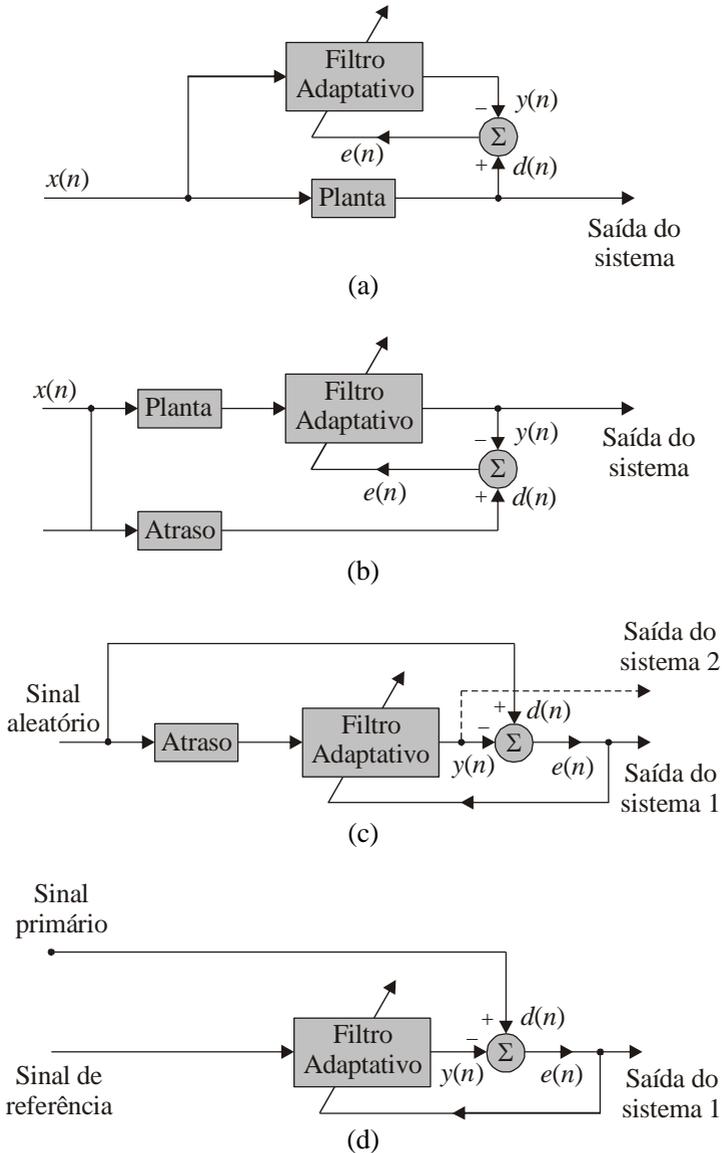
explorando-se algumas características do sistema a ser identificado, como, por exemplo, o tipo de resposta ao impulso do caminho de eco que, em muitos casos, é uma resposta esparsa.

## 1.2 Categorias Básicas de Aplicação da Filtragem Adaptativa

Apesar de a filtragem adaptativa ser atualmente utilizada em diversos campos de aplicação, a forma na qual a resposta desejada é extraída define as categorias básicas de uso da filtragem adaptativa [5]. Assim, a filtragem adaptativa pode ser classificada em quatro categorias principais, conforme a aplicação desejada como ilustrado na Figura 1.1 [2], [4]. Nesses esquemas  $x(n)$  representa o sinal de entrada,  $y(n)$ , o sinal de saída do filtro e  $d(n)$ , a resposta desejada. O erro de estimação  $e(n)$  é definido como  $e(n) = d(n) - y(n)$ .

As quatro classes básicas de aplicações da filtragem adaptativa são resumidamente descritas a seguir:

- I) **Identificação** [Figura 1(a)]. Nesse tipo de aplicação, o filtro adaptativo é utilizado para prover um modelo linear que melhor represente um sistema desconhecido.
- II) **Modelagem inversa** [Figura 1(b)]. A função do filtro adaptativo é obter o melhor modelo inverso de um sistema desconhecido. No caso de sistemas lineares, em uma situação ideal, o modelo inverso representa a função inversa do sistema desconhecido.
- III) **Predição** [Figura 1(c)]. Nessa configuração, o filtro adaptativo visa disponibilizar a melhor predição para a evolução de um sinal tendo como base o conhecimento de seus valores passados.
- IV) **Cancelamento de interferência** [Figura 1(d)]. Nessa aplicação, o filtro adaptativo é usado para cancelar uma interferência desconhecida contida no sinal primário (sinal com a informação pretendida).



**Figura 1.1** - Classes básicas de aplicações de filtragem adaptativa. (a) Classe I: identificação. (b) Classe II: modelagem inversa. (c) Classe III: predição. (d) Classe IV: cancelamento de interferência.

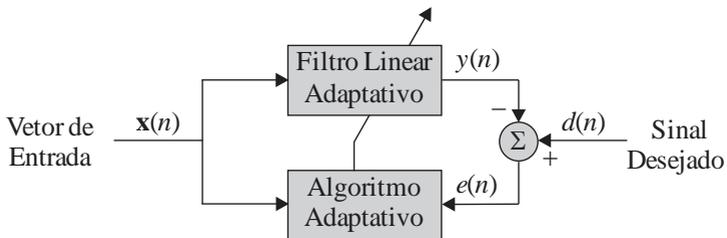
### 1.3 Algoritmos LMS e NLMS

Em um contexto de reconhecimento de padrões, o algoritmo LMS [4], resultante dos primeiros esforços de pesquisa na área de filtragem adaptativa, introduzido por Bernard Widrow e Marcian E. Hoff Jr em 1960, tem como critério de desempenho (função custo) o valor quadrático médio do sinal de erro. O algoritmo LMS utiliza uma estimativa do gradiente da função custo baseada em uma única amostra do sinal de erro para atualizar os coeficientes do filtro adaptativo. A ampla utilização desse algoritmo em sistemas práticos vem principalmente de sua simplicidade computacional, convergência não polarizada e robustez [4], [5], [22]. Aplicado com sucesso primeiramente em antenas adaptativas e cancelamento de ruído [22], [23], a utilização do algoritmo LMS em aplicações práticas teve um forte impulso após o desenvolvimento do equalizador adaptativo por R. W. Lucky em 1965 (referido na literatura técnica como *decision-direct learning*) [4], [22] e do cancelador de eco [22], [24] implementado por M. M. Sondhi e A. J. Presti em 1966.

Considerando um problema geral de filtragem adaptativa conforme mostrado na Figura 1.2, o algoritmo LMS pode ser sucintamente descrito pela equação de atualização dos coeficientes do filtro adaptativo, dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \beta e(n) \mathbf{x}(n) \quad (1.1)$$

onde  $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]^T$  denota o vetor do sinal de entrada,  $\beta$  representa o valor do passo de adaptação que controla a estabilidade e velocidade de convergência e  $\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \dots \ w_N(n)]^T$  o vetor de dimensão  $N$  correspondente aos coeficientes do filtro adaptativo.



**Figura 1.2** - Diagrama de blocos de um problema geral de filtragem adaptativa.

Contudo, apesar da baixa complexidade computacional, o LMS apresenta velocidade de convergência baixa e seu desempenho é fortemente afetado pelo nível de potência do sinal de entrada. Assim, para reduzir os efeitos causados pela variação da potência do sinal de entrada no algoritmo LMS, foi proposto o algoritmo NLMS [5]. O NLMS normaliza o passo de adaptação levando em conta a potência do sinal de entrada resultando em maior velocidade de convergência. Além disso, o algoritmo NLMS possui baixa complexidade computacional quando comparado com outros algoritmos adaptativos e exibe muito boas características de estabilidade. A equação de atualização dos coeficientes do filtro adaptativo do algoritmo NLMS é dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\mathbf{x}^T(n)\mathbf{x}(n)} e(n) \mathbf{x}(n). \quad (1.2)$$

Ainda hoje, o algoritmo LMS e suas variantes, como o NLMS, são muito utilizados e objetos de diversos estudos de pesquisa na área de processamento adaptativo de sinais.

#### 1.4 Filtragem Adaptativa Proporcional

Além das aplicações já mencionadas anteriormente, o algoritmo LMS e o NLMS vêm sendo utilizados em muitas outras aplicações, tais como localização de fontes, sistemas de televisão digital e processos sísmicos e químicos [5]. Uma característica comum em todas essas aplicações é a natureza esparsa da resposta ao impulso do sistema a ser estimado [25]–[31]. Dessa forma, a concepção de filtros que exploram tal característica tem se tornado uma área de pesquisa ativa e de grande relevância. Para essa classe de respostas ao impulso, os algoritmos adaptativos clássicos, utilizando o mesmo passo de adaptação para todos os coeficientes do filtro, tais como o LMS e o NLMS, são superados em termos de velocidade de convergência e habilidade de rastreamento por aqueles algoritmos especialmente projetados para tirar proveito da natureza esparsa da resposta ao impulso [29]. Por exemplo, algoritmos do tipo proporcional representam um importante segmento de algoritmos adaptativos que utilizam a esparsidade da planta como forma de melhorar suas características de convergência.

A seguir, plantas<sup>1</sup> esparsas são sucintamente discutidas, assim como o grau de esparsidade é definido. Adicionalmente, um breve histórico dos algoritmos tipo proporcional é apresentado.

### 1.4.1 Plantas Esparsas

Respostas ao impulso esparsas são comumente encontradas em diversas aplicações práticas, tais como comunicação sem fio, radar, sonar, cancelamento de eco, dentre outros. Uma resposta ao impulso (planta) é qualitativamente classificada como esparsa se a maioria de seus coeficientes apresenta magnitude nula ou próxima de zero e somente alguns poucos coeficientes têm magnitudes significativas. Para quantificar a esparsidade de uma planta, uma métrica típica encontrada na literatura é o grau de esparsidade. Tal métrica é baseada na relação entre a norma 1 e norma 2 da resposta ao impulso, definida como [26], [32]

$$S(\mathbf{w}^o) \triangleq \frac{N}{N - \sqrt{N}} \left( 1 - \frac{\|\mathbf{w}^o\|_1}{\sqrt{N} \|\mathbf{w}^o\|_2} \right) \quad (1.3)$$

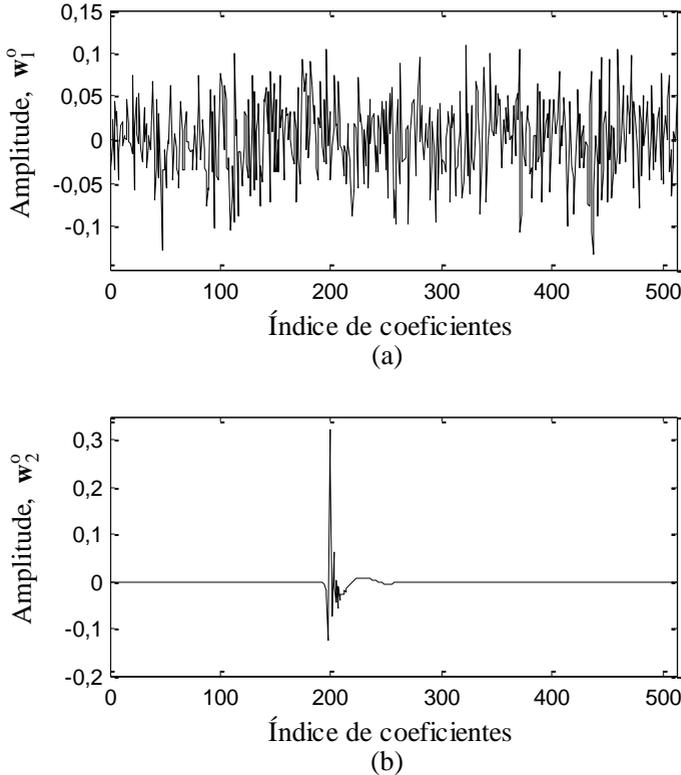
onde o vetor  $\mathbf{w}^o$  representa uma resposta ao impulso de dimensão  $N$ , e  $\|\mathbf{w}^o\|_1$  e  $\|\mathbf{w}^o\|_2$  denotam, respectivamente, a norma  $\ell_1$  e a norma  $\ell_2$  de  $\mathbf{w}^o$ . Note que o valor do grau de esparsidade  $S(\mathbf{w}^o)$  varia entre 0 (esparsidade do filtro uniforme, isto é, todos os coeficientes com mesma magnitude) a 1 (esparsidade do filtro de Dirac).

Para exemplificar a esparsidade da resposta ao impulso da planta, considere duas plantas ( $\mathbf{w}_1^o$  e  $\mathbf{w}_2^o$ ) cujas respostas ao impulso, com comprimento de 512 coeficientes, são ilustradas na Figura 1.3 [33]. Observe que a resposta ao impulso  $\mathbf{w}_1^o$  mostrada na Figura 1.3(a) (também chamada de planta dispersiva) possui muitos coeficientes significativos, ao passo que a resposta ao impulso  $\mathbf{w}_2^o$  mostrada na Figura 1.3(b) possui muitos coeficientes com magnitude zero (ou próximas de zero) e somente alguns poucos coeficientes com magnitude significativa [31]. Agora, aplicando-se (1.3) para a determinação do grau de esparsidade dessas respostas, obtém-se  $S(\mathbf{w}_1^o) = 0,1991$  para a primeira planta e  $S(\mathbf{w}_2^o) = 0,8970$  para a segunda, refletindo a nítida

---

<sup>1</sup> Na literatura especializada, os termos planta e resposta ao impulso da planta são usados indistintamente.

diferença de esparsidade entre tais plantas. Plantas com resposta ao impulso esparsa são aqui chamadas, por simplicidade, plantas esparsas.



**Figura 1.3** – Tipos de resposta ao impulso de planta. (a) Resposta não esparsa (ou dispersiva). (b) Resposta esparsa.

### 1.4.2 Algoritmos Tipo Proporcional

Considerando o grande número de plantas esparsas observadas em aplicações práticas [34]-[36], a concepção de algoritmos adaptativos que explorem tal característica tem se tornado cada vez mais relevante. O algoritmo denominado NLMS proporcional (PNLMS - *proportionate NLMS*), proposto por Duttweiler em 2000 [38], foi um dos primeiros a explorar a natureza esparsa da planta. Utilizando ganhos individuais para atualizar cada coeficiente do filtro adaptativo de forma proporcional à sua

magnitude, o algoritmo PNLMS exibe uma maior velocidade de convergência para sistemas cuja planta é esparsa, quando comparado com aqueles que usam o mesmo passo de adaptação para todos os coeficientes do filtro [25], [37]–[43].

Apesar de o algoritmo PNLMS exibir maior velocidade de convergência e melhor capacidade de rastreamento do que o LMS e NLMS para os casos de plantas esparsas, essas melhores características não são mantidas em todo o processo de adaptação [39], [44]. Além disso, o algoritmo PNLMS tem seu desempenho deteriorado à medida que a esparsidade da planta diminui [45]. Para contornar tais problemas, versões melhoradas do algoritmo PNLMS vêm sendo propostas na literatura especializada [37], [39], [40], [46–48]. Por exemplo, o algoritmo PNLMS melhorado (IPNLMS – *improved* PNLMS) considera uma mistura de ganhos de adaptação proporcional e não proporcional, a qual é ponderada por um parâmetro específico [39]. Como resultado, o desempenho do algoritmo IPNLMS pode ser controlado de acordo com a esparsidade da planta, conduzindo a melhores características de convergência do que o PNLMS padrão para uma faixa maior de valores de esparsidade. Em [46], uma nova versão do algoritmo IPNLMS, denominada IPNLMS melhorado (IIPNLMS - *improved* IPNLMS), é proposta. Tal versão introduz dois parâmetros de ponderação e, com base na magnitude do coeficiente, mistura os ganhos de adaptação proporcional e não proporcional com pesos distintos melhorando, assim, as características de convergência do filtro adaptativo. Em [49] e [50], buscando maior robustez com respeito à variação de esparsidade da planta, foram propostos os algoritmos SC-PNLMS (*sparseness controlled* PNLMS) e SC-IPNLMS (*sparseness controlled* IPNLMS), respectivamente. Esses algoritmos possuem desempenho satisfatório para plantas com valores de esparsidade elevados, assim como para aquelas com valores de esparsidade médio. Entretanto, tal comportamento é obtido às custas de uma elevada carga computacional quando comparada com aquela do algoritmo PNLMS original. Objetivando preservar a rápida convergência inicial durante todo o processo de adaptação, o PNLMS com lei  $\mu$  (MPNLMS -  $\mu$ -*law* PNLMS) e MPNLMS adaptativo (AMPNLMS – *adaptive* MPNLMS) foram propostos em [41] e [51], respectivamente; porém, tais algoritmos são computacionalmente muito mais complexos do que o PNLMS original. Em [48], o algoritmo PNLMS com fatores de ativação individuais (IAF-PNLMS) tem sido introduzido, o qual usa fatores de ativação individuais para cada coeficiente do filtro adaptativo ao invés de um fator global como o utilizado pelo PNLMS

original. Tal estratégia conduz a um aumento de ganho atribuído a coeficientes ativos em relação ao algoritmo PNLMS e, como consequência, maior velocidade de convergência é obtida. O algoritmo IAF-PNLMS melhorado (EIAF-PNLMS – *enhanced* IAF-PNLMS), dado em [52], usa uma versão suavizada do erro quadrático instantâneo  $e^2(n)$  como medida de proximidade do regime permanente para trocar a política de ganho proporcional para ganho uniforme e, desse modo, aumentar o ganho atribuído a coeficientes inativos, melhorando assim a convergência global do filtro adaptativo.

Em [53], é proposto o algoritmo  $l_0$ -LMS ( $l_0$  norm LMS) e em [54], os algoritmos ZA-LMS (*zero attracting* LMS) e RZA-LMS (*reweighted zero attracting* LMS). Tais algoritmos são baseados na inclusão de uma restrição na função custo sendo que, no primeiro algoritmo, é utilizada como restrição a norma  $l_0$  dos coeficientes do filtro adaptativo e, nos dois últimos, a norma  $l_1$ . Durante a adaptação, esses algoritmos reduzem o valor dos coeficientes cuja magnitude seja próxima de zero e, com isso, aumentam a velocidade de convergência em relação ao algoritmo LMS para plantas esparsas.

Recentemente, dois novos algoritmos da classe proporcional foram propostos em [33]. Eles são denominados *water-filling* e proporcional ao desvio quadrático médio dos coeficientes<sup>2</sup> (PDQMC). O primeiro, considerando sinal de entrada branco, baseia-se na minimização do erro quadrático médio a cada iteração com respeito ao ganho alocado aos coeficientes do filtro adaptativo. O segundo (denominado por seus autores *z<sup>2</sup>-proportionate*), é derivado do primeiro e possui desempenho global melhor do que seu predecessor com reduzida complexidade computacional. Em contraste a outros algoritmos disponíveis na literatura que utilizam a magnitude dos coeficientes para definir o ganho, os dois algoritmos apresentados em [33] usam uma estimativa do desvio quadrático médio dos coeficientes obtida a partir do sinal de erro. A distribuição de ganho utilizada no algoritmo PDQMC combina o ganho proporcional ao desvio quadrático médio dos coeficientes com ganho unitário, tal que sua distribuição se torne uniforme em regime permanente. Essa mistura é controlada por um parâmetro que depende do conhecimento do valor da variância do ruído de medição. Assim, uma estimativa dessa variância se faz necessária para uma operação adequada

---

<sup>2</sup> Desvio do coeficiente: diferença entre o valor ótimo do coeficiente e seu valor corrente.

do algoritmo. Além disso, o algoritmo PDQMC tem seu desempenho severamente degradado sempre que a estimativa da variância do ruído de medição não é precisamente determinada. Outros trabalhos relacionados a algoritmos adaptativos em aplicações cuja resposta ao impulso da planta é esparsa podem ser encontrados em [55]-[59].

## 1.5 Objetivos do Trabalho

Ao longo dos últimos anos, visando melhorar o desempenho de algoritmos adaptativos para casos em que a resposta ao impulso da planta exibe alta ou média esparsidade, diversos algoritmos da classe proporcional vêm sendo propostos. Contudo, problemas decorrentes de variações da esparsidade da planta, alto erro em excesso<sup>3</sup>, carga computacional elevada, dentre outros, são comuns em algoritmos da família PNLMS. Nesse contexto, o objetivo deste trabalho é formular novas estratégias visando melhorar algumas características dos algoritmos adaptativos da classe proporcional, tais como desempenho, robustez e aplicabilidade, seja pelo aprimoramento de algoritmos já consagrados na literatura ou pela proposição de novos algoritmos.

## 1.6 Trabalhos Publicados

Durante a realização deste trabalho de pesquisa, três artigos foram publicados em fóruns especializados, os quais são listados a seguir:

- 1) F. L. Perez, F. C. de Souza, R. Seara, “Algoritmo com ganho proporcional ao desvio médio quadrático dos coeficientes baseado na autocorrelação do erro, ” in *XXX Simpósio Brasileiro de Telecomunicações (SBrT 2012)*, Brasília-DF, Brazil, Sep. 2012.
- 2) F. L. Perez, F. C. de Souza, R. Seara, “Nova estratégia de distribuição de ganho de adaptação aplicada ao algoritmo IAF-PNLMS, ” in *XXXI Simpósio Brasileiro de Telecomunicações (SBrT 2013)*, Fortaleza-CE, Brazil, Sep. 2013.
- 3) F. L. Perez, F. C. de Souza, R. Seara, “An improved mean-square weight deviation-proportionate gain algorithm based on error autocorrelation,” *Signal Process.*, vol. 94, no. 1, pp. 503-513, Jan. 2014.

---

<sup>3</sup> Erro decorrente da flutuação dos coeficientes do filtro adaptativo.

## 1.7 Organização do Trabalho

Este trabalho está organizado conforme descrito a seguir. No Capítulo 2, alguns dos mais relevantes algoritmos da classe proporcional são revisitados, destacando suas principais características e particularidades. No Capítulo 3, os efeitos dos ganhos de adaptação sobre o comportamento do algoritmo PNLMS são discutidos bem como uma forma de estimativa da taxa de variação da magnitude dos coeficientes do filtro adaptativo. No Capítulo 4, uma nova política de distribuição de ganho para algoritmos tipo proporcional baseada na taxa de variação da magnitude do coeficiente é proposta. Além disso, três versões melhoradas dos algoritmos PNLMS, IPNLMS e IAF-PNLMS são apresentadas, discutindo suas características, propriedades e desempenho. No Capítulo 5, uma modificação do algoritmo PDQMC é proposta visando reduzir sua forte dependência com respeito à variância do ruído de medição. Ainda no Capítulo 5, a eficácia do método proposto é verificada através de resultados de simulação considerando níveis de ruído de medição distintos. No Capítulo 6, dois novos algoritmos adaptativos para operação em plantas apresentando alta esparsidade são introduzidos, assim como são discutidas suas políticas de distribuição de ganho e desempenho. Finalmente, no Capítulo 7 são apresentados os comentários e as conclusões finais desta tese de doutorado.

## 1.8 Conclusões

Neste capítulo, foram apresentados os conceitos básicos concernentes à filtragem adaptativa bem como a classificação típica de suas aplicações. Além disso, a definição de grau de esparsidade e as expressões de atualização dos coeficientes do filtro adaptativo dos algoritmos LMS e NLMS são descritas. Também, as principais versões do algoritmo PNLMS e suas características foram sucintamente discutidas assim como os objetivos gerais deste trabalho de pesquisa.

## ALGORITMOS DA CLASSE PROPORCIONAL

### 2.1 Introdução

Conforme descrito no Capítulo 1, os algoritmos adaptativos da classe proporcional fazem parte de um importante grupo de algoritmos adaptativos em que o passo de adaptação de cada coeficiente do filtro é ajustado por uma função de proporcionalidade. O PNLMS exibe maior velocidade de convergência do que o algoritmo NLMS para aplicações em que as respostas ao impulso da planta são esparsas [26–28], [37], [38]. Contudo, para plantas com média ou baixa esparsidades, o algoritmo PNLMS sofre uma forte diminuição de sua velocidade de convergência [37], [39]; além do mais, sua rápida convergência inicial não é mantida durante todo o processo de adaptação [40], [41]. Assim, visando contornar tais desvantagens, diversas versões do algoritmo PNLMS vêm sendo propostas, tais como o IPNLMS [39], IIPNLMS [46], IAF-PNLMS [48], EIAF-PNLMS [52], SC-PNLMS [49], PDQMC [33]. Neste capítulo, inicialmente as expressões gerais de algoritmos do tipo proporcional são mostradas e, na sequência, alguns importantes algoritmos da classe proporcional são revisitados.

### 2.2 Expressões Gerais dos Algoritmos da Classe Proporcional

A formulação geral dos algoritmos tipo proporcional pode ser descrita pelas seguintes expressões [46]:

Equação de atualização dos coeficientes ( $N \times 1$ )

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}(n)e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n) + \xi} \quad (2.1)$$

Sinal de erro

$$e(n) = d(n) - y(n) + v(n) \quad (2.2)$$

Saída do filtro adaptativo

$$y(n) = \mathbf{w}^T(n) \mathbf{x}(n) = \mathbf{x}^T(n) \mathbf{w}(n) \quad (2.3)$$

Matriz de distribuição de ganho

$$\mathbf{G}(n) = \text{diag}[g_1(n) \ g_2(n) \ \cdots \ g_N(n)] \quad (2.4)$$

Ganho individual

$$g_i(n) = \frac{\gamma_i(n)}{\Phi(n)}, \quad i = 1, 2, \dots, N \quad (2.5)$$

Fator de normalização

$$\Phi(n) = \sum_{j=1}^N \gamma_j(n) \quad (2.6)$$

onde  $\xi > 0$  é um parâmetro de regularização que previne divisão por zero em (2.1) e estabiliza a solução, e  $v(n)$  representa um ruído de medição independente e identicamente distribuído (i.i.d.) com média zero, variância  $\sigma_v^2$  e não correlacionado com qualquer outro sinal no sistema. A matriz diagonal  $\mathbf{G}(n)$  distribui os ganhos dentre os coeficientes de acordo com uma dada regra considerando um específico tipo de algoritmo proporcional. Note que definindo  $\mathbf{G}(n)$  igual à matriz identidade, os algoritmos tipo proporcional se comportam como o algoritmo NLMS.

### 2.3 Algoritmo PNLMS

A regra específica para computar a função de proporcionalidade do  $i$ -ésimo coeficiente do filtro adaptativo no algoritmo PNLMS é dada por [48]

$$\gamma_i^P(n) = \max[\gamma_{\min}(n), |w_i(n)|] \quad (2.7)$$

com fator de ativação

$$\gamma_{\min}^P(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_{\infty}] \quad (2.8)$$

onde o sobrescrito P representa o algoritmo PNLMS e  $\|\cdot\|_\infty$  denota a norma infinita<sup>4</sup>. As variáveis  $\rho$  e  $\delta$  são parâmetros de proporcionalidade e inicialização, respectivamente. O parâmetro de proporcionalidade  $\rho$ , através do fator de ativação  $\gamma_{\min}^P(n)$ , evita que um coeficiente “congele” quando sua magnitude é nula ou muito menor do que a magnitude do maior coeficiente do filtro adaptativo. O parâmetro de inicialização  $\delta$  permite o início do processo de adaptação quando  $n=0$ , isto é, quando todos os coeficientes do filtro são inicializados com zero [37], [39]. Na Tabela 2.1, um resumo do algoritmo PNLMS é apresentado.

**Tabela 2.1 - ALGORITMO PNLMS**

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_{\min}^P(n) = \rho \max[\delta, |w_1(n)|, |w_2(n)|, \dots, |w_N(n)|]$$

$$\gamma_i^P(n) = \max[\gamma_{\min}^P(n), |w_i(n)|], \quad i = 1, 2, \dots, N$$

$$g_i^P(n) = \frac{\gamma_i^P(n)}{\phi_i^P(n)}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^P(n) = \text{diag}[g_1^P(n) \ g_2^P(n) \ \dots \ g_N^P(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^P(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^P(n) \mathbf{x}(n) + \xi}$$


---

O fator de ativação  $\gamma_{\min}^P(n)$ , originalmente considerado um fator secundário para o algoritmo [39], afeta de forma significativa o

---

<sup>4</sup>  $\|\vec{x}\|_\infty = \max_{i=1}^n (|x_i|)$

comportamento do PNLMS como mostrado em recentes trabalhos de pesquisa [43], [48] e [61]. Nesse contexto, tem sido verificado que tal fator e seus parâmetros associados podem ser utilizados para prover maior robustez ao algoritmo com vistas à variação de esparsidade da planta [43].

## 2.4 Algoritmo PNLMS++

O algoritmo PNLMS++, proposto em [37], foi concebido visando melhorar a velocidade de convergência do algoritmo PNLMS para os casos em que a resposta ao impulso da planta é dispersiva. Para tal, o PNLMS++ atualiza os coeficientes usando tanto a filosofia proporcional (PNLMS) quanto a filosofia do ganho uniforme (NLMS), isto é, não proporcional. Duas formas distintas de mistura de ganho proporcional e não proporcional foram discutidas e analisadas em [37]; na primeira, a atualização dos coeficientes no algoritmo é obtida alternando os algoritmos PNLMS e NLMS a cada período de amostragem e, na segunda, combinando os dois algoritmos no mesmo período de amostragem. A implementação do algoritmo PNLMS++ através da atualização alternada, denominada PNLMS++ (AU), pode ser realizada de diversas maneiras, como, por exemplo, utilizando o algoritmo NLMS e, então, a cada  $n$  iterações usar o PNLMS, ou vice-versa [essa alternância é feita comumente a cada iteração]. A outra forma, denominada PNLMS++ (DU), mistura os dois tipos de ganho a cada iteração usando a seguinte formulação para a atualização do vetor de coeficientes do filtro adaptativo:

$$w_i(n+1) = w_i(n) + \beta e(n) \mathbf{x}(n) \left( \frac{g_i^P(n)}{\mathbf{x}^T(n) \mathbf{G}^P(n) \mathbf{x}(n) + \xi} + \frac{1-\beta}{\mathbf{x}^T(n) \mathbf{x}(n) + \xi} \right), \quad i = 1, 2, \dots, N. \quad (2.9)$$

Como resultado, ambas as formas de implementação do algoritmo PNLMS++ superam o desempenho exibido pelo algoritmo PNLMS quando a planta possui média ou baixa esparsidade; além disso, a complexidade computacional do algoritmo PNLMS++, em relação à complexidade do algoritmo PNLMS, possui um acréscimo de  $N$  adições para o tipo (DU) e a mesma complexidade para o tipo (AU).

## 2.5 Algoritmo IPNLMS

Considerando o conceito de mistura do ganho proporcional e não proporcional, visando reduzir a degradação de desempenho do algoritmo PNLMS para casos cuja resposta ao impulso da planta não apresenta alta esparsidade, Benesty e Gay propuseram em [39] o algoritmo IPNLMS. O princípio básico do algoritmo IPNLMS é semelhante ao empregado pelo algoritmo PNLMS++ [37], cuja atualização dos coeficientes é efetuada alternando ou misturando o algoritmo NLMS e o PNLMS; porém, no caso do IPNLMS, a mistura de ganho proporcional e não proporcional é ponderada e sempre realizada a cada iteração.

A formulação do algoritmo IPNLMS a partir do PNLMS é realizada substituindo em (2.8) a norma infinita do vetor de coeficientes do filtro adaptativo por  $\|\mathbf{w}(n)\|_1$  e usando  $\rho = 1/N$ . Assim, considerando que o parâmetro  $\delta$  é utilizado apenas na primeira iteração para inicializar o algoritmo quando  $\mathbf{w}(0) = 0$ , o fator de ativação é expresso por

$$\begin{aligned} \gamma_{\min}^{\text{IP}}(n) &= \frac{1}{N} \max[\delta, \|\mathbf{w}(n)\|_1] \\ &= \frac{\|\mathbf{w}(n)\|_1}{N}, \quad n \geq 1, \end{aligned} \quad (2.10)$$

onde o sobrescrito IP é o acrônimo que referencia o algoritmo IPNLMS.

Note que o fator de ativação do algoritmo IPNLMS considera a magnitude de todos os coeficientes do filtro em contraste com o PNLMS, que considera a magnitude do maior coeficiente, resultando na média aritmética das magnitudes dos elementos do vetor de coeficientes. Dessa forma, a função de proporcionalidade do algoritmo IPNLMS é dada por

$$\gamma_i^{\text{IP}}(n) = \max\left[\frac{\|\mathbf{w}(n)\|_1}{N}, |w_i(n)|\right], \quad i = 1, 2, \dots, N. \quad (2.11)$$

Agora, considerando a média ponderada entre o fator de ativação e a magnitude do coeficiente  $w_i(n)$ , a função de proporcionalidade dada em (2.11) é modificada para

$$\gamma_i^{\text{IP}}(n) = (1 - \alpha) \frac{\|\mathbf{w}(n)\|_1}{N} + (1 + \alpha) |w_i(n)|, \quad -1 \leq \alpha \leq 1 \quad (2.12)$$

onde  $\alpha$  é um parâmetro de ponderação que controla o peso de cada uma das parcelas do lado direito de (2.12). Pode-se observar que o primeiro termo do lado direito de (2.12), comum a todos os coeficientes do filtro adaptativo, corresponde a uma parcela não proporcional; já o segundo

termo, corresponde à parcela proporcional. Assim, a partir de (2.5), o ganho individual do algoritmo IPNLMS é expresso como

$$g_i^{\text{IP}}(n) = \frac{\gamma_i^{\text{IP}}(n)}{\Phi^{\text{IP}}(n)}, \quad i = 1, 2, \dots, N \quad (2.13)$$

com

$$\Phi^{\text{IP}}(n) = \sum_{j=1}^N \gamma_j^{\text{IP}}(n). \quad (2.14)$$

Substituindo (2.12) em (2.14), obtém-se

$$\begin{aligned} \Phi^{\text{IP}}(n) &= \sum_{j=1}^N \left[ (1-\alpha) \frac{\|w(n)\|_1}{N} + (1+\alpha) |w_j(n)| \right] \\ &= \left[ \frac{(1-\alpha)}{N} \sum_{j=1}^N \|w(n)\|_1 \right] + \left[ (1+\alpha) \sum_{j=1}^N |w_j(n)| \right] \\ &= (1-\alpha) \|w(n)\|_1 + (1+\alpha) \|w(n)\|_1 \\ &= 2 \|w(n)\|_1. \end{aligned} \quad (2.15)$$

Por fim, aplicando (2.12) e (2.15) em (2.13), o ganho individual do algoritmo IPNLMS é escrito como

$$g_i^{\text{IP}}(n) = \frac{(1-\alpha)}{2N} + \frac{(1+\alpha) |w_i(n)|}{2 \|w(n)\|_1}, \quad i = 1, 2, \dots, N. \quad (2.16)$$

Para evitar divisão por zero, principalmente quando  $\mathbf{w}(0) = 0$ , a expressão final do ganho é obtida modificando (2.16) para

$$g_i^{\text{IP}}(n) = \frac{(1-\alpha)}{2N} + \frac{(1+\alpha) |w_i(n)|}{2 \|w(n)\|_1 + \varphi}, \quad i = 1, 2, \dots, N. \quad (2.17)$$

onde  $\varphi$  é uma pequena constante positiva [39].

A Tabela 2.2 apresenta um resumo do algoritmo IPNLMS. Pode-se notar que, para  $\alpha = -1$ , a matriz de ganho  $\mathbf{G}^{\text{IP}}(n)$  resulta em  $\frac{1}{N} \mathbf{I}$  e com isso o algoritmo IPNLMS torna-se idêntico ao NLMS. Por outro lado, quando  $\alpha = 1$ , o algoritmo IPNLMS apresenta um comportamento semelhante ao algoritmo PNLMS. Tipicamente, valores de  $\alpha$  compreendidos entre 0 e  $-0,5$  resultam em filtros com velocidades de convergência superiores às dos algoritmos PNLMS e NLMS para o caso

de planta esparsa. Embora o algoritmo IPNLMS possa ser utilizado sobre uma gama maior de esparsidade de planta ao combinar ganho proporcional e não proporcional, o peso dessa mistura é efetuado usando um parâmetro fixo, o que pode ocasionar perda de rendimento em virtude de variações do grau de esparsidade.

**Tabela 2.2 - ALGORITMO IPNLMS**

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$g_i^{\text{IP}}(n) = \frac{(1 - \alpha)}{2N} + (1 + \alpha) \frac{|w_i(n)|}{2 \|\mathbf{w}(n)\|_1 + \phi}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^{\text{IP}}(n) = \text{diag}[g_1^{\text{IP}}(n) \ g_2^{\text{IP}}(n) \ \dots \ g_N^{\text{IP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{IP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{IP}}(n) \mathbf{x}(n) + \xi}$$


---

## 2.6 Algoritmo IIPNLMS

O algoritmo IPNLMS, ao usar o mesmo parâmetro  $\alpha$  para obter os ganhos individuais  $g_i(n)$ , estabelece igual distribuição de pesos entre a parcela proporcional e não proporcional a todos os coeficientes do filtro adaptativo, independentemente de seu valor. Isso proporciona um ganho maior aos coeficientes de maior magnitude e ganhos menores aos de menor magnitude o que, em geral, resulta em uma mais rápida convergência inicial. Contudo, após esse período, em virtude dos baixos valores de ganho atribuídos aos coeficientes inativos<sup>5</sup>, a velocidade de

---

<sup>5</sup> Coeficientes cuja magnitude é zero ou muito pequena.

convergência é reduzida. Com o intuito de modificar a característica de ganho entre coeficientes de grandes e pequenas magnitudes, o algoritmo IPNLMS melhorado (IIPNLMS), proposto em [46], define dois parâmetros de ponderação para distribuição de pesos. Tal distribuição visa atribuir maior peso à parcela proporcional aos coeficientes de pequena magnitude e aos de grande magnitude, maior peso à parcela não proporcional. Assim, para a definição do valor do parâmetro de ponderação a ser utilizado na determinação do ganho associado ao  $i$ -ésimo coeficiente do filtro adaptativo, o algoritmo IIPNLMS utiliza o seguinte procedimento:

$$\alpha_i(n) = \begin{cases} \alpha_1, & g_i(n) > \zeta \max[\mathbf{a}(n)] \\ \alpha_2, & g_i(n) \leq \zeta \max[\mathbf{a}(n)] \end{cases} \quad (2.18)$$

com

$$\mathbf{a}(n) = [a_1(n) \ a_2(n) \ \cdots \ a_N(n)] \quad (2.19)$$

definido por

$$\alpha_i(n) = \max[\rho \max(|\mathbf{w}(n)|), |w_i(n)|], \quad i = 1, 2, \dots, N \quad (2.20)$$

onde  $\zeta$  (tipicamente igual a 0,1) é utilizado para estabelecer o limiar que define se o coeficiente é inativo ou não,  $\alpha_1 = -0,5$  e  $\alpha_2 = 0,5$  são valores típicos dos pesos e  $\rho$  é o mesmo parâmetro de proporcionalidade usado no algoritmo PNLMS. A Tabela 2.3 resume o algoritmo IIPNLMS sendo o IIP o acrônimo que o referencia.

Apesar do aumento de complexidade computacional em relação ao algoritmo IPNLMS, a melhoria no desempenho do IIPNLMS é pequena; além disso, ele não resolve o problema de variação de esparsidade da planta durante o processo adaptativo, pois o IIPNLMS também utiliza parâmetros fixos para ajustar o seu comportamento de acordo com as características da resposta ao impulso do sistema.

## 2.7 Algoritmo MPNLMS

Para contornar o problema de redução da velocidade de convergência do algoritmo PNLMS após o período inicial, o algoritmo PNLMS com lei  $\mu$  (MPNLMS) é proposto em [40] e [41]. Tal algoritmo foi concebido visando maximizar a taxa de convergência global do filtro adaptativo para sinal de entrada branco. Derivado do algoritmo *Steepest Descent* (SD) proporcional [41], o algoritmo MPNLMS não leva em

**Tabela 2.3 - ALGORITMO IIPNLMS**

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$a_i(n) = \max[\rho \max(|\mathbf{w}(n)|), |w_i(n)|], \quad i = 1, 2, \dots, N$$

$$\mathbf{a}(n) = [a_1(n) \ a_2(n) \ \dots \ a_N(n)]$$

$$\alpha_i(n) = \begin{cases} \alpha_1, & a_i(n) > \zeta \max[\mathbf{a}(n)] \\ \alpha_2, & a_i(n) \leq \zeta \max[\mathbf{a}(n)] \end{cases}$$

$$g_i^{\text{IIP}}(n) = \frac{[1 - \alpha_i(n)]}{2N} + [1 + \alpha_i(n)] \frac{|w_i(n)|}{2 \|\mathbf{w}(n)\|_1 + \varphi}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^{\text{IIP}}(n) = \text{diag}[g_1^{\text{IIP}}(n) \ g_2^{\text{IIP}}(n) \ \dots \ g_N^{\text{IIP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{IIP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{IIP}}(n) \mathbf{x}(n) + \xi}$$


---

consideração diretamente a magnitude do coeficiente para a determinação do ganho individual, mas sim a função dada por

$$F[|w_i(n)|] = \ln[1 + \mu |w_i(n)|], \quad i = 1, 2, \dots, N \quad (2.21)$$

onde  $\mu = 1/\varepsilon$  ( $\varepsilon > 0$ ) é o parâmetro que define se o coeficiente do filtro adaptativo convergiu, isto é, se o valor do coeficiente atingiu a vizinhança  $\varepsilon$  de seu valor ótimo. O algoritmo MPNLMS é descrito na Tabela 2.4.

A escolha do valor de  $\mu$  deve ser realizada considerando o nível de ruído de medição do sistema; tipicamente, em aplicações de cancelamento de eco de linha, é recomendado utilizar  $\mu = 1000$ , uma vez que níveis de eco abaixo de  $-60$  dB podem ser desconsiderados.

Por fim, a aplicação de (2.21) no algoritmo MPNLMS objetiva minimizar o número de iterações necessárias para que todos os

coeficientes atinjam a vizinhança  $\varepsilon$  do seu valor ótimo; como resultado, o MPNLMS apresenta uma velocidade de convergência uniforme durante todo o processo de adaptação. Entretanto, o algoritmo MPNLMS apresenta uma alta complexidade computacional (em virtude da utilização de uma função logarítmica para determinar o valor do passo de adaptação dos coeficientes), assim como pode apresentar uma degradação de desempenho para sinais de entrada altamente correlacionados [48].

**Tabela 2.4 - ALGORITMO MPNLMS**

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$F[|w_i(n)|] = \ln[1 + \mu |w_i(n)|], \quad i = 1, 2, \dots, N$$

$$\gamma_{\min}^{\text{MP}}(n) = \rho \max[\delta, F(|w_1(n)|), \dots, F(|w_N(n)|)]$$

$$\gamma_i^{\text{MP}}(n) = \max[\gamma_{\min}^{\text{MP}}(n), F(|w_i(n)|)], \quad i = 1, 2, \dots, N$$

$$g_i^{\text{MP}}(n) = \frac{\gamma_i^{\text{MP}}(n)}{\phi_i^{\text{MP}}(n)}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^{\text{MP}}(n) = \text{diag}[g_1^{\text{MP}}(n) \ g_2^{\text{MP}}(n) \ \dots \ g_N^{\text{MP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{MP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{MP}}(n) \mathbf{x}(n) + \xi}$$


---

## 2.8 Algoritmo SC-PNLMS

Visando prover maior robustez aos algoritmos adaptativos proporcionais com respeito a variações de esparsidade da resposta ao impulso da planta, o algoritmo SC-PNLMS (*sparseness controlled PNLMS*) é introduzido em [43] e [49]. Tais variações de esparsidade

podem ser oriundas, por exemplo, de mudanças de temperatura, pressão, movimentação da fonte acústica ou mesmo variações do ambiente. Assim, o algoritmo SC-PNLMS propõe o uso da medida de esparsidade definida em (1.3) para determinar o valor do parâmetro de proporcionalidade utilizado no algoritmo PNLMS. Entretanto, como geralmente a resposta ao impulso da planta  $\mathbf{w}^o$  não é conhecida *a priori*, o algoritmo SC-PNLMS considera uma estimativa de  $S(\mathbf{w}^o)$ ; para tal, em (1.3),  $\mathbf{w}^o$  é substituído pelo valor instantâneo dos coeficientes do filtro adaptativo  $\mathbf{w}(n)$ , resultando em

$$\hat{S}[\mathbf{w}(n)] = \frac{N}{N - \sqrt{N}} \left[ 1 - \frac{\|\mathbf{w}(n)\|_1}{\sqrt{N} \|\mathbf{w}(n)\|_2} \right] \quad (2.22)$$

onde  $\hat{S}[\mathbf{w}(n)]$  denota a estimativa instantânea de  $S[\mathbf{w}(n)]$ .

Agora, com a ajuda de uma função exponencial baseada na estimativa instantânea do grau de esparsidade da planta, o algoritmo SC-PNLMS define o valor da constante de proporcionalidade  $\rho$  (a cada iteração) como

$$\rho^{\text{SCP}}(n) = e^{-\lambda \hat{S}[\mathbf{w}(n)]} \quad (2.23)$$

onde SCP representa o algoritmo SC-PNLMS e o expoente  $\lambda$ , utilizado para o cálculo de  $\rho$ , possui uma faixa recomendada de valores que varia entre 4 e 8 [49]. O algoritmo SC-PNLMS é descrito na Tabela 2.5.

Note que, quanto mais esparsa a resposta ao impulso da planta, menor o valor do parâmetro de proporcionalidade [veja (2.23)]. Além disso, a determinação do parâmetro  $\rho^{\text{SCP}}(n)$  com base no grau de esparsidade da planta é inicializado somente após um período de aprendizagem do algoritmo adaptativo, isto é, para  $n \geq N$ .

## 2.9 Algoritmo IAF-PNLMS

Conforme descrito em [44], o algoritmo PNLMS tem seu desempenho impactado tanto pelo parâmetro de proporcionalidade quanto pelo fator de ativação  $\gamma_{\min}^P(n)$ . A função do fator de ativação no algoritmo PNLMS, durante o processo de adaptação, é prevenir que os coeficientes do filtro “congelem” quando suas magnitudes são nulas ou muito pequenas. Em contraste com o algoritmo PNLMS que utiliza um

**Tabela 2.5 - ALGORITMO SC-PNLMS**


---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_{\min}^{\text{SCP}}(n) = \rho^{\text{SCP}}(n) \max[\delta, |w_1(n)|, |w_2(n)|, \dots, |w_N(n)|]$$

$$\gamma_i^{\text{SCP}}(n) = \max[\gamma_{\min}^{\text{SCP}}(n), |w_i(n)|], \quad i = 1, 2, \dots, N$$

$$g_i^{\text{SCP}}(n) = \frac{\gamma_i^{\text{SCP}}(n)}{\phi_i^{\text{SCP}}(n)}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^{\text{SCP}}(n) = \text{diag}[g_1^{\text{SCP}}(n) \ g_2^{\text{SCP}}(n) \ \dots \ g_N^{\text{SCP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{SCP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{SCP}}(n) \mathbf{x}(n) + \xi}$$

$$\hat{S}[\mathbf{w}(n)] = \frac{N}{N - \sqrt{N}} \left[ 1 - \frac{\|\mathbf{w}(n)\|_1}{\sqrt{N} \|\mathbf{w}(n)\|_2} \right], \quad n \geq N$$

$$\rho^{\text{SCP}}(n) = \begin{cases} \rho^{\text{SCP}}(0), & n < N \\ e^{-\lambda \hat{S}[\mathbf{w}(n)]}, & n \geq N \end{cases}$$


---

fator de ativação  $\gamma_{\min}^{\text{P}}(n)$  comum a todos os coeficientes independentemente da sua magnitude, o algoritmo IAF-PNLMS (*individual activation factors* – PNLMS), proposto em [48], utiliza fatores de ativação individuais  $f_i(n)$ . Tal estratégia visa obter ganho proporcional à magnitude para todos os coeficientes do filtro adaptativo (ativos e inativos) e, com isso, uma maior velocidade de convergência para plantas com alta esparsidade. A idéia básica do algoritmo IAF-PNLMS é aumentar o ganho atribuído a coeficientes de maior magnitude e reduzir o ganho daqueles de menor magnitude.

Conseqüentemente, maiores velocidades de convergência inicial são atingidas para plantas com alta esparsidade. Assim, o algoritmo IAF-PNLMS propõe uma alteração na função de proporcionalidade do algoritmo PNLMS conforme mostrado a seguir:

$$\gamma_i^{\text{IAF}}(n) = \max[f_i(n), |w_i(n)|], \quad i = 1, 2, \dots, N \quad (2.24)$$

com

$$f_i(n) = \begin{cases} \frac{1}{2} |w_i(n)| + \frac{1}{2} \gamma_i^{\text{IAF}}(n-1), & n = mN, \quad m = 1, 2, 3, \dots \\ f_i(n-1), & \text{caso contrário} \end{cases} \quad (2.25)$$

onde o acrônimo IAF caracteriza o algoritmo IAF-PNLMS.

Assumindo convergência, quando  $n \rightarrow \infty$ , pode-se observar a partir de (2.24) e (2.25) que  $\gamma_i^{\text{IAF}}(n) \rightarrow |w_i(n)|$  tanto para coeficientes ativos quanto inativos [48]. Em contraste com o algoritmo PNLMS cuja proporcionalidade do ganho é observada apenas nos coeficientes ativos, o ganho individual do algoritmo IAF-PNLMS tende a ser proporcional a  $|w_i(n)|$  para todos os coeficientes. Como consequência, o algoritmo IAF-PNLMS atinge maior velocidade de convergência do que a observada pelo PNLMS quando a resposta ao impulso da planta exhibe alta esparsidade; contudo, em virtude do aumento do ganho atribuído aos coeficientes de maior magnitude (acelerando a convergência inicial com respeito ao PNLMS), um aumento do erro em excesso em regime permanente no algoritmo IAF-PNLMS é constatado. A Tabela 2.6 apresenta uma síntese do algoritmo IAF-PNLMS, sendo que o fator de ativação individual  $f_i(n)$  deve ser inicializado com um pequeno valor positivo.

Uma importante modificação no algoritmo IAF-PNLMS, visando melhorar a velocidade de convergência global e reduzir o erro em excesso, é apresentada em [52]. Tal modificação, apresentada no algoritmo EIAF-PNLMS (*enhanced individual-activation-factor* PNLMS), aumenta o ganho dos coeficientes inativos (à medida que o regime permanente se aproxima) através da mudança da política de distribuição de ganho proporcional para ganho uniforme, isto é, nessa fase o mesmo ganho é atribuído a todos os coeficientes do filtro adaptativo. Baseado em [62], [63], o algoritmo EIAF-PNLMS usa uma versão suavizada do erro quadrático instantâneo (como indicador de proximidade do regime permanente) e, comparando-a com um limiar

**Tabela 2.6 - ALGORITMO IAF-PNLMS**


---

 Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

 Para  $n = 1, 2, 3, \dots$ 

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

$$\gamma_i^{\text{IAF}}(n) = \max[f_i(n), |w_i(n)|], \quad i = 1, 2, \dots, N$$

$$g_i^{\text{IAF}}(n) = \frac{\gamma_i^{\text{IAF}}(n)}{\phi_i^{\text{IAF}}(n)}, \quad i = 1, 2, \dots, N$$

$$\mathbf{G}^{\text{IAF}}(n) = \text{diag}[g_1^{\text{IAF}}(n) \ g_2^{\text{IAF}}(n) \ \dots \ g_N^{\text{IAF}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{IAF}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{IAF}}(n) \mathbf{x}(n) + \xi}$$


---

pré-definido, estabelece o instante da mudança do ganho proporcional para o ganho não proporcional (uniforme). A estratégia utilizada pelo algoritmo EIAF-PNLMS para a determinação do instante de mudança de política de distribuição do ganho é dada por

$$\psi_e(n) \leq K \quad (2.26)$$

com

$$\psi_e(n) = \tau \psi_e(n-1) + (1-\tau) e^2(n) \quad (2.27)$$

onde  $\psi_e(n)$  denota o erro quadrático instantâneo suavizado e  $\tau$  ( $0 < \tau < 1$ ) é o fator de suavização.

Assim, tomando como base (2.26), o ganho individual dos coeficientes do filtro adaptativo do algoritmo EIAF-PNLMS é obtido por

$$g_i^{\text{EIAF}}(n) = \begin{cases} \frac{1}{N}, & \psi_e(n) \leq K \\ \frac{\gamma_i^{\text{IAF}}(n)}{\phi^{\text{IAF}}(n)}, & \text{caso contrário} \end{cases} \quad (2.28)$$

onde o sobrescrito EIAF referencia o algoritmo EIAF-PNLMS.

Como resultado, o algoritmo EIAF-PNLMS melhora as características de convergência e supera o desempenho do IAF-PNLMS para respostas ao impulso com alto e médio grau de esparsidade [52]. Por outro lado, a escolha do valor apropriado do parâmetro de limiar  $K$ , utilizado para estimar a proximidade do regime permanente, não é uma tarefa trivial, visto que tal valor depende da variância do ruído de medição presente no sistema, grau de esparsidade da planta, comprimento da resposta ao impulso, dentre outros. Na prática, o valor do limiar deve ser sempre maior do que a variância do ruído de medição, isto é,  $K > \sigma_v^2$ , visto que a mudança para ganho uniforme deve ocorrer antes que o algoritmo atinja seu regime permanente; tal imposição obriga o conhecimento ou estimativa da potência do ruído de medição  $\sigma_v^2$  a fim de garantir uma adequada operação do algoritmo EIAF-PNLMS, o que na prática pode representar um inconveniente.

## 2.10 Algoritmo proporcional ao desvio quadrático médio dos coeficientes (PDQMC)

Em [33], dois novos algoritmos da classe proporcional, denominados *water-filling* e proporcional ao desvio quadrático médio dos coeficientes (PDQMC), foram propostos. O algoritmo *water-filling* é derivado da minimização do erro quadrático médio a cada iteração para sinais de entrada brancos considerando duas restrições, que são  $g_i(n) \geq 0 \quad \forall i, n$  e  $\sum_{j=1}^N g_j(n) = N \quad \forall n$ . Como resultado, o ganho individual do algoritmo *water-filling* para sinal de entrada branco é dado por [64]

$$g_i^{\text{WF}}(n) = \frac{\frac{2\beta\sigma_x^2 E[z_i^2(n)]}{\sigma_x^2 N + \xi} - \lambda}{2\sigma_x^4 \left( \frac{\beta}{\sigma_x^2 N + \xi} \right)^2 \left( 2E[z_i^2(n)] + \sum_{j=1}^N E[z_j^2(n)] + \frac{\sigma_v^2}{\sigma_x^2} \right)} \quad (2.29)$$

onde WF denota o algoritmo *water-filling*,  $\lambda$  denota o multiplicador de Lagrange e  $E[z_i^2(n)]$  representa o desvio quadrático médio dos coeficientes definido como

$$E[z_i^2(n)] = E\{[w_i^o - w_i(n)]^2\} \quad (2.30)$$

com  $E[\cdot]$  denotando o valor esperado.

O multiplicador de Lagrange  $\lambda$  é computado como segue:

- i) Classifique em ordem crescente de valor os elementos do vetor  $\mathbf{c}(n) = [c_1(n) \ c_2(n) \ \dots \ c_N(n)]$  dado por

$$c_i(n) = \frac{2\beta\sigma_x^2 E[z_i^2(n)]}{\sigma_x^2 N + \xi}, \quad i = 1, 2, \dots, N \quad (2.31)$$

para formar um novo vetor, tal que  $c_{(1)}(n) < c_{(2)}(n) < \dots < c_{(N)}(n)$ .

- ii) Rearranje o vetor  $\mathbf{q}(n) = [q_1(n) \ q_2(n) \ \dots \ q_N(n)]$  dado por

$$q_i(n) = 2\sigma_x^4 \left( \frac{\beta}{\sigma_x^2 N + \xi} \right)^2 \left( 2E\{z_i^2(n)\} + \sum_{j=1}^N E\{z_j^2(n)\} + \frac{\sigma_v^2}{\sigma_x^2} \right) \quad (2.32)$$

seguindo o mesmo ordenamento do vetor  $\mathbf{c}(n)$ , mantendo assim a correspondência dos índices originais, isto é,  $[q_{(1)}(n) \ q_{(2)}(n) \ \dots \ q_{(N)}(n)]$ .

- iii) Calcule as soluções candidatas para o valor do multiplicador de Lagrange  $\lambda$  através de

$$\lambda_i = \frac{\sum_{j=i}^N \frac{c_{(j)}}{q_{(j)}} - N}{\sum_{j=i}^N \frac{1}{q_{(j)}}}. \quad (2.33)$$

- iv) Escolha  $\lambda = \lambda_i$  se  $c_{(i-1)} < \lambda_i < c_{(i)}$ , onde  $c_{(0)} = -\infty$ .

Visto que a determinação de  $E[z_i^2(n)]$  por (2.30) requer o conhecimento da resposta ao impulso da planta  $w_i^o$ , um procedimento

para estimar o valor esperado do desvio quadrático médio é apresentado em [33]. Tal procedimento de estimação é realizado como segue:

$$E[z_i^2(n)]_{\text{est}} = \frac{E[p_i(n)]_{\text{est}}}{\sigma_x^2} \quad (2.34)$$

com

$$E[p_i(n)]_{\text{est}} = \alpha E[p_i(n-1)]_{\text{est}} + (1-\alpha)p_i(n) \quad (2.35)$$

sendo

$$p_i(n) \triangleq x_i(n)e(n) \quad (2.36)$$

e ( $0 < \alpha < 1$ ) um parâmetro de suavização. O operador  $E[\cdot]_{\text{est}}$  denota uma estimativa do valor esperado  $E[\cdot]$ . Agora, substituindo o valor esperado do desvio quadrático médio dos coeficientes pelo quadrado da estimativa do valor esperado do desvio dos coeficientes, isto é,

$$E[z_i^2(n)] \cong \{E[z_i(n)]\}_{\text{est}}^2 \quad \text{e} \quad \sum_{j=1}^N E[z_j^2(n)] \cong \sum_{j=1}^N \{E[z_j(n)]\}_{\text{est}}^2, \quad \text{pode-se}$$

computar (2.29).

Em virtude da alta carga computacional do algoritmo *water-filling*, visando reduzir sua complexidade, o algoritmo PDQMC é derivado considerando  $\beta=1$ , o que implica  $\lambda=0$ . Dessa forma, o ganho do algoritmo PDQMC é computado por [33]

$$g_i^{\text{PDQMC}} = \frac{E[z_i^2(n)]_{\text{est}}}{\frac{1}{N} \sum_{j=1}^N E[z_j^2(n)]_{\text{est}}}. \quad (2.37)$$

O algoritmo PDQMC pode ser implementado conforme apresentado na Tabela 2.7, onde  $\sigma_x^2$  e  $\sigma_v^2$  representam, respectivamente, o valor da variância do sinal de entrada e do ruído de medição e  $\mathbf{1}$  é um vetor unitário  $N$ -dimensional. Visto que o algoritmo PDQMC apresenta desempenho superior ao seu predecessor, os estudos aqui realizados estão direcionados somente ao algoritmo PDQMC.

Como se pode verificar na Tabela 2.7, o algoritmo PDQMC usa uma política de distribuição de ganho  $\mathbf{g}_c(n)$  que combina o ganho proporcional ao desvio quadrático médio dos coeficientes com o ganho uniforme, utilizando uma regra denominada combinação de ganho convexa adaptativa [33], [64], de forma que a distribuição de ganho se

**Tabela 2.7** - ALGORITMO PDQMC

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$p_i(n) = x_i(n) e(n)$$

$$E[p_i(n)]_{\text{est}} = \alpha E[p_i(n-1)]_{\text{est}} + (1 - \alpha) p_i(n)$$

$$E[z_i(n)]_{\text{est}} = \frac{E[p_i(n)]_{\text{est}}}{\sigma_x^2}$$

$$g_i^{\text{PDQMC}}(n) = \frac{\{E[z_i(n)]_{\text{est}}\}^2}{\frac{1}{N} \sum_{j=1}^N \{E[z_j(n)]_{\text{est}}\}^2}, \quad i = 1, 2, \dots, N$$

$$\mathbf{g}^{\text{PDQMC}}(n) = [g_1^{\text{PDQMC}}(n) \ g_2^{\text{PDQMC}}(n) \ \dots \ g_N^{\text{PDQMC}}(n)]$$

$$\zeta(n) = \min \left[ 1, \frac{\omega \sigma_v^2}{\sigma_x^2 \sum_{j=1}^N \{E[z_j(n)]_{\text{est}}\}^2 + \sigma_v^2} \right]$$

$$\mathbf{g}_c(n) = [1 - \zeta(n)] \mathbf{g}^{\text{PDQMC}}(n) + \zeta(n) \mathbf{1}$$

$$\mathbf{G}^{\text{PDQMC}}(n) = \text{diag}[g_{c_1}(n) \ g_{c_2}(n) \ \dots \ g_{c_N}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{PDQMC}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{PDQMC}}(n) \mathbf{x}(n) + \xi}$$


---

torna uniforme em regime permanente. No entanto, essa mistura é controlada por um parâmetro  $\zeta(n)$  que depende do conhecimento do valor da variância do ruído de medição  $\sigma_v^2$ , o qual geralmente não está prontamente disponível; com isso, uma estimativa da variância do ruído de medição é necessária para uma correta operação do algoritmo.

## 2.11 Conclusões

Neste capítulo, alguns algoritmos da classe proporcional foram revisitados. Aproveitando o conceito de proporcionalidade introduzido por Duttweiler no algoritmo PNLMS [38], diversas versões de algoritmos tipo proporcional têm sido propostas. Tais algoritmos apresentam estratégias de distribuição de ganho distintas com o intuito de melhorar seu desempenho, adequá-los a uma aplicação com características específicas ou, simplesmente, tentar sanar alguma deficiência intrínseca ao PNLMS. Contudo, apesar dos avanços alcançados por essas novas versões, ainda há muitos pontos dos algoritmos adaptivos proporcionais que podem ser aprimorados, tais como velocidade de convergência, erro em excesso, complexidade computacional, robustez, dentre outros.



## ANÁLISE DA DISTRIBUIÇÃO DE GANHO E COMPORTAMENTO DOS COEFICIENTES DO ALGORITMO PNLMS

### 3.1 Introdução

Uma das grandes contribuições do algoritmo PNLMS [38] para o avanço da filtragem adaptativa em ambientes com planta esparsa foi atribuir ganhos de adaptação individuais aos coeficientes, em contraste aos algoritmos LMS e NLMS que atribuem ganhos iguais. A distribuição desses ganhos está diretamente associada à magnitude dos coeficientes e aos parâmetros de proporcionalidade  $\rho$  e inicialização  $\delta$ . Assim, neste capítulo, uma análise da política de distribuição de ganho e seus efeitos sobre os coeficientes são apresentados e discutidos. Além disso, uma metodologia para a determinação da taxa de variação de magnitude do coeficiente visando estimar a convergência de cada coeficiente do filtro adaptativo é proposta.

### 3.2 Ganhos de Adaptação do Algoritmo PNLMS

Conforme mostrado no capítulo anterior, a relação que determina o ganho individual do  $i$ -ésimo coeficiente do algoritmo PNLMS pode ser expressa como [27], [39]

$$g_i^P(n) = \frac{\gamma_i^P(n)}{\Phi_i^P(n)}, \quad i = 1, 2, \dots, N \quad (3.1)$$

com

$$\gamma_i^P(n) = \max[\gamma_{\min}^P(n), |w_i(n)|], \quad i = 1, 2, \dots, N \quad (3.2)$$

e

$$\gamma_{\min}^P(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_{\infty}] \quad (3.3)$$

sendo o denominador de (3.1) definido em (2.6).

Analisando (3.1), verifica-se que o ganho do  $i$ -ésimo coeficiente é diretamente proporcional a  $\gamma_i^P(n)$  e essa função, por sua vez, é igual seja ao fator de ativação  $\gamma_{\min}^P(n)$  ou à magnitude do coeficiente  $|w_i(n)|$ , conforme (3.2) e (3.3). Assim, a análise do ganho pode ser separada em duas partes como mostrada a seguir [48]:

a) Quando  $|w_i(n)| < \gamma_{\min}^P(n)$ .

Visto que  $\gamma_{\min}^P(n)$  possui, em geral, um valor pequeno, é razoável assumir que o  $i$ -ésimo coeficiente  $w_i(n)$  seja inativo, isto é, sua magnitude é muito menor do que a magnitude do maior coeficiente do vetor  $\mathbf{w}(n)$ . Com isso, a partir de (3.1)-(3.3), o seu ganho associado pode ser reescrito como

$$g_i^P(n) = \frac{1}{\Phi^P(n)} \gamma_{\min}^P(n). \quad (3.4)$$

Uma vez que o fator de ativação  $\gamma_{\min}^P(n)$  é comum a todos os coeficientes do filtro, um ganho mínimo e comum é atribuído a todos os coeficientes inativos; assim, pode-se definir o ganho dos coeficientes inativos como

$$g_i^{\text{inativo}}(n) \triangleq g_i^P(n) = \frac{1}{\Phi^P(n)} \gamma_{\min}^P(n), \quad |w_i(n)| < \gamma_{\min}^P(n) \quad (3.5)$$

onde  $\gamma_{\min}^P(n)$ , descrito em (3.3), é função da norma infinita da magnitude dos coeficientes do filtro adaptativo e dos parâmetros de proporcionalidade  $\rho$  e inicialização  $\delta$ .

b) Quando  $|w_i(n)| \geq \gamma_{\min}^P(n)$ .

Assumindo que os coeficientes que não se enquadram como inativos são considerados ativos, então, um coeficiente será ativo se  $|w_i(n)| \geq \gamma_{\min}^P(n)$ , isto é, quando sua magnitude for maior ou igual ao fator de ativação  $\gamma_{\min}^P(n)$ . O seu ganho associado, obtido a partir (3.1), é dado por

$$g_i^{\text{ativo}}(n) \triangleq g_i^P(n) = \frac{1}{\Phi^P(n)} |w_i(n)|, \quad |w_i(n)| \geq \gamma_{\min}^P(n). \quad (3.6)$$

A equação (3.6) representa a filosofia básica do algoritmo PNLMS, ou seja, cada ganho  $g_i^{\text{ativo}}(n)$  é proporcional à magnitude do correspondente coeficiente ativo  $|w_i(n)|$ .

Além disso, o ganho total distribuído dentre os coeficientes do filtro adaptativo a cada iteração pode ser determinado por

$$\begin{aligned} g_{\text{total}}^{\text{P}}(n) &= \text{tr}[\mathbf{G}^{\text{P}}(n)] = g_1^{\text{P}}(n) + g_2^{\text{P}}(n) + \cdots + g_N^{\text{P}}(n) \\ &= \frac{\gamma_1^{\text{P}}(n) + \gamma_2^{\text{P}}(n) + \cdots + \gamma_N^{\text{P}}(n)}{\Phi^{\text{P}}(n)} \\ &= \frac{\gamma_1^{\text{P}}(n) + \gamma_2^{\text{P}}(n) + \cdots + \gamma_N^{\text{P}}(n)}{\sum_{j=1}^N \gamma_j^{\text{P}}(n)} \\ &= 1. \end{aligned} \tag{3.7}$$

Como resultado da normalização dos ganhos individuais por  $\Phi^{\text{P}}(n)$ , o ganho total distribuído a cada iteração dentre os coeficientes é constante ao longo do tempo. Note que o fator de normalização  $\Phi(n)$ , dado em (2.6), pode ser escalado por  $1/N$  (veja [27] e [42]), implicando que o ganho total neste caso seja igual a  $N$ ; aqui é adotada a mesma formulação de [39], o que resulta em (3.7). No entanto, salienta-se que ambas formulações são equivalentes.

A partir da subdivisão dos ganhos em dois grupos distintos, ativos e inativos, a equação de ganho total distribuído dentre os coeficientes do filtro adaptativo a cada iteração pode ser redefinida como

$$\begin{aligned} g_{\text{total}}^{\text{P}}(n) &= \sum_{i \in A} g_i^{\text{P}}(n) + \sum_{i \in B} g_i^{\text{P}}(n) = \sum_{i \in A} g_i^{\text{P}}(n) + \sum_{i \in B} \frac{1}{\Phi^{\text{P}}(n)} \gamma_{\text{min}}^{\text{P}}(n) \\ &= \sum_{i \in A} g_i^{\text{P}}(n) + \frac{N - N_{\text{ativos}}}{\Phi^{\text{P}}(n)} \gamma_{\text{min}}^{\text{P}}(n) \\ &= 1 \end{aligned} \tag{3.8}$$

onde  $N_{\text{ativos}}$  é o número de coeficientes ativos e  $A$  e  $B$  são os conjuntos de índices associados às posições dos coeficientes ativos e inativos, respectivamente.

Analisando (3.8), nota-se que o primeiro termo do lado direito representa o ganho total distribuído dentre os coeficientes ativos e o segundo, o ganho total distribuído dentre os inativos. Observe que

qualquer alteração no fator de ativação  $\gamma_{\min}^P(n)$  modifica a distribuição de ganho atribuída aos coeficientes ativos, bem como aos inativos. Logo, verifica-se que  $\gamma_{\min}^P(n)$  tem um papel preponderante no comportamento do algoritmo adaptativo.

### 3.3 Impacto da Distribuição de ganho e Convergência dos Coeficientes

Nesta seção, o comportamento do algoritmo PNLMS em relação à sua distribuição de ganho é estudado. Para isso, considerando um problema de identificação de sistemas, simulações de Monte Carlo (MC) são realizadas (média de 100 realizações independentes) comparando-o com o algoritmo NLMS. O cenário para todas as simulações deste capítulo consiste de uma resposta ao impulso esparsa com comprimento  $N = 128$ , mostrada na Figura 3.1, obtida a partir do Modelo 1 de caminho de eco (para testes de canceladores de eco de fala) da norma ITU-T Recomendações G.168 [65], denominada aqui  $\mathbf{w}_A^o$ . O grau de esparsidade dessa resposta ao impulso, calculado a partir de (1.3), é igual a 0,78. O sinal de entrada é branco com variância  $\sigma_x^2$  unitária. O ruído de medição  $v(n)$  também é branco com média zero e variância  $\sigma_v^2 = 10^{-3}$  (SNR = 30 dB)<sup>6</sup>. O desempenho do algoritmo PNLMS é avaliado utilizando o desalinhamento normalizado (em dB) dado por [39]

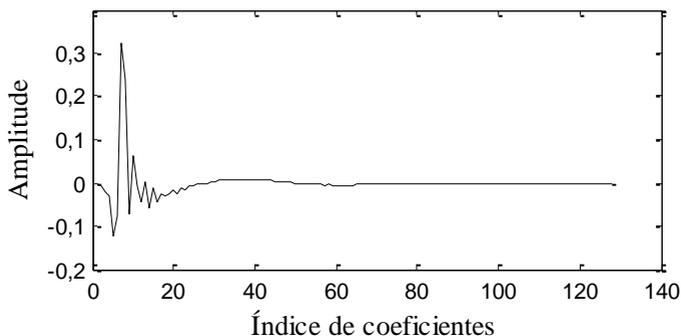
$$\kappa(n) = 10 \log_{10} \frac{\|\mathbf{w}^o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}^o\|_2^2}. \quad (3.9)$$

Através das curvas mostradas na Figura 3.2, pode-se avaliar o comportamento dos algoritmos PNLMS e NLMS. Ambos algoritmos possuem passo de adaptação  $\beta = 0,1$  e constante de regularização  $\xi = 10^{-4}$ . Além disso, para o PNLMS, são usados os parâmetros de proporcionalidade  $\rho = 0,01$  e inicialização  $\delta = 0,01$ . O comportamento típico do algoritmo PNLMS considerando plantas esparsas pode ser observado da Figura 3.2, isto é, velocidade de convergência inicial elevada e uma forte redução nessa velocidade após o período inicial. Note

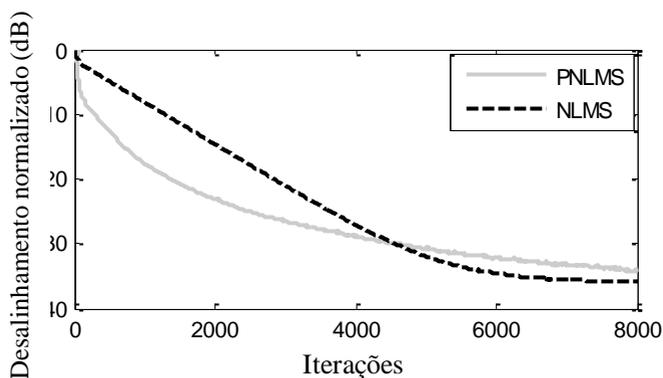
---

<sup>6</sup> SNR =  $10 \log(\sigma_x^2 / \sigma_v^2)$

que, em virtude dessa redução de velocidade de convergência, o PNLMS atinge o regime permanente após o algoritmo NLMS ter atingido.



**Figura 3.1** - Resposta ao impulso  $\mathbf{w}_A^0$  obtida a partir do Modelo 1 de caminho de eco da norma ITU-T Recomendação G.168 com comprimento  $N=128$  e esparsidade igual a 0,78.



**Figura 3.2** - Curvas de desalinhamento normalizado do algoritmo PNLMS e NLMS para sinal de entrada branco com variância unitária usando  $\beta=0,1$ ,  $\xi=10^{-4}$ ,  $\rho=0,01$  e  $\delta=0,01$ .

Para melhor compreensão das características do algoritmo PNLMS, uma avaliação do comportamento do ganho  $g_i^P(n)$  dado por (3.1) e a evolução dos valores dos coeficientes do filtro adaptativo é aqui descrita. Para isso, três diferentes faixas de magnitude dos coeficientes da resposta ao impulso da planta  $\mathbf{w}^0$  são avaliados, isto é, alta, média e baixa

magnitude. Tais coeficientes são definidos tomando como base os seguintes critérios:

- i) Coeficiente de maior magnitude  $w_{\max}$ : Caracteriza o coeficiente do filtro adaptativo correspondente ao coeficiente de maior magnitude da resposta ao impulso da planta  $\mathbf{w}^o$ , obtido por

$$w_{\max} = \|\mathbf{w}^o\|_{\infty} = \max_i (|w_i^o|). \quad (3.10)$$

- ii) Coeficiente com magnitude intermediária  $w_{\text{int}}$ : Representa a coeficiente do filtro adaptativo correspondente a um coeficiente da resposta ao impulso da planta  $\mathbf{w}^o$  cuja magnitude é a maior dentre os coeficientes com magnitude igual ou menor a  $0,5|w_{\max}|$ , isto é, metade da maior magnitude, obtido por

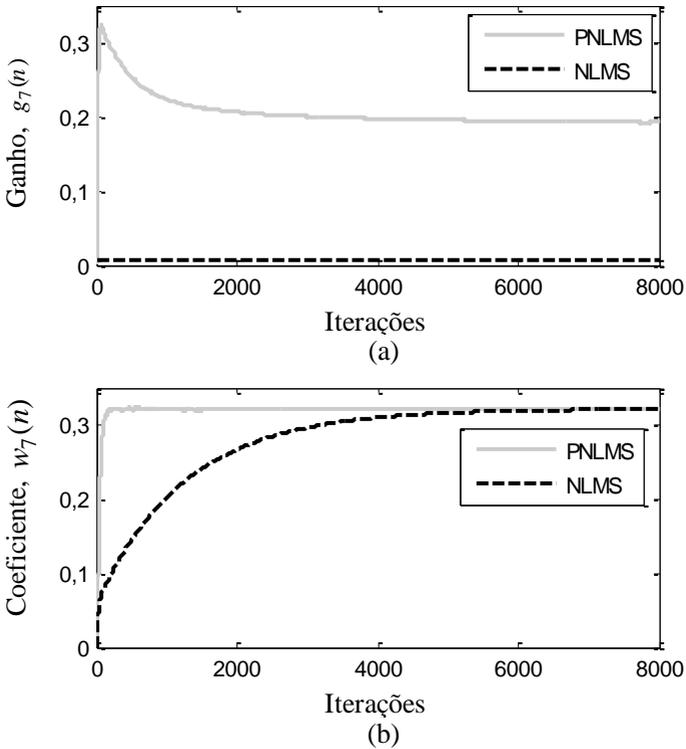
$$w_{\text{int}} = \max_i |w_i^o|, \quad |w_i^o| \leq 0,5|w_{\max}|. \quad (3.11)$$

- iii) Coeficientes inativos: Diz respeito aos coeficientes do filtro adaptativo correspondentes a coeficientes da resposta ao impulso da planta  $\mathbf{w}^o$  cuja magnitude é menor do que  $\gamma_{\min}^P(n)$ .

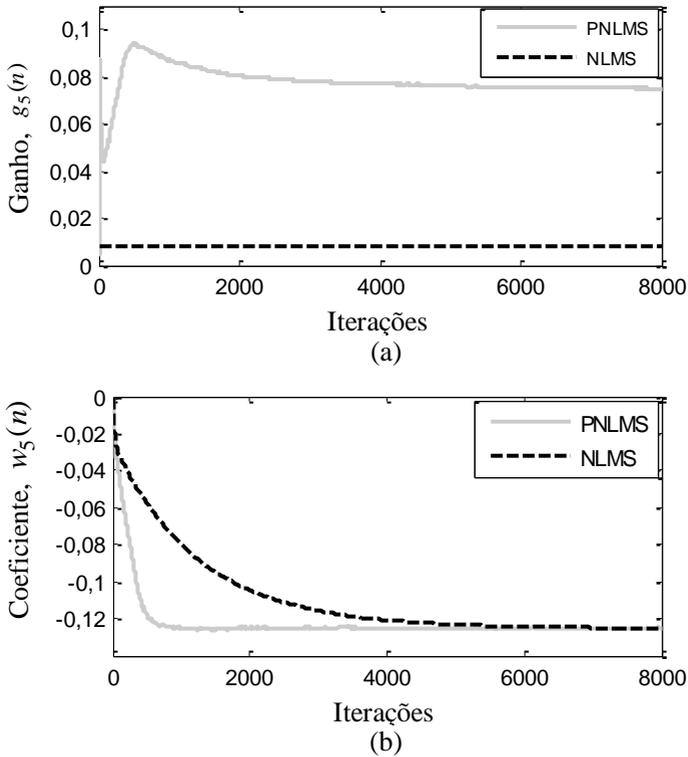
No caso de plantas esparsas, em geral, muitos coeficientes são classificados como inativos. Com isso, para uma melhor avaliação do comportamento desses coeficientes, aqui são avaliados três coeficientes inativos com magnitudes distintas, isto é, os coeficientes do filtro adaptativo correspondentes à  $w_{13}^o$ ,  $w_{28}^o$  e  $w_{52}^o$ . Dessa forma, a tendência e não o comportamento de um coeficiente isoladamente, pode ser melhor observada. Assim, os ganhos associados aos coeficientes  $w_{13}^o$ ,  $w_{28}^o$  e  $w_{52}^o$  e suas respectivas evoluções de valores são mostrados nas Figuras 3.3 a 3.5.

As Figuras 3.3(a) e 3.4(a) mostram de forma clara diferenças entre os algoritmos NLMS e PNLMS, isto é, atribuição de ganhos maiores a coeficientes com maiores magnitudes (mais significativos) decorrente da proporcionalidade entre a magnitude do coeficiente do filtro adaptativo e o ganho a ele associado. No caso do coeficiente mais significativo (veja Figura 3.3), esse ganho chega a ser aproximadamente 42 vezes maior do que o ganho alocado ao algoritmo NLMS (no pico) e 24 vezes maior em regime permanente. O mesmo padrão de comportamento pode ser observado para o coeficiente de magnitude intermediária, o qual atinge

aproximadamente 12 vezes o ganho do NLMS no pico e 10 vezes em regime permanente (veja Figura 3.4). O efeito do aumento dos ganhos individuais atribuídos aos coeficientes de maior magnitude  $w_{\max}$  e de magnitude intermediária  $w_{\text{int}}$  pode ser observado nas Figuras 3.3(b) e 3.4(b), onde são mostradas as evoluções dos seus valores durante o processo de adaptação. Note que o coeficiente de maior magnitude [veja Figura 3.3(b)] do algoritmo PNLMS converge para seu valor ótimo muito mais rápido do que o do algoritmo NLMS. Além disso, no PNLMS o coeficiente mais significativo converge mais rápido do que o coeficiente com magnitude intermediária [veja Figura 3.4(b)], sendo que esse também converge mais rápido do que o do NLMS. É por esse motivo que o algoritmo PNLMS atinge velocidade de convergência inicial



**Figura 3.3** - Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\max} = w_7(n)$ ]. (a) Ganho atribuído. (b) Evolução do coeficiente.

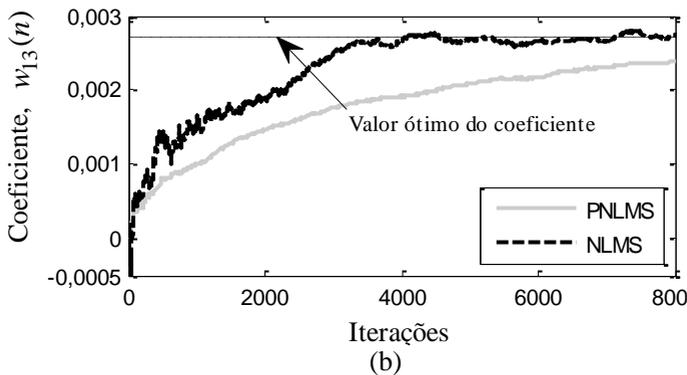
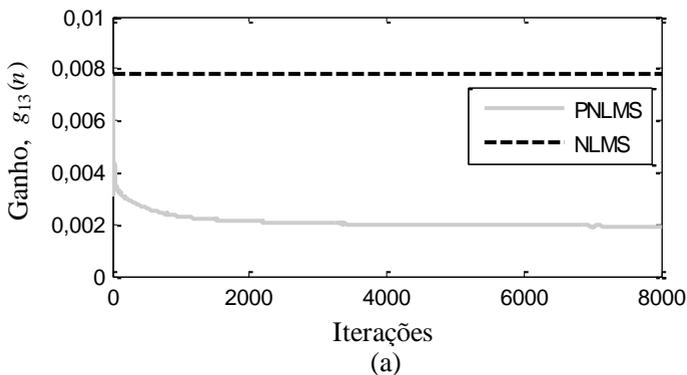


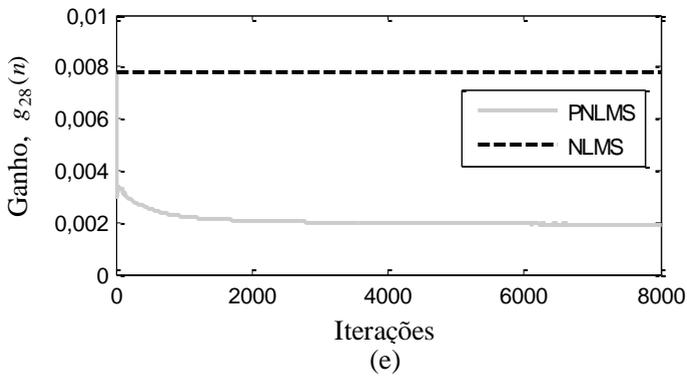
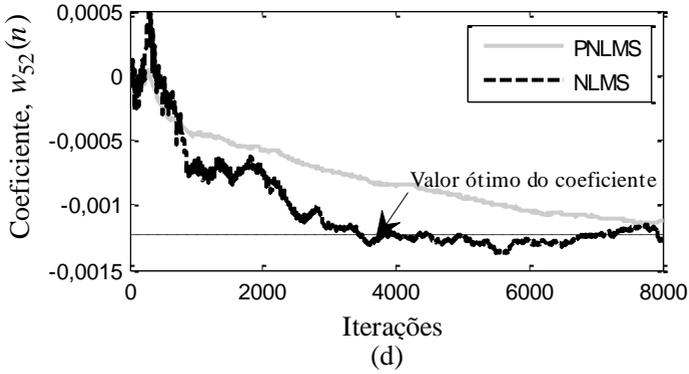
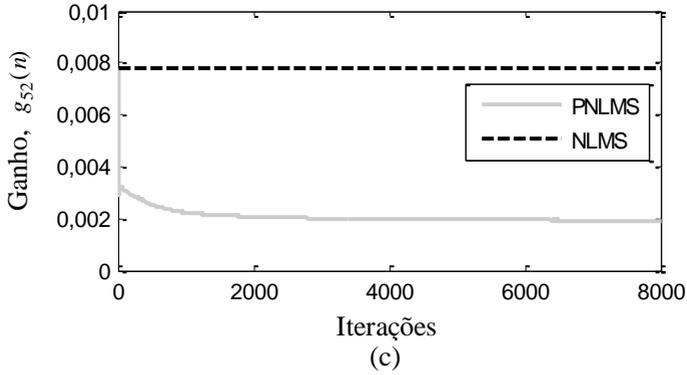
**Figura 3.4** - Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\text{int}} = w_5(n)$ ]. (a) Ganho atribuído. (b) Evolução do coeficiente.

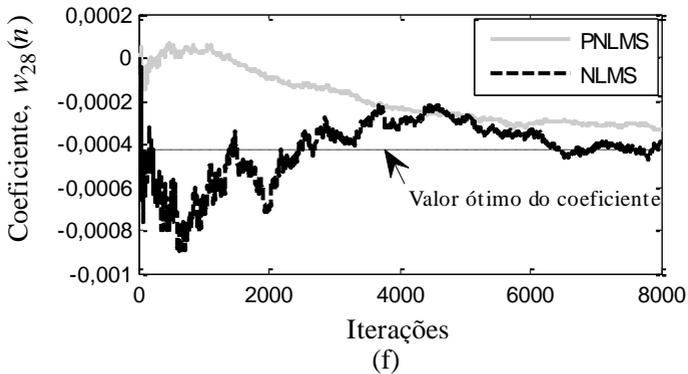
superior ao do NLMS, isto é, os coeficientes mais significativos da planta convergem mais rapidamente, pois recebem ganhos maiores.

Contudo, a mesma razão que estabelece a rápida convergência inicial é a que traz um dos grandes problemas de convergência para o algoritmo PNLMS, isto é, a forte redução da velocidade de convergência após o período inicial. Como mostrado em (3.7), o ganho total distribuído dentre os coeficientes do filtro é constante e igual a 1, assim para atribuir maiores ganhos aos coeficientes mais significativos, o algoritmo PNLMS “retira” ganho dos coeficientes inativos. Isso pode ser visto através das Figuras 3.5(a), 3.5(c) e 3.5(e), em que os ganhos alocados aos coeficientes inativos no algoritmo PNLMS são bem menores do que os atribuídos pelo algoritmo NLMS. O efeito desse ganho pequeno conferido aos

coeficientes inativos é observado nas Figuras 3.5(b), 3.5(d) e 3.5(f), onde constata-se a convergência mais lenta desses coeficientes. Como para uma planta esparsa há muitos coeficientes inativos, após a convergência dos ativos, o algoritmo PNLMS apresenta uma forte desaceleração em sua velocidade de convergência e, com isso, tende ao regime permanente mais lentamente. Em alguns casos, como mostrado na Figura 3.2, tal redução na taxa de convergência do algoritmo PNLMS faz com que este último seja ultrapassado pelo algoritmo NLMS durante o processo adaptativo.







**Figura 3.5** - Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^0$ . (a), (c) e (e) Ganho atribuído aos coeficientes inativos. (b), (d) e (f) Evolução do coeficiente.

Outro ponto importante a ser discutido analisando o comportamento dos ganhos individuais do algoritmo PNLMS, é que o valor do ganho alocado aos coeficientes mais significativos se mantém elevado durante todo o processo adaptativo, sem considerar o fato do coeficiente já ter alcançado a vizinhança de seu valor ótimo. Em outras palavras, do total de ganho disponível para atualização dos coeficientes, a maior parte é usada para a convergência dos coeficientes mais significativos e esse ganho continua sendo alocado mesmo após o coeficiente ter alcançado o seu objetivo, isto é, estar nas proximidades de seu valor ótimo. Por outro lado, a “falta” de ganho disponível aos coeficientes de menor magnitude, durante todo o processo adaptativo, faz com que esses coeficientes convirjam muito lentamente, impactando negativamente a convergência global do algoritmo PNLMS.

### 3.4 Taxa de Variação da Magnitude do Coeficiente do Filtro Adaptativo

Uma forma de verificar a evolução da convergência dos coeficientes do filtro adaptativo é através da análise de suas características de convergência, como, por exemplo, a derivada de sua magnitude, visto que, conforme o valor esperado do coeficiente do filtro adaptativo se aproxima de seu valor ótimo em regime, essa derivada tende para zero. Entretanto, além de ser relativamente custoso computacionalmente, em virtude da oscilação típica presente nos coeficientes adaptativos, a

avaliação da derivada da magnitude do coeficiente não refletiria sua real situação com respeito à convergência. Assim, buscando prover maior confiabilidade na estimativa de convergência do coeficiente do filtro adaptativo assim como robustez com respeito às flutuações de sua magnitude, uma metodologia para avaliar o processo de convergência do coeficiente, baseada na taxa de variação suavizada e normalizada (NSVR) de sua magnitude, é aqui apresentada e discutida.

### 3.4.1 Taxa de Variação Suavizada e Normalizada (NSVR) da Magnitude do Coeficiente

Aqui, para reduzir o efeito da flutuação típica do valor dos coeficientes inerente ao processo adaptativo, a estimativa da taxa de variação da magnitude do coeficiente é realizada a cada  $\Delta$  iterações. Desse modo, pode-se utilizar  $\Delta$  valores distintos da magnitude do coeficiente para melhor avaliação do seu comportamento no intervalo considerado. Assim,  $\Delta$  define o número de amostras de  $w_i(n)$  utilizadas para calcular a taxa de variação da magnitude do correspondente coeficiente. Na estratégia aqui apresentada, a variável  $\Delta$  é o único parâmetro dessa estimação.

A NSVR da magnitude dos coeficientes do filtro adaptativo, realizada a cada  $\Delta$  iterações, é computada usando as seguintes expressões:

$$V_i(n) = \begin{cases} \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, & n = m\Delta, \quad m = 1, 2, 3, \dots \\ V_i(n-1), & \text{caso contrário} \end{cases} \quad (3.12)$$

com

$$Q_i(n) = \sum_{j=0}^{(\Delta-2)/2} |w_i(n-j)|, \quad i = 1, 2, \dots, N \quad (3.13)$$

e

$$P_i(n) = \sum_{j=\Delta/2}^{\Delta-1} |w_i(n-j)|, \quad i = 1, 2, \dots, N \quad (3.14)$$

onde  $P_i(n)$  e  $Q_i(n)$  são versões suavizadas de  $|w_i(n)|$  computadas considerando, respectivamente, a primeira e segunda metade do

intervalo  $\Delta$ . A variável  $V_i(n)$  representa a estimativa da taxa de variação da magnitude de cada coeficiente  $w_i(n)$ , isto é, a NSVR de  $|w_i(n)|$ . O uso do módulo ( $|\cdot|$ ) e da operação de mínimo no cálculo da NSVR garantem que o valor de  $V_i(n)$  reflita apenas a taxa de variação da magnitude de cada coeficiente, independentemente do coeficiente ser positivo ou negativo ou, ainda, se  $|w_i(n)|$  está aumentando ou diminuindo o seu valor.

### 3.4.2 Complexidade Computacional e Memória Requerida

Visto que a complexidade computacional é uma característica importante em qualquer algoritmo de filtragem adaptativa, uma análise desse quesito se faz necessária. Uma vez que a especificação de um processador está diretamente relacionada ao número de operações efetuadas em um mesmo intervalo de tempo, as operações de soma em (3.13) e (3.14) podem ser diluídas se efetuadas a cada iteração, e não todas as operações em uma só vez a cada  $\Delta$  iterações. Desse modo, a complexidade computacional e memória requerida associadas à implementação de (3.12)–(3.14), mostrada na Tabela 3.1, é separada em duas partes, isto é, operações realizadas a cada iteração e, a cada  $\Delta$  iterações.

Como sintetizado na Tabela 3.1, a estimativa da taxa de variação da magnitude dos coeficientes não requer operações de multiplicação e existem apenas  $N$  operações de adição, ora para  $P_i(n)$  ora para  $Q_i(n)$ , a cada iteração. Além disso,  $N$  comparações, subtrações e sendo, sua complexidade computacional é dependente do valor de  $\Delta$  e somente poderá ser computada quando definido o valor do intervalo  $\Delta$ . Note que, quanto maior o valor de  $\Delta$ , menor a carga computacional, visto que menor será o número de operações matemáticas realizadas por iteração.

### 3.4.3 Análise da Métrica NSVR

A NSVR visa avaliar a tendência de variação da magnitude do coeficiente durante o processo adaptativo. Contudo, como a convergência do coeficiente não é, em geral, um processo monotônico, tal avaliação deve ser feita considerando o valor médio da magnitude do coeficiente. Assim, conforme mostrado no Apêndice I, o valor esperado da NSVR da magnitude do coeficiente, dado por

**Tabela 3.1**

CARGA COMPUTACIONAL E MEMÓRIA REQUERIDA PARA A DETERMINAÇÃO DA NSVR DA MAGNITUDE DOS COEFICIENTES DO FILTRO ADAPTATIVO

Intervalo	Multiplificações	Adições	Subtrações	Divisões	Comparações	Memória
A cada iteração	0	$N$	0	0	0	$3N$
A cada $\Delta$ iterações	0	0	$N$	$N$	$N$	0

$$E[V_i(n)] = \begin{cases} E\left[\frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}\right], & n = m\Delta, \quad m = 1, 2, 3, \dots \\ E[V_i(n-1)] \end{cases} \quad (3.15)$$

com

$$E\left[\frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}\right] \cong \frac{\left| \overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}+1} - \overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}} \right|}{\min\{E[\overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}+1}], E[\overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}}]\}} \quad (3.16)$$

onde  $\overline{|\cdot|}_a^b$  denota o valor médio da magnitude no período entre  $a$  e  $b$ ,

é proporcional à diferença da magnitude média dos coeficientes adaptativos nos intervalos considerados.

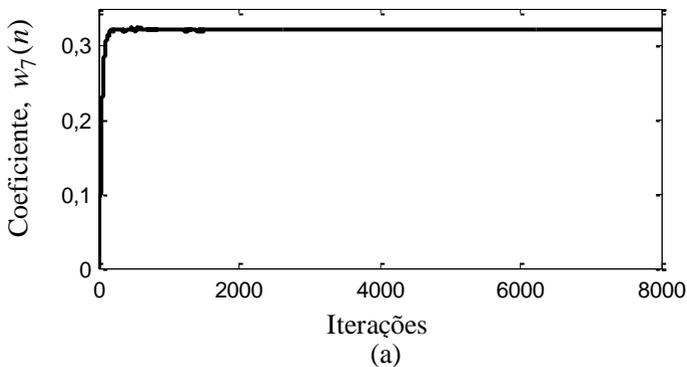
Note que a normalização na taxa de variação objetiva transformar a NSVR em uma métrica percentual, isto é, variação da magnitude média do coeficiente em relação ao seu valor médio. Dessa forma, a NSVR pode ser utilizada para avaliação da variação da magnitude do coeficiente independentemente do valor do coeficiente. Por exemplo, se a média das magnitudes avalida passou de 1 para 2 ou de 0,1 para 0,2, em ambas situações a NSVR será igual a 1.

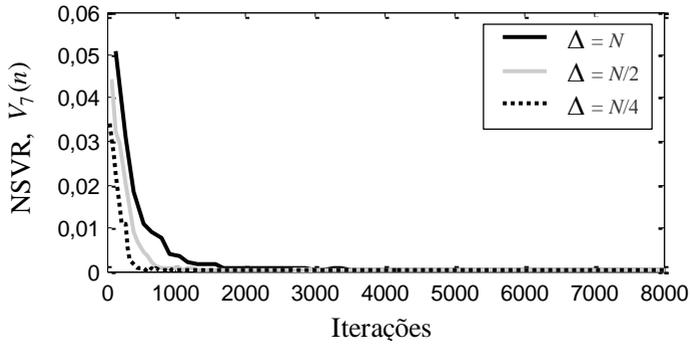
Com vistas ao impacto do valor de  $\Delta$  na determinação da NSVR, a partir de (3.16) pode-se verificar que, para o caso de  $\Delta = 2$ , a NSVR é

baseada na diferença da magnitude entre amostras adjacentes do coeficiente e, em função da característica oscilatória típica do processo de estimação, não sendo capaz de “captar” a tendência da variação da magnitude do coeficiente. Por outro lado, para o caso de  $\Delta = N$ , a NSVR compara o valor médio da magnitude dos coeficientes em um intervalo de  $N/2$  amostras e, dessa forma, suaviza a oscilação presente no valor do coeficiente avaliado.

### 3.4.4 Eficácia do Método de Estimação da Taxa de Variação da Magnitude dos Coeficientes

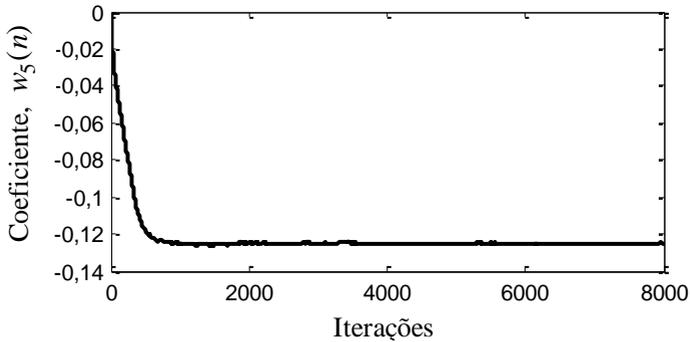
Para verificar a eficácia do método de estimação da taxa de variação da magnitude dos coeficientes do filtro adaptativo usando a NSVR de  $|w_i(n)|$ , simulações de Monte Carlo (MC) (média de 100 realizações independentes) do algoritmo PNLMS são realizadas considerando o mesmo cenário apresentado na Seção 3.3. Para tal, no cálculo da NSVR da magnitude dos coeficientes, três valores de  $\Delta$  distintos são utilizados, mais especificamente igual ao comprimento do filtro  $N$ ,  $N/2$  e  $N/4$ . Como resultado, a evolução dos coeficientes e suas correspondentes taxas de variação obtidas para os diferentes valores de  $\Delta$  considerando o coeficiente de maior magnitude  $w_{\max}$ , de magnitude intermediária  $w_{\text{int}}$  e o coeficiente inativo  $w_{13}$  são mostrados nas Figuras 3.6 a 3.8.



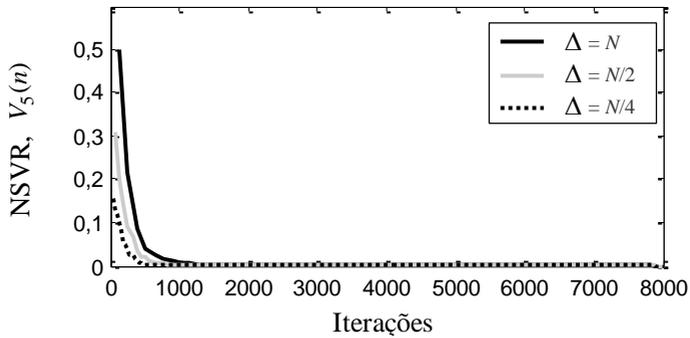


(b)

**Figura 3.6** - Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^0$  [ $w_{\max} = w_7(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ .

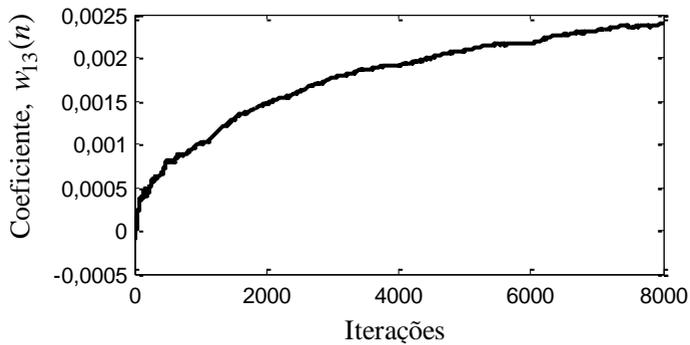


(a)

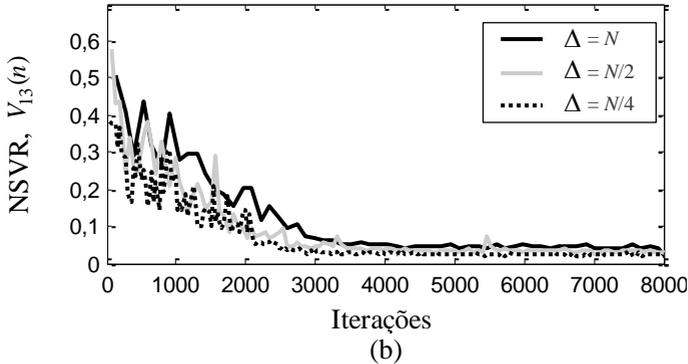


(b)

**Figura 3.7** - Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  [ $w_{\text{int}} = w_5(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ .



(a)



**Figura 3.8** - Comportamento do coeficiente inativo da resposta ao impulso  $w_A^o$  [ $w_i = w_{13}(n)$ ]. (a) Evolução do coeficiente. (b) NSVR da magnitude considerando valores distintos de  $\Delta$ .

Note que, independente do valor de  $\Delta$ , a NSVR da magnitude dos coeficientes ativos apresentam praticamente o mesmo valor em regime permanente, isto é, resultam na mesma taxa de variação da magnitude do coeficiente [veja Figuras 3.6(b) e 3.7(b)]. Em contraste, a NSVR da magnitude dos coeficientes inativos, apresenta valores diferentes da NSVR em regime permanente para os diferentes valores de  $\Delta$  [veja Figura 3.8(b)]. Isso se explica visto que os coeficientes inativos possuem uma maior variação em sua magnitude durante o processo adaptativo e, quanto maior o valor de  $\Delta$ , maior será o valor da taxa de variação. De qualquer forma, independente de  $\Delta$  e do valor do coeficiente, pode-se verificar que a NSVR de  $|w_i(n)|$  é capaz de fornecer informações sobre a convergência do coeficiente e, com isso, representar um importante componente para a definição da política de alocação de ganho durante o processo adaptativo.

### 3.5 Conclusões

Neste capítulo, uma análise da distribuição de ganho e comportamento dos coeficientes do algoritmo PNLMS foi realizada, assim como expressões do ganho total e de cada coeficiente, diferenciando-os entre ativos e inativos foram apresentadas. Através de simulações numéricas, considerando grupos de coeficientes com magnitudes distintas, as relações entre ganho do coeficiente e convergência foram

mostradas e discutidas. Também, uma estratégia para avaliação de convergência individual do coeficiente utilizando uma taxa de variação suavizada e normalizada (NSVR) da magnitude do coeficiente do filtro adaptativo foi proposta. Além disso, estudo da complexidade computacional e eficácia da NSVR da magnitude do coeficiente na estimação do processo de convergência do coeficiente foram apresentadas.

No próximo capítulo, utilizando a NSVR da magnitude dos coeficientes para redefinir a política de distribuição de ganho de algoritmos da classe proporcional, novas versões dos algoritmos PNLMS, IPNLMS e IAF-PNLMS são introduzidas.



## NOVA POLÍTICA DE DISTRIBUIÇÃO DE GANHO PARA ALGORITMOS TIPO PROPORCIONAL BASEADA NA CONVERGÊNCIA DO COEFICIENTE

### 4.1 Introdução

Como mencionado anteriormente, o algoritmo PNLMS [38] explora a característica esparsa da planta através da matriz de ganhos, a qual estabelece um passo de adaptação para cada coeficiente do filtro adaptativo através de uma função de proporcionalidade entre o ganho atribuído e sua magnitude. Tal estratégia resulta em uma melhoria na velocidade de convergência para plantas esparsas quando comparado aos algoritmos que utilizam uma distribuição uniforme de ganho. Apesar de a estratégia de distribuição de ganho utilizada pelo algoritmo PNLMS alcançar uma rápida convergência inicial para plantas esparsas, sua maior velocidade de convergência não se mantém por todo o processo de adaptação (veja Figura 3.2). Essa redução da velocidade de convergência deve-se ao fato que, após a convergência dos coeficientes ativos, os coeficientes inativos convergem lentamente, uma vez que o ganho alocado a coeficientes inativos pelo algoritmo PNLMS é muito pequeno (veja Figura 3.5) [39], [44]. Além disso, o PNLMS tem seu desempenho deteriorado à medida que a esparsidade da resposta ao impulso diminui [45]. Com isso, uma alternativa para reduzir essa desaceleração na taxa de convergência do algoritmo PNLMS é alterar sua política de distribuição de ganho durante a adaptação, visto que atribuir ganhos elevados a coeficientes que já atingiram a vizinhança de seus valores ótimos praticamente não afeta a convergência global do algoritmo.

Assim, neste capítulo, uma nova política de distribuição de ganho proporcional baseada na avaliação da convergência individual do coeficiente é apresentada. A política aqui proposta reduz o ganho atribuído aos coeficientes ativos que já atingiram a vizinhança de seus valores ótimos e o redistribui dentre os coeficientes que ainda não convergiram, dessa forma, melhorando as características de convergência do algoritmo. Para a avaliação de convergência do coeficiente, a NSVR (taxa de variação suavizada e normalizada) da magnitude do coeficiente, definida no capítulo anterior, é utilizada como métrica. Então,

considerando o uso da métrica NSVR, três novas versões dos algoritmos PNLMS, IPNLMS e IAF-PNLMS são apresentadas. Levando em conta distintos cenários de operação, resultados de simulações numéricas são mostrados atestando a eficácia da política de ganho proporcional proposta.

## 4.2 Política de Distribuição de Ganho

Nesta seção, uma estratégia visando realocar ganhos atribuídos a coeficientes ativos que tenham alcançado a vizinhança de seus valores ótimos, isto é, coeficientes que tenham quase atingido a convergência e não mais necessitam receber um ganho elevado é aqui apresentada. Para tal, a convergência do coeficiente é definida como

$$|w_i^o - w_i(L_i)| \leq \psi_o, \quad i = 1, 2, \dots, N \quad (4.1)$$

onde  $L_i$  é o número de iterações necessárias para o  $i$ -ésimo coeficiente do filtro adaptativo atingir a vizinhança  $\psi_o$  de seu valor ótimo. Agora, usando (4.1) para avaliar a convergência individual do coeficiente, o ganho de adaptação proposto para algoritmos do tipo proporcional é dado por

$$g_i(n) = \frac{\gamma_i(n)}{\sum_{j=1}^N \gamma_j(n)} \quad (4.2)$$

com

$$\gamma_i(n) = \begin{cases} \gamma_o(n), & |w_i^o - w_i(L_i)| \leq \psi_o \\ \gamma_i(n), & \text{caso contrário} \end{cases} \quad (4.3)$$

onde  $\gamma_o(n)$  é uma variável comum associada aos coeficientes que já tenham alcançado a vizinhança  $\psi_o$  de seus valores ótimos e  $\gamma_i(n)$  denota a função de proporcionalidade relativa ao algoritmo considerado. Contudo, o uso de (4.1) para avaliação da convergência individual do coeficiente se torna um desafio, visto que essa condição requer o conhecimento *a priori* da resposta ao impulso da planta  $\mathbf{w}^o$ , o que usualmente não está disponível em aplicações do mundo real. Assim, visando contornar tal problema, a NSVR da magnitude do coeficiente é utilizada como métrica de avaliação da convergência individual do coeficiente, visto que a computação da NSVR não necessita do

conhecimento da resposta ao impulso da planta. Além disso, considerando o fato da NSVR da magnitude do coeficiente diminuir conforme o coeficiente se aproxima de seu valor ótimo, pode-se estimar, através de um limiar pré-definido, o instante no qual o coeficiente atinge a proximidade de seu valor ótimo. A seguir, levando em conta a métrica NSVR definida por (3.12), a função de proporcionalidade do  $i$ -ésimo coeficiente, dada em (4.3), é revisada para

$$\gamma_i(n) = \begin{cases} \gamma_o(n), & V_i(n) \leq \varepsilon \\ \gamma_i(n), & \text{caso contrário} \end{cases} \quad (4.4)$$

com  $\varepsilon$  sendo o parâmetro de limiar. Tal parâmetro será utilizado para determinar o instante no qual o  $i$ -ésimo coeficiente atingiu a vizinhança de seu valor ótimo.

### 4.3 Algoritmo PNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente

O algoritmo PNLMS possui a mesma política de distribuição de ganho durante todo o processo adaptativo, isto é, se a magnitude do coeficiente for maior do que um dado  $\gamma_{\min}^P(n)$  seu ganho será sempre proporcional a  $|w_i(n)|$ , caso contrário será proporcional a  $\gamma_{\min}^P(n)$  [veja (3.1)], o que proporciona algumas características indesejáveis ao PNLMS. Para melhorar tais características, nesta seção, uma nova versão do algoritmo PNLMS baseada na taxa de variação da magnitude do coeficiente, aqui denominado algoritmo NSVR-PNLMS, é introduzida, assim como uma breve análise de sua política de distribuição de ganho é apresentada.

#### 4.3.1 Algoritmo NSVR-PNLMS

A nova política de ganho proporcional visa reduzir o ganho atribuído a coeficientes que já tenham quase atingido a convergência e, por consequência, aumentar o ganho atribuído aos demais coeficientes. Para tal, o ganho comum atribuído aos coeficientes que alcançaram a vizinhança de seus valores ótimos é definido como  $\gamma_o(n) = \gamma_{\min}^P(n)$ , uma vez que o fator de ativação  $\gamma_{\min}^P(n)$  é um valor pequeno e comum a todos os coeficientes inativos [veja (3.3)]. Como resultado, o algoritmo NSVR-PNLMS atribui um pequeno ganho de manutenção aos coeficientes que atingiram a vizinhança de seus valores ótimos,

prevenindo com isso o seu “congelamento” durante o processo de adaptação. Além disso, o uso de  $\gamma_o(n) = \gamma_{\min}^P(n)$  evita a inclusão de um novo parâmetro no algoritmo. Assim, para o algoritmo NSVR-PNLMS, o ganho individual é expresso por

$$g_i^{\text{NP}}(n) = \frac{\gamma_i^{\text{NP}}(n)}{\Phi^{\text{NP}}(n)} \quad (4.5)$$

com

$$\gamma_i^{\text{NP}}(n) = \begin{cases} \gamma_{\min}^P(n), & V_i(n) \leq \varepsilon \\ \gamma_i^P(n), & \text{caso contrário} \end{cases} \quad (4.6)$$

onde  $\Phi^{\text{NP}}(n)$  é o fator de normalização definido em (2.6),  $\gamma_i^P(n)$  é computado por (3.2) e o superescrito NP denota o algoritmo NSVR-PNLMS.

A Tabela 4.1 apresenta o algoritmo NSVR-PNLMS. Note que a nova abordagem mantém a idéia de proporcionalidade do PNLMS, isto é, quanto maior a magnitude do coeficiente maior o ganho atribuído a ele, exceto aos coeficientes cuja taxa de variação seja menor do que  $\varepsilon$ , sendo  $\varepsilon$  a constante que define o limiar de convergência. Note que o algoritmo descrito na Tabela 4.1 realiza todas as operações de cálculo das variáveis  $P_i(n)$  e  $Q_i(n)$  de uma única vez a cada  $\Delta$  iterações; contudo, ressalta-se aqui que esse cálculo pode ser efetuado a cada iteração, reduzindo assim o número máximo de operações realizadas por ciclo do algoritmo. Além disso, para evitar divisão por zero em (3.12), a determinação da NSVR é realizada após um período de aprendizagem de  $N$  iterações, isto é, igual ao comprimento do filtro adaptativo.

**Tabela 4.1** – ALGORITMO NSVR-PNLMS

---

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_{\min}^P(n) = \rho \max[\delta, |w_1(n)|, |w_2(n)|, \dots, |w_N(n)|]$$

$$\gamma_i^{\text{NP}}(n) = \max[\gamma_{\min}^P(n), |w_i(n)|], \quad 1 \leq i \leq N$$

$$Q_i(n) = |w_i(n)| + |w_i(n-1)| + \dots + |w_i(n - \frac{\Delta}{2} + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$P_i(n) = |w_i(n - \frac{\Delta}{2})| + |w_i(n - \frac{\Delta}{2} - 1)| + \dots + |w_i(n - \Delta + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$V_i(n) = \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, \quad \begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

Se  $V_i(n) \leq \varepsilon$

$$\gamma_i^{\text{NP}}(n) = \gamma_{\min}^P(n), \quad 1 \leq i \leq N \text{ e } n \geq N + \Delta$$

$$g_i^{\text{NP}}(n) = \frac{\gamma_i^{\text{NP}}(n)}{\sum_{j=1}^N \gamma_j^{\text{NP}}(n)}, \quad 1 \leq i \leq N$$

$$\mathbf{G}^{\text{NP}}(n) = \text{diag}[g_1^{\text{NP}}(n) \ g_2^{\text{NP}}(n) \ \dots \ g_N^{\text{NP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{NP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{NP}}(n) \mathbf{x}(n) + \xi}$$


---

### 4.3.2 Análise da Distribuição de Ganho do Algoritmo NSVR-PNLMS

A distribuição de ganho do algoritmo PNLMS pode ser dividida em duas parcelas conforme mostrado em (3.8), isto é, ganho atribuído aos coeficientes ativos e inativos. Aqui, uma nova classificação no processo de alocação de ganhos considerando a proximidade do valor ótimo é realizada, sendo que tanto coeficientes ativos quanto inativos podem assumir essa nova classificação durante o processo adaptativo. Assim, analisando o novo algoritmo, as seguintes definições são primeiramente apresentadas:

- i) **Ganho dos coeficientes que alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  tenha quase atingido a convergência e seu ganho associado é dado por

$$g_c^{\text{NP}}(n) = \frac{1}{\Phi^{\text{NP}}(n)} \gamma_{\min}^{\text{P}}(n). \quad (4.7)$$

- ii) **Ganho dos coeficientes que ainda não alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) > \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  não atingiu a convergência e seu ganho associado é dado por

$$g_{\text{nc},i}^{\text{NP}}(n) = \begin{cases} \frac{1}{\Phi^{\text{NP}}(n)} \gamma_{\min}^{\text{P}}(n), & |w_i(n)| < \gamma_{\min}^{\text{P}}(n) \\ \frac{1}{\Phi^{\text{NP}}(n)} |w_i(n)|, & |w_i(n)| \geq \gamma_{\min}^{\text{P}}(n). \end{cases} \quad (4.8)$$

Agora, levando em conta que o ganho para coeficientes que tenham alcançado a vizinhança de seus valores ótimos é o mesmo ganho atribuído a coeficientes inativos, o ganho total distribuído entre os coeficientes do filtro adaptativo a cada iteração no algoritmo NSVR-PNLMS pode ser escrito como

$$g_{\text{total}}^{\text{NP}}(n) = \frac{1}{\Phi^{\text{NP}}(n)} \left[ \sum_{i \in (A-C)} |w_i(n)| + \sum_{i \in (B \cup C)} \gamma_{\min}^{\text{P}}(n) \right] = 1 \quad (4.9)$$

onde  $C$  é o conjunto de índices associado às posições dos coeficientes ativos que convergiram,  $A$ , o conjunto de índices associado às posições

dos coeficientes ativos e,  $B$ , o conjunto de índices associado às posições dos coeficientes inativos.

Visto que o número de coeficientes que atingem a proximidade de seus valores ótimos, aqui denominado  $N_c(n)$ , se altera durante o processo adaptativo, os valores e a quantidade de elementos das duas parcelas à direita de (4.9) são variáveis e apresentam o seguinte comportamento:

- i) No início do processo adaptativo não há coeficientes que atingiram a convergência, portanto  $N_c(n) = 0$  e conseqüentemente  $C$  é um conjunto vazio. Nessa condição, (4.9) é igual a (3.8), isto é, a primeira parcela representa a soma dos ganhos dos coeficientes ativos e a segunda, a dos inativos. Assim, a distribuição de ganho do novo algoritmo é igual à do PNLMS; com isso, a rápida convergência inicial, uma característica importante do algoritmo PNLMS, é mantida no novo algoritmo.
- ii) Conforme os coeficientes se aproximam de seu valor ótimo, a NSVR diminui e, assim que  $V_j(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente alcançou a proximidade de seu valor ótimo. Com isso,  $N_c(n)$  aumenta o seu valor e  $C$  deixa de ser um conjunto vazio. Isso diminui o número de elementos associados à primeira parcela à direita de (4.9) elevando o número de coeficientes com o ganho  $\gamma_{\min}(n)$  e, portanto, reduzindo os ganhos atribuídos aos coeficientes ativos (primeira parcela à direita). Em uma situação ideal (em regime permanente) todos os coeficientes atingiriam a proximidade de seus valores ótimos e, assim,  $N_c(n) = N$  e  $(A - C) = \emptyset$ , de forma que todos os coeficientes receberiam o mesmo ganho e o novo algoritmo se comportaria como o algoritmo NLMS.
- iii) Sempre que o coeficiente ativo  $w_i(n)$  atinge a vizinhança do seu valor ótimo, o ganho transferido aos demais coeficientes é

$$T_{\text{ativo}, i}^{\text{NP}}(n) = \frac{|w_i(n)|}{\sum_{\substack{j=1 \\ j \neq i}}^N \gamma_j^{\text{NP}}(n) + |w_i(n)|} - \frac{\gamma_{\min}(n)}{\sum_{\substack{j=1 \\ j \neq i}}^N \gamma_j^{\text{NP}}(n) + \gamma_{\min}(n)} \quad (4.10)$$

onde o primeiro termo do lado direito de (4.10) representa o ganho proporcional à magnitude do coeficiente e o segundo termo é o ganho atribuído a  $w_i(n)$  dado pela nova política de distribuição. Note que, quanto maior a magnitude do coeficiente ativo, maior é o valor do ganho distribuído aos demais coeficientes.

- iv) Finalmente, quando um coeficiente inativo atinge a vizinhança de seu valor ótimo, não há transferência de ganho a outros coeficientes, visto que a função de proporcionalidade para coeficientes inativos é  $\gamma_{\min}^P(n)$  e, como consequência, o algoritmo mantém inalterada a sua distribuição de ganho.

A nova estratégia de distribuição de ganho sintetizada em (4.9), apesar de inicialmente ser igual a do algoritmo PNLMS, altera substancialmente a alocação de ganhos aos coeficientes do filtro conforme o processo adaptativo evolui. Basicamente, após o coeficiente ativo atingir as proximidades de seu valor ótimo, seu ganho é reduzido e redistribuído visando acelerar a convergência dos demais coeficientes, sendo que quanto maior o ganho, maior sua contribuição. Na próxima seção, resultados numéricos atestam a eficácia do novo algoritmo.

#### 4.4 Resultados de Simulação do Algoritmo NSVR-PNLMS

Nesta seção, considerando um problema de identificação de sistemas, simulações de Monte Carlo (média de 100 realizações independentes) dos algoritmos PNLMS e NSVR-PNLMS são apresentadas. Os objetivos dessas simulações são:

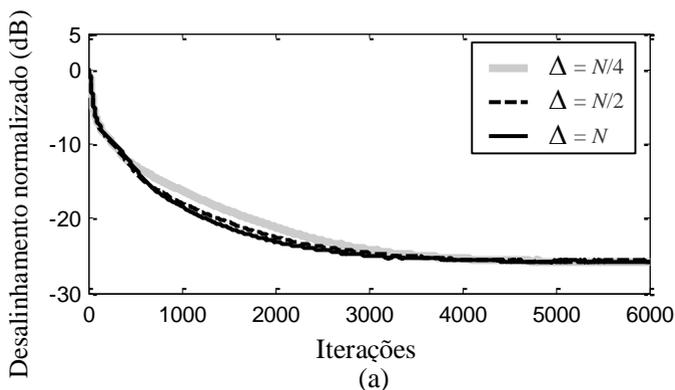
- i) Comparar os algoritmos em termos de velocidade de convergência, resposta a perturbações na planta e habilidade de rastreamento.
- ii) Comparar o comportamento dos algoritmos em plantas com alta e média esparsidades.
- iii) Avaliar as distribuições de ganhos dos algoritmos.
- iv) Verificar o comportamento dos coeficientes ativos e inativos dos algoritmos.
- v) Comparar o desempenho dos algoritmos para sinais de entrada brancos e correlacionados.
- vi) Verificar a sensibilidade do algoritmo frente a alterações dos parâmetros de limiar  $\varepsilon$  e número de amostras  $\Delta$ .

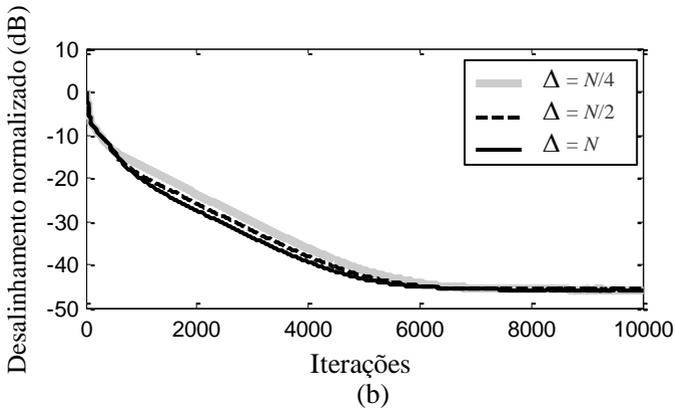
Para tal, seis exemplos são considerados. Em todos os exemplos são usados, tanto para o algoritmo PNLMS quanto para o algoritmo NSVR-PNLMS, passo de adaptação  $\beta = 0,1$ ,  $\rho = 0,01$ ,  $\delta = 0,01$  e constante de regularização  $\xi = 10^{-4}$ . Além disso, duas respostas ao impulso da planta, com grau de esparsidade distintos, são aqui utilizadas. Tais respostas, denominadas  $\mathbf{w}_A^o$  e  $\mathbf{w}_B^o$ , são obtidas a partir dos Modelos 1 e 4 da norma ITU-T Recomendações G.168 [65], com comprimento  $N = 128$  e grau de esparsidade, calculado a partir de (1.3) [26], igual a 0,78 e 0,42, respectivamente.

#### 4.4.1 Exemplo 4.1

Neste exemplo, o comportamento da convergência do algoritmo NSVR-PNLMS com respeito ao número de amostras  $\Delta$  e ao limiar  $\varepsilon$  é verificado. Além disso, a capacidade de  $V_i(n)$  estimar o instante no qual o coeficiente ativo atinge a vizinhança de seu valor ótimo é avaliada, assim como sua confiabilidade e robustez. Para tal, utiliza-se a resposta ao impulso da planta  $\mathbf{w}_A^o$ , dados de entrada branco com variância  $\sigma_x^2 = 1$  e diferentes valores de SNR.

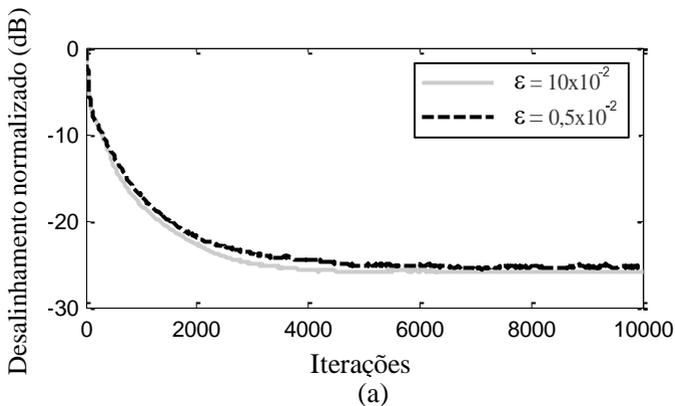
A Figura 4.1 mostra as curvas de desalinhamento normalizado para  $\varepsilon = 5 \times 10^{-2}$  e três valores de  $\Delta$ , isto é,  $N/4$ ,  $N/2$  e  $N$ . Pode-se observar que o ajuste de  $\Delta$  não é crítico, visto que o algoritmo apresenta comportamento similar para os três valores de  $\Delta$  selecionados, independente da variância do ruído de medição considerada.

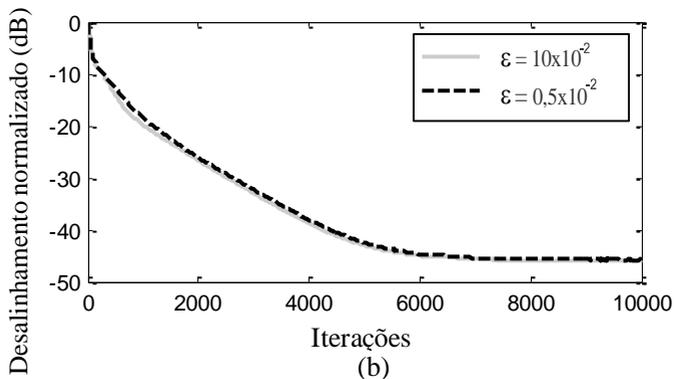




**Figura 4.1** – Exemplo 4.1. Curvas de desalinamento normalizado do algoritmo NSVR-PNLMS usando  $\Delta$  igual a  $N/4$ ,  $N/2$  e  $N$  com  $\varepsilon = 5 \times 10^{-2}$ . (a) SNR = 20 dB. (b) SNR = 40 dB.

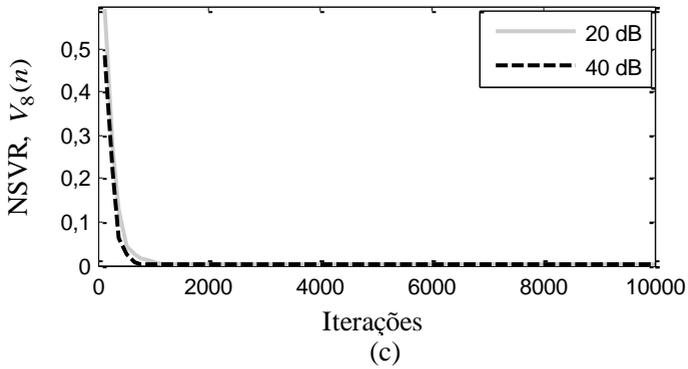
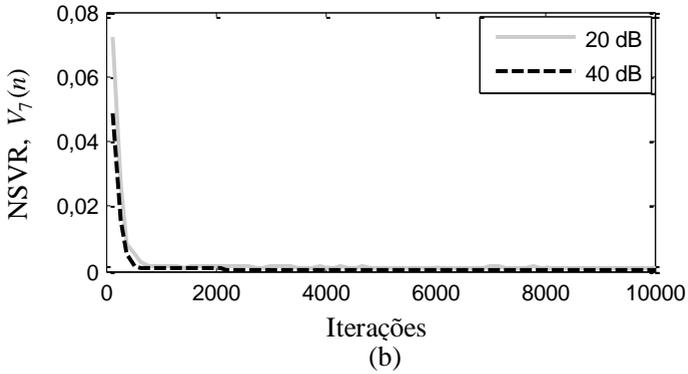
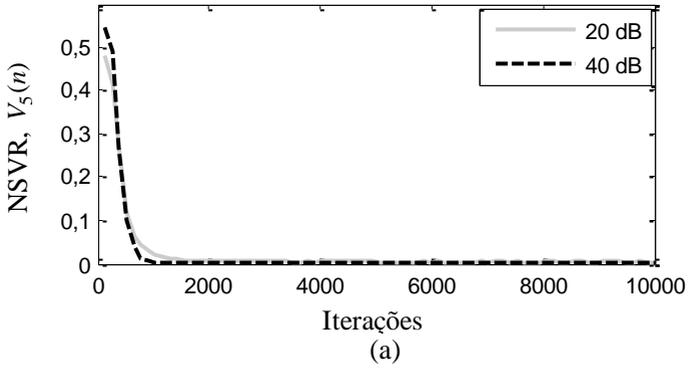
A Figura 4.2 ilustra as curvas de desalinamento normalizado para  $\Delta = N$ , usando  $\varepsilon$  igual a  $0,5 \times 10^{-2}$  e  $10 \times 10^{-2}$  para valores de SNR correspondentes a 20 e 40 dB. Dessa figura, observa-se que o limiar  $\varepsilon$  pode ser escolhido dentro de uma faixa de valores  $0,5 \times 10^{-2} \leq \varepsilon \leq 10 \times 10^{-2}$  sem praticamente alterar o comportamento do algoritmo, independente do valor de SNR considerado.

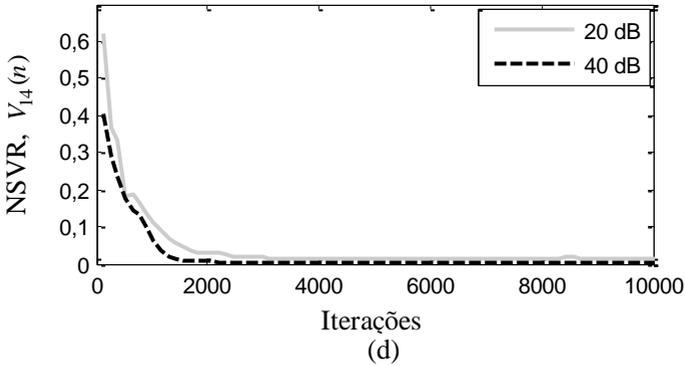




**Figura 4.2** – Exemplo 4.1. Curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS para  $\Delta = N$  e diferentes valores do limiar  $\varepsilon$ . (a) SNR = 20 dB. (b) SNR = 40 dB.

A Figura 4.3 descreve o comportamento da variável  $V_i(n)$  (NSVR de  $|w_i(n)|$ ) considerando dois valores de SNR, isto é, 20 e 40 dB para quatro coeficientes ativos com magnitudes distintas. Note que, no início do processo adaptativo,  $V_i(n)$  possui um valor maior e, conforme  $w_i(n) \rightarrow w_i^o$ ,  $V_i(n)$  diminui para um valor menor em regime permanente. Tal comportamento se deve ao fato de que coeficientes próximos aos seus valores ótimos exibem menor variação de magnitude. Dessa forma, a NSVR reflete adequadamente a característica de convergência do coeficiente e pode ser utilizada como uma métrica para a avaliação de sua convergência. Além disso, observa-se que  $V_i(n)$  exibe comportamento similar independente do valor de SNR e da magnitude do coeficiente.





**Figura 4.3** – Exemplo 4.1 – Comportamento da taxa de variação suavizada e normalizada (NSVR) para quatro coeficientes ativos com magnitudes distintas, considerando SNR igual a 20 e 40 dB.

A Tabela 4.2 mostra a distância relativa entre coeficientes ativos e seus correspondentes valores ótimos no instante em que o algoritmo NSVR-PNLMS troca a política de ganho individual do coeficiente. A correspondente distância é definida como

$$D_i = \frac{|w_i(\tau_i) - w_i^o|}{|w_i^o|} \quad (4.11)$$

onde  $\tau_i$  denota o instante no qual o algoritmo NSVR-PNLMS considera que o  $i$ -ésimo coeficiente quase atingiu a convergência. A Tabela 4.3 apresenta os valores do instante  $\tau_i$  em que o algoritmo NSVR-PNLMS altera o ganho do  $i$ -ésimo coeficiente para um ganho mínimo atribuído aos coeficientes que já tenham alcançado a vizinhança de seus valores ótimos.

**Tabela 4.2** – DISTÂNCIA  $D_i$  PARA SNR IGUAL A 20, 30 E 40 dB

USANDO  $\varepsilon = 5 \times 10^{-2}$

Coeficientes	20 dB	30 dB	40 dB
$w_5(\tau_5)$ $w_5^o = -0,1252$	0,08	0,06	0,05
$w_7(\tau_7)$ $w_7^o = 0,3215$	0,04	0,04	0,03
$w_8(\tau_8)$ $w_8^o = 0,2402$	0,05	0,04	0,05
$w_{14}(\tau_{14})$ $w_{14}^o = -0,0571$	0,17	0,09	0,08

**Tabela 4.3** – VALORES DE  $\tau_i$  PARA SNR IGUAL A 20, 30 E 40 dB

USANDO  $\varepsilon = 5 \times 10^{-2}$

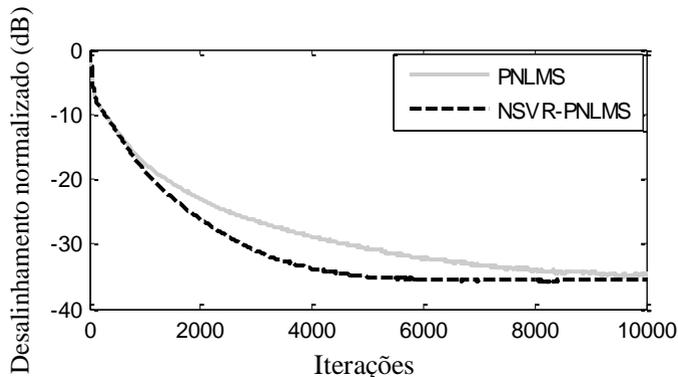
Coeficientes	20 dB	30 dB	40 dB
$w_5(\tau_5)$ $w_5^o = -0,1252$	584	554	556
$w_7(\tau_7)$ $w_7^o = 0,3215$	357	330	332
$w_8(\tau_8)$ $w_8^o = 0,2402$	433	394	385
$w_{14}(\tau_{14})$ $w_{14}^o = -0,0571$	955	928	881

#### 4.4.2 Exemplo 4.2

Este exemplo tem por objetivo comparar a velocidade de convergência dos algoritmos PNLMS e NSVR-PNLMS para uma planta

esparso, assim como ilustrar o comportamento dos coeficientes do filtro adaptativo e seus ganhos associados. Para tal, usa-se a resposta ao impulso da planta  $w_A^0$ , sinal de entrada branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Para a avaliação de convergência dos coeficientes no algoritmo NSVR-PNLMS, o valor do limiar é definido  $\varepsilon = 1 \times 10^{-2}$ .

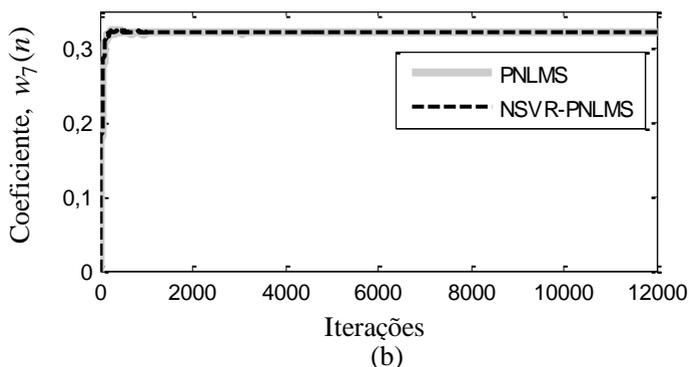
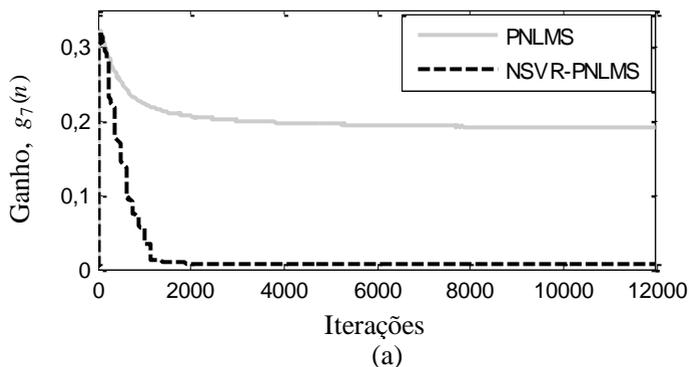
A Figura 4.4 mostra as curvas de desalinhamento normalizado dos algoritmos avaliados. Note que o algoritmo NSVR-PNLMS possui a mesma velocidade de convergência inicial do PNLMS, sendo que após esse período o novo algoritmo atinge velocidade de convergência maior. Além disso, pode-se observar que o algoritmo NSVR-PNLMS exibe um menor desalinhamento em regime permanente.



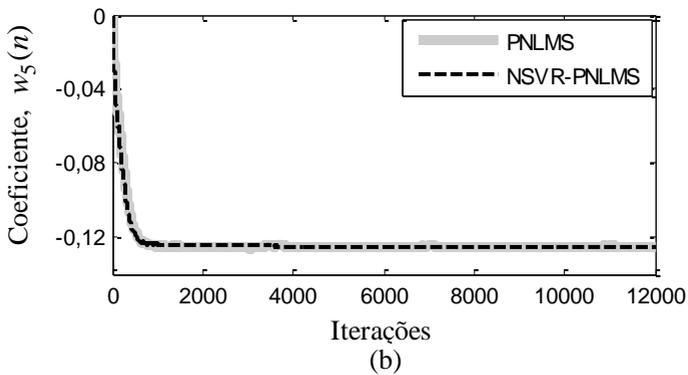
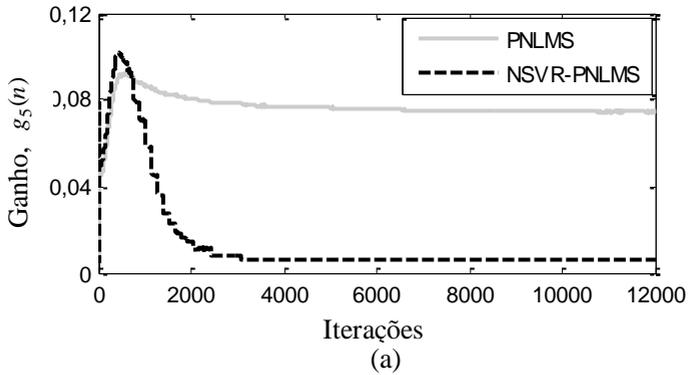
**Figura 4.4** - Exemplo 4.2. Curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ .

As Figuras 4.5 a 4.7 mostram a evolução dos valores dos coeficientes do filtro adaptativo e o comportamento de seus ganhos correspondentes. Note que o algoritmo proposto reduz o ganho atribuído aos coeficientes ativos após a sua convergência [veja Figuras 4.5(a) e 4.6(a)] e o redistribui aos demais coeficientes. Como resultado, o ganho do coeficiente com magnitude intermediária  $w_{int}$  atinge valor de pico superior ao do algoritmo PNLMS antes de sua convergência [veja Figura 4.6(a)] e o ganho dos coeficientes inativos aumenta seu valor (à medida que os coeficientes ativos se aproximam do seu valor ótimo) como consequência dessa redistribuição de ganhos [veja Figuras 4.7(a), 4.7(c)]

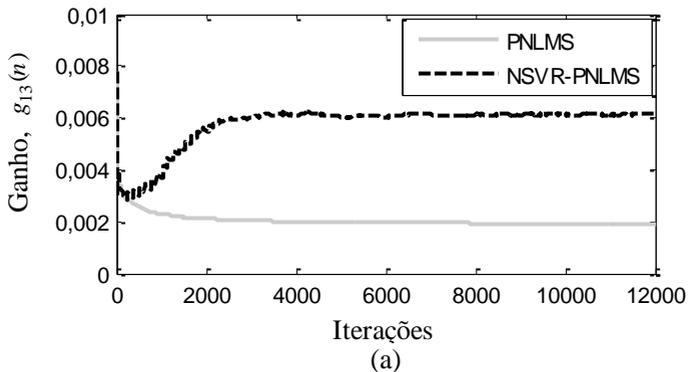
e 4.7(e)]. Além disso, pode-se observar que a evolução dos coeficientes ativos praticamente não é afetada pela redução de ganho efetuada pelo algoritmo proposto [veja Figuras 4.5(b) e 4.6(b)]. Por outro lado, devido ao aumento de ganho dos coeficientes inativos durante o processo adaptativo, esses coeficientes melhoram suas taxas de convergência [veja Figuras 4.7(b), 4.7(d) e 4.7(f)]. Esse aumento de ganho dos coeficientes inativos e ativos que ainda não convergiram é responsável pela maior aceleração da convergência do algoritmo NSVR-PNLMS quando comparado com o algoritmo PNLMS.

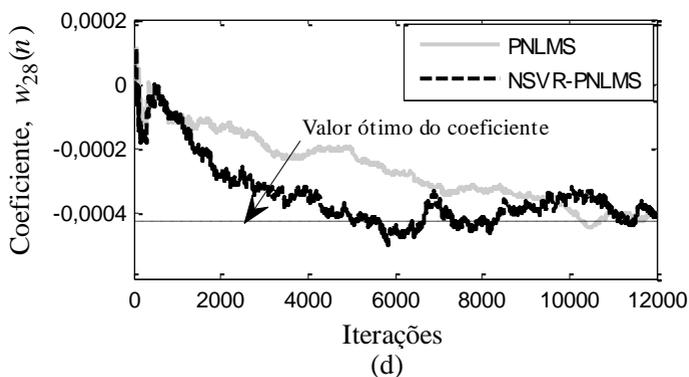
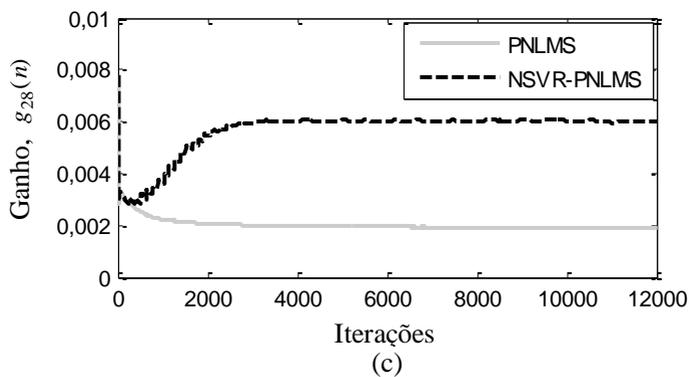
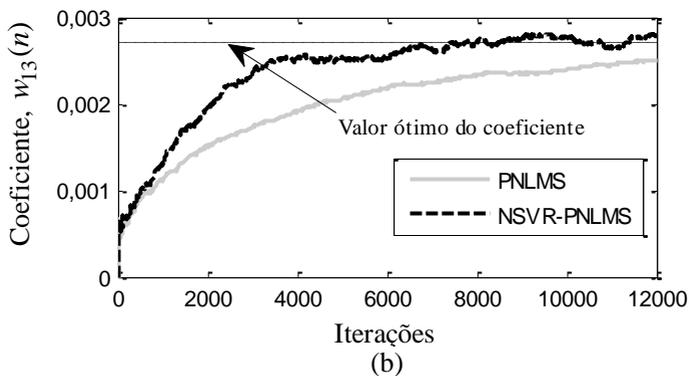


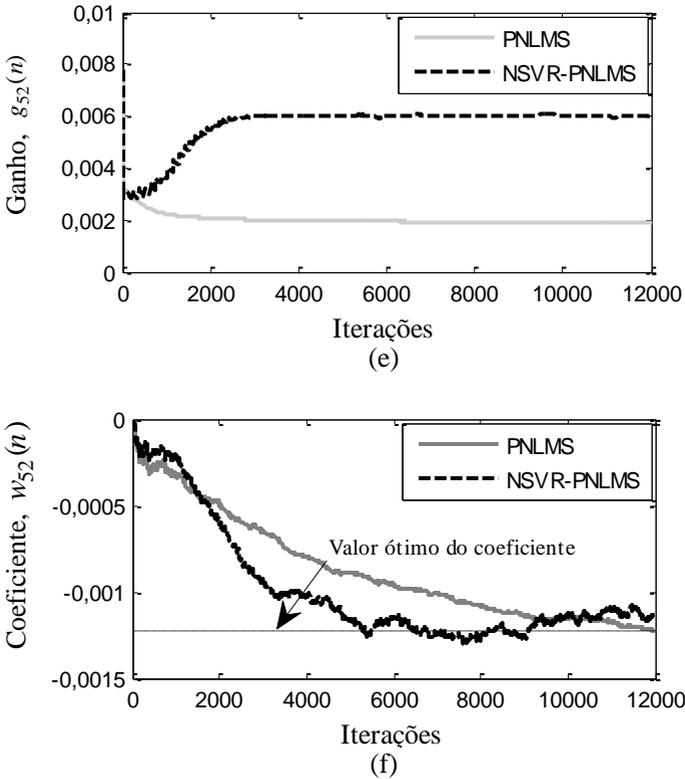
**Figura 4.5** - Exemplo 4.2. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $w_A^o$  para os algoritmos PNLMS e NSVR-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente.



**Figura 4.6** - Exemplo 4.2. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos PNLMs e NSVR-PNLMs. (a) Ganho atribuído. (b) Evolução do coeficiente.







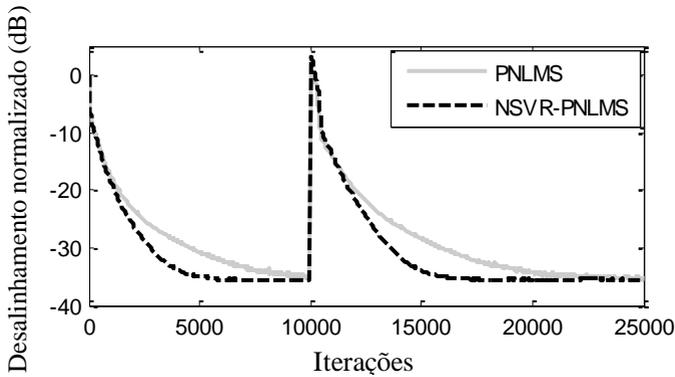
**Figura 4.7** - Exemplo 4.2. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos PNLMS e NSVR-PNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente.

#### 4.4.3 Exemplo 4.3

Neste exemplo, o comportamento do algoritmo NSVR-PNLMS é avaliado considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário do exemplo anterior é utilizado, porém agora, o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita no instante  $n = 10000$ , alterando a posição de todos os coeficientes ativos.

A Figura 4.8 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS e proposto. Pode-se observar que o algoritmo NSVR-PNLMS supera o PNLMS mesmo na presença de perturbações abruptas na resposta ao impulso da planta. Tal melhoria na taxa de

convergência do algoritmo NSVR-PNLMS é decorrente da sua melhor distribuição de ganho, isto é, após a convergência dos coeficientes ativos o seu ganho é reduzido e redistribuído aos demais coeficientes acelerando assim a convergência global do filtro adaptativo. Observe que, mesmo na presença de distúrbio na planta, o algoritmo proposto supera em termos de velocidade de convergência o algoritmo PNLMS.



**Figura 4.8** - Exemplo 4.3. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . Quando  $n = 10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^0$  é deslocado de 12 amostras para a direita.

#### 4.4.4 Exemplo 4.4

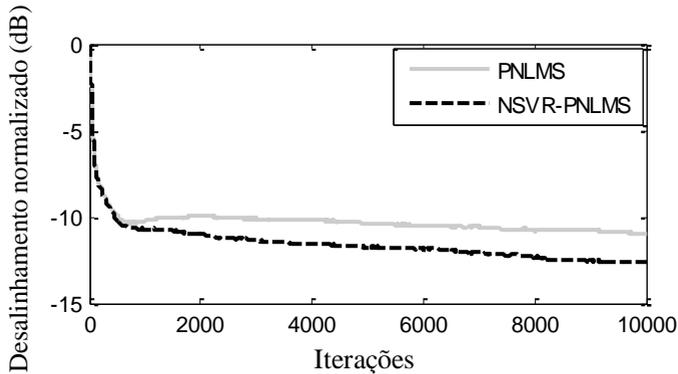
Neste exemplo, a habilidade de rastreamento do algoritmo NSVR-PNLMS é avaliada. Para tal, a resposta ao impulso da planta é variante no tempo, obtida através de um modelo modificado de primeira ordem de Markov dado por [66]–[68]

$$\mathbf{w}^o(n+1) = \alpha \mathbf{w}^o(n) + \sqrt{1 - \alpha^2} \mathbf{s}(n) \quad (4.12)$$

onde  $0 << \alpha < 1$  é o parâmetro que controla a contribuição relativa da parcela de inovação (segunda parcela à direita) e  $\mathbf{s}(n)$ , denota uma sequência randômica de comprimento  $N$  com distribuição Gaussiana de média zero e variância  $\sigma_s^2$ . O valor inicial de  $\mathbf{w}^o(n)$  é obtido a partir da resposta ao impulso da planta  $\mathbf{w}_A^0$ , sendo que o sinal de entrada, a variância do ruído de medição e todos os demais parâmetros são idênticos

aos utilizados no Exemplo 4.2. Para obter a variação temporal da planta, dada por (4.12), são considerados  $\alpha = 1 - 1 \times 10^{-7}$  e  $\sigma_s^2 = 1$ .

A Figura 4.9 mostra a habilidade de rastreamento do novo algoritmo. Note que o desalinhamento normalizado do algoritmo NSVR-PNLMS, em virtude da redistribuição de ganhos dos coeficientes ativos que atingiram a vizinhança de seus valores ótimos, é menor do que o alcançado pelo algoritmo PNLMS após a convergência inicial. Com isso, o algoritmo NSVR-PNLMS apresenta maior capacidade de rastrear variações da resposta ao impulso da planta do que o algoritmo PNLMS.



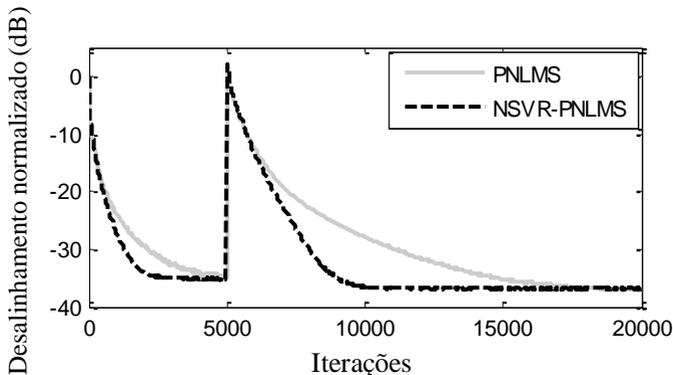
**Figura 4.9** - Exemplo 4.4. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . Inicialmente a planta variante no tempo  $\mathbf{w}^o$  é igual a  $\mathbf{w}_A^o$  com grau de esparsidade 0,78. Para  $n = 10000$ , o grau de esparsidade da planta é modificado para 0,32.

#### 4.4.5 Exemplo 4.5

Neste exemplo, compara-se a habilidade de rastreamento dos algoritmos em um ambiente (planta) de esparsidade média. Para tal, considerando uma variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB), a resposta ao impulso  $\mathbf{w}_A^o$  com grau de esparsidade  $S(\mathbf{w}_A^o) = 0,78$  é abruptamente alterada em  $n = 10000$  para a resposta ao impulso  $\mathbf{w}_B^o$ , cujo grau de esparsidade é  $S(\mathbf{w}_B^o) = 0,42$ . Aqui, o sinal de

entrada e todos os outros parâmetros são idênticos aos utilizados no Exemplo 4.2.

A Figura 4.10 mostra as curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS quando a planta esparsa muda subitamente para uma de menor grau de esparsidade. Pode-se observar que o algoritmo proposto apresenta uma velocidade de convergência superior ao do algoritmo PNLMS (tanto para planta esparsa quanto dispersa) e, como consequência, atinge o regime permanente muito antes do que o PNLMS. Note que, em virtude do maior número de coeficientes ativos, o algoritmo NSVR-PNLMS operando em um cenário de esparsidade média tem desempenho superior ao PNLMS, principalmente, devido ao aumento do ganho dos coeficientes de menor magnitude e, com isso, acelerando ainda mais sua convergência em plantas menos esparsas.



**Figura 4.10** - Exemplo 4.5. Desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-2}$ . A planta  $\mathbf{w}_A^o$  [com esparsidade  $S(\mathbf{w}_A^o) = 0,78$ ] é trocada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^o$  [com esparsidade  $S(\mathbf{w}_B^o) = 0,42$ ].

#### 4.4.6 Exemplo 4.6

Neste exemplo, o comportamento do algoritmo NSVR-PNLMS para sinais de entrada correlacionados é discutido. Para tal, considera-se um sinal de entrada, de média zero e variância unitária, obtido através de um processo AR(2) dado por

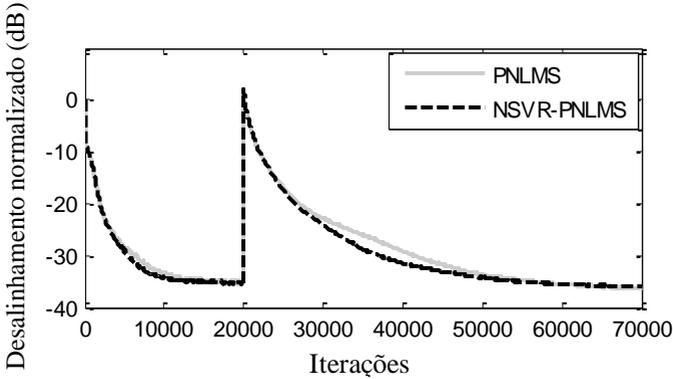
$$x(n) = b_1 x(n-1) + b_2 x(n-2) + \eta(n) \quad (4.13)$$

onde  $\eta(n)$  é um ruído branco com variância  $\sigma_{\eta}^2 = 0,77$ ,  $b_1 = 0,40$  e  $b_2 = -0,40$ , resultando em uma dispersão dos autovalores da matriz de autocorrelação do vetor de entrada  $\chi = 10$ . O cenário aqui utilizado é o mesmo do Exemplo 4.3, isto é, no início do processo adaptativo a resposta ao impulso esparsa  $\mathbf{w}_A^o$  é abruptamente deslocada de 12 amostras para a direita em  $n = 20000$ . Com exceção do limiar de convergência dos coeficientes do algoritmo proposto, que agora é definido como  $\varepsilon = 1 \times 10^{-3}$ , todos os demais parâmetros são os mesmos utilizados no Exemplo 4.3.

A Figura 4.11 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS para sinal de entrada correlacionado com dispersão  $\chi = 10$ . Note que o algoritmo proposto novamente exibe velocidade de convergência superior ao PNLMS independente da esparsidade da planta. A redistribuição de ganho dos coeficientes ativos (na proximidade de seus valores ótimos) para os coeficientes inativos é responsável pela melhora de desempenho do algoritmo NSVR-PNLMS. Contudo, em virtude da convergência mais lenta do filtro adaptativo (decorrente do uso de um sinal de entrada correlacionado), o limiar de convergência  $\varepsilon$  foi reduzido para  $1 \times 10^{-3}$  evitando, assim, considerar convergência de um coeficiente que ainda esteja um tanto longe do seu valor ótimo.

#### 4.5 Algoritmo IPNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente

O algoritmo IPNLMS usa uma mistura controlada de ganhos de adaptação proporcional e não proporcional. Tal estratégia melhora as características de convergência do algoritmo IPNLMS para uma larga faixa de valores de esparsidade da planta em relação ao PNLMS. Conforme mostrado na Seção 2.4, a regra específica para computar o ganho individual do algoritmo IPNLMS é dada por (2.17). Contudo, uma característica da distribuição de ganho do IPNLMS é atribuir ganho proporcional para todos os coeficientes mesmo após esses coeficientes já terem alcançado a vizinhança de seus valores ótimos. A seguir, para contornar tal deficiência, uma nova versão do algoritmo IPNLMS, denominada NSVR-IPNLMS, é introduzida.



**Figura 4.11** - Exemplo 4.6. Curvas de desalinhamento normalizado dos algoritmos PNLMS e NSVR-PNLMS considerando um sinal de entrada AR(2) com variância  $\sigma_x^2$  unitária e dispersão  $\chi=10$  usando  $\Delta=N$  e  $\varepsilon=1 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$ , em  $n=20000$ , é deslocada para direita em 12 posições.

#### 4.5.1 Algoritmo NSVR-IPNLMS

Visto que a distribuição de ganho do IPNLMS é composta de duas partes, isto é, uma não proporcional (comum a todos os coeficientes) e outra proporcional à magnitude do coeficiente, o algoritmo NSVR-IPNLMS retira a parte proporcional do ganho dos coeficientes que já atingiram a vizinhança de seus valores ótimos, mantendo somente a parte não proporcional; como resultado, o algoritmo proposto aumenta o ganho dos coeficientes que ainda não convergiram. Dessa forma, para obter o ganho do algoritmo NSVR-IPNLMS, a seguinte formulação é utilizada:

$$g_i^{\text{NIP}}(n) = \frac{\gamma_i^{\text{NIP}}(n)}{\Phi^{\text{NIP}}(n)}, \quad i = 1, 2, \dots, N \quad (4.14)$$

com

$$\gamma_i^{\text{NIP}}(n) = (1 - \alpha) \frac{\max(\delta, \|\mathbf{w}(n)\|_1)}{N} + (1 + \alpha) |w_i^c(n)| \quad (4.15)$$

onde o sobrescrito NIP caracteriza o algoritmo NSVR-IPNLMS,  $\delta$  denota o parâmetro de inicialização e o vetor auxiliar  $\mathbf{w}^c(n) = [w_1^c(n) \ w_2^c(n) \ \dots \ w_N^c(n)]$  é definido como

$$w_i^c(n) = \begin{cases} 0, & V_i(n) \leq \varepsilon \\ w_i(n), & \text{caso contrário.} \end{cases} \quad (4.16)$$

Note de (4.14)-(4.16) que, após o  $i$ -ésimo coeficiente atingir a vizinhança de seu valor ótimo, um pequeno ganho de manutenção é atribuído ao correspondente coeficiente, o que está alinhado com a política de distribuição proposta em (4.3) e (4.4).

A Tabela 4.4 descreve o algoritmo NSVR-IPNLMS. Note que o novo algoritmo mantém, no início do processo adaptativo, a mesma filosofia do IPNLMS, isto é, ganho composto por uma mistura ponderada entre uma parcela não proporcional e outra proporcional à magnitude do coeficiente. À medida que os coeficientes atingem a vizinhança de seus valores ótimos, ou seja, tão logo  $V_i(n) \leq \varepsilon$ , apenas a parcela comum de ganho a todos os coeficientes do filtro adaptativo (parcela não proporcional) é mantida.

#### 4.5.2 Análise da Distribuição de Ganho do Algoritmo NSVR-IPNLMS

Inicialmente, levando em conta que a política de alocação de ganho do algoritmo NSVR-IPNLMS reduz o ganho atribuído a coeficientes que já alcançaram a vizinhança de seus valores ótimos, as seguintes definições são estabelecidas:

- i) **Ganho dos coeficientes que alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  tenha quase atingido a convergência e seu ganho associado é dado por

$$g_c^{\text{NIP}}(n) = \frac{1}{\Phi^{\text{NIP}}(n)} \left[ \frac{(1-\alpha) \|w(n)\|_1}{N} \right]. \quad (4.17)$$

- ii) **Ganho dos coeficientes que ainda não alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) > \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  não atingiu a convergência e seu ganho associado é dado por

$$g_{nc,i}^{\text{NIP}}(n) = \frac{1}{\Phi^{\text{NIP}}(n)} \left[ \frac{(1-\alpha) \|\mathbf{w}(n)\|_1}{N} + (1+\alpha) |w_i(n)| \right]. \quad (4.18)$$

**Tabela 4.4** – ALGORITMO NSVR-IPNLMS

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$


---

Para  $n = 1, 2, 3, \dots$

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

$$Q_i(n) = |w_i(n)| + |w_i(n-1)| + \dots + |w_i(n - \frac{\Delta}{2} + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$P_i(n) = |w_i(n - \frac{\Delta}{2})| + |w_i(n - \frac{\Delta}{2} - 1)| + \dots + |w_i(n - \Delta + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$V_i(n) = \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, \quad \begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

Se  $V_i(n) \leq \varepsilon$

$$w_i^c(n) = 0, \quad 1 \leq i \leq N \text{ e } n \geq N + \Delta$$

$$g_i^{\text{NIP}}(n) = \frac{\gamma_i^{\text{NIP}}(n)}{\sum_{j=1}^N \gamma_j^{\text{NIP}}(n)}, \quad 1 \leq i \leq N$$

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

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{NIP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{NIP}}(n) \mathbf{x}(n) + \xi}$$


---

O ganho total distribuído dentre os coeficientes do filtro adaptativo por iteração no algoritmo NSVR-IPNLMS pode ser descrito como

$$g_{\text{total}}^{\text{NIP}}(n) = \frac{1}{\Phi^{\text{NIP}}(n)} \left\{ \frac{(1-\alpha) \|\mathbf{w}(n)\|_1}{N} + (1+\alpha) \sum_{i \in (A+B-D)} |w_i(n)| \right\} = 1 \quad (4.19)$$

onde  $D$  é o conjunto de índices associado às posições dos coeficientes (ativos e inativos) que já tenham alcançado a vizinhança de seus valores ótimos e,  $A$  e  $B$  são os mesmos conjuntos de posições definidos em (4.9).

Uma vez que o número de coeficientes que tenham quase atingido a convergência  $N_c(n)$  varia durante o processo adaptativo, o segundo termo do lado direito de (4.19) se altera, apresentando as seguintes características:

- i) No início do processo adaptativo,  $N_c(n) = 0$  visto que  $D$  é um conjunto vazio. Nessa condição, a distribuição de ganho do algoritmo NSVR-IPNLMS é similar àquela do algoritmo IPNLMS.
- ii) Conforme os coeficientes tendem a seus valores ótimos,  $V_i(n)$  diminui seu valor e, tão logo  $V_i(n) \leq \varepsilon$  (implicando que o  $i$ -ésimo coeficiente já tem quase atingido a convergência),  $N_c(n)$  é incrementado e  $D$  não é mais um conjunto vazio; portanto, o número de elementos associados à segunda parcela do lado direito de (4.19) diminui. Em uma situação ideal (em regime permanente) todos os coeficientes atingiriam a proximidade de seus valores ótimos e, assim,  $N_c(n) = N$  e  $(A+B-D) = \emptyset$ , de forma que todos os coeficientes receberiam o mesmo ganho e o novo algoritmo se comportaria como o algoritmo NLMS.
- iii) Sempre que o  $i$ -ésimo coeficiente atinge a vizinhança de seu valor ótimo,  $w_i^c(n)$  muda de  $w_i(n)$  para 0. Com isso, o ganho transferido aos demais coeficientes é agora dado por

$$T_i^{\text{NIP}}(n) = \frac{\frac{(1-\alpha)\|\mathbf{w}(n)\|_1}{N} + (1+\alpha)|w_i(n)|}{\sum_{j=1}^N \gamma_j^{\text{NIP}}(n)} - \frac{(1-\alpha)\|\mathbf{w}(n)\|_1}{N} \cdot \frac{\sum_{\substack{j=1 \\ j \neq i}}^N \gamma_j^{\text{NIP}}(n) + \frac{(1-\alpha)\|\mathbf{w}(n)\|_1}{N}}{N}. \quad (4.20)$$

Observe que, quanto maior a magnitude do coeficiente  $w_i(n)$ , maior o valor de  $T_i^{\text{NIP}}(n)$  e, como consequência, mais ganho é transferido aos demais coeficientes.

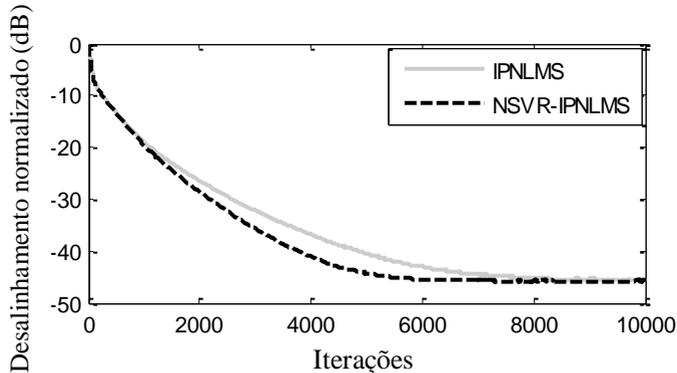
#### 4.6 Resultados de Simulação do Algoritmo NSVR-IPNLMS

Nesta seção, considerando um problema de identificação de sistemas, resultados de simulações de Monte Carlo (média de 100 realizações independentes) dos algoritmos IPNLMS e NSVR-IPNLMS são mostrados. Para tal, cinco exemplos são considerados. Em todos os exemplos a constante de regularização é  $\xi = 10^{-4}$ . Para o algoritmo IPNLMS, o valor do parâmetro de peso  $\alpha$  entre as parcelas de ganho proporcional e não proporcional é igual a zero e, para o algoritmo proposto,  $\Delta = N$ .

##### 4.6.1 Exemplo 4.7

Este exemplo compara a velocidade de convergência dos algoritmos IPNLMS e NSVR-IPNLMS para uma planta esparsa, assim como mostra o comportamento dos coeficientes do filtro adaptativo e seus ganhos associados. Para tal, usa-se passo de adaptação  $\beta = 0,1$ , resposta ao impulso da planta  $\mathbf{w}_A^o$ , sinal de entrada branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 0,0001$  (SNR = 40 dB). Para avaliação da convergência dos coeficientes do algoritmo NSVR-IPNLMS, o valor do limiar usado é  $\varepsilon = 5 \times 10^{-3}$ .

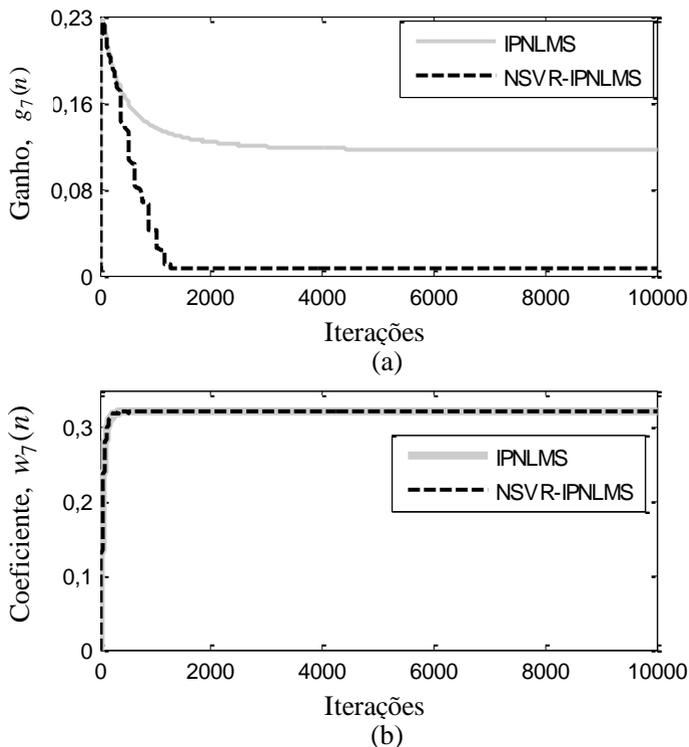
A Figura 4.12 mostra as curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS. Note que o algoritmo NSVR-IPNLMS possui a mesma velocidade de convergência inicial do IPNLMS, sendo que após esse período o algoritmo proposto atinge velocidade de convergência maior.



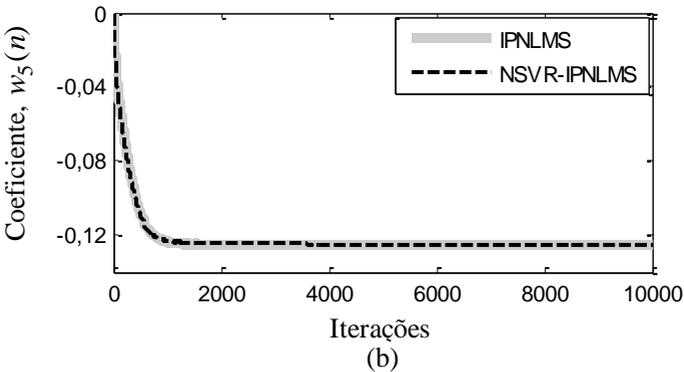
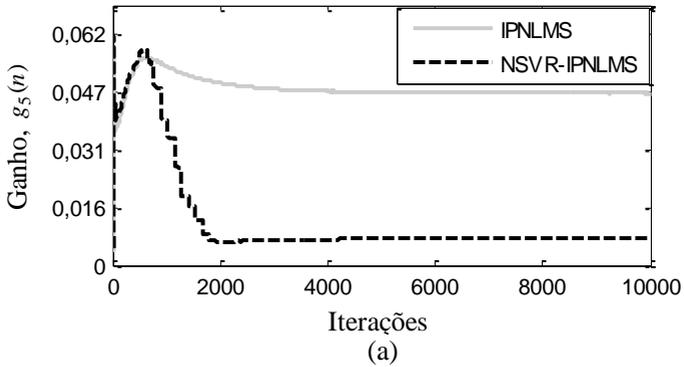
**Figura 4.12** – Exemplo 4.7. Curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ .

As Figuras 4.13 a 4.15 mostram a evolução dos valores dos coeficientes do filtro adaptativo e o comportamento de seus ganhos correspondentes. Observe que o algoritmo proposto reduz o ganho atribuído aos coeficientes ativos após a sua convergência [veja Figuras 4.13(a) e 4.14(a)] e o redistribui aos demais coeficientes. Como resultado, o ganho do coeficiente com magnitude intermediária  $w_{int}$  atinge valor de pico superior ao do algoritmo IPNLMS antes de sua convergência [veja Figura 4.14(a)] e o ganho dos coeficientes inativos aumenta seu valor (à medida que os coeficientes ativos se aproximam de seus valores ótimos) como consequência dessa redistribuição de ganhos [veja Figuras 4.15(a), 4.15(c) e 4.15(e)]. Além disso, pode-se verificar que a evolução dos coeficientes ativos praticamente não é afetada pela redução de ganho realizada pelo algoritmo proposto [veja Figuras 4.13(b) e 4.13(b)]. Por outro lado, os coeficientes inativos apresentam um aumento na velocidade de convergência decorrente do aumento de ganho [veja Figuras 4.15(b), 4.15(d) e 4.15(f)]. A redistribuição de ganho dos coeficientes ativos que alcançaram a vizinhança de seus valores ótimos para os coeficientes inativos e ativos que ainda não convergiram é a

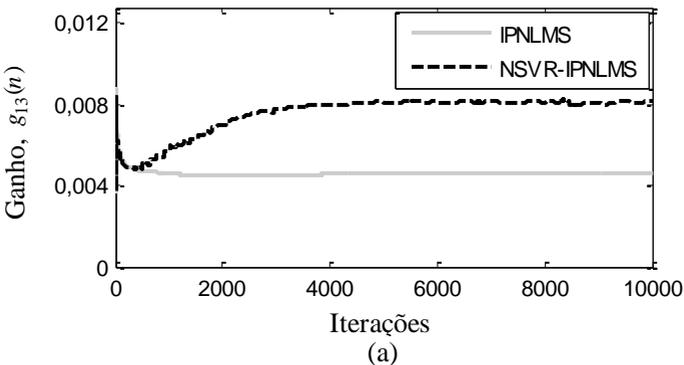
responsável pela melhoria da taxa de convergência do algoritmo NSVR-IPNLMS quando comparada com o IPNLMS.

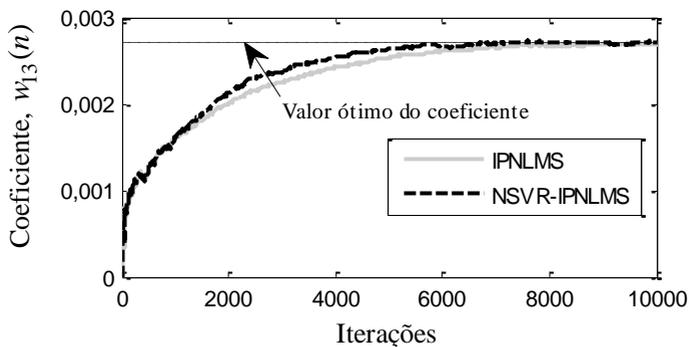


**Figura 4.13** - Exemplo 4.7. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente.

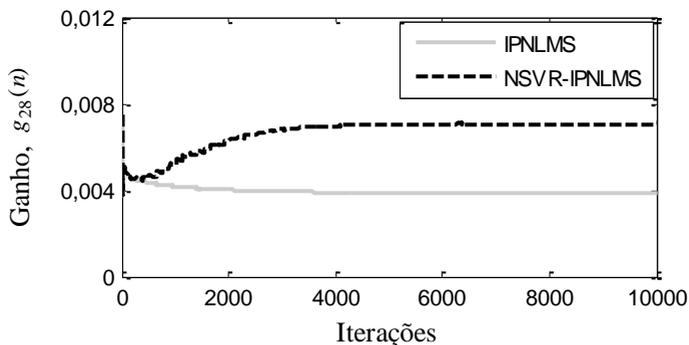


**Figura 4.14** - Exemplo 4.7. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $w_A^o$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente.

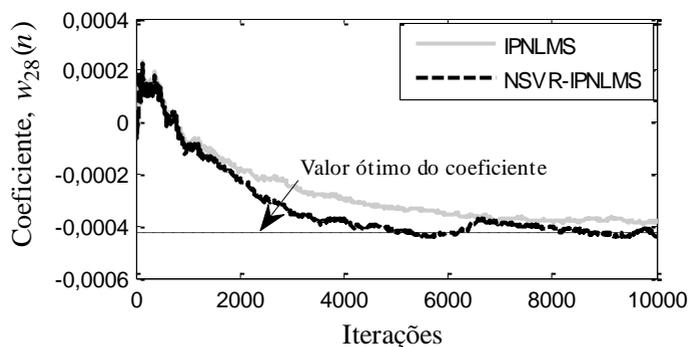




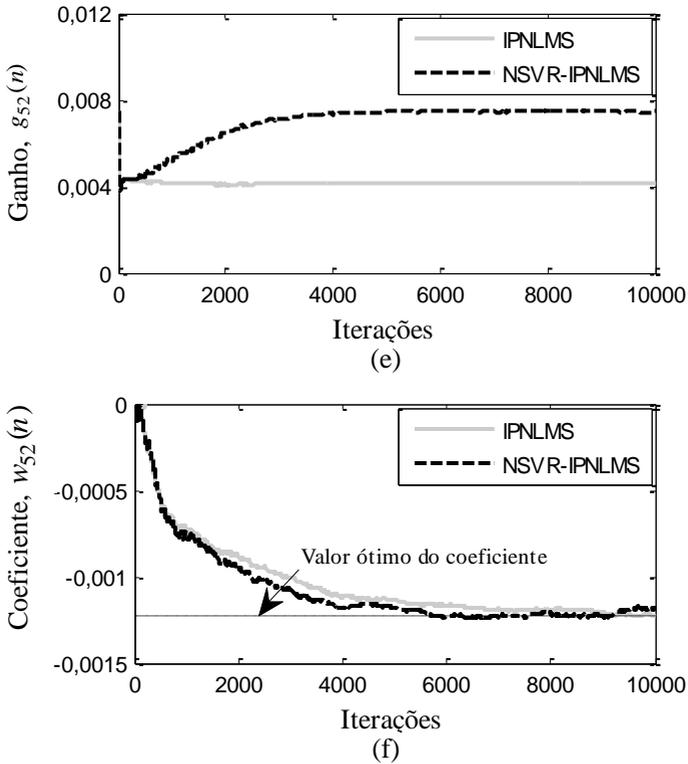
(b)



(c)



(d)



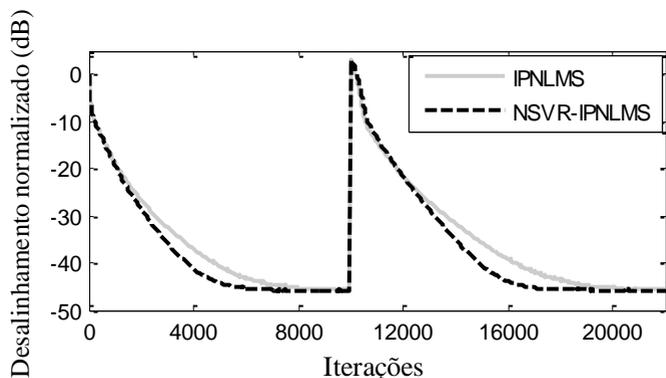
**Figura 4.15** - Exemplo 4.7. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^0$  para os algoritmos IPNLMS e NSVR-IPNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente.

#### 4.6.2 Exemplo 4.8

Neste exemplo, o comportamento do algoritmo NSVR-IPNLMS é avaliado considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário do exemplo anterior é utilizado, contudo, no instante  $n=10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^0$  é deslocado de 12 amostras para a direita.

A Figura 4.16 mostra as curvas de desalinhamento normalizado dos algoritmos avaliados. Pode-se observar que o algoritmo NSVR-IPNLMS supera o IPNLMS mesmo na presença de perturbações abruptas na resposta ao impulso da planta. O aumento da velocidade de convergência do algoritmo NSVR-IPNLMS é consequência da nova

política de distribuição de ganho que (mesmo na presença de distúrbio na planta) redistribui o ganho dos coeficientes ativos que alcançaram as proximidades de seus valores ótimos aos demais coeficientes do filtro adaptativo e, com isso, atinge maior velocidade de convergência quando comparado com o algoritmo IPNLMS.



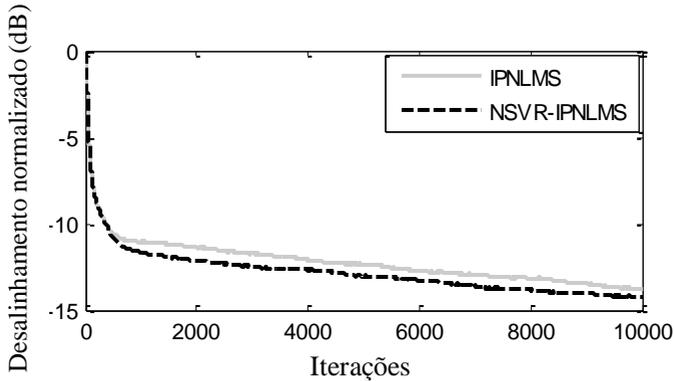
**Figura 4.16** - Exemplo 4.8. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ . Quando  $n = 10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^0$  é deslocado de 12 amostras para a direita.

#### 4.6.3 Exemplo 4.9

Neste exemplo, a habilidade de rastreamento do algoritmo NSVR-IPNLMS é verificada considerando uma variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Para tal, a resposta ao impulso da planta usada é variante no tempo utilizando o modelo modificado de primeira ordem de Markov dado por (4.12), sendo que todos os outros parâmetros para obtenção da resposta ao impulso da planta são idênticos aos utilizados no Exemplo 4.4. Para o algoritmo NSVR-IPNLMS, o valor do limiar de convergência é  $\varepsilon = 5 \times 10^{-2}$ .

A Figura 4.17 mostra a habilidade de rastreamento do algoritmo NSVR-IPNLMS. Note que o desalinhamento normalizado do algoritmo proposto, em virtude da melhor distribuição de ganhos dentre os coeficientes do filtro adaptativo, é menor do que o alcançado pelo

algoritmo IPNLMS após a convergência inicial. Assim, o algoritmo NSVR-IPNLMS exibe maior capacidade de seguir as variações da resposta ao impulso da planta do que o algoritmo IPNLMS.

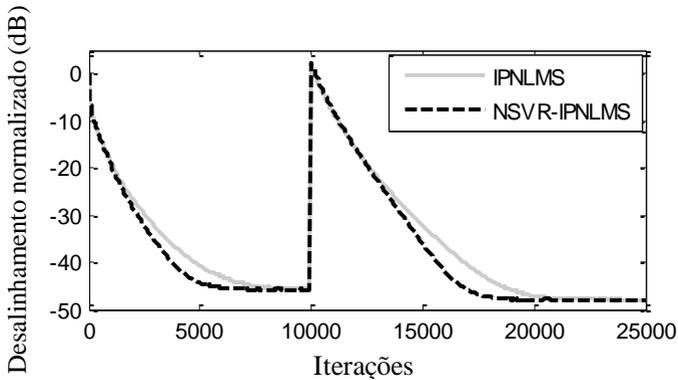


**Figura 4.17** - Exemplo 4.9. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ . A planta  $\mathbf{w}^o$  é variante no tempo e inicialmente igual  $\mathbf{w}_A^o$  [ $S(\mathbf{w}_A^o) = 0,78$ ]. Para  $n = 10000$ , o grau de esparsidade da planta é alterado para 0,32.

#### 4.6.4 Exemplo 4.10

Neste exemplo, compara-se o desempenho dos algoritmos IPNLMS e NSVR-IPNLMS em um ambiente de esparsidade média. Para tal, a resposta ao impulso  $\mathbf{w}_A^o$  com grau de esparsidade  $S(\mathbf{w}_A^o) = 0,78$  é abruptamente mudada em  $n = 10000$  para a resposta ao impulso  $\mathbf{w}_B^o$ , cujo grau de esparsidade é  $S(\mathbf{w}_B^o) = 0,42$ . A entrada, a variância do ruído de medição e todos os outros parâmetros são idênticos aos utilizados no Exemplo 4.7.

A Figura 4.18 mostra as curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS quando a planta esparsa muda subitamente para uma de menor grau de esparsidade. Pode-se verificar que o algoritmo proposto apresenta velocidade de convergência superior ao do algoritmo IPNLMS independente do grau de esparsidade da planta considerado.

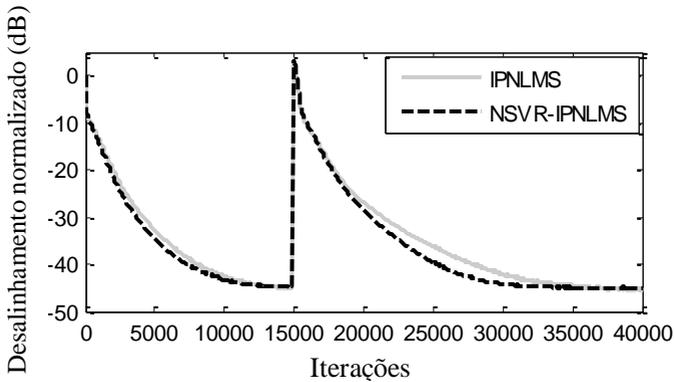


**Figura 4.18** - Exemplo 4.10. Desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$  [com esparsidade  $S(\mathbf{w}_A^o) = 0,78$ ] é trocada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^o$  [com esparsidade  $S(\mathbf{w}_B^o) = 0,42$ ].

#### 4.6.5 Exemplo 4.11

Neste exemplo, o comportamento do algoritmo NSVR-IPNLMS para entradas correlacionadas considerando uma variância do ruído de medição  $\sigma_v^2 = 0,0001$  (SNR = 40 dB) é ilustrado. Para tal, o sinal de entrada, com média zero e variância unitária, é obtido através do processo AR(2) dado por (4.13) usando os mesmos parâmetros do Exemplo 4.6. Com respeito à resposta ao impulso da planta,  $\mathbf{w}_A^o$  é deslocada de 12 amostras para a direita em  $n = 15000$ . Para o algoritmo IPNLMS, o passo de adaptação é  $\beta = 0,11$  e, para o algoritmo NSVR-IPNLMS,  $\beta = 0,1$  e o limiar de convergência  $\varepsilon = 1 \times 10^{-3}$ .

A Figura 4.19 mostra as curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS para sinal de entrada correlacionado com dispersão  $\chi = 10$ . Note que o algoritmo proposto supera novamente o IPNLMS independente do grau de esparsidade da planta considerado.



**Figura 4.19** - Exemplo 4.11. Curvas de desalinhamento normalizado dos algoritmos IPNLMS e NSVR-IPNLMS considerando um sinal de entrada AR(2) com variância  $\sigma_x^2$  unitária e dispersão  $\chi = 10$  usando  $\Delta = N$  e  $\varepsilon = 1 \times 10^{-3}$ . A planta  $\mathbf{w}_A^o$ , em  $n = 15000$ , é deslocada para a direita em 12 posições.

#### 4.7 Algoritmo IAF-PNLMS Baseado na Taxa de Variação da Magnitude do Coeficiente

O algoritmo PNLMS [38] utiliza um fator de ativação  $\gamma_{\min}^P(n)$  comum a todos os coeficientes do filtro adaptativo [veja (2.8)], o que implica um comportamento não proporcional dos coeficientes inativos [44], [48]. Assim, o algoritmo IAF-PNLMS apresenta uma forma alternativa de computar o fator de ativação resgatando a filosofia proporcional em sua determinação e, com isso, o ganho individual do IAF-PNLMS tende a ser proporcional à magnitude do coeficiente tanto para os coeficientes ativos quanto para os inativos [48]. Tal estratégia conduz a uma maior velocidade de convergência para plantas com resposta ao impulso exibindo alta esparsidade. Isso é obtido pelo aumento do ganho atribuído aos coeficientes ativos em relação ao algoritmo PNLMS e, conseqüentemente, pela redução do ganho atribuído para os coeficientes inativos. Entretanto, tal procedimento também acentua os problemas típicos do PNLMS, isto é, uma forte desaceleração após o período inicial e a perda de desempenho para plantas com baixa esparsidade. Outro ponto importante a ser notado é o aumento do erro em excesso em regime permanente do algoritmo IAF-PNLMS, visto que

coeficientes mais significativos possuem ganhos mais elevados e, com isso, provocando uma maior oscilação em torno de seus valores ótimos.

Aqui, visando diminuir os “efeitos colaterais” decorrentes da extrapolação da filosofia proporcional presente no algoritmo IAF-PNLMS, uma nova versão, denominada NSVR-IAF-PNLMS, é proposta. Para tal, o novo algoritmo utiliza a política de distribuição de ganho apresentada na Seção 4.2, na qual diminui o ganho atribuído a coeficientes que já tenham quase atingido a convergência e o redistribui para os que ainda não convergiram.

#### 4.7.1 Algoritmo NSVR-IAF-PNLMS

Para obter a redução do ganho alocado aos coeficientes que já tenham atingido a vizinhança de seus valores ótimos, o novo algoritmo usa  $\gamma_o(k) = 1/N$  para definir o ganho de tais coeficientes. Como resultado, a distribuição de ganho do algoritmo NSVR-IAF-PNLMS é obtida como

$$g_i^{\text{NIAF}}(n) = \frac{\gamma_i^{\text{NIAF}}(n)}{\Phi^{\text{NIAF}}(n)} \quad (4.21)$$

com

$$\gamma_i^{\text{NIAF}}(n) = \begin{cases} \frac{1}{N}, & V_i(n) \leq \varepsilon \\ \gamma_i^{\text{IAF}}(n), & \text{caso contrário} \end{cases} \quad (4.22)$$

onde o sobrescrito NIAF representa o algoritmo NSVR-IAF-PNLMS e a função de proporcionalidade  $\gamma_i^{\text{IAF}}(n)$  é computada de acordo com (2.24).

O algoritmo NSVR-IAF-PNLMS é descrito na Tabela 4.5. Pode-se observar que a nova abordagem mantém a filosofia de fatores de ativação individuais que tendem ao valor da magnitude dos coeficientes durante o processo de adaptação preservando, assim, a principal característica do algoritmo IAF-PNLMS, isto é, maior velocidade de convergência para plantas com alto grau de esparsidade. Contudo, o novo algoritmo realiza um ajuste no ganho dos coeficientes que possuem taxa de variação da magnitude pequena assumindo, dessa forma, que o coeficiente alcançou a vizinhança de seu valor ótimo e não há mais necessidade de manter um passo de adaptação elevado para sua atualização. Note que, ao contrário do algoritmo NSVR-PNLMS em que

**Tabela 4.5** - ALGORITMO NSVR-IAF-PNLMS

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_i^{\text{NIAF}}(n) = \max(f_i(n), |w_i(n)|), \quad 1 \leq i \leq N$$

$$Q_i(n) = |w_i(n)| + |w_i(n-1)| + \dots + |w_i(n - \frac{\Delta}{2} + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$P_i(n) = |w_i(n - \frac{\Delta}{2})| + |w_i(n - \frac{\Delta}{2} - 1)| + \dots + |w_i(n - \Delta + 1)|,$$

$$\begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

$$V_i(n) = \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, \quad \begin{cases} 1 \leq i \leq N \text{ e} \\ n = N + m\Delta, \quad m = 1, 2, 3, \dots \end{cases}$$

Se  $V_i(n) \leq \varepsilon$

$$\gamma_i^{\text{NIAF}}(n) = 1/N, \quad 1 \leq i \leq N \text{ e } n \geq N + \Delta$$

$$g_i^{\text{NIAF}}(n) = \frac{\gamma_i^{\text{NIAF}}(n)}{\sum_{l=1}^N \gamma_l^{\text{NIAF}}(n)}, \quad 1 \leq i \leq N$$

$$\mathbf{g}_i^{\text{NIAF}}(n) = [g_1^{\text{NIAF}}(n) \ g_2^{\text{NIAF}}(n) \ \dots \ g_N^{\text{NIAF}}(n)]$$

$$\mathbf{G}^{\text{NIAF}}(n) = \text{diag}[g_1^{\text{NIAF}}(n) \ g_2^{\text{NIAF}}(n) \ \dots \ g_N^{\text{NIAF}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{NIAF}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{NIAF}}(n) \mathbf{x}(n) + \xi}$$

$$f_i(n) = \begin{cases} \frac{1}{2} |w_i(n)| + \frac{1}{2} \gamma_i^{\text{NIAF}}(n-1), & n = mN, \quad m = 1, 2, 3, \dots \\ f_i(n-1), & \text{caso contrário.} \end{cases}$$

apenas os coeficientes ativos que quase atingem a convergência produzem alteração na distribuição de ganho, aqui tanto a convergência dos coeficientes ativos quanto dos inativos impactam os valores dos ganhos distribuídos. Isso ocorre porque o ganho do coeficiente, seja ativo ou inativo, é proporcional à sua magnitude e, após sua convergência, passa a ser igual a  $1/N$ , alterando assim o ganho de todos os demais coeficientes do filtro.

#### 4.7.2 Análise da Distribuição de Ganho do Algoritmo NSVR-IAF-PNLMS

A principal característica da política de distribuição de ganho do algoritmo IAF-PNLMS é, através de um processo recursivo, atribuir ganho proporcional à magnitude dos coeficientes adaptativos (tanto para os ativos quanto os inativos). Assim, para simplificação da análise de distribuição de ganho do algoritmo NSVR-IAF-PNLMS, é assumido aqui que o fator de proporcionalidade dos coeficientes seja igual a sua magnitude, isto é,  $\gamma_i^{\text{IAF}}(n) = |w_i^{\text{IAF}}(n)|$ . Tal consideração é razoável visto que  $\gamma_i^{\text{IAF}}(n)$  é sempre igual a  $|w_i^{\text{IAF}}(n)|$  para coeficientes ativos e, conforme o algoritmo converge,  $\gamma_i^{\text{IAF}}(n)$  tende a  $|w_i^{\text{IAF}}(n)|$  para coeficientes inativos [48].

Dessa forma, analisando o algoritmo NSVR-IAF-PNLMS, as seguintes definições são primeiramente estabelecidas:

- i) **Ganho dos coeficientes que alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  tenha quase atingido a convergência e seu ganho associado é dado por

$$g_c^{\text{NIAF}}(n) = \frac{1}{N \Phi^{\text{NIAF}}(n)}. \quad (4.23)$$

- ii) **Ganho dos coeficientes que ainda não alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) > \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  não atingiu a convergência e seu ganho associado é

$$g_{\text{nc},i}^{\text{NIAF}}(n) = \frac{1}{\Phi^{\text{NP}}(n)} |w_i(n)|. \quad (4.24)$$

Com isso, considerando que o ganho total por iteração atribuído pelo algoritmo NSVR-IAF-PNLMS aos coeficientes do filtro adaptativo é a soma do ganho dos coeficientes que não convergiram com o ganho dos que quase convergiram, tem-se que

$$g_{\text{total}}^{\text{NIAF}} = \sum_{i \in (A+B-D)} \frac{1}{\Phi(n)} |w_i(n)| + \sum_{i \in D} \frac{1}{N \Phi(n)} \quad (4.25)$$

onde  $A$ ,  $B$  e  $D$  são os conjuntos de índices de posições definidos conforme (4.19).

Agora, analisando (4.25), os seguintes pontos são destacados:

- i) No início do processo adaptativo não há coeficientes que alcançaram a vizinhança dos seus valores ótimos, portanto  $N_c(n) = 0$  e, conseqüentemente,  $D$  é um conjunto vazio. Nessa condição, (4.25) gera uma distribuição de ganho em acordo com a distribuição de ganho proposta pelo algoritmo IAF-PNLMS, isto é, ganho proporcional à magnitude do coeficiente (sejam eles ativos ou inativos).
- ii) Conforme os coeficientes se aproximam de seus valores ótimos,  $V_i(n)$  diminui e, assim que  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente quase atingiu a convergência. Como consequência,  $N_c(n)$  aumenta o seu valor e  $D$  deixa de ser um conjunto vazio. Tal fato reduz o número de elementos associados à primeira parcela à direita de (4.25) elevando o número de coeficientes com ganho  $1/N$ . Em uma situação ideal, em regime permanente, todos os coeficientes atingiriam a proximidade de seus valores ótimos e, portanto,  $N_c(n) = N$  e  $(A + B - D) = \emptyset$ , de forma que a distribuição de ganho seja uniforme e o novo algoritmo se comporta como o algoritmo NLMS.
- iii) Sempre que o  $i$ -ésimo coeficiente alcança a vizinhança de seu valor ótimo, o ganho transferido a outros coeficientes é dado por

$$T_i^{\text{NIAF}}(n) = \frac{|w_i(n)|}{\sum_{\substack{j=1 \\ j \neq i}}^N \gamma_j^{\text{NIAF}}(n) + |w_i(n)|} - \frac{1/N}{\sum_{\substack{j=1 \\ j \neq i}}^N \gamma_j^{\text{NIAF}}(n) + \frac{1}{N}} \quad (4.26)$$

onde o primeiro termo do lado direito de (4.26) corresponde ao ganho proporcional à magnitude do  $i$ -ésimo coeficiente e o

segundo termo representa o ganho comum atribuído a coeficientes que tenham quase atingido a convergência. Note que a diferença de ganho  $T_i^{\text{NIAF}}(n)$  pode ser positiva ou negativa, dependendo se o coeficiente é ativo ou inativo. Quando um coeficiente ativo alcança a proximidade de seu valor ótimo, em geral,  $|w_i(n)| > 1/N$  e, como resultado, há um aumento no ganho atribuído aos demais coeficientes. Por outro lado, quando um coeficiente inativo atinge a vizinhança de seu ótimo, em geral,  $|w_i(n)| < 1/N$  e há uma redução do ganho dos outros coeficientes.

## 4.8 Resultados de Simulação do Algoritmo NSVR-IAF-PNLMS

Nesta seção, considerando um problema de identificação de sistemas, simulações de Monte Carlo (média de 100 realizações independentes) dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS são apresentadas. Para tal, cinco exemplos são considerados. Em todos os exemplos são usados, tanto para o algoritmo IAF-PNLMS quanto para o algoritmo proposto, passo de adaptação  $\beta = 0,1$ , fator de ativação individual inicial  $f_i(0) = 10^{-4}$  e constante de regularização  $\xi = 10^{-4}$ . Para o algoritmo NSVR-IAF-PNLMS, o número de amostras  $\Delta$  utilizadas para o cálculo da NSVR é igual a  $N$ .

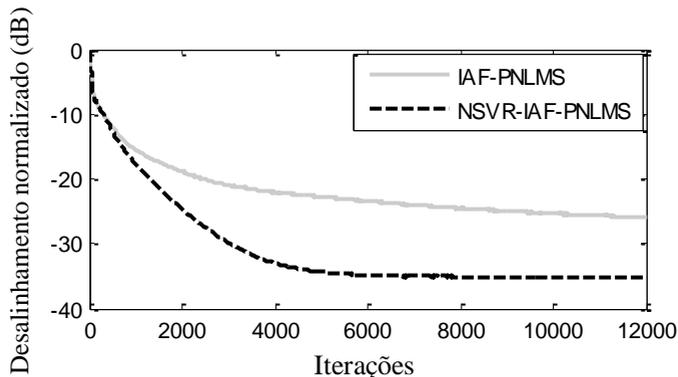
### 4.8.1 Exemplo 4.12

Este exemplo tem por objetivo comparar a velocidade de convergência dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS para planta esparsa. Para tal, usa-se a resposta ao impulso da planta  $\mathbf{w}_A^o$ , dados de entrada branco com variância  $\sigma_x^2$  unitária e variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Para o algoritmo NSVR-IAF-PNLMS, o valor do limiar de convergência é  $\varepsilon = 5 \times 10^{-2}$  e, para o cálculo de  $V_i(n)$ , o número de amostras de  $w_i(n)$  é  $\Delta = N$ . Além disso, o comportamento dos coeficientes ativos e inativos e seus ganhos correspondentes são também apresentados.

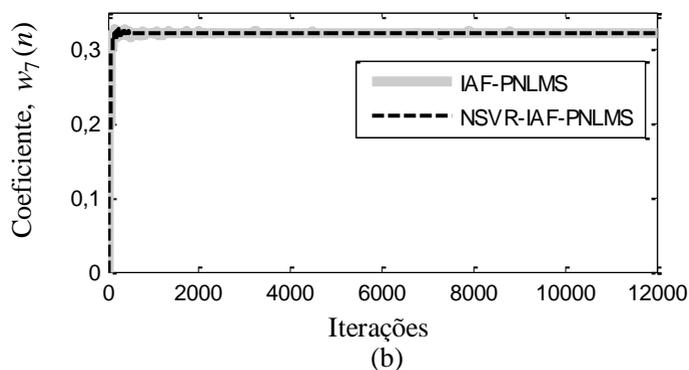
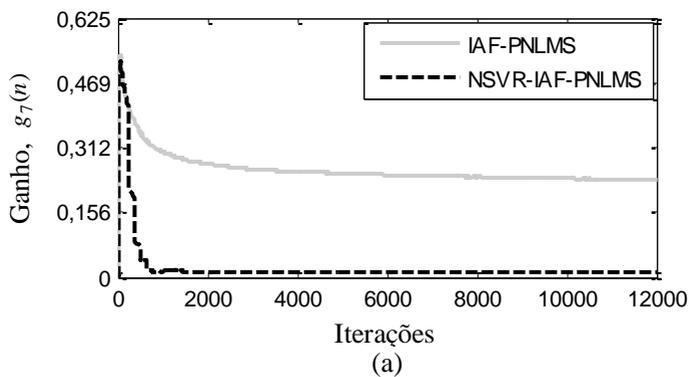
A Figura 4.20 mostra as curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. Dessa figura, observa-se que o algoritmo proposto possui velocidade de convergência inicial igual ao do IAF-PNLMS, sendo que após esse período o novo

algoritmo atinge tanto uma velocidade de convergência maior quanto um menor desalinhamento em regime permanente.

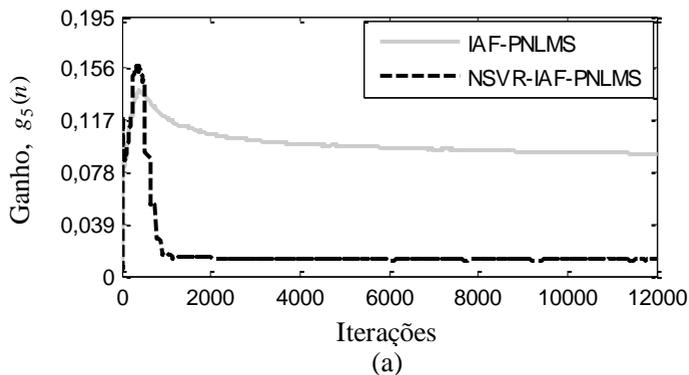
As Figuras 4.21 a 4.23 mostram a evolução dos valores dos coeficientes e o comportamento de seus ganhos correspondentes. De forma semelhante ao realizado anteriormente pelos algoritmos NSVR-PNLMS e NSVR-IPNLMS, o algoritmo NSVR-IAF-PNLMS reduz o ganho atribuído a coeficientes ativos após o coeficiente alcançar a vizinhança de seu valor ótimo e o redistribui aos demais coeficientes [veja Figuras 4.21(a) e 4.22(a)]. Observe que o ganho do coeficiente intermediário atinge valor de pico superior ao do algoritmo IAF-PNLMS antes de sua mudança de política de ganho [veja Figura 4.22(a)] e o ganho dos coeficientes inativos [veja Figura 4.23(a)] aumenta seu valor (à medida que os coeficientes ativos se aproximam de seus valores ótimos) como consequência dessa redistribuição de ganhos. Note que o ganho atribuído aos coeficientes possui maior oscilação do que o apresentado pelos algoritmos NSVR-PNLMS e NSVR-IPNLMS. Essa característica se deve ao comportamento irregular da magnitude de coeficientes inativos, com isso, a cada iteração há uma quantidade variável de coeficientes inativos que o algoritmo assume convergência e, dessa forma, atribui o ganho  $1/N$ , causando mudanças no ganho alocado a todos os demais coeficientes do filtro.

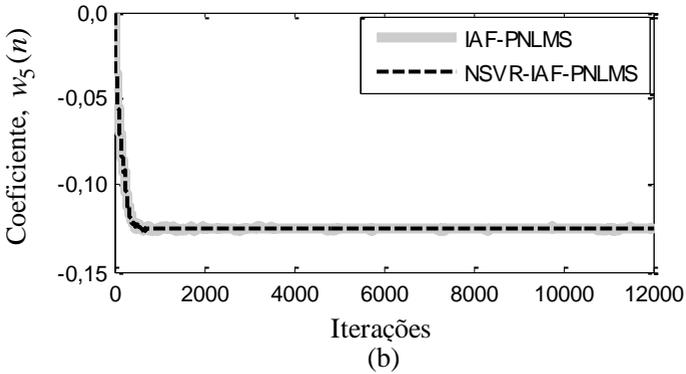


**Figura 4.20** - Exemplo 4.12. Curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ .

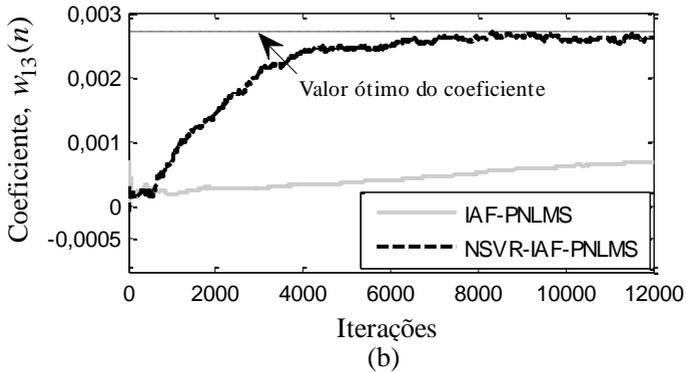
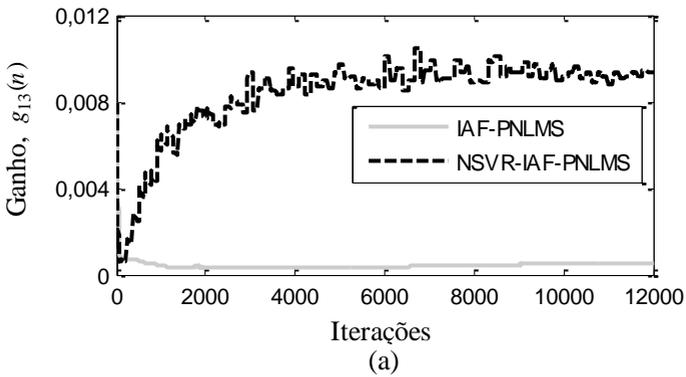


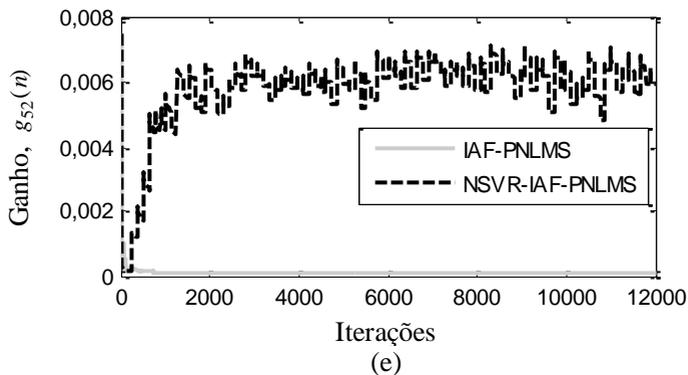
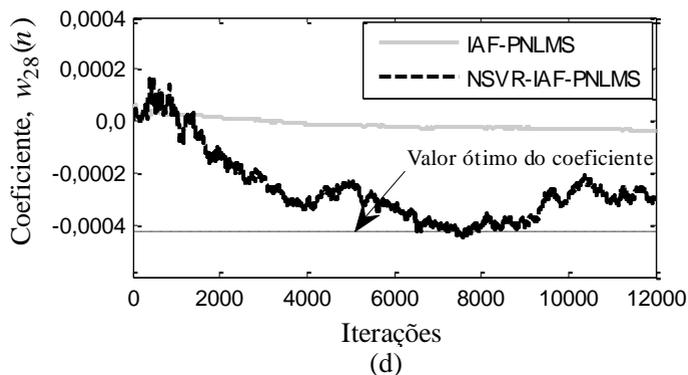
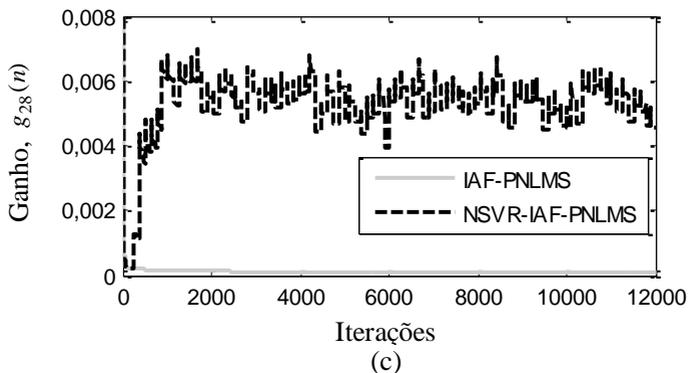
**Figura 4.21** - Exemplo 4.12. Comportamento do coeficiente com maior magnitude da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente.

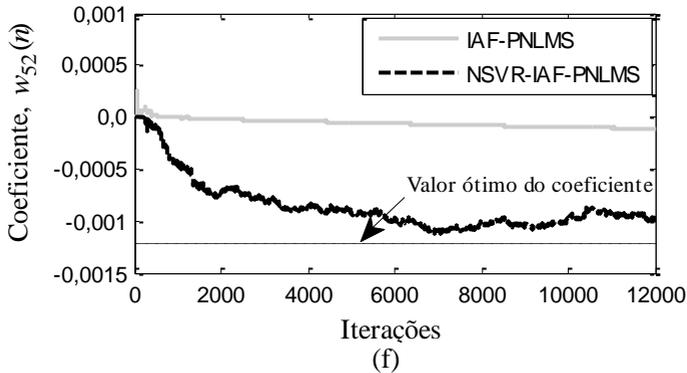




**Figura 4.22** - Exemplo 4.12. Comportamento do coeficiente com magnitude intermediária da resposta ao impulso  $w_A^0$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a) Ganho atribuído. (b) Evolução do coeficiente.







**Figura 4.23** - Exemplo 4.12. Comportamento de coeficientes inativos da resposta ao impulso  $\mathbf{w}_A^o$  para os algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. (a), (c) e (e) Ganho atribuído. (b), (d) e (f) Evolução do coeficiente.

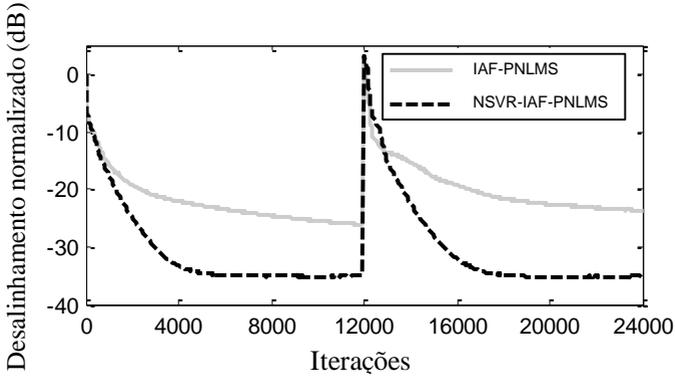
#### 4.8.2 Exemplo 4.13

Neste exemplo, o comportamento do algoritmo NSVR-IAF-PNLMS é avaliado considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário do exemplo anterior é utilizado, contudo, no instante  $n=10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita.

A Figura 4.24 ilustra as curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-IAF-PNLMS. Pode-se observar que o algoritmo proposto supera o IAF-PNLMS mesmo na presença de perturbações abruptas na resposta ao impulso da planta.

#### 4.8.3 Exemplo 4.14

Neste exemplo, a habilidade de rastreamento do algoritmo NSVR-IAF-PNLMS é avaliada. Aqui, a resposta ao impulso da planta variante no tempo  $\mathbf{w}^o$  é a mesma do Exemplo 4.4 (obtida através de um modelo modificado de primeira ordem de Markov), o sinal de entrada é branco e a variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Todos os demais parâmetros são idênticos aos utilizados no Exemplo 4.12.



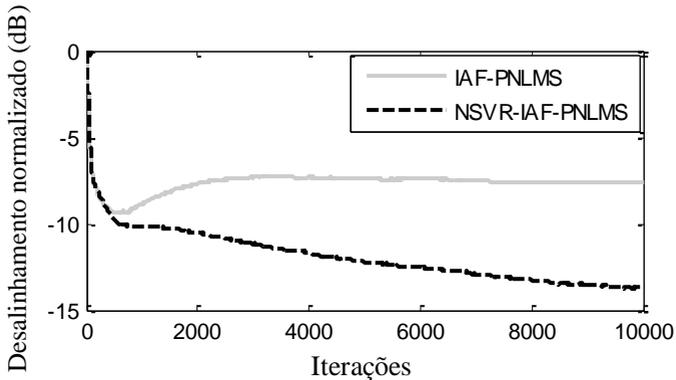
**Figura 4.24** - Exemplo 4.13. Desalinhamento normalizado dos algoritmos IAF-PNLMS e NSVR-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ . Quando  $n = 10000$ , o vetor de coeficientes da planta  $\mathbf{w}_A^o$  é deslocado de 12 amostras para a direita.

#### 4.8.4 Exemplo 4.15

Neste exemplo, o comportamento dos algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS em uma condição de esparsidade média é comparado. Para tal, a resposta ao impulso da planta  $\mathbf{w}_A^o$  [ $S(\mathbf{w}_A^o) = 0,78$ ] é trocada em  $n = 10000$  para  $\mathbf{w}_B^o$  [ $S(\mathbf{w}_B^o) = 0,42$ ]. Aqui, o dado de entrada é branco com variância  $\sigma_x^2 = 1$  e a variância do ruído de medição é  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Para o algoritmo NSVR-IAF-PNLMS, o passo de adaptação é  $\beta = 0,1$ ,  $\varepsilon = 10 \times 10^{-2}$  e  $\Delta = N$ . Para o algoritmo IAF-PNLMS são usados  $\beta = 0,1$  e  $f_i(0) = 0$ . Para o algoritmo EIAF-PNLMS considera-se  $\beta = 0,11$  e  $K$  (parâmetro de limiar desse algoritmo) igual a  $15 \times 10^{-3}$ .

A Figura 4.25 mostra a habilidade de rastreamento do algoritmo NSVR-IAF-PNLMS. Note que o desalinhamento normalizado do algoritmo proposto, em virtude da melhor distribuição de ganhos dentre os coeficientes ativos e inativos, é menor do que o alcançado pelo algoritmo IAF-PNLMS. Assim, o novo algoritmo apresenta uma maior

capacidade de seguir as variações da resposta da planta do que o algoritmo IAF-PNLMS.

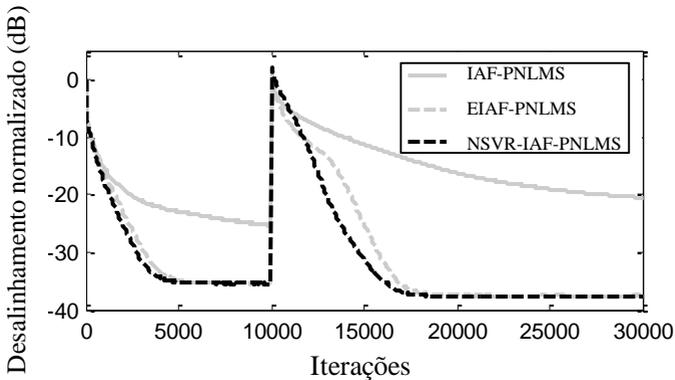


**Figura 4.25** - Exemplo 4.14. Desalinhamento normalizado do algoritmo IAF-PNLMS e NSVR-IAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 5 \times 10^{-2}$ . A planta  $\mathbf{w}^o$  é variante no tempo inicialmente igual  $\mathbf{w}_A^o$  com grau de esparsidade 0,78. Para  $n = 10000$ , o grau de esparsidade da planta é 0,32.

A Figura 4.26 mostra as curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS. Pode-se observar que o algoritmo proposto apresenta velocidade de convergência superior a dos algoritmos IAF-PNLMS e EIAF-PNLMS tanto para planta esparsa quanto dispersa.

#### 4.8.5 Exemplo 4.16

Neste exemplo, o desempenho do algoritmo NSVR-IAF-PNLMS para um sinal de entrada correlacionado é apresentado. Para tal, o sinal de entrada com média zero e variância  $\sigma_x^2$  unitária é obtido através de um processo AR(2) dado por (4.13) utilizando  $b_1 = 0,40$ ,  $b_2 = -0,40$  e  $\sigma_\eta^2 = 0,77$ , resultando em uma dispersão dos autovalores da matriz de autocorrelação do vetor de entrada  $\chi$  igual a 10. O cenário utilizado aqui é o mesmo do Exemplo 4.13, isto é, a resposta ao impulso da planta  $\mathbf{w}_A^o$ , em  $n = 15000$ , é deslocada de 12 amostras para a direita. Para o algoritmo



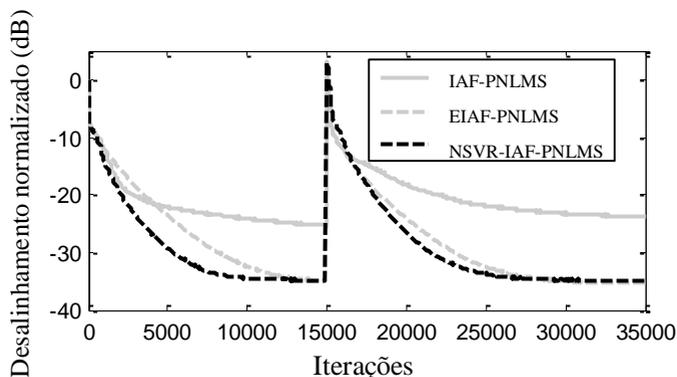
**Figura 4.26** - Exemplo 4.15. Desalinamento normalizado dos algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada branco com variância  $\sigma_x^2$  unitária usando  $\Delta = N$  e  $\varepsilon = 10 \times 10^{-2}$ . A resposta ao impulso da planta  $\mathbf{w}_A^0$  é mudada abruptamente em  $n = 10000$  para  $\mathbf{w}_B^0$ .

NSVR-IAF-PNLMS, o valor do limiar é  $\varepsilon = 5 \times 10^{-2}$  e  $\Delta = N$ . Para os algoritmos IAF-PNLMS e EIAF-PNLMS, os parâmetros são os mesmos considerados no exemplo anterior.

A Figura 4.27 ilustra as curvas de desalinamento normalizado para os algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS quando o sinal de entrada é correlacionado com  $\chi = 10$ . Note que o algoritmo proposto novamente apresenta velocidade de convergência superior a dos algoritmos IAF-PNLMS e EIAF-PNLMS.

## 4.9 Complexidade Computacional

Com respeito à carga computacional dos algoritmos propostos, a computação de (3.12)-(3.14) realizada para determinar  $V_i(n)$  e sua comparação com o limiar  $\varepsilon$  são as principais razões para o aumento da complexidade computacional. Como consequência, as seguintes operações adicionais são requeridas a cada  $\Delta$  iterações:  $N(\Delta - 1)$  adições,  $N$  divisões e  $2N$  comparações. A Tabela 4.6 mostra uma comparação entre os algoritmos PNLMS e NSVR-PNLMS, IPNLMS e NSVR-IPNLMS e IAF-PNLMS e NSVR-IAF-PNLMS. Destaca-se aqui



**Figura 4.27** - Exemplo 4.16. Desalinhamento normalizado para os algoritmos IAF-PNLMS, EIAF-PNLMS e NSVR-IAF-PNLMS considerando sinal de entrada correlacionado com variância  $\sigma_x^2$  unitária e  $\chi=10$  usando  $\Delta=N$  e  $\varepsilon=1 \times 10^{-3}$ . A planta  $\mathbf{w}_A^0$  é deslocada de 12 amostras para a direita em  $n=15000$ .

que as divisões em (2.5) não foram consideradas em todos os algoritmos avaliados (veja Tabela 4.6). Tal resultado é obtido pela multiplicação tanto do numerador quanto do denominador de (2.1) por (2.6). Dessa forma,  $N$  divisões são trocadas por uma multiplicação.

**Tabela 4.6**  
COMPARAÇÃO DA COMPLEXIDADE COMPUTACIONAL  
PARA  $\Delta=N$

Algoritmo	Multiplicações	Adições	Divisões	Comparações
PNLMS	$5N + 3$	$4N - 1$	1	$2N$
NSVR-PNLMS	$5N + 3$	$5N - 2$	2	$2N + 2$
IPNLMS	$6N + 4$	$5N$	1	0
NSVR-IPNLMS	$6N + 3$	$7N - 1$	2	3
IAF-PNLMS	$5N + 3$	$4N$	1	$N$
NSVR-IAF-PNLMS	$5N + 3$	$5N - 1$	2	$N + 2$

### 4.9.1 Redução da Complexidade Computacional

A nova política de distribuição de ganho proporcional reduz o ganho de coeficientes que alcançaram a vizinhança de seus valores ótimos mantendo um ganho mínimo de manutenção. Contudo, para os casos no qual esse ganho mínimo de manutenção é igual ou muito próximo ao ganho já atribuído aos coeficientes inativos, a mudança da política de ganho desses coeficientes (devido à avaliação de convergência) causa nenhum ou muito pouco efeito no comportamento global do algoritmo, visto que o ganho transferido aos demais coeficientes é zero ou próximo de zero. Dessa forma, não há necessidade de computar  $V_i(n)$  para coeficientes inativos assim como alterar a sua distribuição de ganho durante o processo adaptativo. Nesse contexto, a nova política de distribuição de ganho pode ser aplicada tanto a todos os coeficientes do filtro quanto somente aos coeficientes ativos, sem que isso represente perda no desempenho do algoritmo. Com respeito à redução de complexidade computacional para os casos no qual a nova distribuição de ganho seja aplicada apenas a coeficientes ativos, tal estratégia acrescenta  $N$  comparações e reduz as operações de cálculo da NSVR para  $N_{\text{ativos}}(\Delta - 1)$  adições,  $N_{\text{ativos}}$  divisões e  $(N + 2N_{\text{ativos}})$  comparações a cada  $\Delta$  iterações. O acréscimo de  $N$  comparações é decorrente da avaliação da magnitude do coeficiente durante o processo adaptativo a fim de classificá-lo como ativo ou inativo e, com isso, selecionar os coeficientes nos quais a nova política de ganho será aplicada. Dessa forma, para plantas com respostas ao impulso esparsas, a complexidade computacional associada à avaliação de convergência dos coeficientes é substancialmente diminuída, visto que o número de coeficientes ativos é bem menor do que o número total de coeficientes.

Levando em conta os algoritmos propostos, note que a redução de complexidade pode ser considerada nos algoritmos NSVR-PNLMS e NSVR-IPNLMS (veja Seções 4.3.2 e 4.5.2), visto que a transferência de ganho dos coeficientes inativos a outros coeficientes devido à mudança de política de ganho é zero ou muito pequena. Por outro lado, tal consideração não se aplica ao algoritmo NSVR-IAF-PNLMS, uma vez que o ganho atribuído aos coeficientes inativos (proporcional à sua magnitude) pode ser bem diferente de  $1/N$  (veja Seção 4.7.2). Para verificar o comportamento dos algoritmos cuja estratégia de redução de complexidade computacional é utilizada, simulações de MC (média de 100 realizações independentes) são realizadas considerando dois casos, isto é, o primeiro em que a nova política é aplicada a todos os coeficientes

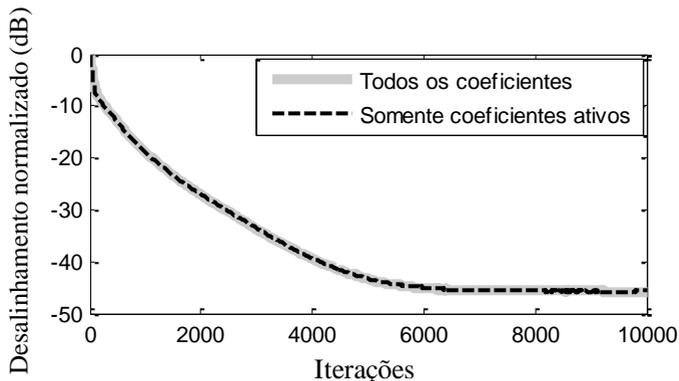
do filtro adaptativo e, o segundo, somente a ativos. Para a realização dessas simulações considera-se a resposta ao impulso da planta  $\mathbf{w}_A^o$ , dados de entrada branco com variância  $\sigma_x^2 = 1$ , variância do ruído de medição  $\sigma_v^2 = 0,0001$  (SNR = 40 dB), passo de adaptação  $\beta = 0,1$ ,  $\Delta = N$  e valor de limiar  $\varepsilon = 1 \times 10^{-2}$ .

A Figura 4.28 ilustra as curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS considerando a aplicação da nova política de distribuição de ganho a todos os coeficientes do filtro adaptativo, assim como somente a coeficientes ativos. Pode-se observar que não há qualquer mudança no comportamento do algoritmo NSVR-PNLMS visto que o ganho atribuído a coeficientes que já atingiram a vizinhança de seus valores ótimos é  $\gamma_{\min}^P(n)$ , isto é, o mesmo já atribuído aos coeficientes inativos e, conseqüentemente, sem alteração de ganho alocado a outros coeficientes.

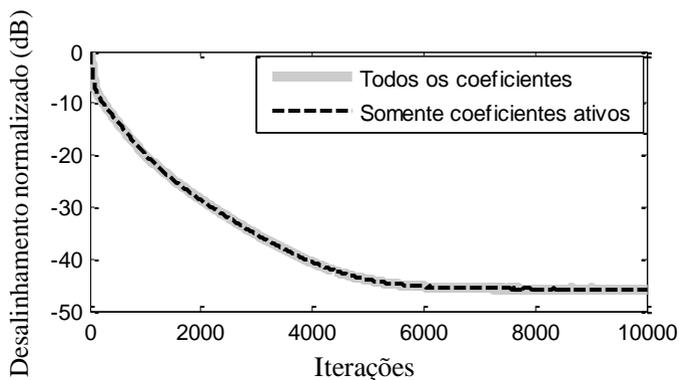
A Figura 4.29 mostra as curvas de desalinhamento normalizado do algoritmo NSVR-IPNLMS utilizando a nova política seja a todos os coeficientes do filtro adaptativo ou somente aos coeficientes ativos. Aqui, são considerados ativos os coeficientes cuja magnitude  $|w_i(n)|$  é maior do que  $|w_{\max}|/100$ . Observe que o comportamento do algoritmo NSVR-IPNLMS praticamente não se altera com a redução de ganho aplicada apenas a coeficientes ativos. Isso se deve ao fato de que, para coeficientes inativos, a parcela comum do ganho (parcela não proporcional) é em geral bem maior do que a parcela proporcional; como resultado, a distribuição de ganho a outros coeficientes que ainda não convergiram é muito pequena e pode ser negligenciada.

A Figura 4.30 compara as curvas de desalinhamento normalizado do algoritmo NSVR-IAF-PNLMS considerando a distribuição de ganho proposta aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos. Em contraste com os algoritmos NSVR-PNLMS e NSVR-IPNLMS, o NSVR-IAF-PNLMS tem seu desempenho prejudicado quando a nova distribuição de ganho é aplicada somente a coeficientes ativos. Tal comportamento é devido ao fato de que sem a mudança do ganho de coeficientes inativos que alcançaram a vizinhança de seus valores ótimos para  $1/N$ , o ganho alocado a esses coeficientes permanece proporcional a  $|w_i(n)|$  durante todo o processo adaptativo; com isso, o algoritmo NSVR-IAF-PNLMS não apresentará

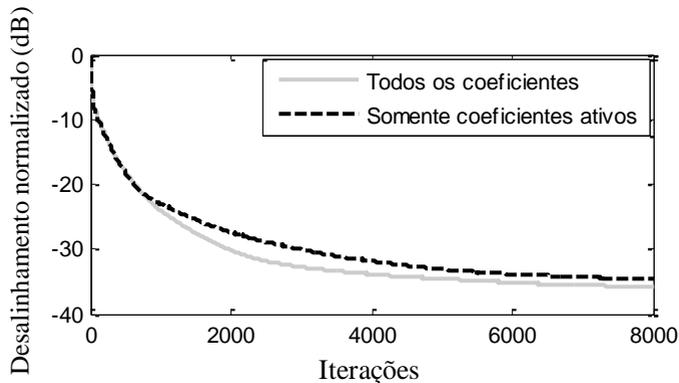
uma distribuição uniforme em regime permanente, o que afeta negativamente o desempenho do algoritmo como pode ser observado na Figura 4.30.



**Figura 4.28** – Curvas de desalinhamento normalizado do algoritmo NSVR-PNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.



**Figura 4.29** - Curvas de desalinhamento normalizado do algoritmo NSVR-IPNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.



**Figura 4.30** - Curvas de desalinhamento normalizado do algoritmo NSVR-IAF-PNLMS considerando a nova política de distribuição de ganho aplicada tanto a todos os coeficientes do filtro adaptativo quanto somente a coeficientes ativos.

#### 4.10 Discussões Sobre a Nova Política de Ganho Proporcional

Aqui, baseado nos resultados de simulações anteriores, algumas considerações são levantadas sobre o limiar de convergência  $\varepsilon$  e o número de amostras  $\Delta$  usados nos algoritmos NSVR-PNLMS, NSVR-IPNLMS e NSVR-IAF-PNLMS. Além disso, uma breve discussão sobre aspectos filosóficos da nova abordagem é apresentada.

##### A. Parâmetro de Limiar $\varepsilon$

Este parâmetro é usado como um indicador de convergência do coeficiente adaptativo e define o ponto no qual o algoritmo muda a política de ganho do coeficiente, tendo um importante papel no desempenho dos algoritmos propostos. A determinação do valor do limiar depende das características da planta, do sinal de entrada, assim como da potência do ruído de medição presente no sistema. Na prática, a escolha de  $\varepsilon$  deve ser realizada de forma a garantir que o ganho dos coeficientes mais significativos mude durante o processo adaptativo sempre que esses coeficientes alcançarem a vizinhança de seus valores ótimos. Por outro lado, valores de  $\varepsilon$  que mudam a política de ganho de coeficientes que ainda não atingiram as proximidades de seus valores ótimos prejudicam o desempenho do algoritmo. No entanto, salienta-se que a escolha do valor de  $\varepsilon$  não é crítica visto que tal valor pode ser escolhido dentro de

uma faixa de valores, de acordo com o cenário de aplicação, para uma adequada operação do algoritmo conforme mostrado no Exemplo 4.1.

### B. Número de Amostras $\Delta$

A variável  $\Delta$  define o número de amostras de  $w_i(n)$  usada para computar a NSVR para os algoritmos propostos. O valor de  $\Delta$  deve ser escolhido de tal forma que se “capture” o comportamento da convergência individual do coeficiente durante o processo adaptativo. Note que grandes valores de  $\Delta$  implicam uma super filtragem da magnitude do coeficiente, resultando em perda de importante informação sobre o comportamento da convergência do coeficiente. Por outro lado, pequenos valores de  $\Delta$  conduzem a uma sub filtragem da magnitude do coeficiente originando forte flutuação em  $V_i(n)$ , o que impede a avaliação correta do processo de convergência. Assim, destaca-se aqui que há um compromisso na escolha do valor de  $\Delta$  para realizar uma mudança adequada da política de distribuição de ganho. Simulações numéricas têm mostrado que  $\Delta$  igual a  $N/4$ ,  $N/2$  e  $N$  resultam em um adequado desempenho do algoritmo (veja Exemplo 4.1).

### C. Filosofia da Política de Ganho Proporcional Proposta

Em contraste com os demais algoritmos da classe proporcional que utilizam uma política de distribuição de ganho global, a nova abordagem usa uma política individual e independente dos demais coeficientes do filtro adaptativo. Além disso, note que misturar ou trocar a política de ganho do algoritmo é uma prática comum em algoritmos do tipo proporcional. Por exemplo, o algoritmo IPNLMS mistura ganho proporcional e não proporcional visando melhorar as características de convergência para uma larga faixa de esparsidade da planta. O algoritmo EIAF-PNLMS usa uma versão suavizada do erro instantâneo quadrático  $e^2(n)$  como uma medida de proximidade do regime permanente e, com isso, muda sua política de distribuição de ganho para uniforme. Os algoritmos *water-filling*, *water-filling* colorido (CWF) e PDQMC combinam o ganho proporcional com um ganho uniforme usando uma regra específica de mistura para obter a distribuição de ganho do algoritmo. Contudo, todos esses algoritmos misturam ou trocam a política de ganho para todos os coeficientes do filtro adaptativo ao mesmo tempo, ao passo que a abordagem proposta usa uma avaliação da convergência

do coeficiente para definir uma política de ganho que pode ser trocada a qualquer instante durante o processo adaptativo independentemente da política dos demais coeficientes.

#### **4.11 Conclusões**

Neste capítulo, uma nova política de distribuição de ganho para algoritmos tipo proporcional baseada na taxa de variação da magnitude dos coeficientes foi discutida. A nova estratégia reduz o ganho alocado a coeficientes ativos que já alcançaram a vizinhança de seus valores ótimos, isto é, que quase atingiram a convergência, e o redistribui dentre os coeficientes que ainda não convergiram. Para tal, a taxa de variação suavizada e normalizada (NSVR) da magnitude dos coeficientes é utilizada como uma métrica de avaliação individual de convergência. Dessa forma, considerando a política de ganho proposta, três versões dos algoritmos PNLMS, IPNLMS e IAF-PNLMS, denominadas respectivamente NSVR-PNLMS, NSVR-IPNLMS e NSVR-IAF-PNLMS, foram introduzidas. Os novos algoritmos apresentam uma melhor distribuição de ganho do que os algoritmos originais, resultando em melhores características de convergência. Para diferentes cenários de operação, resultados de simulações numéricas atestaram a eficácia da política de distribuição de ganho proposta.



## ALGORITMO PROPORCIONAL AO DESVIO QUADRÁTICO MÉDIO DOS COEFICIENTES BASEADO NA AUTOCORRELAÇÃO DO ERRO

### 5.1 Introdução

O algoritmo *water-filling*, obtido a partir da minimização do erro quadrático médio a cada iteração para sinal de entrada branco, apresenta uma elevada carga computacional conforme mostrado na Seção 2.9. Para contornar tal problema, o algoritmo proporcional ao desvio quadrático médio dos coeficientes (PDQMC) é introduzido em [33]. Em contraste com outros algoritmos da classe proporcional que usam a magnitude dos coeficientes para computar seus ganhos, o PDQMC usa uma estimativa do desvio quadrático médio dos coeficientes  $E[z_i^2(n)]$  combinado com o ganho unitário. Essa combinação é realizada através de uma regra proposta pelos autores, denominada combinação de ganho convexa adaptativa [33], [69]. Tal regra é utilizada para se obter o mesmo nível de desalinhamento do algoritmo NLMS em regime permanente. Como se pode ver na Tabela 2.7, a matriz diagonal de distribuição de ganho  $\mathbf{G}(n)$  do algoritmo PDQMC é formada pelos elementos  $g_{c_i}(n)$ , os quais são calculados como

$$g_{c_i}(n) = [1 - \zeta(n)]g_i(n) + \zeta(n)1, \quad i = 1, 2, \dots, N \quad (5.1)$$

com

$$g_i(n) = \frac{\{E[z_i(n)]_{\text{est}}\}^2}{\frac{1}{N} \sum_{j=1}^N \{E[z_j(n)]_{\text{est}}\}^2} \quad (5.2)$$

e

$$\zeta(n) = \min \left[ 1, \frac{\omega \sigma_v^2}{\sigma_x^2 \sum_{j=1}^N \{E[z_j(n)]_{\text{est}}\}^2 + \sigma_v^2} \right] \quad (5.3)$$

onde  $\omega \geq 0$  é um parâmetro que deve ser ajustado para uma operação adequada do algoritmo, e  $\sigma_x^2$  e  $\sigma_v^2$  são as variâncias do sinal de entrada e do ruído de medição, respectivamente.

Observe que a mistura dos ganhos em (5.1) é controlada pelo parâmetro  $\zeta(n)$  dado em (5.3) tal que a política de distribuição de ganho se torne uniforme em regime permanente. A migração do ganho proporcional a  $E[z_i^2(n)]$  para o ganho uniforme é realizada devido ao fato que a estimativa do desvio quadrático médio dos coeficientes  $E[z_i^2(n)]_{\text{est}}$  [obtida a partir do sinal de erro] se degrada à medida que o regime permanente se aproxima. Tal degradação na estimativa de  $E[z_i^2(n)]$  prejudica o desempenho do algoritmo PDQMC e ocorre porque, à medida que  $\mathbf{w}(n) \rightarrow \mathbf{w}^o$ , o ruído de medição se torna predominante no sinal de erro. Contudo, para realizar o cálculo de  $\zeta(n)$  é necessário conhecer o valor da variância do ruído de medição  $\sigma_v^2$  presente no sistema, o qual geralmente não está prontamente disponível em aplicações práticas. Com isso, uma estimativa dessa variância se faz indispensável.

Assim, visando reduzir a dependência do conhecimento da variância do ruído de medição  $\sigma_v^2$ , uma nova estratégia para migrar do ganho proporcional a  $E[z_i^2(n)]$  para o ganho unitário é proposta. Tal estratégia é baseada na avaliação da autocorrelação do sinal de erro entre amostras adjacentes.

## 5.2 Impacto da Estimativa da Variância do Ruído de Medição e do Parâmetro $\omega$ sobre o Comportamento do Algoritmo PDQMC

Como mostrado em [33], o algoritmo PDQMC tem melhor desempenho do que os algoritmos PNLMS e NLMS para sinal de entrada branco. Entretanto, a determinação de  $\zeta(n)$  requer o conhecimento ou

estimativa da variância do ruído de medição  $\sigma_v^2$ , o que representa uma desvantagem em relação aos demais algoritmos.

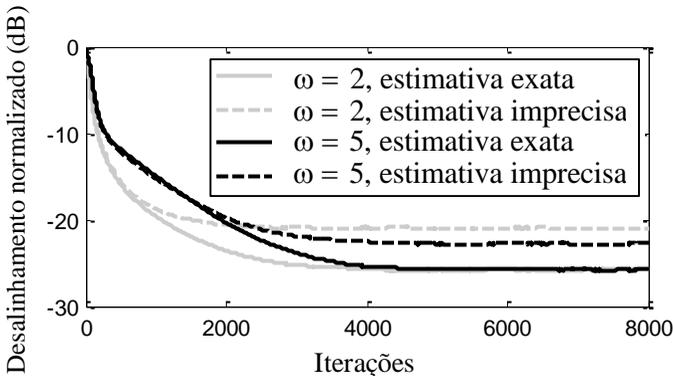
Outro ponto a ser considerado é a sensibilidade do algoritmo PDQMC com respeito a erros de estimativa da variância do ruído de medição. Para verificar tal questão, simulações de MC (média de 100 realizações independentes) são realizadas considerando um problema de identificação de sistemas e passo de adaptação  $\beta=0,1$ . Para isso, a resposta ao impulso da planta  $\mathbf{w}_A^0$  [ $S(\mathbf{w}_A^0)=0,78$ ] é considerada.

A Figura 5.1 ilustra a relação entre o parâmetro  $\omega$  e o efeito do erro de estimativa da variância do ruído de medição no comportamento do algoritmo PDQMC considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária e variância do ruído de medição  $\sigma_v^2=17$  dB. Para tal, curvas de desalinhamento normalizado são mostradas para dois valores distintos de  $\omega$  considerando uma estimativa exata e uma imprecisa de  $\sigma_v^2$ . Em relação à estimativa imprecisa de  $\sigma_v^2$ , o valor estimado utilizado no algoritmo PDQMC é de 20 dB, isto é, uma diferença de +3 dB. Pode-se observar que  $\omega=2$  resulta em maior velocidade de convergência do que  $\omega=5$ , contudo  $\omega=2$  apresenta pior desempenho do que  $\omega=5$  considerando o mesmo erro na estimativa da variância do ruído de medição  $\sigma_v^2$ . Em outras palavras, reduzir o parâmetro  $\omega$  aumenta a velocidade de convergência do algoritmo PDQMC, mas o deixa mais sensível a erros de estimação de  $\sigma_v^2$ . Portanto, existe um compromisso entre velocidade de convergência e sensibilidade a erros de estimativa da variância do ruído de medição. Salienta-se que, conforme mostrado a seguir, valores de  $\omega$  menores do que 2 não são aconselháveis neste caso, pois podem não garantir que  $\zeta(n)$  seja igual a 1 em regime permanente, conduzindo a um maior desalinhamento em regime permanente se comparado com outros algoritmos, tais como NLMS e PNLMS.

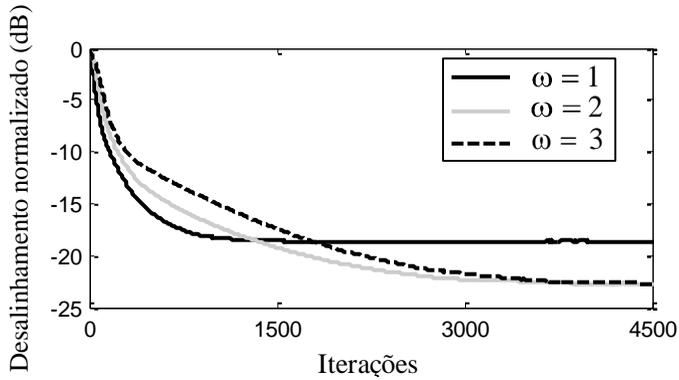
Note que o parâmetro  $\omega$  tem um importante papel no desempenho do algoritmo PDQMC, sendo que a definição do valor de tal parâmetro não é uma tarefa trivial, visto que  $\omega$  depende da precisão da estimativa da variância do ruído de medição, das características da resposta ao impulso da planta, do tipo de sinal de entrada aplicado ao sistema, dentre outros. Para avaliar esse ponto, simulações de MC (média de 100

realizações independentes) são realizadas para três valores distintos de  $\omega$  considerando o mesmo cenário da Figura 5.1, porém agora a estimativa da variância do ruído de medição é perfeita, isto é,  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ . A Figura 5.2 mostra as curvas de desalinhamento normalizado do algoritmo PDQMC para os três valores de  $\omega$  utilizados. Observe que  $\omega=1$  resulta em desalinhamento em regime permanente muito acima dos outros dois valores de  $\omega$  avaliados. Para  $\omega=3$ , a redução de velocidade de convergência é evidente. Com isso, o melhor ajuste deste parâmetro, para esse cenário, é  $\omega=2$ . Contudo, qualquer alteração do sistema no qual o algoritmo está inserido, um novo ajuste desse parâmetro é necessário.

Assim, a necessidade de conhecimento da variância do ruído de medição, sensibilidade do algoritmo com respeito a erros de estimativa desta variância e, por fim, dificuldade de ajuste do parâmetro  $\omega$ , prejudicam o uso do algoritmo PDQMC em aplicações práticas. Na próxima seção, uma nova abordagem para contornar tais desvantagens é proposta.



**Figura 5.1** - Curvas de desalinhamento normalizado do algoritmo PDQMC considerando um sinal de entrada branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 17$  dB.



**Figura 5.2** – Curvas de desalinhamento normalizado do algoritmo PDQMC considerando três valores do parâmetro  $\omega$  para sinal de entrada branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 17$  dB. A estimativa da variância do ruído de medição  $\hat{\sigma}_v^2$  é perfeita, isto é,  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ .

### 5.3 Abordagem Proposta

Para estimar o desvio quadrático médio dos coeficientes  $E[z_i^2(n)]$  o algoritmo PDQMC (veja Tabela 2.7) utiliza o sinal de erro obtido por

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n) + v(n) \quad (5.4)$$

com sinal desejado  $d(n)$  dado por

$$d(n) = \mathbf{x}^T(n) \mathbf{w}^o. \quad (5.5)$$

Substituindo (5.5) em (5.4), o sinal de erro pode ser reescrito como

$$e(n) = \mathbf{x}^T(n) [\mathbf{w}^o - \mathbf{w}(n)] + v(n). \quad (5.6)$$

Note que o sinal de erro é formado por uma parcela do erro devido ao processo adaptativo relacionada diretamente com o desvio dos coeficientes [primeira parcela à direita em (5.6)] e pelo ruído de medição [segunda parcela à direita em (5.6)]. Contudo, assumindo convergência, conforme a adaptação do filtro evolui e  $\mathbf{w}(n) \rightarrow \mathbf{w}^o$ , há uma redução da parcela do erro vinculada ao desvio dos coeficientes. Assim, o sinal de

erro tende a ser predominantemente formado pelo ruído de medição impedindo uma estimativa razoável de  $E[z_i^2(n)]$ . Dessa forma, à medida que o regime permanente se aproxima, a mudança do ganho proporcional a  $E[z_i^2(n)]$  para o ganho uniforme se faz necessária. No caso do algoritmo PDQMC, a estratégia usada para realizar a mudança de ganho durante o processo adaptativo é dependente do conhecimento do valor da variância do ruído de medição  $\sigma_v^2$  presente no sistema [veja (5.1)-(5.3)], o que obriga uma estimação de  $\sigma_v^2$ . Assim, visando eliminar as dificuldades na utilização prática do algoritmo PDQMC, causadas pela dependência do conhecimento da variância do ruído de medição presente no sistema, um novo algoritmo é aqui proposto.

### 5.3.1 Autocorrelação do Erro

Considerando o desvio dos coeficientes  $\mathbf{z}(n) = \mathbf{w}^o - \mathbf{w}(n)$ , (5.6) pode ser reescrita como

$$e(n) = \mathbf{x}^T(n)\mathbf{z}(n) + v(n). \quad (5.7)$$

com  $v(n)$  sendo um ruído de medição independente e identicamente distribuído (i.i.d.) de média zero e descorrelacionado com qualquer outro sinal no sistema. Agora, definindo  $c(n)$  como uma versão suavizada da autocorrelação entre amostras adjacentes do sinal de erro  $e(n)$  e  $e(n-1)$  [70], obtém-se

$$c(n) = \gamma c(n-1) + (1-\gamma)e(n)e(n-1) \quad (5.8)$$

onde  $0 < \gamma < 1$  denota um parâmetro de suavização.

Assumindo convergência, isto é,  $\mathbf{w}(k) \rightarrow \mathbf{w}^o$ ,  $e(n)$  tende ao ruído de medição  $v(n)$  [veja (5.6)] e, portanto,  $c(n)$  tende para zero. Assim, considerando o comportamento de  $e(n)$  e  $c(n)$ , uma nova estratégia para migrar do ganho proporcional para o ganho unitário à medida que o regime permanente se aproxima é mostrada a seguir.

### 5.3.2 Algoritmo Proposto

Em contraste com o algoritmo PDQMC, que faz uma migração gradual do ganho proporcional a  $E[z_i^2(n)]$ , a nova abordagem realiza

uma migração abrupta dos ganhos. Essa migração abrupta de ganho, realizada através da avaliação da autocorrelação do sinal de erro, não necessita do conhecimento da variância do ruído de medição. Para tal, uma nova estratégia para determinar o instante  $n$  no qual o algoritmo deve migrar do ganho proporcional ao desvio quadrático médio para o ganho uniforme é discutida. A nova estratégia utiliza uma versão normalizada e suavizada da autocorrelação quadrática do erro dada por

$$m(n) = \gamma m(n-1) + (1-\gamma) \frac{c^2(n)}{q^2(n)} \quad (5.9)$$

com

$$q(n) = \gamma q(n-1) + (1-\gamma)e^2(n) \quad (5.10)$$

onde  $q(n)$  é uma versão suavizada do erro quadrático e  $c(n)$  é obtida de (5.8). Então, comparando  $m(n)$  com um valor de limiar predefinido  $0 < T_{lo} < 1$ , é estabelecida uma condição de mudança de ganho dada por

$$m(n) \leq T_{lo}. \quad (5.11)$$

Dessa forma, sempre que tal condição ocorrer em  $N$  iterações consecutivas, um ganho unitário é atribuído a todos os coeficientes. Esse período de aprendizado  $N$  evita uma troca prematura da política de ganho devido a flutuações de  $m(n)$ . Uma vez que o ganho é modificado, o retorno para o ganho proporcional a  $E[z_i^2(n)]$  ocorre se, e somente se,  $m(n) \geq T_{hi}$ , assumindo aqui  $T_{hi} \geq T_{lo}$ . Note que  $T_{hi}$  é usado para criar uma simples histerese evitando, assim, que pequenas flutuações de  $m(n)$  levem o algoritmo inapropriadamente a atribuir ganhos proporcionais a  $E[z_i^2(n)]$ .

Para melhor compreensão da operação do algoritmo proposto, uma breve discussão do comportamento de (5.9) e sua inter-relação com os limiares  $T_{lo}$  e  $T_{hi}$ , os quais definem a política de ganho durante o processo adaptativo, é apresentada a seguir. Em geral, no início da convergência,  $e(n)$  é correlacionado com  $e(n-1)$ ; então, de (5.8) e (5.10), nota-se que  $c^2(n) \cong q^2(n)$  e, portanto,  $c^2(n)/q^2(n) \cong 1$ , resultando em  $m(n) > T_{lo}$ . Para essa condição, os ganhos do algoritmo são proporcionais a  $E[z_i^2(n)]$ . Conforme o regime permanente se aproxima,

isto é,  $\mathbf{w}(n) \rightarrow \mathbf{w}^o$ , verifica-se, a partir de (5.6), que  $e(n)$  tende para  $v(n)$ , forçando  $c(n)$  e, conseqüentemente,  $m(n)$  a tenderem para zero. Quando  $m(n) \leq T_{lo}$ , a política de ganho do algoritmo é alterada para uma distribuição uniforme, isto é, 1 para todos os coeficientes. Agora, se uma perturbação ocorre fazendo a autocorrelação do erro  $c(n)$  aumentar, como consequência,  $m(n)$  também aumenta. Assim que  $m(n) \geq T_{hi}$ , o algoritmo volta a atribuir ganhos proporcionais a  $E[z_i^2(n)]$ , seguindo o processo.

Note que a filosofia do algoritmo proposto para realizar a mudança de ganho é conceitualmente diferente da usada pelo PDQMC. Observe que o PDQMC usa uma mistura gradual (combinação convexa adaptativa) baseada na variância do erro de medição, ao passo que o algoritmo proposto considera uma política de distribuição de ganho (mudança abrupta) baseada em uma versão suavizada e normalizada da autocorrelação entre amostras adjacentes do sinal de erro. Além disso, como a nova estratégia não usa qualquer informação da variância do ruído de medição, o algoritmo proposto pode ser aplicado em ambientes no qual tal variância seja desconhecida, pobremente estimada ou mesmo em casos de ruído de medição não estacionário.

### 5.3.3 Influência da Variância do Ruído de Medição sobre $m(n)$

A variável  $m(n)$  desempenha um papel fundamental na operação do algoritmo proposto, visto que a mudança da distribuição de ganho é realizada pela comparação de  $m(n)$  com o limiar  $T_{lo}$  [veja (5.11)]. Assim, o impacto do ruído de medição presente no sistema sobre a variável  $m(n)$  é aqui discutido. Tal discussão visa mostrar que o comportamento de  $m(n)$  sofre muito pouca influência do valor da variância do ruído de medição  $\sigma_v^2$ . Levando em conta que  $m(n)$  é uma versão suavizada de  $c^2(n)/q^2(n)$  [veja (5.9)], a análise do impacto de  $\sigma_v^2$  sobre  $m(n)$  é realizada considerando apenas a expressão de  $E[c^2(n)/q^2(n)]$ . Então, conforme mostrado no Apêndice II, o valor esperado da autocorrelação quadrática do sinal de erro normalizada é dado por [71]

$$\begin{aligned}
 E\left[\frac{c^2(n)}{q^2(n)}\right] &\cong \\
 &\frac{\sum_{i=0}^{n-1} \gamma^{2i} \{\text{tr}[\mathbf{R}\mathbf{K}(n-i-1) + \mathbf{R}\mathbf{K}(n)]\} + \dots}{\sum_{i=0}^{n-1} \gamma^{2i} 6\text{tr}[\mathbf{R}\mathbf{K}(n-i)] + \dots} \\
 &\frac{\sum_{i=1}^{n-1} \gamma^{2i-1} \text{tr}[\mathbf{R}_2 \mathbf{K}_2^T(n-i+1)] + \sum_{i=0}^{n-2} \gamma^{2i+1} \text{tr}[\mathbf{R}_2^T \mathbf{K}_2(n-i)]}{\sum_{\substack{i=0 \\ i \neq j}}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \text{tr}[\mathbf{R}\mathbf{K}(n-i) + \mathbf{R}\mathbf{K}(n-j)]}
 \end{aligned} \tag{5.12}$$

com  $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ ,  $\mathbf{R}_2 = E[\mathbf{x}(n)\mathbf{x}^T(n-2)]$ ,  $\mathbf{K}(n) = E[\mathbf{z}(n)\mathbf{z}^T(n)]$ , e  $\mathbf{K}_2(n) = E[\mathbf{z}(n)\mathbf{z}^T(n-2)]$ .

Note de (5.12) a independência de  $E[c^2(n)/q^2(n)]$  com respeito a  $\sigma_v^2$ . Esse resultado é obtido assumindo algumas aproximações; de fato, uma pequena dependência de  $\sigma_v^2$  existe, a qual na prática pode ser negligenciada; como consequência, a variável  $m(n)$  é fracamente dependente da variância do ruído de medição. Visto que  $T_{lo}$  e  $T_{hi}$  são utilizados juntamente com  $m(n)$  para definir o ponto no qual a política de distribuição de ganho deve ser alterada, o conhecimento *a priori* da variância do ruído de medição  $\sigma_v^2$  não é essencial para a determinação dos valores de limiar  $T_{lo}$  e  $T_{hi}$ .

### 5.3.4 Complexidade Computacional

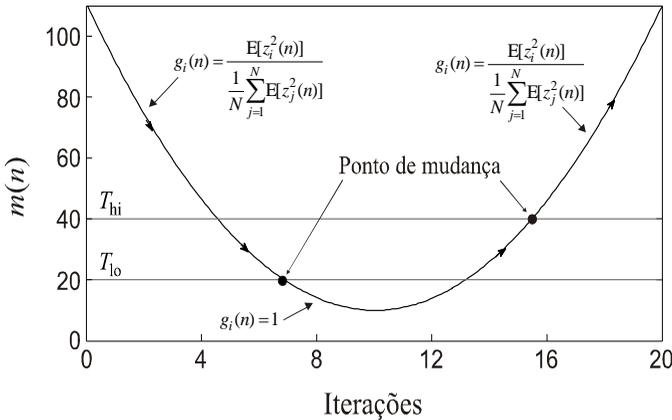
Em relação à carga computacional, a Tabela 5.1 mostra que o algoritmo proposto requer aproximadamente  $N$  adições e  $N$  multiplicações a menos, e duas comparações a mais do que o PDQMC.

**Tabela 5.1** - COMPARAÇÃO DE COMPLEXIDADE COMPUTACIONAL

Algoritmo	Multiplicações	Adições	Divisões	Comparações
PDQMC	$8N + 2$	$6N + 1$	2	1
Proposto	$7N + 8$	$5N + 6$	2	3

**5.4 Distribuição de ganho e Escolha dos Valores de  $T_{lo}$  e  $T_{hi}$  do Algoritmo Proposto**

A distribuição de ganho do algoritmo PDQMC é resultante da mistura do ganho proporcional a  $E[z_i^2(n)]$  com o ganho unitário. O novo algoritmo, em contraste com o PDQMC, leva a uma distribuição proporcional a  $E[z_i^2(n)]$  ou uniforme. A definição de qual é a política de ganho utilizada depende da inter-relação de  $m(n)$  com  $T_{lo}$  e  $T_{hi}$ . A Figura 5.3 ilustra como essa mudança de ganhos no novo algoritmo é efetuada.



**Figura 5.3** - Princípio da distribuição de ganho do novo algoritmo resultante da inter-relação de  $m(n)$  com  $T_{lo}$  e  $T_{hi}$ .

Assumindo aqui ruído de medição não correlacionado, a variável  $m(n)$  tende a um valor muito pequeno à medida que o regime permanente se aproxima e, devido à normalização por  $q^2(n)$ , tal variável sofre muito pouca influência do ruído de medição presente no sistema. Como consequência, levando em conta que os limiares  $T_{lo}$  e  $T_{hi}$  são usados

juntamente com a variável  $m(n)$  para definir a política de ganho, a escolha do valor dos limiares torna-se fracamente dependente da variância do ruído de medição. Dessa forma, o limiar  $T_{lo}$  pode ser escolhido dentro de uma faixa de valores sem praticamente afetar o desempenho do algoritmo (veja Exemplo 5.1). Em relação à escolha de  $T_{hi}$ , o qual evita que pequenas flutuações em  $m(n)$  possam afetar o comportamento do algoritmo, a regra básica é selecionar esse limiar como  $4T_{lo} \geq T_{hi} \geq 2T_{lo}$ .

### 5.5 Resultados de Simulação

Nesta seção, considerando um problema de identificação de sistemas, simulações de MC (média de 100 realizações independentes) dos algoritmos PDQMC e proposto são realizadas. Os objetivos dessas simulações são os seguintes:

- i) Avaliar o impacto do valor dos limiares  $T_{lo}$  e  $T_{hi}$  no algoritmo proposto.
- ii) Comparar os algoritmos em termos de velocidade de convergência, resposta a perturbações na planta e habilidade de rastreamento.
- iii) Comparar o comportamento dos algoritmos com plantas de alta e média esparsidades.
- iv) Avaliar a robustez do novo algoritmo frente a diferentes valores da variância do ruído de medição.
- v) Avaliar o desempenho do algoritmo proposto considerando ambientes com ruído de medição não estacionário.

Para isso, seis exemplos são apresentados. Em todos os exemplos são considerados passo de adaptação  $\beta = 0,1$  e constante de regularização  $\xi = 10^{-4}$ . O fator de suavização para o algoritmo PDQMC é  $\alpha = 0,99$  e, para o algoritmo proposto,  $\gamma = 0,99$ .

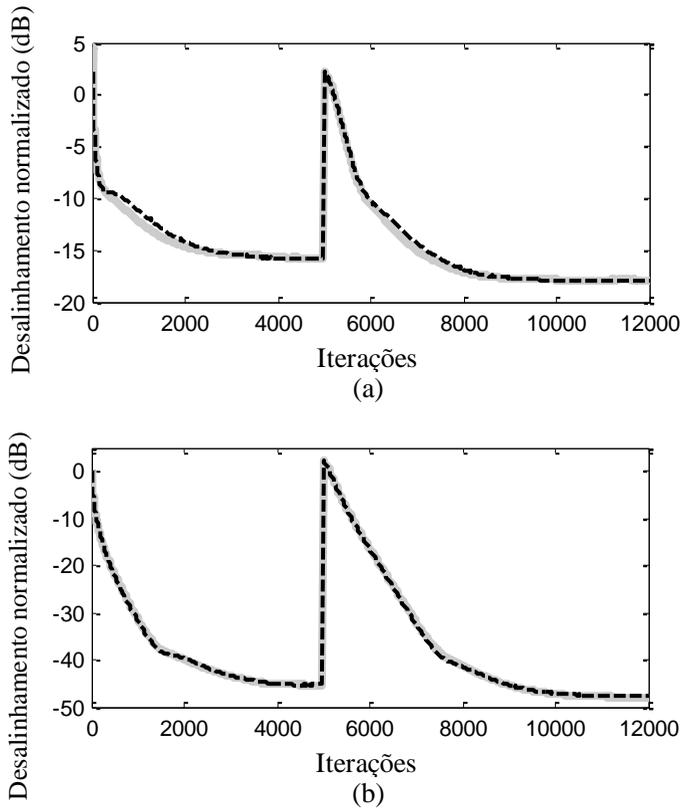
#### 5.5.1 Exemplo 5.1

Neste exemplo, o comportamento do algoritmo proposto é avaliado com vistas aos limiares  $T_{hi}$  e  $T_{lo}$ , considerando dois valores de SNR: 10 e 40 dB. Para tal, um sinal de entrada branco com variância

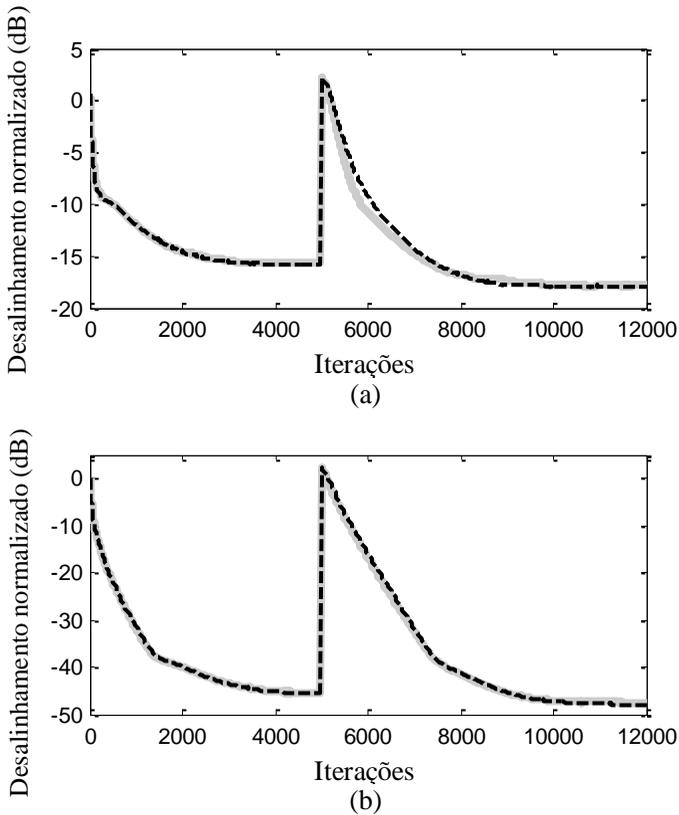
$\sigma_x^2 = 1$  é usado. A Figura 5.4 mostra as curvas de desalinhamento normalizado para  $T_{hi} = 3,5 \times 10^{-2}$  e dois valores de  $T_{lo}$ , isto é,  $1,2 \times 10^{-2}$  e  $0,6 \times 10^{-2}$ . A Figura 5.5 descreve o desalinhamento normalizado usando agora  $T_{hi}$  igual a  $3 \times 10^{-2}$  e  $4,5 \times 10^{-2}$ , considerando  $T_{lo} = 0,9 \times 10^{-2}$ . Ambas simulações iniciam com resposta ao impulso de planta  $\mathbf{w}_A^0$  e, na iteração  $n = 5000$  uma mudança para  $\mathbf{w}_B^0$  ocorre. Observe que uma faixa típica para os limiares, considerando 10 e 40 dB de SNR, são  $0,6 \times 10^{-2} \leq T_{lo} \leq 1,2 \times 10^{-2}$  e  $3 \times 10^{-2} \leq T_{hi} \leq 4,5 \times 10^{-2}$ . Note que a seleção dos limiares  $T_{hi}$  e  $T_{lo}$  não é um ponto crítico visto que eles podem ser escolhidos dentro de uma faixa de valores sem praticamente alterar o comportamento do algoritmo, independente tanto do nível do ruído de medição quanto do grau de esparsidade da planta.

### 5.5.2 Exemplo 5.2

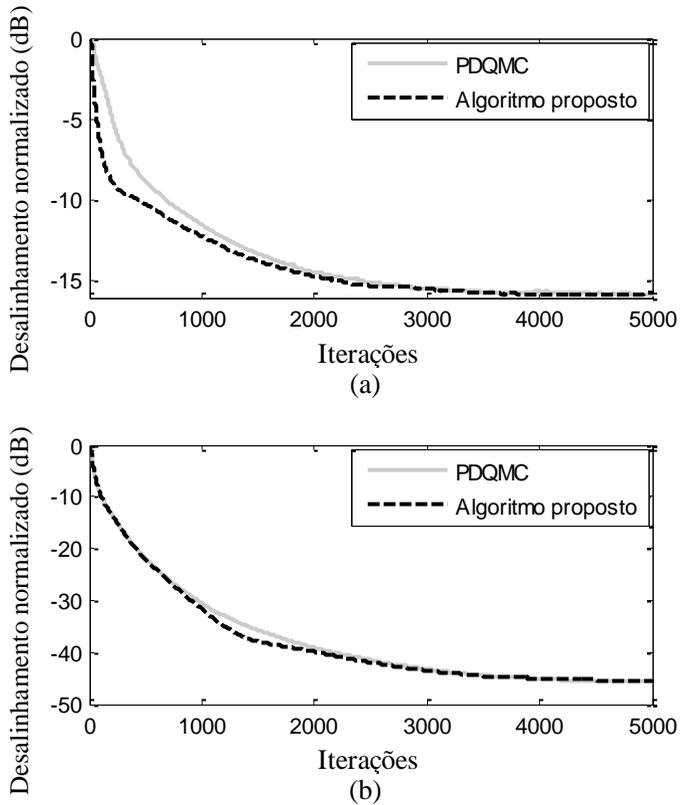
Este exemplo compara o comportamento entre o algoritmo PDQMC e o algoritmo proposto considerando distintos valores de SNR usando a planta  $\mathbf{w}_A^0$ . O sinal de entrada é branco com variância  $\sigma_x^2 = 1$  e os limares  $T_{lo}$  e  $T_{hi}$  são, respectivamente,  $1,2 \times 10^{-2}$  e  $3,5 \times 10^{-2}$ . Para o algoritmo PDQMC, o parâmetro  $\zeta(n)$  é computado por (5.3) usando  $\omega = 2$  para obter a melhor velocidade de convergência (veja Figura 5.1). A Figura 5.6 mostra as curvas de desalinhamento normalizado dos dois algoritmos para SNR igual a 10 e 40 dB, considerando aqui que a estimativa da variância do ruído de medição  $\sigma_v^2$  é perfeita, isto é,  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ . As Figuras 5.7 e 5.8 ilustram as curvas de desalinhamento levando em conta estimativas imprecisas da variância do ruído de medição para o algoritmo PDQMC, considerando casos de variância subestimadas e superestimadas, respectivamente. Dessas figuras, pode-se observar que, além de não requerer uma estimação do ruído de medição presente no sistema, o algoritmo proposto apresenta velocidade de convergência ligeiramente superior à exibida pelo PDQMC quando a estimativa da variância do ruído de medição é perfeita; por outro lado, para casos de estimativas imperfeitas, o algoritmo proposto possui características de convergência muito superiores ao do PDQMC.



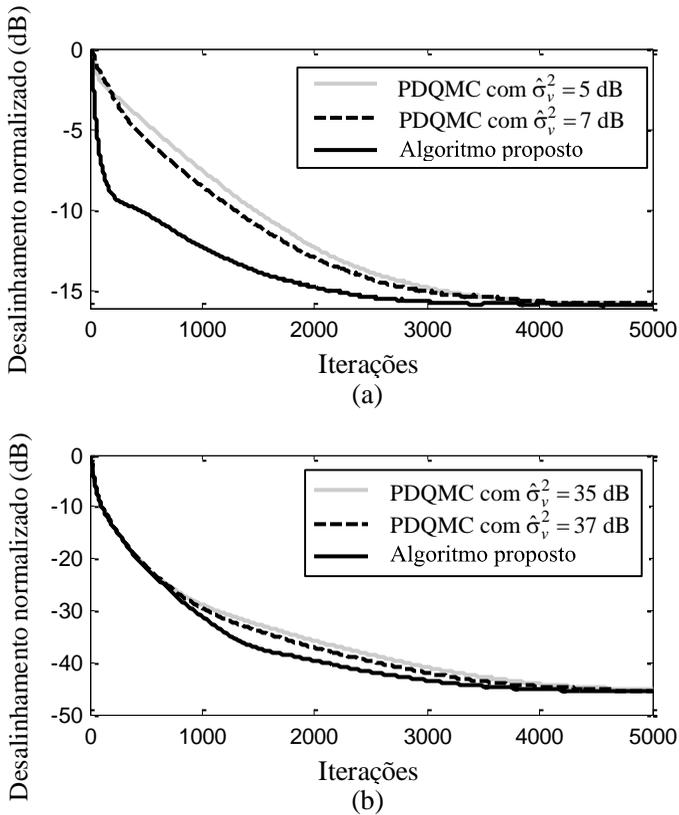
**Figura 5.4** - Exemplo 5.1. Curvas de desalinhamento normalizado do algoritmo proposto usando  $T_{lo}$  igual a  $1,2 \times 10^{-2}$  (linha cinza) e  $0,6 \times 10^{-2}$  (linha tracejada) com  $T_{hi} = 3,5 \times 10^{-2}$ . A resposta ao impulso da planta  $\mathbf{w}_A^o$  é trocada para  $\mathbf{w}_B^o$  em  $n = 5000$ . (a) SNR = 10 dB. (b) SNR = 40 dB.



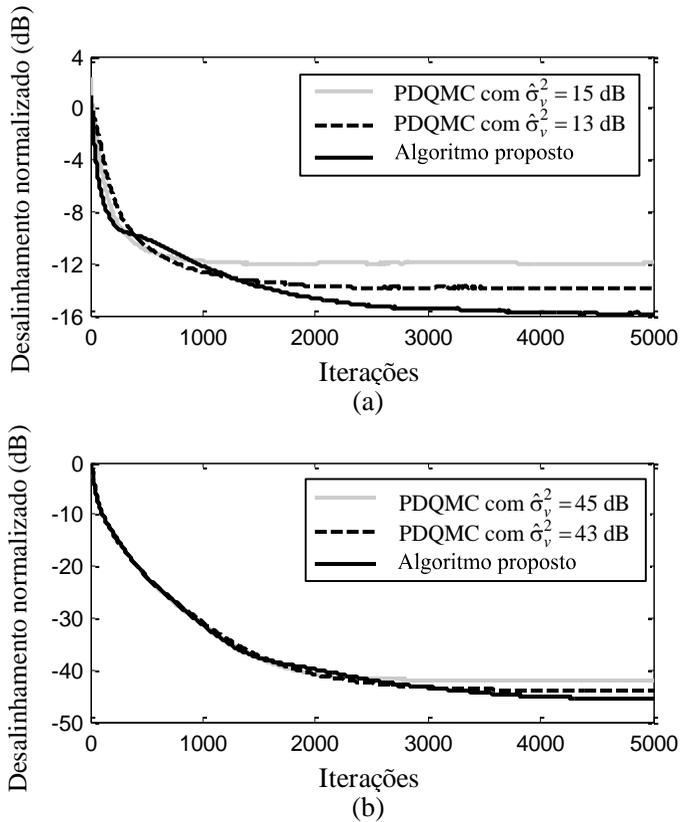
**Figura 5.5** - Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo proposto usando  $T_{hi}$  igual a  $3 \times 10^{-2}$  (linha cinza) e  $4,5 \times 10^{-2}$  (linha tracejada) com  $T_{lo} = 0,9 \times 10^{-2}$ . A resposta ao impulso da planta é chaveada de  $\mathbf{w}_A^o$  para  $\mathbf{w}_B^o$  em  $n = 5000$ . (a) SNR = 10 dB. (b) SNR = 40 dB.



**Figura 5.6** - Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando sinal de entrada branco com variância  $\sigma_x^2 = 1$  e uma planta esparsa. Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata. (a) SNR = 10 dB. (b) SNR = 40 dB.



**Figura 5.7** - Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando casos de subestimação da variância do ruído de medição. (a) SNR = 10 dB. (b) SNR = 40 dB.



**Figura 5.8** - Exemplo 5.2. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto, considerando casos de superestimação da variância do ruído de medição. (a) SNR = 10 dB. (b) SNR = 40 dB.

### 5.5.3 Exemplo 5.3

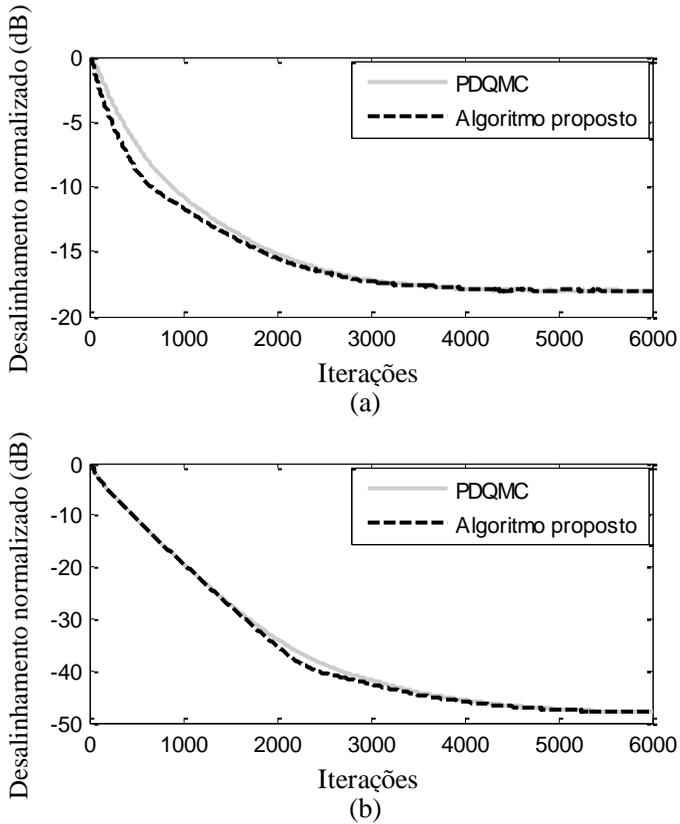
Neste exemplo, o comportamento do algoritmo proposto considerando uma planta com média esparsidade ( $\mathbf{w}_B^o$ ) e dados de entrada branco com variância  $\sigma_x^2 = 1$  é avaliado. Do mesmo modo, dois níveis de variância do ruído de medição são utilizados, isto é,  $\sigma_v^2 = 0,1$  (SNR = 10 dB) e  $\sigma_v^2 = 0,0001$  (SNR = 40 dB). Os limiares  $T_{lo}$  e  $T_{hi}$  são os mesmos utilizados no exemplo anterior. Para o algoritmo PDQMC,  $\omega = 2$  e a estimativa da variância do ruído de medição  $\sigma_v^2$  é novamente perfeita, isto é,  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ .

A Figura 5.9 ilustra as curvas de desalinhamento normalizado entre o algoritmo PDQMC e o algoritmo proposto. Observa-se que o algoritmo proposto novamente supera o PDQMC independentemente da potência do ruído de medição e da menor esparsidade da planta.

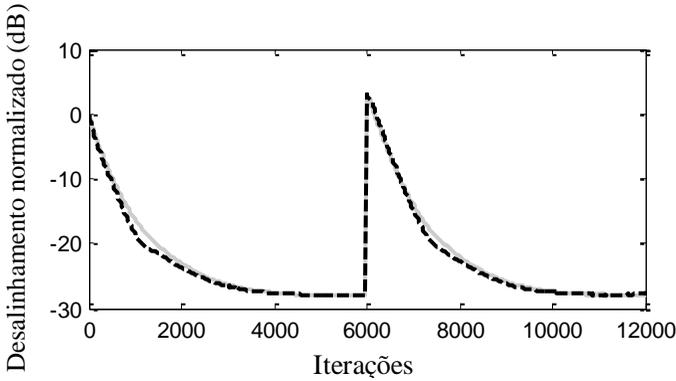
### 5.5.4 Exemplo 5.4

Neste exemplo, a habilidade de rastreamento do algoritmo proposto é verificada. Para tal, considerando dados de entrada branco com variância  $\sigma_x^2 = 1$ , a resposta ao impulso  $\mathbf{w}_B^o$  é deslocada de 12 amostras para a direita no instante  $n = 6000$ . A variância do ruído de medição é  $\sigma_v^2 = 0,01$  (SNR = 20 dB). Os valores dos limiares  $T_{lo}$  e  $T_{hi}$  são os mesmos do Exemplo 5.2. Novamente, para o algoritmo PDQMC,  $\omega = 2$  e a estimativa da variância do ruído de medição  $\sigma_v^2$  é perfeita, isto é,  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ .

A Figura 5.10 mostra as curvas de desalinhamento normalizado dos dois algoritmos avaliados. Pode-se verificar que o algoritmo proposto supera o algoritmo PDQMC em termos de velocidade de convergência para a condição de operação considerada, isto é, estimativa perfeita da variância do ruído de medição  $\sigma_v^2$  e parâmetro  $\omega$  ajustado para obter a melhor velocidade de convergência.



**Figura 5.9** - Exemplo 5.3. Curvas de desalinhamento normalizado do algoritmo PDQMC e do algoritmo proposto considerando sinal de entrada branco com variância  $\sigma_x^2 = 1$  e uma planta com média esparsidade  $\mathbf{w}_B^0$ . Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata. (a) SNR = 10 dB. (b) SNR = 40 dB.



**Figura 5.10** - Exemplo 5.4. Curvas de desalinhamento normalizado do algoritmo PDQMC (linha cinza) e do algoritmo proposto (linha tracejada) considerando um sinal de entrada branco com variância  $\sigma_x^2=1$ . A resposta ao impulso da planta  $\mathbf{w}_B^0$  é deslocada de 12 amostras para a direita em  $n = 6000$ . Para o algoritmo PDQMC, a estimativa do ruído de medição usada é exata.

### 5.5.5 Exemplo 5.5

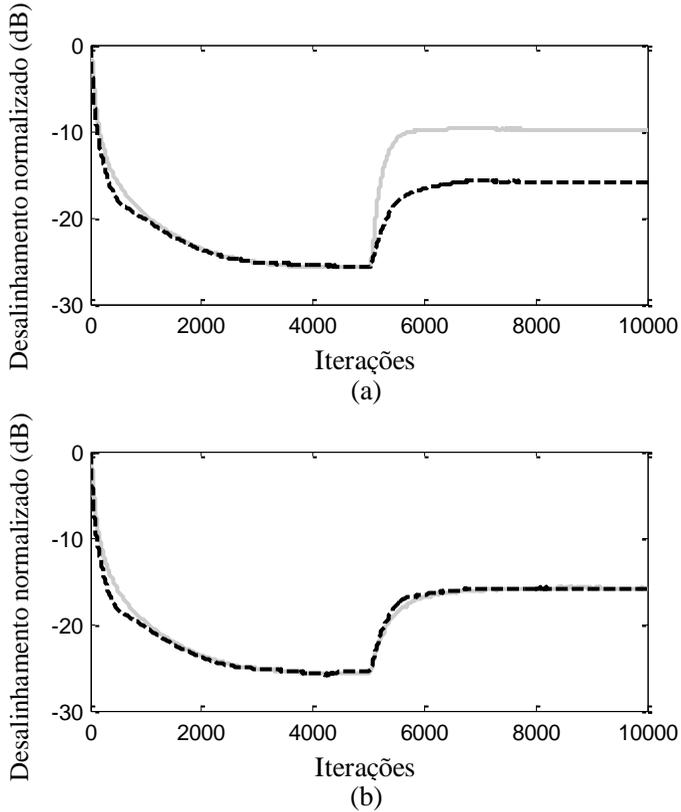
Este exemplo ilustra o comportamento do algoritmo proposto para casos de ruído de medição não estacionário usando sinal de entrada branco com variância  $\sigma_x^2 = 1$  e resposta ao impulso da planta esparsa  $\mathbf{w}_A^0$ . Para tal, dois diferentes cenários para o ruído de medição não estacionário são considerados.

A Figura 5.11 mostra as curvas de desalinhamento normalizado entre o algoritmo PDQMC e o algoritmo proposto quando a variância do ruído de medição é dada por

$$\sigma_v^2(n) = \begin{cases} 0,01, & 0 \leq n < 5000 \\ 0,01 + 0,09[1 - 0,99^{(n-5000)}], & n \geq 5000. \end{cases} \quad (5.13)$$

A Figura 5.11(a) retrata o caso em que a estimativa do ruído de medição  $\hat{\sigma}_v^2$  para o algoritmo PDQMC é mantida igual à variância do ruído de medição inicial (caso de estimação imprecisa), isto é,  $\hat{\sigma}_v^2 = 0,01$  (20 dB SNR) durante todo o processo adaptativo. A Figura 5.11(b)

mostra o caso em que a estimativa da variância do ruído de medição  $\hat{\sigma}_v^2$  é atualizada para ser perfeita a cada iteração.



**Figura 5.11** - Exemplo 5.5. Curvas de desalinhamento normalizado do algoritmo PDQMC (linha cinza) e do algoritmo proposto (linha tracejada) usando sinal de entrada branco com variância  $\sigma_x^2 = 1$  considerando o caso de ruído de medição não estacionário dado por (5.13). (a) Estimativa imprecisa. (b) Estimativa perfeita.

A Figura 5.12 ilustra as curvas de desalinhamento normalizado para os dois algoritmos avaliados quando a variância do ruído de medição é obtida por

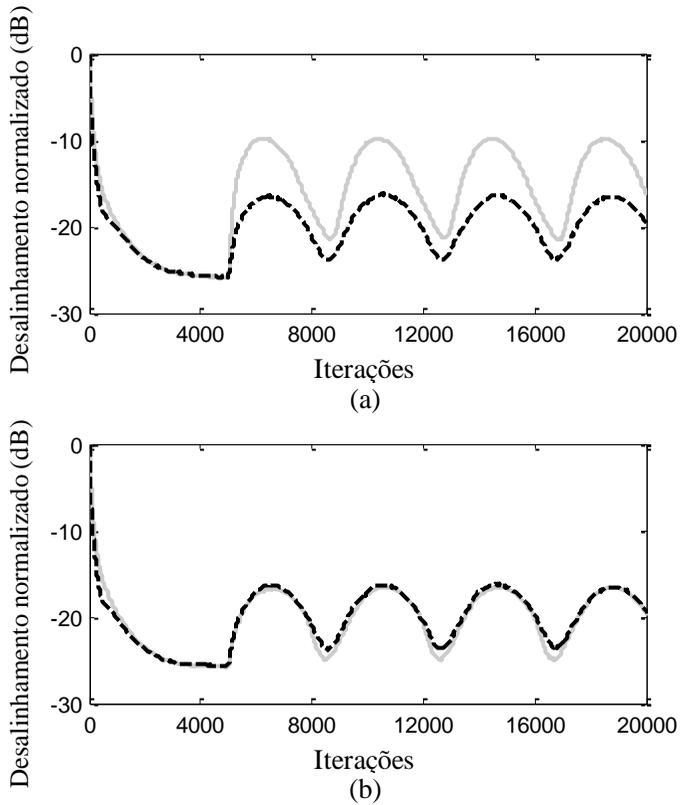
$$\sigma_v^2(n) = \begin{cases} 0,01, & 0 \leq n < 3000 \\ 0,001 + \frac{0,099}{2} \left\{ 1 + \sin \left[ \frac{2\pi(n-3000)}{4096} \right] \right\} & n \geq 3000. \end{cases} \quad (5.14)$$

Essa regra conduz a uma variação da amplitude do ruído de medição entre  $\sigma_v^2 = 0,1$  (10 dB SNR) e  $\sigma_v^2 = 0,001$  (30 dB SNR). A Figura 5.12(a) mostra o caso em que a estimativa da variância do ruído de medição  $\hat{\sigma}_v^2$  para o algoritmo PDQMC é mantida igual à variância do ruído de medição inicial, isto é,  $\hat{\sigma}_v^2 = 0,01$  (20 dB SNR) durante todo o processo adaptativo. Por outro lado, a Figura 5.12(b) mostra o caso em que a estimativa  $\hat{\sigma}_v^2$  é atualizada para ser perfeita a cada iteração, isto é,  $\hat{\sigma}_v^2(n) \equiv \sigma_v^2(n)$ .

Observe que independente da característica do ruído de medição não estacionário, o algoritmo PDQMC somente possui um comportamento similar ao algoritmo proposto quando a variância de tal ruído é perfeitamente estimada a cada iteração.

### 5.5.6 Exemplo 5.6

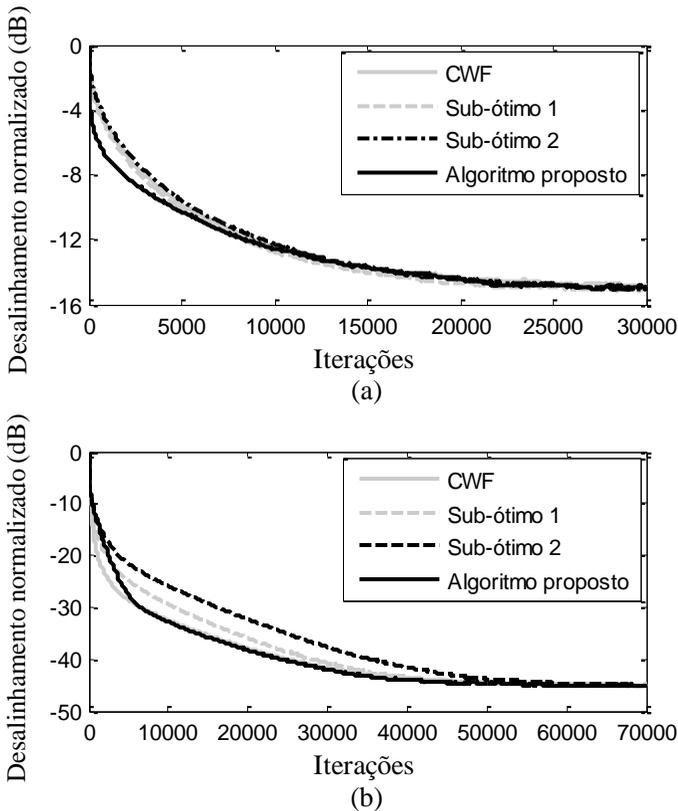
Neste exemplo, usando dados de entrada coloridos, o desempenho do algoritmo proposto é avaliado comparando-o com os algoritmos CWF, Sub-ótimo 1 e Sub-ótimo 2, discutidos em [69], os quais são derivados a partir da minimização do erro quadrático médio, a cada iteração, para entrada correlacionada, sendo também usada a combinação de ganho convexa adaptativa. Aqui, a resposta ao impulso da planta é  $\mathbf{w}_A^o$  e o sinal de entrada colorido é obtido por (4.13) utilizando  $\sigma_{\eta_1}^2 = 0,36$ ,  $b_1 = 0,75$  e  $b_2 = -0,75$ . A dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é  $\chi = 120$ . A resposta ao impulso da planta usada é  $\mathbf{w}_A^o$  e os parâmetros  $T_{lo}$  e  $T_{hi}$  são, respectivamente,  $1,2 \times 10^{-2}$  e  $3,5 \times 10^{-2}$ . Visando atingir um rápida velocidade de convergência, são definidos  $\omega = 5,6$  para o algoritmo CWF e  $\omega = 4$  tanto para o Sub-ótimo 1 quanto para o Sub-ótimo 2. Também, para os algoritmos dados em [69],  $\psi = 0,9$  (um parâmetro desses algoritmos) é usado. Novamente, dois casos de estimativa da variância do ruído de medição  $\hat{\sigma}_v^2$  são considerados, isto é, estimativa perfeita e imperfeita.



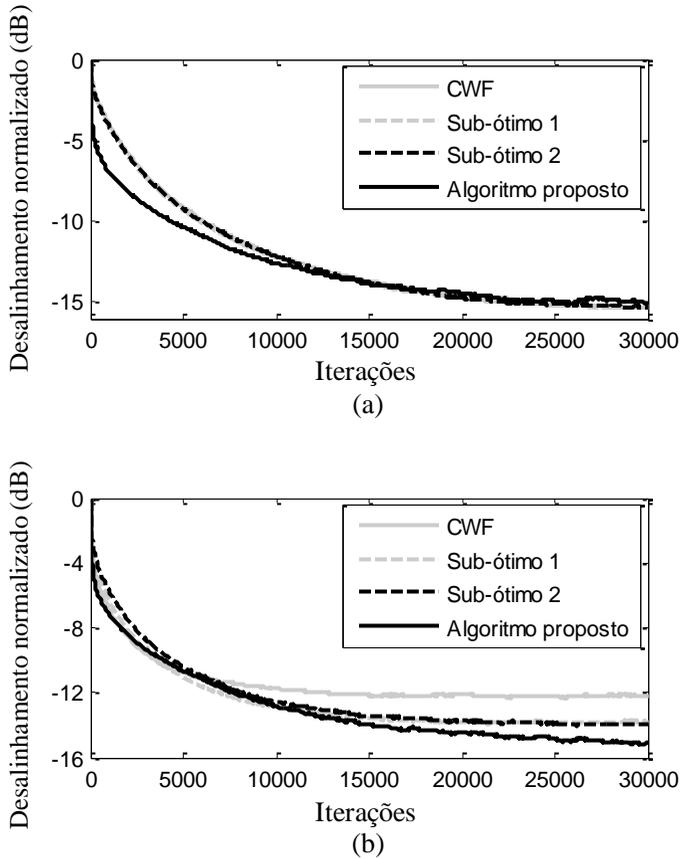
**Figura 5.12** - Exemplo 5.5. Curvas de desalinhamento normalizado do algoritmo PDQMC (linha cinza) e do algoritmo proposto (linha tracejada) usando sinal de entrada branco com variância  $\sigma_x^2 = 1$  considerando um caso de ruído de medição não estacionário dado por (5.14). (a) Estimativa imprecisa. (b) Estimativa perfeita.

A Figura 5.13 ilustra as curvas de desalinhamento normalizado assumindo estimativa perfeita da variância do ruído de medição para os algoritmos CWF, Sub-ótimo 1 e Sub-ótimo 2. A Figura 5.14 mostra dois casos de erro de estimativa, isto é, caso subestimado e superestimado com erros de  $-3$  e  $+3$  dB, respectivamente, na variância do ruído de medição. Analisando a Figura 5.13, pode-se observar que para  $\text{SNR} = 10\text{dB}$ , o algoritmo proposto supera todos os algoritmos avaliados

e, para  $\text{SNR} = 40\text{dB}$ , o algoritmo proposto exibe desempenho similar ao do CWF (embora esse último tenha uma complexidade computacional muito maior) e é melhor do que os outros dois algoritmos sub-ótimos em termos de velocidade de convergência. Para os casos de estimativa imperfeita, o algoritmo proposto mantém seu ótimo desempenho, ao passo que o CWF e os algoritmos sub-ótimos sofrem redução da velocidade de convergência [veja Figura 5.14(a)] ou degradação do nível de desalinhamento em regime permanente [veja Figura 5.14(b)].



**Figura 5.13** – Exemplo 5.6. Curvas de desalinhamento normalizado dos algoritmos CWF, Sub-ótimo 1, Sub-ótimo 2 e do algoritmo proposto para dados de entrada coloridos considerando  $\hat{\sigma}_v^2 \equiv \sigma_v^2$ . (a)  $\text{SNR} = 10\text{dB}$ . (b)  $\text{SNR} = 40\text{dB}$ .



**Figura 5.14** – Exemplo 5.6. Curvas de desalinhamento normalizado dos algoritmos CWF, Sub-ótimo 1, Sub-ótimo 2 e do algoritmo proposto para dados de entrada coloridos, considerando  $\text{SNR} = 10 \text{ dB}$  e erros de estimativa na variância do ruído de medição. (a) Caso subestimado (erro de  $-3 \text{ dB}$ ). (b) Caso superestimado (erro de  $+3 \text{ dB}$ ).

## 5.6 Conclusões

Neste capítulo, uma versão melhorada do algoritmo PDQMC foi apresentada. O novo algoritmo visa reduzir a dependência da variância do ruído de medição do algoritmo PDQMC. Para tal, através da medida da autocorrelação do sinal de erro, uma nova estratégia para migrar do ganho proporcional à estimativa do desvio quadrático médio dos coeficientes para o ganho unitário é introduzida. Essa estratégia, além de eliminar a dependência da variância do ruído de medição, apresenta maior velocidade de convergência tanto para plantas esparsas quanto dispersas, mesmo para os casos em que o algoritmo PDQMC é ajustado para sua melhor condição de operação com vistas à velocidade de convergência. Além disso, a nova abordagem pode ser aplicada a outros algoritmos que utilizam combinação de ganho convexa adaptativa, tais como o *water-filling* [33], CWF, Sub-ótimo 1 e Sub-ótimo 2 [69]. Resultados de simulações numéricas atestaram a eficácia do novo algoritmo para os casos em que a variância do ruído de medição é desconhecida ou pobremente estimada.

### **NOVA ABORDAGEM PARA ALGORITMOS TIPO PROPORCIONAL PARA OPERAÇÃO EM AMBIENTES COM ELEVADA ESPARSIDADE**

#### **6.1 Introdução**

Neste capítulo, uma política de distribuição de ganho proporcional para algoritmos operando em ambientes de alta esparsidade é apresentada. A nova política usa uma função de amplificação do ganho atribuído a coeficientes ativos e a NSVR (taxa de variação suavizada e normalizada da magnitude do coeficiente) como estratégia de transferência de ganho ao longo do processo adaptativo. Além disso, considerando a política de ganho proposta, dois novos algoritmos são concebidos. Esses algoritmos apresentam características de convergência superiores a dos algoritmos proporcionais tradicionais quando a resposta ao impulso da planta tem alta esparsidade. Resultados de simulações numéricas atestam a eficácia da nova política de distribuição de ganho proporcional.

#### **6.2 Análise do Comportamento de Algoritmos Adaptativos em Plantas com Elevada Esparsidade**

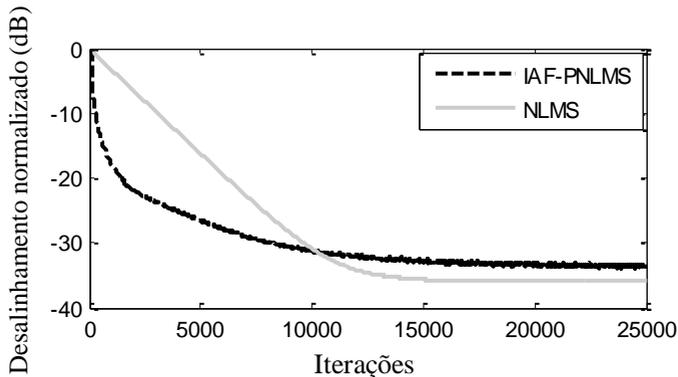
Plantas esparsas com elevado grau de esparsidade são caracterizadas por possuírem alguns poucos coeficientes ativos e muitos coeficientes inativos, o que acentua os problemas já bem conhecidos dos algoritmos adaptativos que operam nesse tipo de ambiente. Tal comportamento se deve ao fato de que a velocidade de convergência inicial é muito dependente da taxa de convergência desses poucos coeficientes ativos e, após a convergência dos coeficientes ativos, a influência da convergência dos coeficientes inativos impacta fortemente a convergência global do filtro adaptativo devido ao grande número

desses coeficientes (inativos). Assim, visando melhorar as características de convergência dos algoritmos adaptativos, estratégias específicas de distribuição de ganho para lidar com plantas com respostas ao impulso de alta esparsidade devem ser consideradas. Contudo, como a soma dos ganhos individuais a cada iteração nos algoritmos adaptativos do tipo proporcional é constante, qualquer aumento ou diminuição do ganho alocado a um coeficiente impacta no ganho atribuído aos demais. Por exemplo, aumentar o ganho dos coeficientes ativos reduz o ganho atribuído aos coeficientes inativos; como resultado, há um aumento da velocidade de convergência dos coeficientes ativos (rápida convergência inicial) e uma diminuição na dos inativos (convergência mais lenta após o período inicial). Além disso, em função do elevado ganho alocado aos coeficientes ativos, um maior desalinhamento em regime permanente é observado. Por outro lado, ao diminuir o ganho dos coeficientes ativos e aumentar o ganho dos coeficientes inativos, uma forte redução da velocidade de convergência do algoritmo é observada, apesar de o desalinhamento em regime diminuir.

Para verificar tal comportamento, simulações de MC (média de 100 realizações independentes) dos algoritmos NLMS e IAF-PNLMS são realizadas considerando um ambiente de alta esparsidade. Para isso, a resposta ao impulso da planta mostrada na Figura 1.3(b), aqui denominada  $\mathbf{w}_C^0$ , com grau de esparsidade  $S(\mathbf{w}_C^0) = 0,90$ , obtida a partir do Modelo 1 da norma ITU-T Recomendações G.168 [65] é utilizada. O sinal de entrada é branco com variância  $\sigma_x^2$  unitária e a variância do ruído de medição é  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Para os dois algoritmos, o passo de adaptação é  $\beta = 0,1$  e, para o algoritmo IAF-PNLMS,  $f_i(0) = 0$ . A escolha desses algoritmos para tal avaliação é baseada no fato que o algoritmo NLMS atribui o mesmo ganho dos coeficientes ativos para os coeficientes inativos, ao passo que o IAF-PNLMS atribui ganho totalmente proporcional à magnitude dos coeficientes e, portanto, ganhos maiores para os coeficientes mais significativos.

A Figura 6.1 ilustra as curvas de desalinhamento normalizado dos algoritmos NLMS e IAF-PNLMS, evidenciando os efeitos no desempenho dos algoritmos decorrentes das políticas de distribuição de ganho filosoficamente opostas. Pode-se observar que, apesar de os dois algoritmos possuírem o mesmo valor de passo de adaptação  $\beta = 0,1$ , o algoritmo IAF-PNLMS possui uma velocidade de convergência inicial

muito maior do que a apresentada pelo algoritmo NLMS; por outro lado, o NLMS possui um comportamento em regime permanente melhor do que o IAF-PNLMS. Tais comportamentos distintos são esperados, visto que esses algoritmos possuem estratégias opostas de alocação de ganho, isto é, um totalmente uniforme e outro totalmente proporcional à magnitude; como consequência, o algoritmo IAF-PNLMS apresenta melhores características iniciais ao passo que o NLMS, melhores características em regime permanente.



**Figura 6.1** – Curvas de desalinhamento normalizado dos algoritmos NLMS e IAF-PNLMS para resposta ao impulso de alta esparsidade, considerando resposta ao impulso da planta  $\mathbf{w}_C^o$ , dados de entrada branco com variância unitária e SNR = 30 dB.

Para melhor compreensão dos resultados mostrados na Figura 6.1, uma análise do desalinhamento dos coeficientes ativos e inativos é realizada a seguir. Para tal, a expressão do desalinhamento normalizado, dada por [39]

$$\Gamma(n) = \frac{\|\mathbf{w}^o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}^o\|_2^2} \quad (6.1)$$

é expandida, resultando em

$$\Gamma(n) = \frac{1}{\|\mathbf{w}^o\|_2^2} \left( |w_1^o - w_1(n)|^2 + |w_2^o - w_2(n)|^2 + \dots + |w_N^o - w_N(n)|^2 \right) \quad (6.2)$$

Note que, para  $\mathbf{w}(n) = \mathbf{0}$ , o desalinhamento normalizado resulta em  $\Gamma(n) = 1$  e, para  $\mathbf{w}(n) = \mathbf{w}^o$ ,  $\Gamma(n) = 0$ .

Assim, visto que o desalinhamento normalizado é composto pelo módulo da diferença entre o valor ótimo e o valor instantâneo de cada coeficiente do filtro adaptativo, os termos  $|w_i^o - w_i(n)|^2$  em (6.2) podem ser reunidos considerando dois grupos de coeficientes, isto é, ativos e inativos. Assim, o desalinhamento normalizado é obtido por

$$\begin{aligned} \Gamma(n) &= \frac{1}{\|\mathbf{w}^o\|_2^2} \left( \sum_{i \in A} |w_i^o - w_i(n)|^2 + \sum_{i \in B} |w_i^o - w_i(n)|^2 \right) \\ &= \frac{\sum_{i \in A} |w_i^o - w_i(n)|^2}{\|\mathbf{w}^o\|_2^2} + \frac{\sum_{i \in B} |w_i^o - w_i(n)|^2}{\|\mathbf{w}^o\|_2^2} \end{aligned} \quad (6.3)$$

$$\triangleq \Gamma_{\text{ativos}}(n) + \Gamma_{\text{inativos}}(n)$$

onde  $A$  e  $B$  são, respectivamente, os conjuntos de índices de posições dos coeficientes ativos e inativos conforme definidos em (3.8).

A partir de (6.3), pode-se estabelecer o valor médio de desalinhamento dos coeficientes ativos e inativos para uma determinada resposta ao impulso da planta. Dessa forma, o desalinhamento médio dos coeficientes ativos é definido como

$$\bar{\Gamma}_{\text{ativos}}(n) \triangleq \frac{\Gamma_{\text{ativos}}(n)}{N_{\text{ativos}}} \quad (6.4)$$

e o desalinhamento médio dos coeficientes inativos como

$$\bar{\Gamma}_{\text{inativos}}(n) \triangleq \frac{\Gamma_{\text{inativos}}(n)}{N_{\text{inativos}}}, \quad (6.5)$$

onde  $N_{\text{ativos}}$  e  $N_{\text{inativos}}$  são, respectivamente, o número de coeficientes ativos e inativos da resposta ao impulso da planta.

Levando em conta (6.4) e (6.5), note que para uma planta uniforme, isto é, com grau de esparsidade  $S(\mathbf{w}^0) = 0$ , no início do processo adaptativo onde  $\mathbf{w}(0) = \mathbf{0}$ , os valores dos desalinhamentos médios são  $\bar{\Gamma}_{\text{inativos}}(0) = 0$  e  $\bar{\Gamma}_{\text{ativos}}(0) = 1/N$ , ou seja, todos os coeficientes contribuem igualmente para o desalinhamento total do filtro adaptativo. Por outro lado, considerando uma planta com resposta ao impulso igual ao delta de Dirac [ $S(\mathbf{w}^0) = 1$ ],  $\bar{\Gamma}_{\text{inativos}}(0) = 0$  e  $\bar{\Gamma}_{\text{ativos}}(0) = 1$ , isto é, todo o desalinhamento está concentrado no único coeficiente ativo da planta. Portanto, pode-se afirmar que quanto maior o grau de esparsidade da planta, maior é o impacto do desalinhamento de um coeficiente ativo no desalinhamento total.

Agora, considerando a planta esparsa  $\mathbf{w}_C^0$  [ $S(\mathbf{w}^0) = 0,90$ ] utilizada na simulação da Figura 6.1, e assumindo que coeficientes ativos são aqueles cuja magnitude é maior ou igual a 1% da magnitude do maior coeficiente, o que resulta em  $N_{\text{ativos}} = 48$  e  $N_{\text{inativos}} = 464$ , para  $n = 0$ , os desalinhamentos médios dos coeficientes inativos e ativos são, respectivamente,  $\bar{\Gamma}_{\text{inativos}}(0) = 7,82 \times 10^{-7}$  e  $\bar{\Gamma}_{\text{ativos}}(0) = 2,08 \times 10^{-2}$ . Observe que, como esperado, o desalinhamento médio de um coeficiente ativo é muito maior do que o desalinhamento médio de um coeficiente inativo, o que justifica a aplicação de ganhos de adaptação maiores aos coeficientes ativos no início do processo adaptativo em relação aos coeficientes inativos. Para quantificar o que representa essa diferença nos desalinhamentos médios, considere que os coeficientes ativos do filtro adaptativo estão relacionados por um valor  $p$  com os coeficientes ativos da planta, isto é,

$$w_i(n) = p w_i^0, \quad i \in A. \quad (6.6)$$

Assim, pode-se calcular qual a distância média  $p$  que os coeficientes ativos precisam estar dos seus valores ótimos para se obter o mesmo desalinhamento dos coeficientes inativos no instante inicial. Com isso,

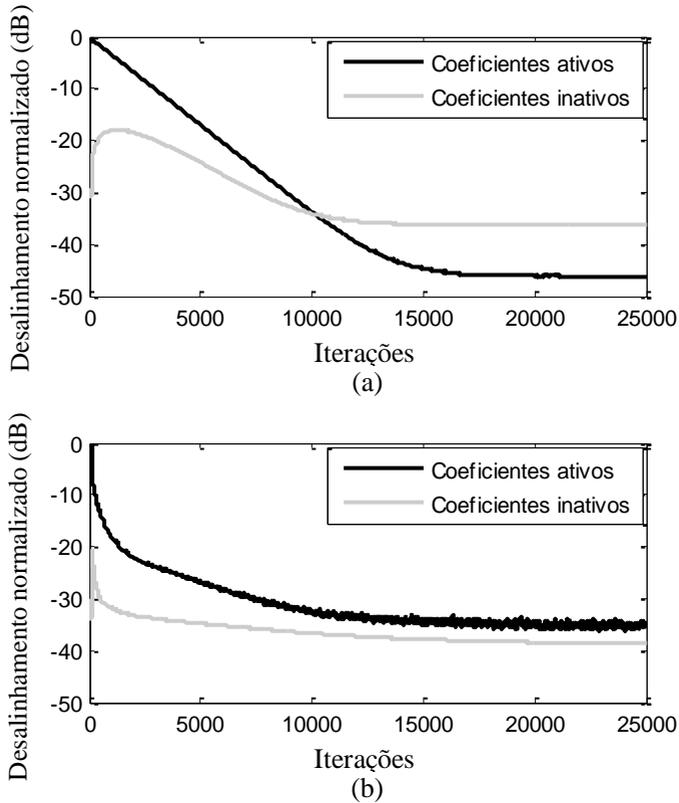
$$\Gamma_{\text{ativos}}(n) = \bar{\Gamma}_{\text{inativos}}(0) N_{\text{inativos}}. \quad (6.7)$$

Agora, aplicando-se as definições dadas em (6.4) e (6.6) em (6.7), resulta em

$$\frac{\sum_{i \in A} |w_i^o - p w_i^o|^2}{\|\mathbf{w}^o\|_2^2} = \bar{\Gamma}_{\text{inativos}}(0) N_{\text{inativos}} \quad (6.8)$$

Com isso, considerando  $p > 0$  e resolvendo (6.8) para  $\bar{\Gamma}_{\text{inativos}}(0)$  obtido a partir da planta  $\mathbf{w}_C^o$ , dois valores de  $p$  satisfazem (6.8), isto é,  $p_1 = 1,019$  e  $p_2 = 0,981$ , ou seja, os valores dos 48 coeficientes ativos do filtro adaptativo devem estar 1,9% acima ou abaixo de seus valores ótimos para igualar o desalinhamento inicial dos coeficientes inativos. Em outras palavras, os coeficientes ativos devem “percorrer” um longo caminho de adaptação em direção a seus valores ótimos para terem o mesmo desalinhamento médio causado pelos coeficientes inativos no início do processo adaptativo.

Note que, na prática, o valor do desalinhamento total dos coeficientes ativos e inativos durante o processo adaptativo depende da política de distribuição de ganho do algoritmo utilizado. Para exemplificar tal afirmação, a Figura 6.2 mostra o desalinhamento dos coeficientes ativos e inativos para os algoritmos NLMS e IAF-PNLMS considerando o mesmo cenário de simulação apresentado na Figura 6.1. Observe que a velocidade de convergência dos coeficientes ativos no algoritmo NLMS [veja Figura 6.2(a)] é baixa; contudo, seu desalinhamento em regime permanente é menor do que o desalinhamento dos coeficientes inativos. Por outro lado, a velocidade de convergência dos coeficientes ativos no algoritmo IAF-PNLMS [veja Figura 6.2(b)] é mais alta inicialmente, entretanto seu desalinhamento em regime permanente é maior do que o desalinhamento dos coeficientes inativos. Além disso, note que a diferença do desalinhamento dos coeficientes inativos entre os algoritmos NLMS e IAF-PNLMS em regime permanente é menor do que a diferença do desalinhamento dos coeficientes ativos; ou seja, para que o algoritmo IAF-PNLMS atinja o mesmo desalinhamento em regime permanente do que o observado no algoritmo NLMS, são os coeficientes ativos que devem percorrer o maior caminho. Para quantificar o “tamanho” do caminho a ser percorrido pelos coeficientes ativos do algoritmo IAF-PNLMS para atingir o mesmo desalinhamento dos coeficientes ativos no algoritmo NLMS, de forma semelhante ao realizado anteriormente, pode-se calcular o valor da distância  $p$  dos coeficientes ativos considerando os valores do desalinhamento desses coeficientes em regime permanente. Dessa forma,



**Figura 6.2** - Curvas de desalinhamento normalizado de coeficientes ativos e inativos considerando resposta ao impulso da planta  $\mathbf{w}_C^o$ , dados de entrada brancos com variância unitária e SNR = 30 dB. (a) Algoritmo NLMS. (b) Algoritmo IAF-PNLMS.

tomando-se como base o valor do desalinhamento em  $k = 50000$ , obtém-se para o algoritmo IAF-PNLMS  $p_1 = 0,9827$  e  $p_2 = 1,0173$  e, para o NLMS,  $p_1 = 0,9951$  e  $p_2 = 1,0049$ ; ou seja, para se obter o mesmo desalinhamento do algoritmo NLMS, os coeficientes ativos do algoritmo IAF-PNLMS devem passar de uma diferença média de 1,73% de seus valores ótimos para 0,49%. Para tal aproximação de seus valores ótimos, o ganho atribuído a esses coeficientes deve ser reduzido.

Assim, a partir da discussão apresentada, conclui-se que estratégias de aumento de ganho atribuído a coeficientes ativos associadas com uma política de transferência de ganho durante o processo adaptativo, em plantas exibindo alta esparsidade, podem melhorar as características de convergência dos algoritmos operando nesse tipo de ambiente.

### 6.3 Abordagem Proposta

Nesta seção, uma nova abordagem para a definição do ganho proporcional de algoritmos adaptativos em aplicações cujas respostas ao impulso da planta possuem elevado grau de esparsidade é proposta. Tal abordagem é baseada na aplicação de uma função de amplificação de ganho atribuído aos coeficientes mais significativos combinada com uma estratégia de transferência de ganho. Aqui, para a realização da transferência de ganho utiliza-se a estratégia discutida no Capítulo 4, isto é, avaliação da convergência individual do coeficiente através da métrica NSVR. Assim, a nova política de distribuição de ganho é definida como

$$g_i(n) = \frac{\gamma_i(n)}{\sum_{j=1}^N \gamma_j(n)} \quad (6.9)$$

com

$$\gamma_i(n) = \begin{cases} K_c \gamma_{\min}^P(n), & V_i(n) \leq \varepsilon \\ \max(\gamma_{\min}^P(n), F_a(|w_i(n)|)), & \text{caso contrário} \end{cases} \quad (6.10)$$

onde  $F_a(\cdot)$  denota a função de amplificação de ganho e  $K_c > 1$  é o parâmetro que define o ganho de coeficientes ativos que atingiram a proximidade de seus valores ótimos.  $V_i(n)$  representa a NSVR da magnitude do coeficiente dada por (3.12).

Note a partir de (6.10) que, ao contrário dos algoritmos introduzidos no Capítulo 4, o ganho atribuído aos coeficientes ativos que atingem a vizinhança de seus valores ótimos é maior do que o ganho atribuído aos coeficientes inativos. Desse modo, a distribuição de ganho proposta não tenderá ao algoritmo NLMS após a convergência dos coeficientes. Contudo, conforme mostrado no Apêndice III, o desequilíbrio do ganho alocado a coeficientes ativos e inativos não impõe

necessariamente uma diferença do erro em excesso em regime permanente quando comparado com o algoritmo NLMS.

## 6.4 Algoritmo Adaptativo Proporcional Baseado em Função de Amplificação de Ganho Linear

Nesta seção, visando aumentar a taxa de convergência dos coeficientes ativos de plantas com elevada esparsidade, um novo algoritmo, aqui denominado LAF-PNLMS, é introduzido. Em seguida, uma breve análise da política de distribuição de ganho do novo algoritmo é apresentada.

### 6.4.1 Algoritmo LAF-PNLMS

O algoritmo LAF-PNLMS usa uma função de amplificação linear para definir a sua política de distribuição de ganho e, como consequência, aumentar a velocidade de convergência dos coeficientes ativos. Além disso, o ganho comum atribuído aos coeficientes que alcançaram a vizinhança de seus valores ótimos é definido ser maior do que o ganho atribuído a coeficientes inativos. O uso de tal estratégia surge do fato de que coeficientes ativos possuem grande impacto no desalinhamento global do algoritmo para os casos em que a resposta ao impulso possui alta esparsidade (veja Seção 6.2) e, portanto, necessitam ter velocidades de convergência maiores do que a dos coeficientes inativos durante todo o processo adaptativo. Assim, para o algoritmo LAF-PNLMS, o ganho individual é expresso por

$$g_i^{\text{LP}}(n) = \frac{\gamma_i^{\text{LP}}(n)}{\Phi^{\text{LP}}(n)} \quad (6.11)$$

com

$$\gamma_i^{\text{LP}}(n) = \begin{cases} K_c \gamma_{\min}^{\text{P}}(n), & V_i(n) \leq \varepsilon \\ \max[\gamma_{\min}^{\text{P}}(n), K_a |w_i(n)|], & \text{caso contrário} \end{cases} \quad (6.12)$$

onde  $K_a$  representa a constante de multiplicação do ganho e o superescrito LP representa o algoritmo LAF-PNLMS.

A Tabela 6.1 resume o algoritmo LAF-PNLMS. Note que o novo método mantém um ganho mínimo para os coeficientes inativos, evitando

assim seu “congelamento” no caso em que a magnitude do coeficiente seja muito menor do que a magnitude do maior coeficiente do filtro adaptativo. Além disso, como qualquer mudança na política de ganho de coeficientes cuja magnitude é muito pequena, praticamente, não altera o comportamento do algoritmo (veja Seção 4.9.1), aqui apenas os coeficientes com magnitude maior ou igual a  $A_{\min} \max[\|\mathbf{w}(n)\|]$  ( $0 < A_{\min} \leq 1$ ) têm sua NSVR computada. Como resultado, a complexidade computacional é consideravelmente reduzida, visto que em uma planta com esparsidade elevada o número de coeficientes ativos é muito pequeno.

#### 6.4.2 Distribuição de Ganho do Algoritmo LAF-PNLMS

A distribuição de ganho do algoritmo LAF-PNLMS pode ser classificada como segue:

- i) **Ganho dos coeficientes que alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  tenha quase atingido a convergência e seu ganho associado é dado por

$$g_c^{\text{LP}}(n) = \frac{1}{\Phi^{\text{LP}}(n)} K_c \gamma_{\min}^{\text{P}}(n). \quad (6.13)$$

- ii) **Ganho dos coeficientes que ainda não alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) > \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  não atingiu a convergência e seu ganho associado é dado por

$$g_{\text{nc},i}^{\text{LP}}(n) = \begin{cases} \frac{1}{\Phi^{\text{LP}}(n)} \gamma_{\min}^{\text{P}}(n), & K_a |w_i(n)| < \gamma_{\min}^{\text{P}}(n) \\ \frac{1}{\Phi^{\text{LP}}(n)} K_a |w_i(n)|, & K_a |w_i(n)| \geq \gamma_{\min}^{\text{P}}(n). \end{cases} \quad (6.14)$$

**Tabela 6.1** – ALGORITMO LAF-PNLMS

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_{\min}^P(n) = \rho \max[\delta, |w_1(n)|, |w_2(n)|, \dots, |w_N(n)|]$$

$$\gamma_i^{\text{LP}}(n) = \max[\gamma_{\min}^P(n), K_a |w_i(n)|], \quad 1 \leq i \leq N$$

Se  $|w_i(n)| \geq A_{\min} \max[|\mathbf{w}(n)|], \quad 1 \leq i \leq N$

$$Q_i(n) = |w_i(n)| + |w_i(n-1)| + \dots + |w_i(n - \frac{\Delta}{2} + 1)|,$$

$$n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

$$P_i(n) = |w_i(n - \frac{\Delta}{2})| + |w_i(n - \frac{\Delta}{2} - 1)| + \dots + |w_i(n - \Delta + 1)|,$$

$$n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

$$V_i(n) = \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, \quad n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

Se  $V_i(n) \leq \varepsilon$

$$\gamma_i^{\text{LP}}(n) = K_c \gamma_{\min}^P(n), \quad n \geq N + \Delta$$

$$g_i^{\text{LP}}(n) = \frac{\gamma_i^{\text{LP}}(n)}{\sum_{j=1}^N \gamma_j^{\text{LP}}(n)}, \quad 1 \leq i \leq N$$

$$\mathbf{G}^{\text{LP}}(n) = \text{diag}[g_1^{\text{LP}}(n) \ g_2^{\text{LP}}(n) \ \dots \ g_N^{\text{LP}}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{\text{LP}}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{\text{LP}}(n) \mathbf{x}(n) + \xi}$$

Agora, levando em conta que o ganho total distribuído dentre os coeficientes do filtro adaptativo (a cada iteração) no algoritmo LAF-PNLMS é constante e igual a 1, o ganho total pode ser escrito como

$$g_{\text{total}}^{\text{LP}}(n) = \frac{1}{\Phi^{\text{LP}}(n)} \left[ \sum_{i \in (A-C)} K_a |w_i(n)| + \sum_{i \in B} \gamma_{\min}^{\text{P}}(n) + \sum_{i \in C} K_c \gamma_{\min}^{\text{P}}(n) \right] = 1 \quad (6.15)$$

onde  $A$ ,  $B$  e  $C$  são, respectivamente, o conjunto de índices associado às posições dos coeficientes ativos, inativos e ativos que quase atingiram a convergência conforme (4.9).

Note que no início do processo adaptativo (isto é, todos os coeficientes ativos ainda não alcançaram a vizinhança de seus valores ótimos) todos os coeficientes ativos têm seus ganhos amplificados linearmente por  $K_a$  em relação aos inativos, isto é, todos os ganhos dos coeficientes ativos são majorados de forma igual independente da sua magnitude. Como resultado, um aumento na velocidade de convergência desses coeficientes em detrimento à velocidade dos coeficientes inativos é esperado. Conforme os coeficientes ativos atingem a vizinhança de seus valores ótimos, seus ganhos são reduzidos para  $K_c \gamma_{\min}^{\text{P}}(n)$ , isto é, um coeficiente ativo sempre irá receber um ganho maior do que o atribuído a um coeficiente inativo. Tal estratégia evita que a velocidade de convergência de um coeficiente ativo diminua excessivamente após a mudança de política de ganho. Observe que, em uma situação ideal (em regime permanente) todos os coeficientes ativos receberiam o ganho

$K_c \frac{\gamma_{\min}^{\text{P}}(n)}{\phi^{\text{LP}}(n)}$ , ao passo que todos os coeficientes inativos receberiam  $\frac{\gamma_{\min}^{\text{P}}(n)}{\phi^{\text{LP}}(n)}$ .

## 6.5 Resultados de Simulação do Algoritmo LAF-PNLMS

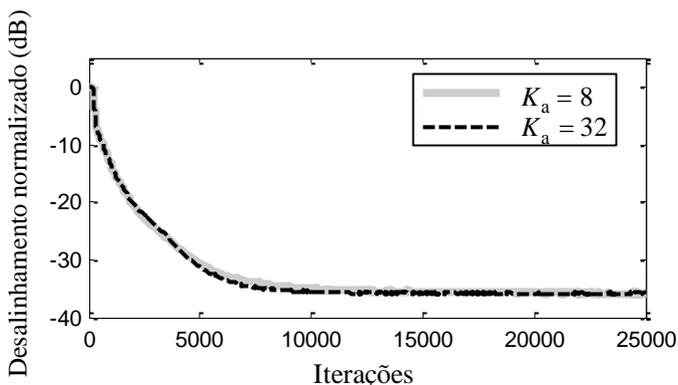
Nesta seção, considerando um problema de identificação de sistemas, simulações de MC (média de 100 realizações independentes) são apresentadas visando avaliar o algoritmo proposto. Para tal, seis exemplos são mostrados. Em todos os exemplos utiliza-se a resposta ao impulso da planta  $\mathbf{w}_C^o$  [ $S(\mathbf{w}_C^o) = 0,90$ ],  $\xi = 10^{-4}$ ,  $\rho = 0,01$  e  $\delta = 0,01$ . Para o algoritmo LAF-PNLMS, o valor do limiar de convergência usado é  $\varepsilon = 5 \times 10^{-2}$ .

### 6.5.1 Exemplo 6.1

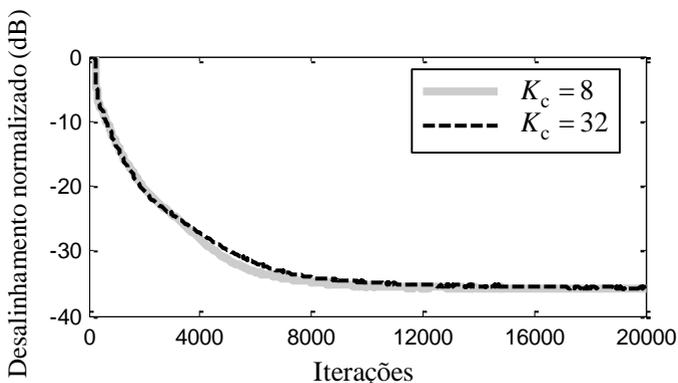
Neste exemplo, o comportamento da convergência do algoritmo LAF-PNLMS com respeito aos parâmetros  $K_a$  e  $K_c$  é verificado. Para tal, utiliza-se dados de entrada branco com variância  $\sigma_x^2 = 1$ , variância do ruído de medição  $\sigma_v^2 = 0,001$  (30 dB SNR), passo de adaptação  $\beta = 0,1$ , número de amostras  $\Delta = N$  e  $A_{\min} = 0,05$ , isto é, apenas coeficientes com magnitudes maiores ou iguais a 5% da magnitude do maior coeficiente do filtro adaptativo têm sua NSVR computada e, conseqüentemente, a política de ganho alterada.

A Figura 6.3 mostra as curvas de desalinhamento normalizado para  $K_c = 14$  e dois valores de  $K_a$ , isto é, 8 e 32. Pode-se verificar que para esses valores de ganho de amplificação ( $8 \leq K_a \leq 32$ ) o algoritmo proposto exhibe comportamento semelhante.

A Figura 6.4 ilustra as curvas de desalinhamento normalizado para  $K_a = 10$  usando agora  $K_c$  igual a 8 e 32. Dessa figura, observa-se que o limiar  $K_c$  pode ser escolhido dentro de uma faixa de valores ( $8 \leq K_c \leq 32$ ) sem praticamente alterar o comportamento do algoritmo.



**Figura 6.3** - Exemplo 6.1. Curvas de desalinhamento normalizado do algoritmo LAF-PNLMS usando  $K_a$  igual a 8 e 32 com  $K_c = 14$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.



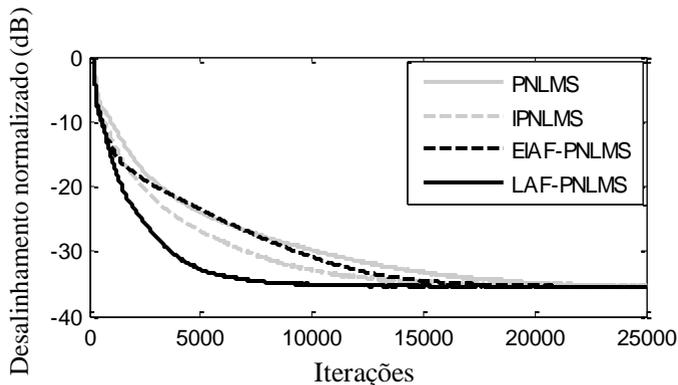
**Figura 6.4** - Exemplo 6.1. Curvas de desalinhamento normalizado do algoritmo LAF-PNLMS usando  $K_c$  igual a 8 e 32 com  $K_a = 10$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.

### 6.5.2 Exemplo 6.2

Este exemplo tem por objetivo avaliar a velocidade de convergência do algoritmo LAF-PNLMS comparando-o com os algoritmos PNLMS, IPNLMS e EIAF-PNLMS para dados de entrada

branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Além disso, o comportamento da distribuição de ganho dos algoritmos é avaliado considerando três coeficientes distintos do filtro adaptativo, isto é, o coeficiente de maior magnitude, o de magnitude intermediária e um coeficiente inativo. Para o algoritmo PNLMS, o passo de adaptação usado é  $\beta = 0,1$ . Para o algoritmo IPNLMS, são utilizados  $\beta = 0,11$  e  $\alpha = 0$ . Para o algoritmo EIAF-PNLMS, usa-se  $\beta = 0,11$  e  $K = 5 \times 10^{-3}$ . Para o algoritmo LAF-PNLMS, os parâmetros são  $\beta = 0,11$ ,  $\Delta = N/4$ ,  $K_a = 10$ ,  $K_c = 14$  e  $A_{\min} = 0,05$ .

A Figura 6.5 mostra as curvas de desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS. Pode-se observar que o algoritmo proposto supera em termos de velocidade de convergência os demais algoritmos avaliados.



**Figura 6.5** - Exemplo 6.2. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária.

A Tabela 6.2 mostra o valor máximo e total do ganho atribuído durante todo o processo adaptativo para o coeficiente de maior magnitude [ $w_{\max} = w_{199}(n)$ ], para o de magnitude intermediária [ $w_{\text{int}} = w_{196}(n)$ ] e para o coeficiente inativo [ $w_{\text{inativo}} = w_{222}(n)$ ]. Note que para os casos dos coeficientes ativos [ $w_{199}(n)$  e  $w_{196}(n)$ ], o valor do ganho máximo

atribuído pelo algoritmo LAF-PNLMS é bem superior ao atribuído pelos algoritmos PNLMS e IPNLMS e semelhante ao atribuído pelo EIAF-PNLMS; contudo, considerando o ganho ao longo do processo adaptativo, tais coeficientes são os que recebem menos ganho dentre todos os algoritmos aqui avaliados. Por outro lado, para o coeficiente inativo, tanto o ganho máximo quanto o ganho total distribuído são superiores ao observado pelos demais. Esse resultado mostra que, independente da magnitude do coeficiente, a política de distribuição de ganho do algoritmo LAF-PNLMS atribui, em algum momento, ganhos elevados a todos os coeficientes do filtro adaptativo; como consequência, aumentando a velocidade de convergência global do algoritmo.

**Tabela 6.2**  
VALOR MÁXIMO E TOTAL DE GANHO DISTRIBUÍDO  
SOBRE 25000 ITERAÇÕES

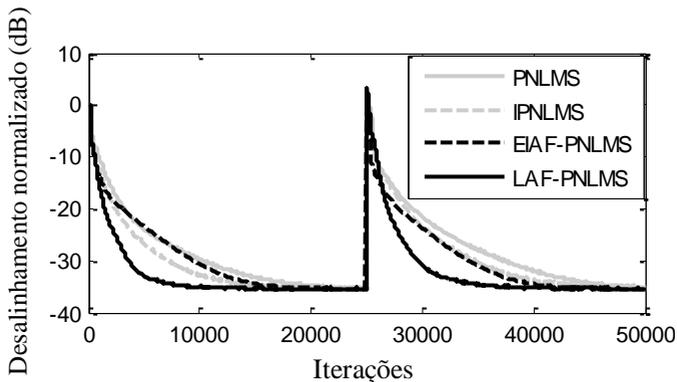
Coeficiente/Indicador		LAF-PNLMS	PNLMS	IPNLMS	EIAF-PNLMS
$w_{199}(n)$	$\max[g_{199}(n)]$	0,4439	0,1511	0,2105	0,4865
	$\sum_n g_{199}(n)$	424,66	2816,2	2673,7	540,47
$w_{196}(n)$	$\max[g_{196}(n)]$	0,1503	0,0491	0,0512	0,1325
	$\sum_n g_{196}(n)$	318,01	1066,0	1032,4	211,20
$w_{222}(n)$	$\max[g_{222}(n)]$	0,0064	0,0020	0,0019	0,0020
	$\sum_n g_{222}(n)$	126,65	28,52	42,05	45,93

### 6.5.3 Exemplo 6.3

Neste exemplo, o comportamento do algoritmo LAF-PNLMS é verificado considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário e parâmetros do exemplo anterior são utilizados, porém agora, no instante  $n = 25000$ , o vetor de coeficientes da planta

$\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita, alterando assim a posição de todos os coeficientes ativos.

A Figura 6.6 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS. Pode-se verificar que o algoritmo LAF-PNLMS apresenta a maior velocidade de convergência dentre os algoritmos avaliados mesmo na presença de perturbações abruptas na resposta ao impulso da planta.



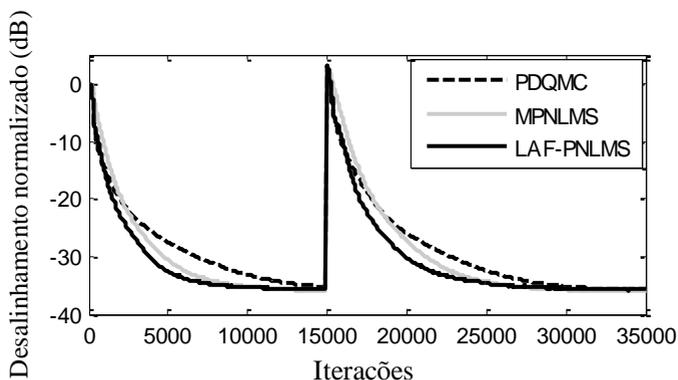
**Figura 6.6** - Exemplo 6.3. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 25000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

### 6.5.4 Exemplo 6.4

Este exemplo avalia a velocidade de convergência do algoritmo LAF-PNLMS comparando-o com os algoritmos PDQMC e MPNLMS considerando o mesmo cenário do exemplo anterior. Para o algoritmo PDQMC, utiliza-se passo de adaptação  $\beta = 0,1$ ,  $\alpha = 0,99$  e  $\omega = 5$ . Para o algoritmo MPNLMS, os parâmetros são  $\beta = 0,1$ ,  $\mu = 1000$  e  $\rho = 0,05$ . Para o algoritmo LAF-PNLMS, considera-se  $\beta = 0,1$ ,  $\Delta = N/4$ ,  $K_a = 10$ ,  $K_c = 14$  e  $A_{\min} = 0,05$ .

A Figura 6.7 mostra as curvas de desalinhamento normalizado dos algoritmos PDQMC, MPNLMS e LAF-PNLMS. Observe que o

algoritmo proposto supera em termos de velocidade de convergência os demais algoritmos avaliados independentemente da perturbação na planta.

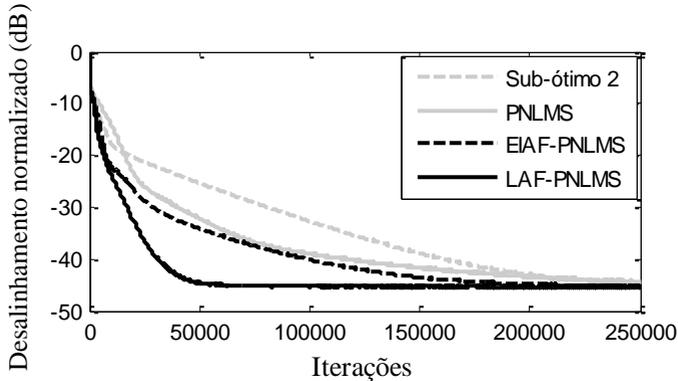


**Figura 6.7** - Exemplo 6.4. Desalinhamento normalizado dos algoritmos PDQMC, MPNLMS e LAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n=15000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

### 6.5.5 Exemplo 6.5

Neste exemplo, o comportamento do algoritmo LAF-PNLMS para entradas correlacionadas é apresentado. Para tal, a variância do ruído de medição é  $\sigma_v^2 = 0,0001$  (SNR = 40 dB) e o sinal de entrada, com média zero e variância unitária, é obtido através do processo AR(2) dado por (4.13) usando  $\sigma_\eta^2 = 0,36$ ,  $b_1 = 0,75$  e  $b_2 = -0,75$ , resultando em uma dispersão dos autovalores da matriz de autocorrelação do vetor de entrada  $\chi = 120$ . Além disso, a velocidade de convergência para entrada correlacionada é comparada com as dos algoritmos PNLMS, EIAF-PNLMS e Sub-ótimo 2. Para o algoritmo PNLMS, o passo de adaptação é  $\beta = 0,1$ . Para o EIAF-PNLMS, usa-se  $\beta = 0,11$  e  $K = 3 \times 10^{-4}$ . Para o Sub-ótimo 2, utiliza-se  $\beta = 0,11$ ,  $\alpha = 0,99$  e  $\omega = 5$ . Para o algoritmo LAF-PNLMS, são utilizados os parâmetros  $\beta = 0,11$ ,  $\Delta = N$ ,  $K_a = 30$ ,  $K_c = 30$  e  $A_{\min} = 0,01$ .

A Figura 6.8 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e LAF-PNLMS. Note que, para o caso de entrada correlacionada, o algoritmo proposto novamente apresenta velocidade de convergência maior do que a observada para os demais algoritmos avaliados.

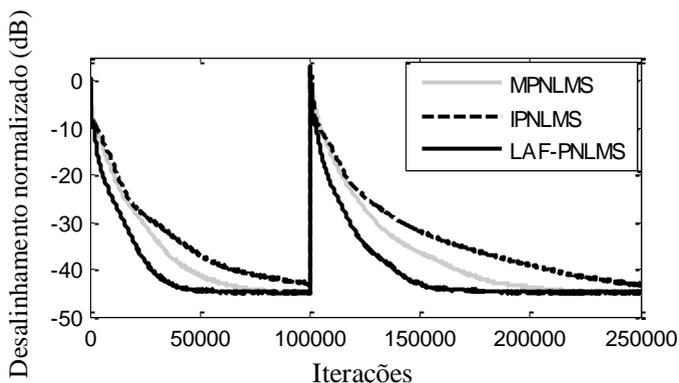


**Figura 6.8** – Exemplo 6.5. Curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e LAF-PNLMS para sinal de entrada correlacionado.

### 6.5.6 Exemplo 6.6

Este exemplo compara a velocidade de convergência dos algoritmos IPNLMS, MPNLMS e LAF-PNLMS para dados de entrada coloridos considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário do exemplo anterior é utilizado, porém, no instante  $n=100000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita. Para o algoritmo IPNLMS, usa-se  $\beta=0,1$  e  $\alpha=0$ . Para o algoritmo MPNLMS, considera-se  $\beta=0,1$ ,  $\mu=1000$  e  $\rho=0,05$ . Para o algoritmo LAF-PNLMS, os mesmos parâmetros do exemplo anterior são utilizados.

A Figura 6.9 ilustra as curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e LAF-PNLMS. Pode-se observar que o LAF-PNLMS supera os algoritmos avaliados para entrada correlacionada independente da perturbação abrupta na planta.



**Figura 6.9** – Exemplo 6.6. Curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e LAF-PNLMS para sinal de entrada correlacionado e SNR = 40 dB. O vetor de coeficientes da planta  $w_C^0$  é deslocado em  $n = 100000$  de 12 amostras para a direita.

## 6.6 Algoritmo Adaptativo Proporcional Baseado em Função de Amplificação de Ganho Proporcional

Nesta seção, utilizando uma função de amplificação proporcional para estabelecer o ganho atribuído a coeficientes ativos, um novo algoritmo, aqui denominado PAF-PNLMS, é introduzido. Na sequência, uma breve discussão sobre a política de distribuição de ganho do novo algoritmo é apresentada.

### 6.6.1 Algoritmo PAF-PNLMS

O algoritmo PAF-PNLMS usa uma função de amplificação proporcional à magnitude do coeficiente do filtro adaptativo para aumentar o ganho atribuído a coeficientes ativos e, como resultado, acelerar sua taxa de convergência. De forma semelhante ao realizado no algoritmo LAF-PNLMS, o ganho comum atribuído aos coeficientes que alcançaram a vizinhança de seus valores ótimos é definido ser maior do que o ganho atribuído a coeficientes inativos e, dessa forma, manter velocidade de convergência maior para coeficientes ativos em relação aos

inativos. Assim, para o algoritmo PAF-PNLMS, o ganho individual é dado por

$$g_i^{\text{PP}}(n) = \frac{\gamma_i^{\text{PP}}(n)}{\Phi^{\text{PP}}(n)} \quad (6.16)$$

com

$$\gamma_i^{\text{PP}}(n) = \begin{cases} K_c \gamma_{\min}^{\text{P}}(n), & V_i(n) \leq \varepsilon \\ \max[\gamma_{\min}^{\text{P}}(n), a |w_i(n)| + b], & \text{caso contrário} \end{cases} \quad (6.17)$$

onde os parâmetros da função de amplificação são definidos como

$$a \triangleq \frac{K_m w_{\max}(n) - \gamma_{\min}^{\text{P}}(n)}{w_{\max}(n) - \gamma_{\min}^{\text{P}}(n)} \quad (6.18)$$

e

$$b \triangleq (1 - a) \gamma_{\min}^{\text{P}}(n), \quad (6.19)$$

sendo  $K_m \geq 1$  o valor da amplificação do ganho do coeficiente de maior magnitude  $w_{\max}(n)$ .

A Tabela 6.3 resume o algoritmo PAF-PNLMS. Observe que o novo algoritmo mantém a estratégia de aplicar o cálculo da NSVR apenas para coeficientes cuja magnitude é maior ou igual a  $A_{\min} \max[|\mathbf{w}(n)|]$ , visando assim reduzir a complexidade computacional do algoritmo proposto. Além disso, para evitar divisão por zero em (6.18), a amplificação do ganho é iniciada somente após um período de aprendizagem de  $N$  amostras, isto é, do comprimento do filtro adaptativo.

**Tabela 6.3** – ALGORITMO PAF-PNLMS

Inicializar:

$$\mathbf{x}(0) = \mathbf{w}(0) = 0$$

Para  $n = 1, 2, 3, \dots$

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

$$\gamma_{\min}^P(n) = \rho \max[\delta, |w_1(n)|, |w_2(n)|, \dots, |w_N(n)|]$$

$$\gamma_i^{PP}(n) = \max[\gamma_{\min}^P(n), a |w_i(n)| + b], \quad 1 \leq i \leq N$$

$$a = \begin{cases} 1, & n < N \\ \frac{K_m w_{\max}(n) - \gamma_{\min}^P(n)}{w_{\max}(n) - \gamma_{\min}^P(n)}, & n \geq N \end{cases}$$

$$b = (1 - a) \gamma_{\min}^P(n)$$

Se  $|w_i(n)| \geq A_{\min} \max[|\mathbf{w}(n)|], \quad 1 \leq i \leq N$

$$Q_i(n) = |w_i(n)| + |w_i(n-1)| + \dots + |w_i(n - \frac{\Delta}{2} + 1)|,$$

$$n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

$$P_i(n) = |w_i(n - \frac{\Delta}{2})| + |w_i(n - \frac{\Delta}{2} - 1)| + \dots + |w_i(n - \Delta + 1)|,$$

$$n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

$$V_i(n) = \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, \quad n = N + m\Delta, \quad m = 1, 2, 3, \dots$$

Se  $V_i(n) \leq \varepsilon$

$$\gamma_i^{PP}(n) = K_c \gamma_{\min}^P(n), \quad n \geq N + \Delta$$

$$g_i^{PP}(n) = \frac{\gamma_i^{PP}(n)}{\sum_{j=1}^N \gamma_j^{PP}(n)}, \quad 1 \leq i \leq N$$

$$\mathbf{G}^{PP}(n) = \text{diag}[g_1^{PP}(n) \ g_2^{PP}(n) \ \dots \ g_N^{PP}(n)]$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta \mathbf{G}^{PP}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}^{PP}(n) \mathbf{x}(n) + \xi}$$

### 6.6.2 Distribuição de Ganho do Algoritmo PAF-PNLMS

A distribuição de ganho do algoritmo PAF-PNLMS pode ser classificada conforme a seguir:

- i) **Ganho dos coeficientes que alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) \leq \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  tenha quase atingido a convergência e seu ganho associado é dado por

$$g_c^{PP}(n) = \frac{1}{\Phi^{PP}(n)} K_c \gamma_{\min}^P(n). \quad (6.20)$$

- ii) **Ganho dos coeficientes que ainda não alcançaram a vizinhança de seus valores ótimos.** Se  $V_i(n) > \varepsilon$ , é assumido que o  $i$ -ésimo coeficiente  $w_i(n)$  não atingiu a convergência e seu ganho associado é dado por

$$g_{nc,i}^{PP}(n) = \begin{cases} \frac{1}{\Phi^{PP}(n)} \gamma_{\min}^P(n), & |w_i(n)| < \gamma_{\min}^P(n) \\ \frac{1}{\Phi^{LP}(n)} (a |w_i(n)| + b), & |w_i(n)| \geq \gamma_{\min}^P(n). \end{cases} \quad (6.21)$$

Agora, considerando que o ganho total distribuído entre os coeficientes do filtro adaptativo a cada iteração no algoritmo PAF-PNLMS é constante e igual a 1, o ganho total pode ser obtido por

$$g_{\text{total}}^{PP}(n) = \frac{1}{\Phi^{PP}(n)} \left[ \sum_{i \in (A-C)} (a |w_i(n)| + b) + \sum_{i \in B} \gamma_{\min}^P(n) + \sum_{i \in C} K_c \gamma_{\min}^P(n) \right] = 1 \quad (6.22)$$

onde  $A$ ,  $B$  e  $C$  são os conjuntos de índices de posições dos coeficientes estabelecidos em (6.15).

Note que no início do processo adaptativo (quando coeficientes ativos ainda não tenham alcançado a vizinhança de seus valores ótimos) todos os coeficientes ativos têm seus ganhos amplificados de forma proporcional à sua magnitude através dos parâmetros  $a$  e  $b$ . Tal amplificação é resultado da função afim  $(a |w_i(n)| + b)$  na qual o ganho

atribuído ao coeficiente de maior magnitude seja igual a  $K_m |w_i(n)|$  e, para os coeficientes com amplitude igual a  $\gamma_{\min}^P$ , o ganho não se altera. Em contraste com o algoritmo LAF-PNLMS, que amplifica todos os coeficientes ativos igualmente, quanto mais significativo for o coeficiente, maior é a amplificação de ganho realizada pelo algoritmo PAF-PNLMS. Sempre que um coeficiente ativo atinge a vizinhança de seu valor ótimo, seu ganho é reduzido para  $K_c \gamma_{\min}^P(n)$ . Em uma situação ideal (em regime permanente) todos os coeficientes ativos receberiam o ganho  $K_c \frac{\gamma_{\min}^P(n)}{\phi_{PP}^P(n)}$ , ao passo que todos os coeficientes inativos receberiam  $\frac{\gamma_{\min}^P(n)}{\phi_{PP}^P(n)}$ .

## 6.7 Resultados de Simulação do Algoritmo PAF-PNLMS

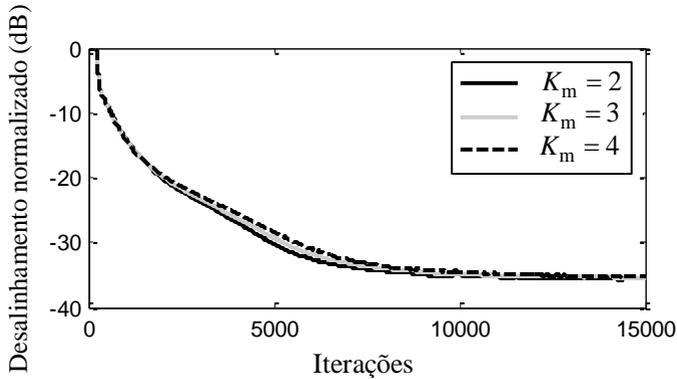
Nesta seção, considerando um problema de identificação de sistemas, simulações de MC (média de 100 realizações independentes) são realizadas para avaliar o algoritmo proposto. Para tal, seis exemplos são apresentados. Em todos os exemplos, utiliza-se resposta ao impulso da planta  $\mathbf{w}_C^0$  [ $S(\mathbf{w}_C^0) = 0,90$ ],  $\xi = 10^{-4}$ ,  $\rho = 0,01$  e  $\delta = 0,01$ . Para o algoritmo PAF-PNLMS, o valor do limiar de convergência usado é  $\varepsilon = 5 \times 10^{-2}$ .

### 6.7.1 Exemplo 6.7

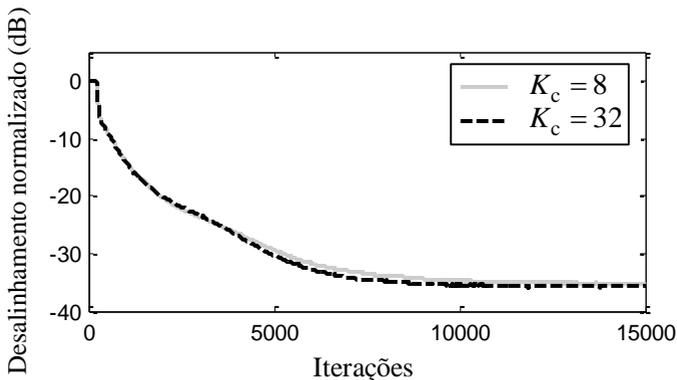
Neste exemplo, o comportamento do algoritmo PAF-PNLMS com respeito aos parâmetros  $K_m$  e  $K_c$  é avaliado. Para tal, considera-se dados de entrada branco com variância  $\sigma_x^2 = 1$ , variância do ruído de medição  $\sigma_v^2 = 0,001$  (30 dB SNR), passo de adaptação  $\beta = 0,1$ , número de amostras  $\Delta = N$  e  $A_{\min} = 0,005$ .

A Figura 6.10 mostra as curvas de desalinhamento normalizado para  $K_c = 18$  e três valores de  $K_m$ , isto é, 2, 3 e 4. Note que para esses valores de ganho de amplificação ( $2 \leq K_m \leq 4$ ) o algoritmo PAF-PNLMS exhibe comportamento semelhante.

A Figura 6.11 ilustra as curvas de desalinhamento normalizado para  $K_m = 2$  usando agora  $K_c$  igual a 8 e 32. Pode-se observar que o limiar  $K_c$  pode ser escolhido dentro de uma faixa de valores  $8 \leq K_c \leq 32$ , o qual mantém o comportamento do algoritmo proposto similar.



**Figura 6.10** - Exemplo 6.7. Curvas de desalinhamento normalizado do algoritmo PAF-PNLMS usando  $K_m$  igual a 2, 3 e 4 com  $K_c = 18$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.

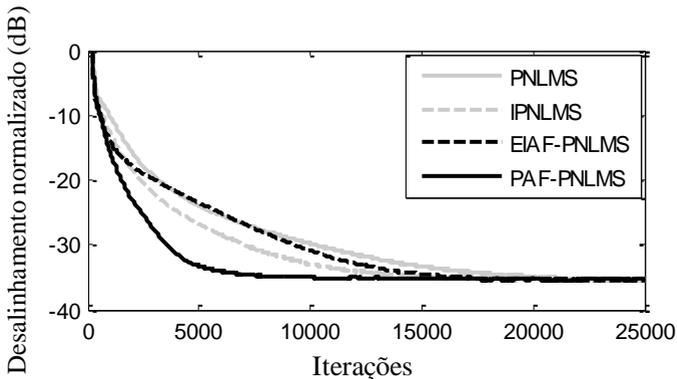


**Figura 6.11** - Exemplo 6.7. Curvas de desalinhamento normalizado do algoritmo PAF-PNLMS usando  $K_c$  igual a 8 e 32 com  $K_m = 2$  para sinal de entrada branco com variância  $\sigma_x^2$  unitária.

### 6.7.2 Exemplo 6.8

Este exemplo visa comparar a velocidade de convergência do algoritmo PAF-PNLMS com os algoritmos PNLMS, IPNLMS e EIAF-PNLMS para um sinal de entrada branco com variância  $\sigma_x^2 = 1$  e variância do ruído de medição  $\sigma_v^2 = 0,001$  (SNR = 30 dB). Além disso, o comportamento da distribuição de ganho dos algoritmos é analisado, considerando três coeficientes distintos do filtro adaptativo, conforme mostrado no Exemplo 6.2. Para o algoritmo PNLMS, o passo de adaptação é  $\beta = 0,1$ . Para o algoritmo IPNLMS, são utilizados  $\beta = 0,11$  e  $\alpha = 0$ . Para o algoritmo EIAF-PNLMS, usa-se  $\beta = 0,11$  e  $K = 5 \times 10^{-3}$ . Para o algoritmo PAF-PNLMS, são considerados  $\beta = 0,11$ ,  $\Delta = N/4$ ,  $K_m = 2$ ,  $K_c = 18$  e  $A_{\min} = 0,005$ .

A Figura 6.12 mostra as curvas de desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS. Observe que o algoritmo proposto apresenta maior velocidade de convergência do que os demais algoritmos avaliados considerando uma planta com elevada esparsidade.



**Figura 6.12** - Exemplo 6.8. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária.

A Tabela 6.4 mostra o valor máximo e total do ganho atribuído durante todo o processo adaptativo para o coeficiente de maior magnitude [ $w_{\max} = w_{199}(n)$ ], para o de magnitude intermediária [ $w_{\text{int}} = w_{196}(n)$ ] e para um coeficiente inativo [ $w_{\text{inativo}} = w_{222}(n)$ ]. Novamente, note que para os casos dos coeficientes ativos [ $w_{199}(n)$  e  $w_{196}(n)$ ], o valor do ganho máximo atribuído pelo algoritmo PAF-PNLMS é bem superior ao atribuído pelos algoritmos PNLMS e IPNLMS e semelhante ao atribuído pelo EIAF-PNLMS; contudo, tais coeficientes são os que recebem menos ganho dentre todos os algoritmos avaliados ao longo do processo adaptativo. Por outro lado, para o coeficiente inativo, o ganho máximo e o ganho total distribuído são superiores ao observado pelos demais algoritmos avaliados. Tal resultado mostra que, independente da magnitude do coeficiente, ao longo do processo adaptativo, o algoritmo PAF-PNLMS atribui ganhos elevados a todos os coeficientes do filtro adaptativo e, dessa forma, alcança maior velocidade de convergência.

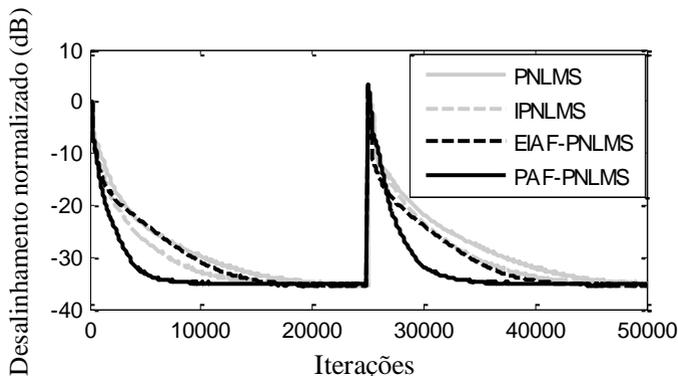
**Tabela 6.4**  
 VALOR MÁXIMO E TOTAL DE GANHO DISTRIBUÍDO  
 SOBRE 25000 ITERAÇÕES

Coeficiente/Indicador		PAF-PNLMS	PNLMS	IPNLMS	EIAF-PNLMS
$w_{199}(n)$	$\max[g_{199}(n)]$	0,4610	0,1511	0,2105	0,4865
	$\sum_n g_{199}(n)$	452,24	2816,2	2673,7	540,47
$w_{196}(n)$	$\max[g_{196}(n)]$	0,1720	0,0491	0,0512	0,1325
	$\sum_n g_{196}(n)$	345,44	1066,0	1032,4	211,20
$w_{222}(n)$	$\max[g_{222}(n)]$	0,0108	0,0020	0,0019	0,0020
	$\sum_n g_{222}(n)$	207,79	28,52	42,05	45,93

### 6.7.3 Exemplo 6.9

Neste exemplo, o comportamento do algoritmo PAF-PNLMS considerando a presença de uma perturbação na planta é avaliado. Para tal, o mesmo cenário e parâmetros do exemplo anterior são utilizados, porém agora, no instante  $n = 25000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

A Figura 6.13 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS. Pode-se verificar que o algoritmo PAF-PNLMS apresenta a maior velocidade de convergência dentre os algoritmos avaliados, independente da perturbação na resposta ao impulso da planta.

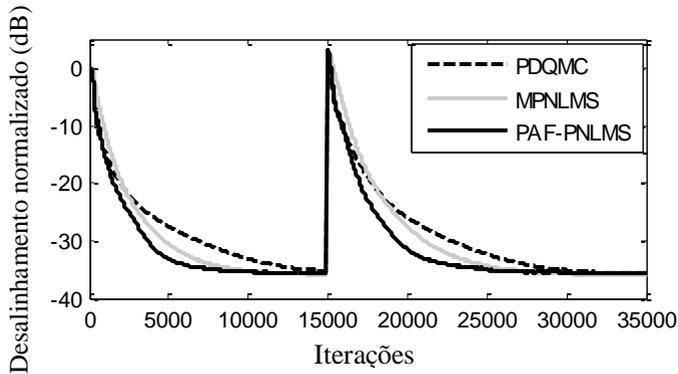


**Figura 6.13** - Exemplo 6.9. Desalinhamento normalizado dos algoritmos PNLMS, IPNLMS, EIAF-PNLMS e PAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 25000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

### 6.7.4 Exemplo 6.10

Este exemplo compara a velocidade de convergência dos algoritmos PAF-PNLMS, PDQMC e MPNLMS considerando o mesmo cenário do exemplo anterior. Para o algoritmo PDQMC, utiliza-se  $\beta = 0,1$ ,  $\alpha = 0,99$  e  $\omega = 5$ . Para o algoritmo MPNLMS, são considerados  $\beta = 0,1$ ,  $\mu = 1000$  e  $\rho = 0,05$ . Para o algoritmo PAF-PNLMS, são utilizados  $\beta = 0,1$ ,  $\Delta = N/4$ ,  $K_m = 2$ ,  $K_c = 18$  e  $A_{\min} = 0,005$ .

A Figura 6.14 mostra as curvas de desalinhamento normalizado dos algoritmos avaliados. Note que o algoritmo proposto supera em termos de velocidade de convergência os demais algoritmos independentemente da perturbação na planta.



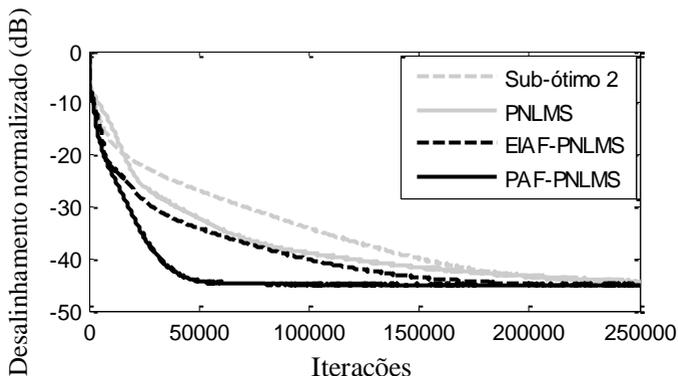
**Figura 6.14** - Exemplo 6.10. Desalinhamento normalizado dos algoritmos PDQMC, MPNLMS e PAF-PNLMS considerando um sinal de entrada branco com variância  $\sigma_x^2$  unitária. Quando  $n = 15000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

### 6.7.5 Exemplo 6.11

Neste exemplo, o algoritmo PAF-PNLMS é avaliado considerando entrada correlacionada e variância do ruído de medição  $\sigma_v^2 = 0,0001$  (SNR = 40 dB). Para tal, o sinal de entrada, com média zero e variância unitária, é obtido através do processo AR(2) dado por (4.13) usando  $\sigma_\eta^2 = 0,36$ ,  $b_1 = 0,75$  e  $b_2 = -0,75$ , resultando em uma dispersão dos autovalores da matriz de autocorrelação do vetor de entrada  $\chi = 120$ . Além disso, sua velocidade de convergência é comparada com os algoritmos PNLMS, EIAF-PNLMS e Sub-ótimo 2. Para o algoritmo PNLMS, o passo de adaptação é  $\beta = 0,1$ . Para o EIAF-PNLMS, os parâmetros utilizados são  $\beta = 0,11$  e  $K = 3 \times 10^{-4}$ . Para o Sub-ótimo 2, são considerados  $\beta = 0,11$ ,  $\alpha = 0,99$  e  $\omega = 5$ . Para o algoritmo

PAF-PNLMS, utiliza-se  $\beta = 0,11$ ,  $\Delta = N$ ,  $K_m = 2$ ,  $K_c = 30$  e  $A_{\min} = 0,01$ .

A Figura 6.15 ilustra as curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e PAF-PNLMS. Note que o algoritmo proposto apresenta velocidade de convergência maior do que a observada nos demais algoritmos avaliados para o caso de entrada correlacionada.



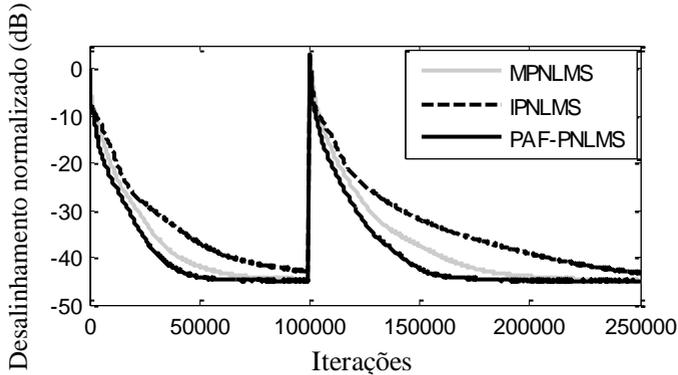
**Figura 6.15** – Exemplo 6.11. Curvas de desalinhamento normalizado dos algoritmos PNLMS, EIAF-PNLMS, Sub-ótimo 2 e PAF-PNLMS para sinal de entrada correlacionado.

### 6.7.6 Exemplo 6.12

Este exemplo compara a velocidade de convergência dos algoritmos IPNLMS, MPNLMS e PAF-PNLMS para dados de entrada correlacionados considerando a presença de uma perturbação na planta. Para tal, o mesmo cenário do exemplo anterior é utilizado, porém, no instante  $n = 100000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^o$  é deslocado de 12 amostras para a direita. Para o algoritmo IPNLMS, usa-se  $\beta = 0,1$  e  $\alpha = 0$ . Para o algoritmo MPNLMS, considera-se  $\beta = 0,1$ ,  $\mu = 1000$  e  $\rho = 0,05$ . Para o algoritmo PAF-PNLMS, são utilizados os mesmos parâmetros do exemplo anterior.

A Figura 6.16 ilustra as curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e PAF-PNLMS. Pode-se observar que o

PAF-PNLMS supera os demais algoritmos avaliados para sinal de entrada correlacionado, independente da perturbação abrupta na planta.



**Figura 6.16** – Exemplo 6.12. Curvas de desalinhamento normalizado dos algoritmos IPNLMS, MPNLMS e PAF-PNLMS para sinal de entrada correlacionado e SNR = 40 dB.  $n = 100000$ , o vetor de coeficientes da planta  $\mathbf{w}_C^0$  é deslocado de 12 amostras para a direita.

## 6.8 Complexidade Computacional

Com respeito à complexidade computacional dos algoritmos propostos, o uso de uma amplitude mínima do coeficiente para a aplicação da política de transferência de ganho durante o processo adaptativo [através do parâmetro  $A_{\min}$ ] diminui a carga computacional decorrente do cálculo da NSVR. Tal estratégia praticamente não afeta o comportamento global do algoritmo, visto que transferir ganho de coeficientes com pequenas magnitudes proporciona pouca ou nenhuma modificação em sua distribuição de ganho. Assim, definindo  $N_{\text{conv}}$  como o número de coeficientes do filtro adaptativo com magnitude maior ou igual a  $A_{\min} \max[\|\mathbf{w}(n)\|]$ , o número de operações para o cálculo da NSVR é reduzido para  $N_{\text{conv}}(\Delta - 1)$  adições,  $N_{\text{conv}}$  divisões e  $2N_{\text{conv}}$  comparações a cada  $\Delta$  iterações. Note que para casos de respostas ao impulso de plantas com grau de esparsidade elevado,  $N_{\text{conv}}$  tende a ser muito menor do que  $N$ . Por exemplo, considerando a planta  $\mathbf{w}_C^0$  com

esparsidade 0,90 e um valor de limiar de amplitude  $A_{\min} = 0,01$ ,  $N_{\text{conv}} = 48$ , isto é, dos 512 coeficientes do filtro adaptativo (em média) apenas em 48 coeficientes [menos do que 10% do total de coeficientes] a nova política de distribuição de ganho é aplicada.

A Tabela 6.5 mostra a complexidade computacional dos algoritmos LAF-PNLMS e PAF-PNLMS para  $\Delta = N$ . Assim como na Seção 4.9, em ambos algoritmos, não foram consideradas no cálculo da complexidade as  $N$  divisões em (2.5), visto que multiplicando o numerador e o denominador de (2.1) por (2.6),  $N$  divisões são trocadas por uma multiplicação.

**Tabela 6.5 - COMPLEXIDADE COMPUTACIONAL PARA  $\Delta = N$**

Algoritmos	Intervalo	Multiplificações	Adições	Divisões	Comparações
LAF-PNLMS	A cada iteração	$6N + 3$	$4N - 1$	1	$3N$
	A cada $N$ iterações	$N_{\text{conv}}$	$N_{\text{conv}}(N - 1)$	$N_{\text{conv}}$	$N + N_{\text{conv}}$
PAF-PNLMS	A cada iteração	$6N + 5$	$5N + 2$	2	$3N$
	A cada $N$ iterações	$N_{\text{conv}}$	$N_{\text{conv}}(N - 1)$	$N_{\text{conv}}$	$N + N_{\text{conv}}$

## **6.9 Conclusões**

Neste capítulo, uma política de distribuição de ganho proporcional para algoritmos operando em ambientes cujas respostas ao impulso da planta exibem elevada esparsidade foi apresentada. Através de uma função de amplificação de ganho, a nova estratégia amplifica o ganho alocado a coeficientes ativos e, conforme esses coeficientes alcançam a proximidade de seus valores ótimos, seu ganho é reduzido e redistribuído aos demais coeficientes do filtro adaptativo. Além disso, considerando a política de ganho proposta, dois novos algoritmos, denominados LAF-PNLMS e PAF-PNLMS, foram introduzidos. Para plantas com elevado grau de esparsidade, os novos algoritmos apresentam maior velocidade de convergência quando comparados com os principais algoritmos do tipo proporcional. Resultados de simulações numéricas comprovam a eficácia da nova política de distribuição de ganho.



### COMENTÁRIOS E CONCLUSÕES FINAIS

Neste trabalho, uma nova política de distribuição de ganho para algoritmos tipo proporcional baseada na convergência do coeficiente foi apresentada. Através da avaliação da taxa de variação da magnitude dos coeficientes três novos algoritmos, denominados NSVR-PNLMS, NSVR-IPNLMS e NSVR-IAF-PNLMS, foram derivados a partir do PNLMS, IPNLMS e IAF-PNLMS. Nos algoritmos propostos, ganhos associados aos coeficientes que já atingiram a proximidade de seus valores ótimos são reduzidos e redistribuídos a outros coeficientes; dessa forma, alcançando uma melhor condição de operação conforme o processo adaptativo evolui. Os novos algoritmos, considerando plantas com alta e média esparsidades, superaram os algoritmos originais em termos de velocidade de convergência, resposta a perturbações na planta e habilidade de rastreamento. Além disso, uma nova abordagem para migrar do ganho proporcional à estimativa do desvio quadrático médio dos coeficientes para o ganho unitário do algoritmo PDQMC foi desenvolvida. Como resultado, o novo algoritmo, além de eliminar a necessidade do conhecimento da variância do ruído de medição presente no sistema, superou o PDQMC em relação à velocidade de convergência e resposta a perturbações na planta, considerando tanto sinal de entrada branco quanto correlacionado. Por fim, uma nova política de distribuição de ganho para algoritmos tipo proporcional para operação em ambientes com elevada esparsidade foi proposta. A nova política utiliza uma função de amplificação do ganho de coeficientes ativos visando aumentar sua velocidade de convergência. A partir da nova política, dois novos algoritmos para operação em ambientes cuja resposta ao impulso da planta apresenta elevada esparsidade foram introduzidos. Tais algoritmos, conforme mostrado através de simulações numéricas, apresentaram velocidade de convergência superior a observada pelos algoritmos PNLMS, IPNLMS, MPNLMS, EIAF-PNLMS, PDQMC e Sub-ótimo 2.

No Capítulo 1, os conceitos básicos envolvendo filtragem adaptativa, tais como características, classificação e aplicações, além de um breve histórico da filtragem adaptativa proporcional foram apresentados. Também, os algoritmos LMS e NLMS foram brevemente discutidos.

No Capítulo 2, as equações gerais dos algoritmos tipo proporcional foram apresentadas, assim como os principais algoritmos da classe proporcional, tais como PNLMS, IPNLMS, IIPNLMS, MPNLMS, SC-PNLMS, IAF-PNLMS, EIAF-PNLMS e PDQMC foram revisitados, destacando-se suas características, vantagens e desvantagens.

No Capítulo 3, uma análise da distribuição de ganho e do comportamento dos coeficientes do filtro adaptativo do algoritmo PNLMS foram realizadas. Para tal, equações do ganho total e de cada coeficiente, diferenciando-os entre ativos e inativos, foram apresentadas. Também, simulações numéricas verificando as relações entre ganho atribuído e convergência foram realizadas. Além disso, uma metodologia para determinação da taxa de variação da magnitude dos coeficientes do filtro adaptativo foi introduzida e analisada em termos de complexidade computacional e eficácia.

No Capítulo 4, uma nova política de distribuição de ganho proporcional foi discutida. Além disso, considerando a política de ganho proposta, três novas versões melhoradas dos algoritmos PNLMS, IPNLMS e IAF-PNLMS foram apresentadas. Os novos algoritmos reduzem o ganho atribuído a coeficientes ativos que alcançaram a vizinhança de seus valores ótimos e o redistribuí dentro os demais coeficientes do filtro adaptativo. A verificação de convergência do coeficiente é efetuada pela avaliação da taxa de variação de sua magnitude. Resultados de simulações numéricas mostraram que os novos algoritmos apresentam melhor distribuição de ganho, superando o desempenho dos algoritmos originais em termos de velocidade de convergência, resposta a perturbações na planta e habilidade de rastreamento, tanto para plantas de alta quanto de média esparsidade.

No Capítulo 5, visando eliminar a dependência do conhecimento da variância do ruído de medição do algoritmo PDQMC, foi apresentada uma forma alternativa de migrar do ganho proporcional ao desvio quadrático médio dos coeficientes para o ganho unitário. A nova estratégia é baseada na autocorrelação do sinal de erro entre amostras adjacentes. O novo algoritmo, além de dispensar a necessidade do conhecimento da variância do ruído de medição, superou o PDQMC em termos de velocidade de convergência e resposta a perturbações na planta, tanto para respostas ao impulso exibindo alta quanto média esparsidade.

Resultados de simulação numérica mostraram que o novo algoritmo mantém seu desempenho independente do nível do ruído de medição presente no sistema.

No Capítulo 6, os algoritmos LAF-PNLMS e PAF-PNLMS foram introduzidos. Tais algoritmos usam uma função de amplificação de ganho visando acelerar a velocidade de convergência de coeficientes ativos. Além disso, para melhorar a distribuição de ganho durante o processo adaptativo, conforme os coeficientes ativos se aproximam de seus valores ótimos, seus ganhos são reduzidos e redistribuídos aos demais coeficientes. Os novos algoritmos apresentaram desempenho superior aos principais algoritmos da classe proporcional para casos em que a resposta ao impulso da planta exibe elevada esparsidade. Simulações numéricas confirmaram a eficácia da abordagem proposta.

Como proposta para trabalhos futuros, sugere-se a aplicação da métrica NSVR (como indicador de convergência individual do coeficiente) a outras classes de algoritmos adaptativos, tais como a de projeções afins e a de restrição por norma. Outra proposta para a continuação deste trabalho seria o desenvolvimento de modelos estocásticos dos novos algoritmos aqui concebidos. Trabalhos futuros podem ainda considerar a concepção de novas técnicas de avaliação da convergência individual dos coeficientes. Por fim, é sugerido o desenvolvimento de novas estratégias de migração de ganho proporcional para não proporcional à medida que o algoritmo se aproxima do regime permanente.



## Análise Estocástica da Taxa de Variação Suavizada e Normalizada da Magnitude do Coeficiente Adaptativo

Neste apêndice, o valor esperado da taxa de variação suavizada e normalizada (NSVR) da magnitude do coeficiente adaptativo é determinado. Assim, da Seção 3.4.1, a expressão da NSVR, é dada por

$$V_i(n) = \begin{cases} \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}, & n = m\Delta, \quad m = 1, 2, 3, \dots \\ V_i(n-1), & \text{caso contrário} \end{cases} \quad (\text{I.1})$$

com

$$Q_i(n) = \sum_{j=0}^{(\Delta-2)/2} |w_i(n-j)|, \quad i = 1, 2, \dots, N \quad (\text{I.2})$$

e

$$P_i(n) = \sum_{j=\Delta/2}^{\Delta-1} |w_i(n-j)|, \quad i = 1, 2, \dots, N. \quad (\text{I.3})$$

Aplicando o valor esperado em (I.1) tem-se

$$E[V_i(n)] = \begin{cases} E\left[\frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}\right], & n = m\Delta, \quad m = 1, 2, 3, \dots \\ E[V_i(n-1)] \end{cases} \quad (\text{I.4})$$

Agora, considerando que

$$E\left[\frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]}\right] \cong \frac{E[|Q_i(n) - P_i(n)|]}{E\{\min[Q_i(n), P_i(n)]\}} \quad (\text{I.5})$$

e ainda

$$E \left[ \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]} \right] \cong \frac{E[|Q_i(n) - P_i(n)|]}{\min\{E[Q_i(n)], E[P_i(n)]\}}, \quad (I.6)$$

pode-se determinar o valor esperado da NSVR a partir da determinação de (I.6).

Dessa forma, assumindo que o valor do coeficiente possui o mesmo sinal durante sua evolução, isto é, o coeficiente está evoluindo positivamente ou negativamente a um valor em regime,  $Q_i(n)$  e  $P_i(n)$  possuem o mesmo sinal e (I.6) pode ser aproximada por

$$E \left[ \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]} \right] \cong \frac{|E[Q_i(n)] - E[P_i(n)]|}{\min\{E[Q_i(n)], E[P_i(n)]\}}. \quad (I.7)$$

Substituindo-se (I.2) e (I.3) em (I.7), a expressão do valor esperado é dada por

$$E \left[ \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]} \right] \cong \frac{\left| E \left[ \sum_{j=0}^{(\Delta-2)/2} |w_i(n-j)| \right] - E \left[ \sum_{j=\Delta/2}^{\Delta-1} |w_i(n-j)| \right] \right|}{\min\{E \left[ \sum_{j=0}^{(\Delta-2)/2} |w_i(n-j)| \right], E \left[ \sum_{j=\Delta/2}^{\Delta-1} |w_i(n-j)| \right]\}}. \quad (I.8)$$

Visto que o valor esperado da soma das magnitudes é igual ao número de elementos vezes o valor médio das magnitudes, tem-se que

$$E \left[ \frac{|Q_i(n) - P_i(n)|}{\min[Q_i(n), P_i(n)]} \right] \cong \frac{\left| \frac{\Delta}{2} \overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}+1} - \frac{\Delta}{2} \overline{|w_i(n)|}_{n-\frac{\Delta}{2}}^{n-\Delta+1} \right|}{\min\{E \left[ \frac{\Delta}{2} \overline{|w_i(n)|}_n^{n-\frac{\Delta}{2}+1} \right], E \left[ \frac{\Delta}{2} \overline{|w_i(n)|}_{n-\frac{\Delta}{2}}^{n-\Delta+1} \right]\}} \quad (I.9)$$

onde  $\overline{|\cdot|}_a^b$  denota o valor médio da magnitude no período entre  $a$  e  $b$ .

Por fim, simplificando a constante comum entre os elementos de (I.9) obtém-se (3.16).

## Análise Estocástica da Autocorrelação Quadrática do Erro Normalizado

Para prover suporte teórico da independência de  $m(n)$  [versão suavizada e normalizada da autocorrelação quadrática do sinal de erro] com respeito à variância do ruído de medição presente no sistema, uma expressão para o valor esperado de  $c^2(n)/q^2(n)$  é aqui mostrada.

Para isso, considerando que o valor esperado da razão  $c^2(n)/q^2(n)$  é aproximadamente igual a razão dos valores esperados, isto é,

$$E\left[\frac{c^2(n)}{q^2(n)}\right] \cong \frac{E[c^2(n)]}{E[q^2(n)]} \quad (\text{II.1})$$

os seguintes valores esperados devem ser obtidos:  $E[c^2(n)]$  e  $E[q^2(n)]$ .

### a) Determinação de $E[c^2(n)]$

A versão suavizada da autocorrelação entre amostras adjacentes do sinal de erro é dada por

$$c(n) = \gamma c(n-1) + (1-\gamma)e(n)e(n-1). \quad (\text{II.2})$$

Agora, reescrevendo (II.2) em uma forma fechada e elevando os dois lados ao quadrado, obtém-se

$$c^2(n) = (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} e(n-j)e(n-j-1)e(n-i)e(n-i-1). \quad (\text{II.3})$$

Substituindo-se o erro instantâneo

$$e(n) = v(n) + \mathbf{z}^T(n)\mathbf{x}(n) \quad (\text{II.4})$$

em (II.3), resulta em

$$\begin{aligned}
 c^2(n) &= (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \{ [v(n-j) + \mathbf{z}^T(n-j)\mathbf{x}(n-j)] \\
 &\quad \times [v(n-j-1) + \mathbf{z}^T(n-j-1)\mathbf{x}(n-j-1)] \\
 &\quad \times [v(n-i) + \mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\
 &\quad \times [v(n-i-1) + \mathbf{z}^T(n-i-1)\mathbf{x}(n-i-1)] \}
 \end{aligned} \tag{II.5}$$

onde  $\mathbf{z}(n) = [z_1(n) \ z_2(n) \ \dots \ z_N(n)]^T$  denota o vetor de desvio dos coeficientes.

Agora, assumindo que  $v(n)$ ,  $\mathbf{z}(n)$  e  $\mathbf{x}(n)$  são descorrelacionados entre si e aplicando o valor esperado nos dois lados de (II.5), obtém-se

$$\begin{aligned}
 E[c^2(n)] &= (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \{ [E[v(n-j)v(n-j-1)v(n-i)v(n-i-1)] \\
 &\quad + E[v(n-j)v(n-i)\mathbf{z}^T(n-j-1)\mathbf{x}(n-j-1)\mathbf{z}^T(n-i-1)\mathbf{x}(n-i-1)] \\
 &\quad + E[v(n-j)v(n-j-1)\mathbf{z}^T(n-j-1)\mathbf{x}(n-j-1)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\
 &\quad + E[v(n-j-1)v(n-i)\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-i-1)\mathbf{x}(n-i-1)] \\
 &\quad + E[v(n-j-1)v(n-i-1)\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\
 &\quad + E[\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-j-1)\mathbf{x}(n-j-1) \\
 &\quad \times \mathbf{z}^T(n-i)\mathbf{x}(n-i)\mathbf{z}^T(n-i-1)\mathbf{x}(n-i-1)] \}.
 \end{aligned} \tag{II.6}$$

Então, usando o teorema da fatoração de momentos Gaussianos, definindo as matrizes  $\mathbf{K}(n) = E[\mathbf{z}(n)\mathbf{z}^T(n)]$ ,  $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ ,  $\mathbf{K}_j(n) = E[\mathbf{z}(n)\mathbf{z}^T(n-l)]$ ,  $\mathbf{R}_2 = E[\mathbf{x}(n)\mathbf{x}^T(n-2)]$  e considerando que os elementos de  $\mathbf{K}_{j-i}$  e  $\mathbf{K}_{i-j}$  pode ser negligenciado para  $|i-j| > 2$ , o requerido valor esperado é dado por

$$\begin{aligned}
 E[c^2(n)] &= (1-\gamma)^2 \left( \sum_{i=0}^{n-1} \gamma^{2i} \{ \sigma_v^4 + \sigma_v^2 \text{tr}[\mathbf{R}\mathbf{K}(n-i-1) + \mathbf{R}\mathbf{K}(n)] \} \right. \\
 &\quad \left. + \sum_{i=1}^{n-1} \gamma^{2i-1} \sigma_v^2 \text{tr}[\mathbf{R}_2 \mathbf{K}_2^T(n-i+1)] + \sum_{i=0}^{n-2} \gamma^{2i+1} \sigma_v^2 \text{tr}[\mathbf{R}_2^T \mathbf{K}_2(n-i)] \right).
 \end{aligned} \tag{II.7}$$

b) Determinação de  $E[q^2(n)]$

Agora, escrevendo a versão suavizada do erro quadrático

$$q(n) = \gamma q(n-1) + (1-\gamma)e^2(n) \quad (\text{II.8})$$

em uma forma fechada e elevando ao quadrado os dois lados, obtém-se

$$q^2(n) = (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} e^2(n-j)e^2(n-i). \quad (\text{II.9})$$

Substituindo (II.4) em (II.9), resulta em

$$q^2(n) = (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \{ [v(n-j) + \mathbf{z}^T(n-j)\mathbf{x}(n-j)]^2 [v(n-i) + \mathbf{z}^T(n-i)\mathbf{x}(n-i)]^2 \} \quad (\text{II.10})$$

sendo que, assumindo novamente que  $v(n)$ ,  $\mathbf{z}(n)$  e  $\mathbf{x}(n)$  são decorrelacionados entre si, o valor esperado de (II.10) pode ser escrito como

$$\begin{aligned} E[q^2(n)] = & (1-\gamma)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \{ [E[v(n-j)v(n-j)v(n-i)v(n-i)] \\ & + E[v(n-j)v(n-j)\mathbf{z}^T(n-i)\mathbf{x}(n-i)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\ & + 2E[v(n-j)v(n-i)\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\ & + 2E[v(n-j)v(n-i)\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \\ & + E[v(n-i)v(n-i)\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-j)\mathbf{x}(n-j)] \\ & + E[\mathbf{z}^T(n-j)\mathbf{x}(n-j)\mathbf{z}^T(n-j)\mathbf{x}(n-j) \\ & \times \mathbf{z}^T(n-i)\mathbf{x}(n-i)\mathbf{z}^T(n-i)\mathbf{x}(n-i)] \}. \end{aligned} \quad (\text{II.11})$$

Então, usando os mesmos passos previamente considerados para obter  $E[c^2(n)]$ , o valor esperado de  $q^2(n)$  é dado por

$$\begin{aligned} E[q^2(n)] = & (1-\gamma)^2 \left( \sum_{i=0}^{n-1} \gamma^{2i} \{ 3\sigma_v^4 + 6\sigma_v^2 \text{tr}[\mathbf{R}\mathbf{K}(n-i)] \} \right. \\ & \left. + \sum_{\substack{i=0 \\ i \neq j}}^{n-1} \sum_{j=0}^{n-1} \gamma^{i+j} \{ \sigma_v^4 + \sigma_v^2 \text{tr}[\mathbf{R}\mathbf{K}(n-i) + \mathbf{R}\mathbf{K}(n-j)] \} \right). \end{aligned} \quad (\text{II.12})$$

Finalmente, o valor esperado de  $E[c^2(n)/q^2(n)]$  é obtido substituindo-se (II.7) e (II.12) em (II.1), o que resulta em

$$\begin{aligned}
 E \left[ \frac{c^2(n)}{q^2(n)} \right] &\cong \frac{\sum_{i=0}^{n-1} \gamma^{2i} \{ \sigma_v^4 + \sigma_v^2 \text{tr}[\mathbf{RK}(n-i-1) + \mathbf{RK}(n)] \} \cdots}{\sum_{i=0}^{n-1} \gamma^{2i} \{ 3\sigma_v^4 + 6\sigma_v^2 \text{tr}[\mathbf{RK}(n-i)] \} \cdots} \\
 &\quad + \frac{\sum_{i=1}^{n-1} \gamma^{2i-1} \sigma_v^2 \text{tr}[\mathbf{R}_2 \mathbf{K}_2^T(n-i+1)] \cdots}{\sum_{i=0}^{n-1} \sum_{\substack{j=0 \\ i \neq j}}^{n-1} \gamma^{i+j} \{ \sigma_v^4 + \sigma_v^2 \text{tr}[\mathbf{RK}(n-i)] \} \cdots} \\
 &\quad + \frac{\sum_{i=0}^{n-2} \gamma^{2i+1} \sigma_v^2 \text{tr}[\mathbf{R}_2^T \mathbf{K}_2(n-i)]}{\sum_{i=0}^{n-1} \gamma^{2i} \{ \sigma_v^4 + \sigma_v^2 \text{tr}[\mathbf{RK}(n-i)] \} \cdots} \cdot \\
 &\quad \quad \quad + \mathbf{RK}(n-j) \} \} \cdot
 \end{aligned} \tag{II.13}$$

Agora, levando em conta que  $\sigma_v^4$  é muito menor do que os outros termos do lado direito dessa expressão, de (II.13) obtém-se (5.12).

## Impacto da Alocação de Ganhos a Coeficientes Ativos e Inativos no Erro em Excesso em Regime Permanente

Para avaliar o impacto da alocação de ganhos a coeficientes ativos e inativos no comportamento de algoritmos tipo proporcional em regime permanente, uma análise estocástica do erro quadrático médio em excesso (EMSE – *excess mean-square error*) é aqui apresentada. Para isso, considerando que  $v(n)$  é um ruído de medição i.i.d. com média zero, variância  $\sigma_v^2$  e descorrelacionado com qualquer outro sinal do sistema, a expressão do EMSE em regime permanente de algoritmos tipo PNLMS para dados de entrada Gaussianos correlacionados é dada por [72]

$$J_{\text{ex}}(\infty) \cong \frac{\beta}{(2-\beta)} \sigma_v^2 \text{tr}[\mathbf{R}\bar{\mathbf{G}}(\infty)] \mathbb{E} \left[ \frac{1}{\mathbf{x}^T(n)\mathbf{G}(\infty)\mathbf{x}(n)} \right] \quad (\text{III.1})$$

onde  $\mathbf{R}$  é a matriz de autocorrelação do sinal de entrada e

$$\bar{\mathbf{G}}(\infty) = \lim_{n \rightarrow \infty} \mathbb{E}[\mathbf{G}(n)] \quad (\text{III.2})$$

denota a matriz de distribuição de ganho média em regime permanente.

Agora, assumindo que para  $N \gg \sqrt{2 \sum_{i=1}^N g_i^2(n)}$  a variância do termo  $\sum_{j=1}^N x_j^2(n)g_j(n)$  se torna muito menor do que o valor esperado [33], pode-se aferir a seguinte aproximação:

$$\mathbb{E} \left[ \frac{1}{\mathbf{x}^T(n)\mathbf{G}(\infty)\mathbf{x}(n)} \right] \cong \frac{1}{N\sigma_x^2}. \quad (\text{III.3})$$

Desse modo, (III.1) pode ser reescrita como

$$J_{\text{ex}}(\infty) \cong \frac{\beta}{(2-\beta)} \frac{\sigma_v^2}{N\sigma_x^2} \text{tr}[\mathbf{R}\bar{\mathbf{G}}(\infty)]. \quad (\text{III.4})$$

Além disso, considerando que a matriz de ganho é uma matriz diagonal e reordenando os elementos da matriz  $\bar{\mathbf{G}}(\infty)$  de forma a agrupar os ganhos associados aos coeficientes ativos e inativos, pode-se escrever a matriz de distribuição de ganho média como

$$\bar{\mathbf{G}}(\infty) = \begin{bmatrix} \bar{\mathbf{G}}^{\text{ativos}}(\infty) & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{G}}^{\text{inativos}}(\infty) \end{bmatrix} \quad (\text{III.5})$$

e, ainda, alterando a ordem dos elementos da matriz  $\mathbf{R}$  em consonância com a alteração realizada em (III.5) para manter a associação entre os elementos de  $\mathbf{R}$  com a matriz  $\bar{\mathbf{G}}(\infty)$ , definindo a matriz  $\mathbf{R}_d$  como a matriz diagonal de  $\mathbf{R}$  e realizando o mesmo agrupamento de (III.5), tem-se

$$\mathbf{R}_d \triangleq \begin{bmatrix} \mathbf{R}_d^{\text{ativos}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_d^{\text{inativos}} \end{bmatrix}. \quad (\text{III.6})$$

Assim, utilizando (III.5) e (III.6) em (III.4), o EMSE em regime permanente pode ser expresso em função do ganho dos coeficientes ativos e inativos como

$$J_{\text{ex}}(\infty) \cong \frac{\beta}{(2-\beta)} \frac{\sigma_v^2}{N\sigma_x^2} \{ \text{tr}[\mathbf{R}_d^{\text{ativos}} \bar{\mathbf{G}}^{\text{ativos}}(\infty)] + \text{tr}[\mathbf{R}_d^{\text{inativos}} \bar{\mathbf{G}}^{\text{inativos}}(\infty)] \}. \quad (\text{III.7})$$

Agora, considerando que algoritmos tipo PNLMS se comportam como o algoritmo NLMS quando  $\mathbf{G}(n) = \mathbf{I}$ , pode-se determinar o valor de  $J_{\text{ex}}(\infty)$  para obter o mesmo EMSE em regime permanente do NLMS; assim, a partir de (III.7) tem-se que

$$\text{tr}[\mathbf{R}_d] = \text{tr}[\mathbf{R}_d^{\text{ativos}} \bar{\mathbf{G}}^{\text{ativos}}(\infty)] + \text{tr}[\mathbf{R}_d^{\text{inativos}} \bar{\mathbf{G}}^{\text{inativos}}(\infty)]. \quad (\text{III.8})$$

Assumindo que em regime permanente o ganho atribuído a todos os coeficientes ativos seja contante e igual a  $g^{\text{ativos}}$  e, para os inativos,  $g^{\text{inativos}}$ , de (III.8) obtém-se

$$\text{tr}[\mathbf{R}_d] = g^{\text{ativos}} \text{tr}(\mathbf{R}_d^{\text{ativos}}) + g^{\text{inativos}} \text{tr}[\mathbf{R}_d^{\text{inativos}}], \quad (\text{III.9})$$

além disso, considerando que a matriz  $\mathbf{R}_d$  possui dimensão  $N$ , tem-se

$$N\sigma_x^2 = N_{\text{ativos}}\sigma_x^2 g^{\text{ativos}} + (N - N_{\text{ativos}})\sigma_x^2 g^{\text{inativos}}. \quad (\text{III.10})$$

Simplificando o termo comum, resulta em

$$N = N_{\text{ativos}}g^{\text{ativos}} + (N - N_{\text{ativos}})g^{\text{inativos}}, \quad (\text{III.11})$$

o que mostra a relação entre número de coeficientes ativos e inativos e seus correspondentes ganhos no valor do EMSE em regime permanente de algoritmos tipo PNLMS. Note que, além da solução  $g^{\text{ativos}} = g^{\text{inativos}} = 1$  em (III.11), soluções onde  $g^{\text{ativos}} > 1$  e  $g^{\text{inativos}} < 1$ , isto é, ganho atribuído a coeficientes ativos maior do que ao atribuído a inativos, ou vice versa, também resultam no mesmo valor de EMSE em regime permanente do que o exibido pelo algoritmo NLMS. Além disso, visto que o ganho do coeficiente é sempre maior do que zero, o valor limite de  $g^{\text{ativos}}$  é  $N/N_{\text{ativos}}$ . Assim, quanto menor o número de coeficientes ativos  $N_{\text{ativos}}$ , maior o valor  $g^{\text{ativos}}$  que pode ser compensado pela diminuição de  $g^{\text{inativos}}$ .



## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] L. G. Morales, *Adaptive Filtering*. Rijeka, Croatia: In-tech, 2011.
- [2] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, Boston, MA, 2008.
- [3] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice-Hall, 1999.
- [4] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [5] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [6] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New Jersey: Wiley, 2003.
- [7] S. M. Kuo and D. R. Morgan, "Active noise control: a tutorial review," *Proc. IEEE*, vol. 87, no. 6, pp. 943-973, Jun. 1999.
- [8] S. J. Elliot, C. C. Boucher, and P. A. Nelson, "The behavior of a multiple channel active control system," *IEEE Trans. Signal Process.*, vol. 40, pp. 1041-1052, May 1992.
- [9] P. L. Feintuch, N. J. Bershad, and A. K. Lo, "A frequency model for filtered LMS algorithm – stability analysis, design, and elimination of the training mode," *IEEE Trans. Signal Process.*, vol. 41, pp. 1518-1531, Apr. 1993.
- [10] S. D. Snyder and H. Hansen, "The effect of transfer function estimation errors on the filtered-X LMS algorithm," *IEEE Trans. Signal Process.*, vol. 42, pp. 950-956, Apr. 1994.

- [11] D. R. Morgan and J. Thi, "A multitone pseudocascade filtered-X LMS adaptive notch filter," *IEEE Trans. Signal Process.*, vol. 41, pp. 946-956, Feb. 1993.
- [12] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*. New-York: Wiley-Interscience, 1996.
- [13] L. J. Eriksson and M. C. Allie, "Use of a random noise for on-line transducer modeling in an adaptive active attenuation system," *Journal Acoust. Soc.*, vol. 85, pp. 797-802, Feb. 1989.
- [14] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Process. Mag.*, vol. 10, no. 4, pp. 12-35, Jun. 1993.
- [15] C. A. Balanis and P. I. Ioannides, *Introduction to Smart Antennas*, 1st ed. San Rafael, CA: Morgan & Claypool, 2007.
- [16] L. C. Godara, *Handbook of Antennas in Wireless Communications*. Boca Raton, FL: CRC Press, 2002.
- [17] L. C. Godara, *Smart Antennas*. Boca Raton, FL: CRC Press, 2004.
- [18] B. Allen and M. Ghavami, *Adaptive Array Systems: Fundamentals and Applications*. Chichester: John Wiley & Sons, 2005.
- [19] S. Applebaum, "Adaptive arrays," *IEEE Trans. Antennas Propagation*, vol. 24, no. 5, pp. 585-598, Sep. 1976.
- [20] M. S. Alencar e F. M. Bernardino Jr., *Telefonia Celular Digital*. São Paulo: Érica, 2004.
- [21] J. Benesty, T. Gansler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Berlin, Germany: Springer, 2001.
- [22] B. Widrow, "Thinking about thinking: the discovery of the LMS algorithm," *IEEE Signal Process. Magazine*, vol. 22, no. 1, pp. 100-106, Jan-2005.
- [23] I. S. Reed, "Brief history of adaptive arrays," *IEEE Military Comm. Conf.*, Boston, MA, Oct. 1985, pp. 515-518.

- [24] M. M. Sondhi, "The history of echo cancellation," *IEEE Signal Process. Magazine*, vol. 23, no. 5, pp. 95-102, Sep. 2006.
- [25] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, Jr., "Exploiting sparsity in adaptive filters," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1883-1894, Aug. 2002.
- [26] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. New York: Springer-Verlag, 2006.
- [27] B. Jelfs, D. P. Mandic, and J. Benesty, "A class of adaptively regularised PNLMS algorithms," in *Proc. 15th Int. Conf. Digital Signal Process.*, Cardiff, UK, Jul. 2007, pp. 19-22.
- [28] B. Jelfs, D. P. Mandic, and A. Cichocki, "A unifying approach to the derivation of the class of PNLMS algorithms," in *Proc. 15th Int. Conf. Digital Signal Process.*, Cardiff, UK, Jul. 2007, pp. 35-38.
- [29] C. Paleologu, J. Benesty, and S. Ciochina, *Sparse Adaptive Filters for Echo Cancellation*. San Rafael, CA: Morgan and Claypool, 2010.
- [30] A. W. H. Khong, P. A. Naylor, and J. Benesty, "A low delay and fast convergence improved proportionate algorithm for sparse system identification," *EURASIP J. Audio, Speech, Music Process.*, vol. 2007, no. 1, Article ID 843776, pp. 1-8, 2007.
- [31] V. V. Krishna, J. Rayala, and B. Slade, "Algorithmic and implementation aspects of echo cancellation in packet voice networks," in *Proc. 36th Asilomar Conf. Signals, Syst., Comput.*, Milpitas, CA, Nov. 2002, vol. 2, pp. 1252-1257.
- [32] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learning Research*, vol. 5, pp. 1457-1469, Nov. 2004.
- [33] K. Wagner and M. Doroslovacki, "Proportionate-type normalized least mean square algorithms with gain allocation motivated by mean-square-error minimization for white input," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2410-2415, May 2011.

- [34] J. Radecki, Z. Zilic, and K. Radecka, "Echo cancellation in IP networks," in *Proc. 45th Midwest Symp. Circuits and Syst.*, Tulsa, USA, Aug. 2002, vol. 2, pp. 219-222.
- [35] V. V. Krishna, J. Rayala, and B. Slade, "Algorithmic and implementation aspects of echo cancellation in packet voice networks," in *Proc. 36th Asilomar Conf. Signals, Syst., Comput.*, Milpitas, CA, Nov. 2002, vol. 2, pp. 1252-1257.
- [36] J. H. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," *IEEE Comm. Magazine*, pp. 36-41, Jul. 2004.
- [37] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Proc. 32th Asilomar Conf. Signals, Syst., Comput.*, Monterey, USA, Nov. 1998, vol. 1, pp. 394-398.
- [38] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech and Audio Process.*, vol. 8, no. 5, pp. 508-518, Sep. 2000.
- [39] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Orlando, USA, May 2002, vol. 2, pp. 1881-1884.
- [40] H. Deng and M. Doroslovacki, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 181-184, Mar. 2005.
- [41] H. Deng and M. Doroslovacki, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Trans. Signal Process.*, vol. 54, no. 5, pp. 1794-1803, May 2006.
- [42] J. Cui, P. A. Naylor, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Process.*, vol. 86, no. 6, pp. 1182-1192, Jun. 2006.
- [43] P. Loganathan, A. W. H. Khong, and P. A. Naylor, "A class of sparseness-controlled algorithms for echo cancellation," *IEEE Trans. Audio, Speech and Language Process.*, vol. 17, no. 8, pp. 1591-1601, Nov. 2009.

- [44] F. das C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, "Alternative approach for computing the activation factor of the PNLMS algorithm," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Glasgow, Scotland, Aug. 2009, pp. 2633-2637.
- [45] A. Deshpande and S. L. Grant, "A new multi-algorithm approach to sparse system adaptation," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Antalya, Turkey, Sep. 2005, vol. 4, pp. 1910-1914.
- [46] P. A. Naylor, M. Brooks, and J. Cui, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Montreal, Canada, May 2004, vol. 4, pp. 141-144.
- [47] H.-C. Shin, A. H. Sayed, and W.-J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132-135, Feb. 2004.
- [48] F. das C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, "A PNLMS algorithm with individual activation factors," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2036-2047, Apr. 2010.
- [49] P. Loganathan, A. W. H. Khong, and P. A. Naylor, "A sparseness controlled proportionate algorithm for acoustic echo cancellation," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008, pp. 1-5.
- [50] A. W. H. Khong and P. A. Naylor, "Efficient use of sparse adaptive filters," in *Proc. 40th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, USA, Oct. 2006, pp. 1375-1379.
- [51] K. T. Wagner and M. I. Doroslovacki, "Gain allocation in proportionate-type NLMS algorithms for fast decay of output error at all times," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3117-3120.
- [52] F. das C. de Souza, R. Seara, and D.R. Morgan, "An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification," *IEEE Transactions on Signal Processing* 60 (6) (2012) 3301-3307.

- [53] Y. Gu, J. Jin, and S. Mei, “ $\|\vec{x}\|_\infty = \max_{i=1}^n(|x_i|)$  constraint LMS algorithm for sparse system identification,” *IEEE Signal Process. Lett.*, vol. 16, no. 3, pp. 774-777, Sep-2009.
- [54] Y. Chen, Y. Gu, and A. O. Hero, “Sparse LMS for system identification,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3125-3128.
- [55] S. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251-276, Feb. 1998.
- [56] S. Amari and S.C. Douglas, "Why natural gradient," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Seattle, USA, May 1998, vol. 2, pp. 1213-1216.
- [57] S. L. Gay and S.C. Douglas, "Normalized natural gradient adaptive filtering for sparse and non-sparse systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Orlando, USA, May 2002, vol. 2, pp. 1406-1408.
- [58] J. Benesty and Y. Huang, "The LMS, PNLMS, and exponentiated gradient algorithms," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Vienna, Austria, Sep. 2004, pp. 721-724.
- [59] K. Wagner, M. Doroslovacki, and H. Deng, "Proportionate-type steepest descent and NLMS algorithms," in *Proc. 41st Annual Conf. Information Sciences and Syst.*, Baltimore, MD, Nov. 2007, pp. 47-50.
- [60] C. Paleologu, S. Ciochina, and J. Benesty, "An efficient proportionate affine projection algorithm for echo cancellation," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 165-168, Feb. 2010.
- [61] K. Wagner, M. Doroslovacki, and H. Deng, "Convergence of proportionate-type LMS adaptive filters and choice of gain matrix," in *Proc. 40th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2006, pp. 243-247.

- [62] T. Gansler, J. Benesty, M. M. Sondhi, and S. L. Gay, "Dynamic resource allocation for network echo cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Salt Lake City, UT, May 2001, vol. 5, pp. 3222-3236.
- [63] K. Wagner and M. Doroslovacki, "On proportionate-type NLMS algorithms for fast decay of output error at all times," in *Proc. 43rd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2009, pp. 186-190.
- [64] K. Wagner and Milos Doroslovacki, *Proportionate-type normalized least mean square algorithms*, London, UK, ISTE Ltd, 2013
- [65] ITU-T Recommendation G.168, *Digital Network Echo Cancellers*. Int. Telecomm. Union, Geneva, Switzerland, 2007.
- [66] N. J. Bershad, S. McLaughlin, and C. F. N. Cowan, "Performance comparison of RLS and LMS algorithms for tracking a first order Markov communications channel," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, New Orleans, USA, May 1990, vol. 1, pp. 266-270.
- [67] A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive filtering with performance analysis for identification of time-varying systems," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 5, pp. 1681-1695, Jul. 2007.
- [68] P. Loganathan, E. A. P. Habets, and P. A. Naylor, "Performance analysis of IPNLMS for identification of time-varying systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Dallas, Texas, Mar. 2010, pp. 317-320.
- [69] K.T. Wagner and M.I. Doroslovacki, "Reduced computational complexity suboptimal gain allocation for proportionate-type NLMS algorithms with colored inputs," in *Proc. of 45th Annual Conf. on Inform. Sciences and Systems (CISS)*, Baltimore, MD, USA, March 2011, pp. 1-6.

- [70] J. G. F. Zipf, O. J. Tobias, and Rui Seara, "Non-parametric VSS-NLMS algorithm with control parameter based on error correlation," *Proc. of the 7<sup>th</sup> IEEE/SBrT International Telecommunications Symposium (ITS)*, Manaus, Brazil, Set 2010, pp. 1-5.
- [71] F. L. Perez, F. C. de Souza, R. Seara, "An improved mean-square weight deviation-proportionate gain algorithm based on error autocorrelation," *Elsevier Signal Process.*, vol. 94, no. 1, pp. 503-513, Jan. 2014.
- [72] E. V. Kuhn, F. das C. de Souza, R. Seara and D. R. Morgan, "On the steady-state analysis of PNLMS-type algorithms for correlated gaussian input data," *IEEE Signal Process. Lett.*, vol. 21, no. 11, pp. 1433-1437, Nov. 2014