

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA**

Francisco das Chagas de Souza

**ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS
PROPORCIONAIS: PROPOSTA DE UM NOVO ALGORITMO E
SUA MODELAGEM ESTOCÁSTICA**

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

Orientador: Prof. Dr. Rui Seara

Co-orientador: Prof. Dr. Orlando José Tobias

Florianópolis, SC

2012

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

Souza, Francisco das Chagas de
Algoritmos Adaptativos LMS Normalizados Proporcionais:
Proposta de um Novo Algoritmo e sua Modelagem Estocástica
[tese] / Francisco das Chagas de Souza ; orientador, Rui
Seara - Florianópolis, SC, 2012.
170 p. ; 21cm

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

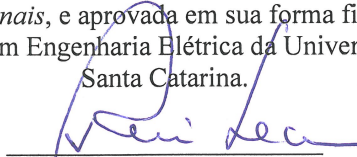
Inclui referências

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

**ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS
PROPORCIONAIS: PROPOSTA DE UM NOVO
ALGORITMO E SUA MODELAGEM ESTOCÁSTICA**

Francisco das Chagas de Souza

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

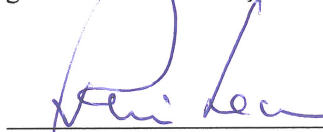


Prof. Rui Seara, Dr.
Orientador

Prof. Patrick Kuo-Peng, Dr

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

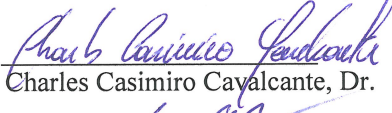
Banca examinadora:



Prof. Rui Seara, Dr.
Presidente



Prof. João Marcos Travassos Romano, Dr.



Prof. Charles Casimiro Cavalcante, Dr.



Prof. Eduardo Luiz Ortiz Batista, Dr.



Prof. Sidnei Noceti Filho, Dr.



Prof. Walter Pereira Carpes Jr., Dr.

Dedico este trabalho de pesquisa à
minha família, pelo incentivo durante
essa jornada.

AGRADECIMENTOS

Deixo aqui os meus sinceros agradecimentos àqueles que contribuíram para a realização deste trabalho de pesquisa, em especial:

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

À co-orientação, apoio e sugestões do Prof. Orlando José Tobias. O sucesso deste trabalho de pesquisa deve-se, em grande parte, às importantes e brilhantes idéias do Prof. Orlando, as quais aqui foram desenvolvidas, resultando na publicação de inúmeros artigos.

A todos os colaboradores do Laboratório de Circuitos e Processamento de Sinais-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 CAPES pelo apoio financeiro recebido, à UFSC e ao LINSE por toda infra-estrutura concedida para a realização do trabalho.

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

A Deus, pelas oportunidades que me foram dadas para a conclusão deste trabalho de pesquisa.

RESUMO

Neste trabalho, um novo algoritmo LMS normalizado proporcional (PNLMS) é proposto. Tal algoritmo usa fatores de ativação individuais para cada coeficiente do filtro adaptativo, em vez de um fator de ativação global como no algoritmo PNLMS padrão. Os fatores de ativação individuais do algoritmo proposto são atualizados recursivamente a partir dos correspondentes coeficientes do filtro adaptativo. Essa abordagem conduz a uma melhor distribuição da energia de adaptação entre os coeficientes do filtro. Dessa forma, para respostas ao impulso com elevada esparsidade, o algoritmo proposto, denominado algoritmo PNLMS com fatores de ativação individuais (IAF-PNLMS), atinge maior velocidade de convergência do que os algoritmos PNLMS padrão e PNLMS melhorado (IPNLMS). Também, uma metodologia de modelagem estocástica dos algoritmos da classe PNLMS é apresentada. Usando essa metodologia, obtém-se um modelo estocástico que prediz satisfatoriamente o comportamento do algoritmo IAF-PNLMS tanto na fase transitória quanto na estacionária. Através de simulações numéricas, a eficácia do modelo proposto é verificada. Adicionalmente, uma versão melhorada do algoritmo IAF-PNLMS, denominada EIAF-PNLMS, é proposta neste trabalho, a qual usa uma estratégia de redistribuição de ganhos durante o processo de aprendizagem, visando aumentar os ganhos atribuídos aos coeficientes inativos quando os ativos aproximam-se da convergência. Resultados de simulação mostram que tal estratégia de redistribuição melhora significativamente as características de convergência do algoritmo.

Palavras-chave: Algoritmo LMS normalizado proporcional (PNLMS), filtragem adaptativa, identificação de sistemas esparsos, modelo estocástico, resposta ao impulso esparsa.

ABSTRACT

In this work, a new proportionate normalized least-mean-square (PNLMS) algorithm is proposed. Such an algorithm uses individual activation factors for each adaptive filter coefficient, instead of a global activation factor as in the standard PNLMS algorithm. The proposed individual activation factors are recursively updated considering the corresponding adaptive filter coefficients. This approach leads to a better distribution of the adaptation energy over the filter coefficients than the standard PNLMS does. Thereby, for impulse responses exhibiting high sparseness, the proposed algorithm, called individual-activation-factor proportionate normalized least-mean-square (IAF-PNLMS), achieves faster convergence than the standard PNLMS and improved PNLMS (IPNLMS) algorithms. Also, a stochastic modeling methodology for the class of PNLMS algorithms is presented. By using this methodology, one obtains a stochastic model that predicts satisfactorily the IAF-PNLMS algorithm behavior in both transient and steady-state phases. Through simulation results, the accuracy of the proposed model is verified. Furthermore, an enhanced version of the IAF-PNLMS algorithm, called EIAF-PNLMS, is proposed, which uses a gain redistribution strategy during the learning process for increasing the gain assigned to inactive coefficients as the active ones approach convergence. Simulation results show that such a strategy significantly improves the convergence characteristics of the algorithm.

Keywords: Adaptive filtering, proportionate normalized least-mean-square (PNLMS) algorithm, sparse impulse response, sparse system identification, stochastic model.

LISTA DE FIGURAS

1.1	Resposta ao impulso típica de um caminho de eco de rede.	26
1.2	Diagrama de blocos de um problema geral de filtragem adaptativa.	27
1.3	Diagrama de blocos simplificado de um sistema de cancelamento de eco.	28
2.1	Desalinhamento normalizado do algoritmo PNLMS padrão usando $\mu = 0,5$ e $\delta = 0,01$. (Linha pontilhada) $\rho = 0,50$ e $a(n)$ dado por (1.10). (Linha tracejada) $\rho = 0,05$ e $a(n)$ dado por (1.10). (Linha escura sólida) $\rho = 0,01$ e $a(n)$ dado por (1.10). (Linha cinza sólida).	41
2.2	Curvas de $w_i(n)$ e $g_i(n)$ para $\mu = 0,5$, $\rho = 0,05$ e $\delta = 0,01$. (a) Coeficientes adaptativos $w_1(n)$, $w_2(n)$, $w_{30}(n)$ e $w_{35}(n)$. (b) Ganhos de adaptação $g_1(n)$, $g_2(n)$, $g_{30}(n)$ e $g_{35}(n)$	44
4.1	Exemplo 4.1. Curvas das variáveis do algoritmo IAF-PNLMS considerando uma perturbação na planta em $n = 2500$ (planta \mathbf{w}^o é mudada para $-\mathbf{w}^o$) com $\mu = 0,5$ e $a_i(0) = 10^{-4}$. (a) Coeficientes ativos. (b) Coeficientes inativos. (c) Fatores de ativação associados aos coeficientes ativos. (d) Fatores de ativação associados aos coeficientes inativos. (e) Funções de proporcionalidade associadas aos coeficientes ativos. (f) Funções de proporcionalidade associadas aos coeficientes inativos.	70
4.2	Exemplo 4.1. Curvas de desalinhamento normalizado considerando a inversão do sinal dos coeficientes da planta no instante $n = 2500$ com $\mu = 0,5$, $\rho = 0,05$, $\delta = 0,01$, $a_i(0) = 10^{-4}$ e $\alpha = 0$	71
4.3	Exemplo 4.2. Curvas das variáveis do algoritmo IAF-PNLMS com um deslocamento nos coeficientes da planta no instante $n = 2500$, usando $\mu = 0,5$ e $a_i(0) = 10^{-4}$. (a) Coeficientes ativos antes da	

perturbação. (b) Coeficientes ativos após a perturbação. (c) Fatores de ativação associados aos coeficientes ativos antes da perturbação. (d) Fatores de ativação associados aos coeficientes ativos após a perturbação. (e) Funções de proporcionalidade associadas aos coeficientes ativos antes da perturbação. (f) Funções de proporcionalidade associadas aos coeficientes ativos após a perturbação. 75

4.4 Exemplo 4.2. Curvas de desalinhamento normalizado considerando um deslocamento nos coeficientes da planta no instante $n = 2500$ com $\mu = 0,5$, $\rho = 0,05$, $\delta = 0,01$, $a_i(0) = 10^{-4}$ e $\alpha = 0$ 76

4.5 Exemplo 4.3. Comportamento em regime estacionário do desalinhamento e dos coeficientes do filtro adaptativo com coeficientes da planta variantes no tempo, considerando $\mu = 0,5$, $\rho = 0,05$, $\delta = 0,01$, $\varepsilon = 0,001$, $a_i(0) = 0,01$ e $\alpha = 0$. (a) Curvas de desalinhamento normalizado. (b) Curvas dos coeficientes adaptativos $w_1(n)$. (c) $w_2(n)$. (d) $w_{30}(n)$ 80

4.6 Exemplo 4.4. Curvas de aprendizagem (EQM) usando $\rho = 0,05$, $\delta = 0,01$, $\varepsilon = 0,001$ e $a_i(0) = 10^{-4}$. (a) Planta **h** [com $\sigma_s^2 = 10^{-8}$ e $S_D(\mathbf{h}) = 0,9428$] considerando $\mu = 0,5$ para o algoritmo MPNLMS e $\mu = 0,30$ para o algoritmo IAF-PNLMS. (b) Planta **h** [com $\sigma_s^2 = 5 \times 10^{-6}$ e $S_D(\mathbf{h}) = 0,9286$] com $\mu = 0,5$ para o algoritmo MPNLMS e $\mu = 0,2$ para o IAF-PNLMS. 84

4.7 Exemplo 4.5. Curvas de aprendizagem (EQM) usando $\rho = 5/N$, $\delta = 0,01$, $\varepsilon = 0,001$, $a_i(0) = 10^{-2} / N$, $\mu = 0,5$ para o algoritmo MPNLMS e $\mu = 0,1$ para o algoritmo IAF-PNLMS [com grau de esparsidade da planta $S_D(\mathbf{r}) = 0,8970$]. 85

5.1 Curvas do comportamento médio dos coeficientes, considerando um sinal de entrada correlacionado com $\chi = 144$, obtidas usando simulações de Monte Carlo (linhas tracejadas pretas) e o modelo proposto (5.20) (linhas contínuas cinzas). De cima para baixo: $w_{17}(n)$, $w_{20}(n)$, $w_{10}(n)$, $w_{22}(n)$ e $w_{24}(n)$ 98

- 5.2 Testes da Hipótese H4 e das Aproximações E1 e E2. (a) Evoluções no tempo de $\{E[\mathbf{G}(n)\mathbf{w}(n)]\}_{20}$ (linha sólida cinza) e $\{E[\mathbf{G}(n)]E[\mathbf{w}(n)]\}_{20}$ (linha tracejada preta). (b) Lados esquerdo (linha sólida cinza) e direito (linha tracejada preta) do 17^o elemento de (5.2). (c) Lados esquerdo (linha sólida cinza) e direito (linha tracejada preta) do 17^o elemento de (5.3). 99
- 5.3 Teste da Aproximação E3. (a) Evoluções no tempo de $E[\phi_1(n)]$ (linha sólida cinza) e $\max\{E[a_1(n)], |E[w_1(n)]|\}$ (linha tracejada preta). (b) $E[\phi_{10}(n)]$ (linha sólida cinza) e $\max\{E[a_{10}(n)], |E[w_{10}(n)]|\}$ (linha tracejada preta). (c) $E[\phi_{17}(n)]$ (linha sólida cinza) e $\max\{E[a_{17}(n)], |E[w_{17}(n)]|\}$ (linha tracejada preta). (d) $E[\phi_{22}(n)]$ (linha sólida cinza) e $\max\{E[a_{22}(n)], |E[w_{22}(n)]|\}$ (linha tracejada preta) 102
- 6.1 Exemplo 6.1. Curvas do comportamento médio dos coeficientes, obtidas usando simulações de Monte Carlo (linhas contínuas cinzas) e o modelo proposto (6.36) (linhas tracejadas pretas) considerando SNR = 20 dB. (a) De cima para baixo: $E[w_{10}(n)]$, $E[w_1(n)]$, $E[w_2(n)]$, $E[w_{24}(n)]$ e $E[w_{15}(n)]$. (b) Detalhe da fase transitória de $E[w_{10}(n)]$ (linhas superiores) e $E[w_2(n)]$ (linhas inferiores). 118
- 6.2 Exemplo 6.1. Curvas de aprendizagem (EQM) obtidas usando simulações de Monte Carlo (linha sólida cinza) e o modelo proposto (6.39) (linha tracejada preta). (a) SNR = 20 dB. (b) SNR = 30 dB. 119
- 6.3 Exemplo 6.2. Curvas do comportamento médio dos coeficientes obtidas usando simulações de Monte Carlo (linha sólida cinza) e modelo proposto (6.36) (linha sólida preta) considerando SNR = 20 dB. (a) De cima para baixo: $E[w_{30}(n)]$, $E[w_1(n)]$, $E[w_2(n)]$, $E[w_{85}(n)]$ e $E[w_{35}(n)]$. (b) Detalhe da fase transitória $E[w_{30}(n)]$ (linhas superiores) e $E[w_2(n)]$ (linhas inferiores). 120

- 6.4 Exemplo 6.2. Curvas de aprendizagem (EQM) obtidas usando simulações de Monte Carlos (linha sólida cinza) e o modelo proposto (6.39) (linha tracejada preta) considerando $SNR = 20$ dB. 121
- 7.1 Exemplo 7.1. Curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS (linhas cinza) e EIAF-PNLMS (linhas pretas) considerando um deslocamento dos coeficientes da planta com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. (a) $SNR = 30$ dB. (b) $SNR = 40$ dB. 133
- 7.2 Exemplo 7.1. Curvas de desalinhamento normalizado do algoritmo EIAF-PNLMS usando $\beta = 0,98$. (a) $K = 2 \times 10^{-2}$ (linha preta) e $K = 3 \times 10^{-2}$ (linha cinza), considerando $SNR = 20$ dB. (b) $K = 3 \times 10^{-3}$ (linha preta) e $K = 7 \times 10^{-3}$ (linha cinza), considerando $SNR = 40$ dB. 134
- 7.3 Exemplo 7.1. Desalinhamento normalizado do algoritmo EIAF-PNLMS considerando um deslocamento dos coeficientes da planta. Fator de suavização $\beta = 0,97$ (linhas pretas) e $\beta = 0,99$ (linhas cinzas). (a) $SNR = 20$ dB com $K = 2,5 \times 10^{-2}$. (b) $SNR = 40$ dB com $K = 5 \times 10^{-3}$ 136
- 7.4 Exemplo 7.2. Desalinhamento normalizado dos algoritmos NLMS (linhas pretas tracejadas), IPNLMS com $\mu = 0,45$ (linhas cinzas sólidas) e EIAF-PNLMS (linhas pretas sólidas), considerando um deslocamento dos coeficientes da planta com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. (a) $SNR = 30$ dB. (b) $SNR = 40$ dB. 138
- 7.5 Exemplo 7.3. Desalinhamento normalizado dos algoritmos MPNLMS (linhas cinzas) e EIAF-PNLMS (linhas pretas) com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. Os coeficientes da resposta ao impulso são deslocados de 12 amostras para a direita. (a) $\chi = 10$ usando $\mu = 0,5$ para o MPNLMS. (b) $\chi = 232$ usando $\mu = 0,43$ para o MPNLMS. 140

LISTA DE TABELAS

2.1. Distribuição de ganhos totais do algoritmo PNLMS padrão para 5000 iterações	42
3.1. Algoritmo IPNLMS	50
3.2. Algoritmo MPNLMS	57
4.1. Algoritmo IAF-PNLMS	64
4.2. Comparação de complexidade computacional e memória requerida	66
4.3. Exemplo 4.1. Número necessário de iterações para o desalinhamento normalizado $\kappa(n)$ atingir -33 dB após a perturbação na iteração $n = 2500$	71
4.4. Exemplo 4.4. Distribuições de ganhos totais dos algoritmos PNLMS padrão, IPNLMS, MPNLMS e IAF-PNLMS para 5000 iterações	82
7.1. Exemplo 7.3. Parâmetros do Processo AR(2) usados para obter dados de entrada com diferentes dispersões de autovalores	139

SUMÁRIO

1. INTRODUÇÃO	23
1.1 Filtragem Adaptativa Proporcional	24
1.1.1 Grau de Esparsidade	25
1.1.2 Adaptação Proporcional.....	26
1.2 Motivações para a Derivação de um Algoritmo PNLMS Melhorado.....	31
1.3 Modelagem dos Algoritmos da Classe PNLMS.....	32
1.4 Trabalhos Publicados	33
1.5 Organização do Manuscrito.....	34
1.6 Conclusões	35
2. IMPACTO DO FATOR DE ATIVAÇÃO NO COMPORTAMENTO DO ALGORITMO PNLMS PADRÃO.....	37
2.1 Introdução	37
2.2 Ganhos de Adaptação	37
2.3 Estudo do Comportamento do Algoritmo	40
2.3.1 Avaliação do Comportamento	40
2.3.2 Distribuição de Ganhos.....	42
2.3.3 Comportamento de Coeficientes e Ganhos	43
2.4 Conclusões	44
3. VARIANTES DO ALGORITMO PNLMS PADRÃO.....	47
3.1 Introdução	47
3.2 Algoritmo IPNLMS	47
3.2.1 Derivação do Algoritmo IPNLMS	47
3.2.2 Forma Fechada para o Cálculo do Ganho Individual	49
3.3 Algoritmo MPNLMS.....	52
3.3.1 Algoritmo <i>Steepest Descent</i>	52
3.3.2 Algoritmo PNLMS com Lei μ	56

3.4	Conclusões.....	58
4.	ALGORITMO PNLMS COM FATORES DE ATIVAÇÃO INDIVIDUAIS.....	59
4.1	Introdução	59
4.2	Derivação do Algoritmo IAF-PNLMS.....	59
4.2.1	Condições Requeridas para o Novo Fator de Ativação.....	60
4.2.2	Abordagem Proposta para Determinar $a_i(n)$	61
4.2.3	Propriedades das Variáveis $a_i(n)$ e $\phi_i(n)$ do Algoritmo IAF-PNLMS	63
4.2.4	Complexidade Computacional e Memória Requerida.....	65
4.3	Resultados de Simulação	66
4.3.1	Exemplo 4.1	67
4.3.2	Exemplo 4.2	72
4.3.3	Exemplo 4.3	76
4.3.4	Exemplo 4.4	80
4.3.5	Exemplo 4.5	82
4.4	Conclusões.....	86
5.	MODELO ESTOCÁSTICO DA EVOLUÇÃO DOS COEFICIENTES DO ALGORITMO IAF-PNLMS	87
5.1	Introdução	87
5.2	Metodologia de Modelagem.....	88
5.2.1	Definição do Problema	88
5.2.2	Hipóteses e Aproximações Gerais	90
5.3	Modelo da Evolução dos Coeficientes do Filtro Adaptativo: Caso do Algoritmo IAF-PNLMS	92
5.3.1	Determinação de $\bar{\mathbf{G}}(n)$	92
5.3.2	Determinação de $\mathbf{R}_{\mathbf{G}}(n)$	93

5.3.3	Determinação de $\mathbf{p}_G(n)$	94
5.3.4	Expressão do Modelo.....	96
5.4	Simulações Numéricas.....	96
5.5	Conclusões.....	102
6.	MODELO ESTOCÁSTICO PARA O MOMENTO DE SEGUNDA ORDEM DO VETOR DE ERRO DOS COEFICIENTES DO ALGORITMO IAF-PNLMS.....	103
6.1	Introdução.....	103
6.2	Equação Geral do Momento de Segunda Ordem do Vetor de Erro dos Coeficientes.....	103
6.3	Determinação dos valores Esperados Associados aos Dados de Entrada.....	104
6.3.1	Determinação de $\mathbf{R}_G(n)$	105
6.3.2	Determinação de $\mathbf{p}_G(n)$	108
6.3.3	Determinação de $\mathbf{R}_2(n)$	111
6.4	Expressões do Modelo.....	113
6.4.1	Comportamento Médio do Vetor de Coeficientes.....	114
6.4.2	Curva de Aprendizagem.....	114
6.5	Resultados de Simulação.....	115
6.5.1	Exemplo 6.1.....	116
6.5.2	Exemplo 6.2	117
6.6	Conclusões.....	121
7.	UMA NOVA ESTRATÉGIA DE DISTRIBUIÇÃO DE GANHOS PARA O ALGORITMO IAF-PNLMS.....	123
7.1	Introdução.....	123
7.2	Formulação do Algoritmo.....	123
7.2.1	Expressões Gerais dos Algoritmos da Classe PNLMS.....	124

7.2.2	Algoritmo PNLMS com Fatores de Ativação Individuais.....	125
7.2.3	Algoritmo PNLMS com Fatores de Ativação Individuais Melhorado.....	126
7.3	Resultados de Simulação.....	130
7.3.1	Exemplo 7.1.....	132
7.3.2	Exemplo 7.2.....	137
7.3.3	Exemplo 7.3.....	138
7.3.4	Discussão.....	140
7.4	Conclusões.....	142
8.	COMENTÁRIOS E CONCLUSÕES FINAIS.....	143
	APÊNDICES.....	147
	REFERÊNCIAS BIBLIOGRÁFICAS.....	163

Introdução

Uma resposta ao impulso (planta) é qualitativamente classificada como esparsa se a maioria de seus coeficientes tem magnitude nula (ou próxima de zero) e apenas alguns poucos coeficientes apresentam magnitudes significativas. Tais respostas esparsas são encontradas em muitas aplicações, tais como cancelamento de eco, localização de fontes, sistemas de transmissão de televisão digital, processos sísmicos, dentre outras [1]-[4]. Assim, a concepção de filtros adaptativos que exploram a natureza esparsa dessas respostas tem se tornado uma área de pesquisa ativa e de grande relevância. Para essa classe de respostas ao impulso, os algoritmos adaptativos clássicos que utilizam o mesmo passo de adaptação para todos os coeficientes do filtro, tal como o algoritmo LMS normalizado (NLMS - *normalized least-mean-square*), são superados em termos de velocidade de convergência e habilidade de rastreamento por aqueles algoritmos especialmente concebidos para tirar proveito da natureza esparsa da resposta ao impulso [5]-[12].

Um dos primeiros algoritmos a explorar a natureza esparsa da resposta ao impulso foi proposto por Donald L. Duttweiler em [6]. Nesse algoritmo, denominado NLMS proporcional (PNLMS - *proportionate NLMS*), cada coeficiente do filtro adaptativo é atualizado proporcionalmente à sua magnitude, resultando em uma melhoria nas características de convergência. Contudo, o algoritmo PNLMS tem seu desempenho deteriorado à medida que a esparsidade da resposta ao impulso decresce [5], [8]. Assim, versões melhoradas do algoritmo PNLMS, tais como o PNLMS++ [5] e o PNLMS melhorado (IPNLMS - *improved PNLMS*), objetivando lidar com respostas ao impulso com esparsidade média, vêm sendo propostas [8]. Entretanto, esses algoritmos não exibem a mesma velocidade inicial de convergência que é obtida com o algoritmo PNLMS para respostas ao impulso com elevada esparsidade [13]. Uma versão do algoritmo

PNLMS que considera a variação de esparsidade da planta¹ é o SC-PNLMS (*sparseness-controlled* PNLMS) [12], [14]. Esse algoritmo apresenta um satisfatório comportamento para plantas com esparsidade elevada, assim como para aquelas com esparsidade média; contudo, tal comportamento é obtido às custas de um aumento de complexidade computacional. Em [9] e [10], outra versão do algoritmo PNLMS, denominada PNLMS com lei μ (MPNLMS - μ -law PNLMS), é proposta. Tal algoritmo provê uma convergência rápida durante todo o processo de adaptação, superando os algoritmos PNLMS e PNLMS++; entretanto, o MPNLMS é computacionalmente mais complexo do que o PNLMS. Outros trabalhos relacionados aos algoritmos adaptativos para aplicações com plantas esparsas podem ser encontrados em [15]-[22]. Nesse contexto, o presente trabalho tem os seguintes objetivos gerais:

- i) Desenvolver algoritmos adaptativos melhorados para aplicações nas quais a planta do sistema é esparsa.
- ii) Desenvolver uma metodologia de modelagem estocástica para algoritmos adaptativos aplicados em plantas esparsas.

A seguir, são apresentados os conceitos básicos envolvendo filtragem adaptativa proporcional, tais como grau de esparsidade, estrutura do algoritmo PNLMS e modelagem estocástica dessa classe de algoritmos. Além disso, são ressaltadas as principais contribuições e a organização deste trabalho de pesquisa.

1.1. Filtragem Adaptativa Proporcional

Nesta seção, inicialmente, o grau de esparsidade de uma resposta ao impulso é definido; em seguida, a estrutura de um dos primeiros algoritmos especialmente concebidos para lidar com plantas esparsas, o algoritmo PNLMS [6], é apresentada. Finalmente, as motivações para a proposição de um novo algoritmo com melhores características de convergência são descritas.

¹ Na literatura especializada, os termos planta e resposta ao impulso são usados indistintamente.

1.1.1. Grau de Esparsidade

Uma medida típica, encontrada na literatura, que quantifica a esparsidade de uma resposta ao impulso é o grau de esparsidade. Tal medida, baseada na relação entre a norma 1 e norma 2 da resposta ao impulso, é definida como [1], [23]

$$S_D(\mathbf{w}^o) \triangleq \frac{N}{N - \sqrt{N}} \left(1 - \frac{\|\mathbf{w}^o\|_1}{\sqrt{N} \|\mathbf{w}^o\|_2} \right) \quad (1.1)$$

onde o vetor \mathbf{w}^o representa uma resposta ao impulso de dimensão N , e $\|\mathbf{w}^o\|_1$ e $\|\mathbf{w}^o\|_2$ são, respectivamente, a norma 1 e a norma 2 de \mathbf{w}^o . A medida $S_D(\mathbf{w}^o)$ varia de 0 (grau de esparsidade de um filtro uniforme) a 1 (grau de esparsidade do filtro de Dirac) [1], [23].

Agora, é exemplificado o uso de (1.1), avaliando o grau de esparsidade de uma resposta ao impulso. Para tal, considere o exemplo de uma resposta ao impulso típica de um caminho de eco de rede (amostrada em 8kHz), ilustrada na Figura 1.1, a qual é originada quando redes de pacotes para comunicação de voz são conectadas com os sistemas PSTN (*public switched telephone network*). Nessa figura, a resposta ao impulso do caminho de eco exhibe as seguintes características:

- i) Uma região ativa devido à resposta ao impulso da híbrida (modelo do caminho de eco #1 definido nas recomendações ITU-T G.168) com tempo de dispersão de 8 ms, representado pelos 64 coeficientes centrais [24].
- ii) Duas regiões inativas devido ao *bulk delay*², com duração de 56 ms, correspondendo a 448 coeficientes [22], [25]-[27].

² Na literatura técnica, o termo *bulk delay* é usado para denominar o atraso total dos pacotes de voz devido à codificação, propagação e variação do tempo de *buffering* (*jitter buffer delays*).

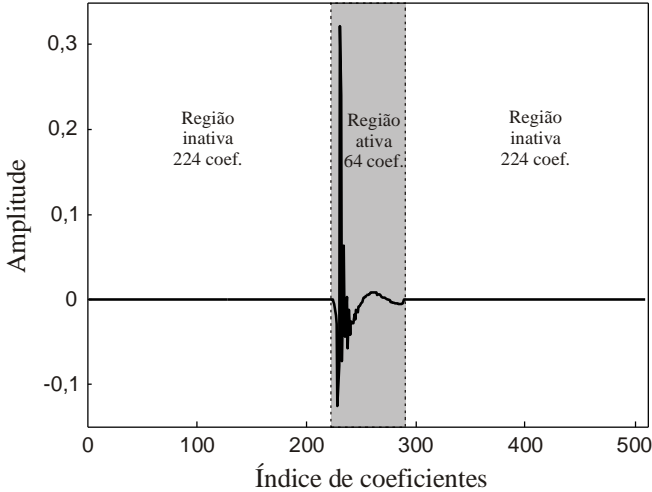


Figura 1.1. Resposta ao impulso típica de um caminho de eco de rede.

Note, da Figura 1.1, que a resposta ao impulso do caminho de eco é dominada pelas duas regiões inativas, o que a torna esparsa. Então, através de (1.1), o grau de esparsidade dessa resposta ao impulso é $S_D(\mathbf{w}^0) = 0.8969$, que é considerado elevado, próximo do máximo valor 1.

1.1.2. Adaptação Proporcional

A. Breve Histórico

O algoritmo LMS (*least-mean-square*), proposto por B. Widrow e M. E. Hoff no contexto de reconhecimento de padrões [28]-[30], é o mais popular algoritmo adaptativo conhecido na literatura. A sua essência pode ser sucintamente descrita pela expressão de atualização do i -ésimo coeficiente do filtro adaptativo, a qual é dada por

$$w_i(n+1) = w_i(n) + \underbrace{\overbrace{\mu}^{\text{passo de adaptação}} e(n)x_i(n)}_{\text{termo de correção}} \quad (1.2)$$

onde μ é o parâmetro de passo, $x_i(n)$ é o i -ésimo elemento do vetor de entrada e $e(n)$ é o sinal de erro, definido como a diferença entre o sinal desejado e a saída do filtro adaptativo (veja Figura 1.2), sendo o sinal desejado obtido de acordo com a aplicação específica. Em (1.2), note que o passo de adaptação é global (mesmo passo para todos os coeficientes do filtro adaptativo) e constante no tempo.

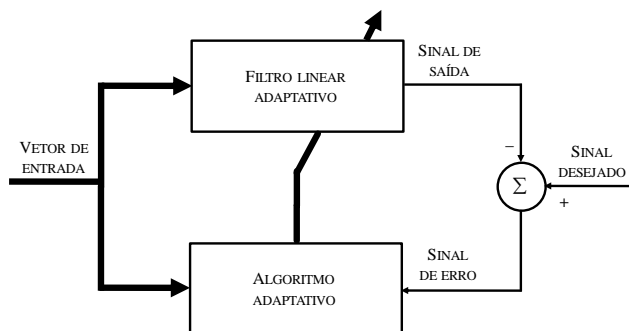


Figura 1.2. Diagrama de blocos de um problema geral de filtragem adaptativa.

Inicialmente, o algoritmo LMS foi empregado com sucesso em antenas adaptativas e cancelamento de ruído [28]-[31]. A sua utilização em aplicações práticas cresceu significativamente após a invenção das seguintes aplicações no Bell Telephone Laboratories:

- a) **Equalizador adaptativo.** Essa aplicação foi desenvolvida por R. W. Lucky em 1965, sendo, na literatura técnica, usualmente referida como *decision-direct learning* [28], [30].
- b) **Cancelador de eco.** A idéia do cancelamento de eco, ilustrada na Figura 1.3, é gerar uma réplica sintética do sinal de eco (saída do filtro adaptativo) e subtraí-la do sinal de retorno (soma do sinal de eco com o sinal de voz do usuário *near-end*), melhorando assim a qualidade das conversações telefônicas. Tal idéia foi inicialmente proposta por J. L. Kelly

e, posteriormente, implementada por M. M. Sondhi e A. J. Presti em 1966 [30], [32].

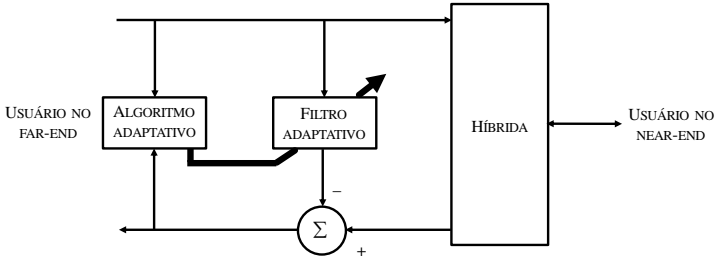


Figura 1.3. Diagrama de blocos simplificado de um sistema de cancelamento de eco.

Conforme (1.2), no algoritmo LMS, o ajuste do vetor de coeficientes é diretamente proporcional ao vetor de entrada, tornando o comportamento do algoritmo sensível às variações na potência desse sinal. Para contornar tal problema, os canceladores de eco começaram a ser implementados usando uma variante do algoritmo LMS denominada LMS normalizado (NLMS). Tal algoritmo pode ser representado por [6]:

$$w_i(n+1) = w_i(n) + \underbrace{\frac{\mu}{N\hat{\sigma}_x^2(n) + \xi}}_{\text{passo de adaptação}} e(n)x_i(n) \quad (1.3)$$

termo de correção

onde N é o comprimento do filtro adaptativo, $\hat{\sigma}_x^2(n)$ é uma estimativa da variância do sinal de entrada e $\xi > 0$ é um parâmetro de regularização (prevenindo divisão por zero e estabilizando a solução). Agora, o passo de adaptação é variante no tempo, porém ainda global.

Em 2000, no Bell Laboratories, Donald L. Duttweiler propôs uma variante do algoritmo NLMS, aqui denominado NLMS proporcional (PNLMS) original, na qual o passo de adaptação associado ao i -ésimo coeficiente é proporcional à magnitude do correspondente coeficiente

[6], [32]. A equação de adaptação do algoritmo PNLMS original [3] é dada por

$$w_i(n+1) = w_i(n) + \underbrace{\frac{\mu g_i(n)}{N\sigma_x^2(n)\bar{g}(n) + \xi}}_{\text{passo de adaptação}} e(n)x_i(n) \quad (1.4)$$

$\underbrace{\hspace{10em}}_{\text{termo de correção}}$

onde $g_i(n)$ é o ganho de adaptação individual associado ao coeficiente $w_i(n)$ e $\bar{g}(n)$ é a média aritmética dos ganhos individuais. A forma como os ganhos individuais são calculados é mostrada na próxima seção. Basicamente, a adaptação proporcional visa fazer $g_i(n)$ proporcional a $|w_i(n)|$. Conseqüentemente, os coeficientes com maior magnitude tendem a convergir mais rapidamente. Dessa forma, quando a planta é esparsa, verifica-se que a velocidade de convergência do algoritmo PNLMS original tende a ser superior à do algoritmo NLMS.

A idéia da adaptação proporcional consiste nos seguintes pontos fundamentais:

- i) Habilidade de aprendizado de características da planta (como, por exemplo, grau de esparsidade) a partir de dados estimados pelo algoritmo.
- ii) Exploração deste aprendizado para aumentar a velocidade de convergência.
- iii) Utilização de passo de adaptação individual e variante no tempo, conforme (1.4). Note que essa característica contrasta com aquelas observadas nos algoritmos LMS [passo de adaptação global e constante conforme (1.2)] e NLMS [passo de adaptação global e variante no tempo de acordo com (1.3)].

Aqui, ressalta-se a importância histórica do algoritmo PNLMS original, visto que, a partir dele, muitas variantes vêm sendo desenvolvidas, constituindo a classe PNLMS de algoritmos adaptativos [3], [20]-[21].

B. Algoritmo PNLMS Padrão

O algoritmo PNLMS padrão (forma compacta e elegante do algoritmo, sendo comumente adotada na literatura aberta) é formulado pelo seguinte conjunto de equações [2], [8]:

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

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

Sinal de erro

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n) + z(n) \quad (1.6)$$

Matriz de distribuição de ganhos ($N \times N$)

$$\mathbf{G}(n) = \text{diag}[g_1(n) \ g_2(n) \ \cdots \ g_N(n)] \quad (1.7)$$

Ganho individual

$$g_i(n) = \frac{\phi_i(n)}{\frac{1}{N} \sum_{j=1}^N \phi_j(n)}, \quad i = 1, 2, \dots, N \quad (1.8)$$

Função de proporcionalidade

$$\phi_i(n) = \max[a(n), |w_i(n)|] \quad (1.9)$$

Fator de ativação

$$a(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_\infty] \quad (1.10)$$

onde $0 < \mu < 2$ é o parâmetro de passo e $\|\cdot\|_\infty$ caracteriza a norma infinita de um vetor. A variável $d(n)$ representa o sinal desejado e

$z(n)$, um ruído de medição independente e identicamente distribuído (i.i.d.) com média zero, variância σ_z^2 , e não correlacionado com qualquer outro sinal do sistema. O vetor de entrada é $\mathbf{x}(n)=[x(n) \ x(n-1) \cdots \ x(n-N+1)]^T$ e o vetor contendo os N coeficientes do filtro adaptativo é dado por $\mathbf{w}(n)=[w_1(n) \ w_2(n) \cdots \ w_N(n)]^T$. A matriz diagonal $\mathbf{G}(n)$ distribui os ganhos $g_i(n)$ entre os coeficientes, dessa forma, governando o ajuste dos passos de adaptação. O fator de ativação $a(n)$, dado em (1.10), depende do vetor de coeficientes do filtro adaptativo, bem como de ρ e δ , os quais são, respectivamente, os parâmetros de proporcionalidade (ou ativação) e inicialização. O parâmetro de inicialização permite a partida do processo de adaptação para $n=0$, quando todos os coeficientes do filtro são inicializados em zero. O parâmetro de proporcionalidade evita que um coeficiente individual “congele” quando sua magnitude é nula ou muito próxima de zero [5], [8].

1.2. Motivações para a Derivação de um Algoritmo PNLMS Melhorado

Nesta seção, destaca-se uma das contribuições específicas deste trabalho para a classe de algoritmos proporcionais.

O comportamento do algoritmo PNLMS padrão depende dos parâmetros que controlam a proporcionalidade e a inicialização, os quais são de difícil ajuste [2]. Assim, um ponto central é como estabelecer valores adequados para esses parâmetros, visto que eles afetam a velocidade de convergência do algoritmo. Tais parâmetros estão relacionados ao fator de ativação, cuja função é prevenir que os coeficientes do filtro adaptativo “congelem”. No algoritmo PNLMS padrão, o fator de ativação é comum a todos os coeficientes, calculado amostra por amostra e dependente também da norma infinita do vetor de coeficientes do filtro adaptativo [veja (1.10)]; conseqüentemente, essa abordagem para obter o fator de ativação conduz a uma distribuição de ganhos que não atende plenamente o conceito de proporcionalidade, o qual é um atributo desejável para os algoritmos da classe PNLMS (veja Capítulo 2).

No Capítulo 4, uma nova estratégia para determinar o fator de ativação é proposta, melhorando o comportamento do algoritmo. Em contraste com o algoritmo PNLMS padrão, a nova abordagem usada para determinar o fator de ativação apresenta as seguintes características:

- i) Utilização de um fator de ativação individual.
- ii) Cada fator de ativação individual é calculado em função da magnitude do correspondente coeficiente do filtro adaptativo.
- iii) O fator de ativação individual não depende dos parâmetros de proporcionalidade e inicialização, tendo em vista que tais parâmetros não estão mais presentes na formulação proposta.

Como consequência, a velocidade de convergência do algoritmo proposto aumenta significativamente, visto que agora existe um fator de ativação individual para cada coeficiente. A nova versão do algoritmo é denominada PNLMS com fator de ativação individual (IAF-PNLMS) [33]-[34].

Para respostas ao impulso com elevado grau de esparsidade, resultados de simulação mostram que o algoritmo proposto apresenta uma maior velocidade de convergência bem como resposta mais rápida a perturbações na planta do que os algoritmos PNLMS padrão e IPNLMS.

1.3. Modelagem dos Algoritmos da Classe PNLMS

A modelagem estocástica é uma ferramenta importante na área de processamento adaptativo de sinais. Seus principais propósitos são prever o desempenho do algoritmo adaptativo sob diferentes condições de operação e obter valores apropriados para os parâmetros do algoritmo, tais como valor do passo, desajuste, comprimento do filtro, dentre outros [35]-[38]. Dessa forma, tendo-se as expressões do modelo em mãos, simulações exaustivas de Monte Carlo (MC) podem ser evitadas. Além do mais, o modelo pode ser usado para identificar algum comportamento indesejado do algoritmo; em seguida, fazendo as modificações adequadas, pode-se contornar tal comportamento, resultando em um algoritmo melhorado [37].

Os algoritmos da classe proporcional (tais como PNLMS padrão, IPNLMS, IAF-PNLMS) vêm sendo amplamente discutidos na literatura técnica [5]-[12]. Entretanto, somente alguns poucos trabalhos de pesquisa tratam da modelagem desses algoritmos, devido à complexidade matemática envolvida no desenvolvimento dos modelos. Visando obter expressões matematicamente tratáveis, os modelos existentes são baseados em algumas simplificações severas, o que freqüentemente resulta em um modelo que não reflete o comportamento do algoritmo. Por exemplo, um modelo simplificado para os algoritmos PNLMS é apresentado em [6]. Para obter tal modelo, o sinal de entrada é assumido branco e a matriz de ganhos, invariante no tempo. Obviamente, esta última restrição, também usada em [39], permite derivar expressões que predizem o comportamento do algoritmo apenas em regime estacionário. Um modelo mais realista é apresentado em [40], no qual a matriz de ganhos é agora considerada variante no tempo, levando a um modelo que prevê o comportamento transitório do algoritmo. Contudo, tal modelo é obtido para condições restritas, isto é, usando uma versão simplificada do algoritmo e assumindo sinal de entrada branco. Assim sendo, em nosso conhecimento, a modelagem dos algoritmos da classe PNLMS ainda é um tema em aberto na literatura especializada. Portanto, no que diz respeito à modelagem dos algoritmos da classe PNLMS, este trabalho tem os seguintes objetivos específicos:

- i) Derivar um modelo estocástico para a evolução dos coeficientes adaptativos do algoritmo IAF-PNLMS.
- ii) Determinar as expressões de um modelo estocástico para o momento de segunda ordem do vetor de erro dos coeficientes e da curva de aprendizagem do algoritmo IAF-PNLMS.
- iii) Formular uma metodologia de análise geral que possa ser aplicada a outros algoritmos da classe PNLMS.

1.4. Trabalhos Publicados

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

- 1) F. 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.
- 2) F. C. de Souza, O. J. Tobias, R. Seara, "Considerações sobre o algoritmo PNLMS com fatores de ativação individuais," in *XXVII Simpósio Brasileiro de Telecomunicações (SBrT 2009)*, Blumenau-SC, Brazil, Out. 2009.
- 3) F. 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.
- 4) F. C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, "Stochastic model for the mean weight evolution of the IAF-PNLMS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5895-5901, Nov. 2010.
- 5) F. C. de Souza, R. Seara, "Considerações sobre a modelagem estocástica dos coeficientes do algoritmo adaptativo IAF-PNLMS," in *XXIX Simpósio Brasileiro de Telecomunicações (SBrT 2011)*, Curitiba-PR, Brazil, Out. 2011.
- 6) F. C. de Souza, R. Seara, and D. R. Morgan, "An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3301-3307, Jun. 2012.

1.5. Organização do Manuscrito

O trabalho de pesquisa aqui apresentado é organizado como segue. O Capítulo 2 apresenta um estudo do algoritmo PNLMS padrão, mostrando os efeitos dos parâmetros desse algoritmo em seu comportamento. Através de simulações numéricas, é mostrado que o fator de ativação tem um impacto importante no comportamento do algoritmo. O Capítulo 3 apresenta as principais variantes do algoritmo PNLMS padrão discutidas na literatura. No Capítulo 4, uma das

principais contribuições deste trabalho é apresentada: a derivação do algoritmo IAF-PNLMS, verificando as suas características, propriedades e desempenho. No Capítulo 5, é proposta uma metodologia geral para a modelagem estocástica dos algoritmos da classe PNLMS, derivando expressões para o momento de primeira ordem do vetor de coeficientes do algoritmo IAF-PNLMS. O Capítulo 6 apresenta as expressões de um modelo estocástico para o momento de segunda ordem do vetor de erro dos coeficientes; também, a expressão de modelo para a curva de aprendizagem do algoritmo IAF-PNLMS é obtida. No Capítulo 7, outra importante contribuição deste trabalho tese é apresentada: a derivação de uma versão melhorada do algoritmo IAF-PNLMS, denominada EIAF-PNLMS (*enhanced* IAF-PNLMS), a qual considera que a filosofia proporcional não precisa ser aplicada durante todo o processo de adaptação, em contraste aos outros algoritmos da classe PNLMS. Finalmente, no Capítulo 8, são apresentados os comentários e as conclusões finais de tese.

1.6. Conclusões

Neste capítulo, os conceitos gerais concernentes aos algoritmos da classe PNLMS (como filtragem adaptativa proporcional e grau de esparsidade), a motivação para o desenvolvimento de tais algoritmos, bem como os objetivos gerais e específicos deste trabalho de pesquisa foram apresentados. Também, os aspectos históricos e filosóficos da filtragem adaptativa proporcional foram brevemente discutidos.

Impacto do Fator de Ativação no Comportamento do Algoritmo PNLMS Padrão

2.1. Introdução

Neste capítulo, o comportamento do algoritmo PNLMS padrão é discutido e analisado, visando obter uma versão melhorada desse algoritmo. Para tal, o ponto central é estudar o efeito do fator de ativação $a(n)$ e dos seus parâmetros associados (proporcionalidade ρ e inicialização δ) sobre o comportamento do algoritmo.

O fator de ativação $a(n)$ tem como funções possibilitar a partida do algoritmo como também impedir o congelamento dos coeficientes inativos durante o processo de adaptação, evitando o funcionamento inadequado do algoritmo em certas situações. Originalmente, $a(n)$ foi considerando um fator secundário dentro do algoritmo [6]. Entretanto, recentes trabalhos têm mostrado que $a(n)$ afeta o comportamento do algoritmo de forma significativa [20], [33]-[34]. Por outro lado, tem sido verificado que tal fator e seus parâmetros associados podem ser utilizados para prover o algoritmo de propriedades desejáveis, como, por exemplo, robustez em relação à variação da esparsidade da planta [12].

A seguir são discutidos os efeitos do fator de ativação e dos ganhos de adaptação do algoritmo PNLMS padrão.

2.2. Ganhos de Adaptação

A partir de (1.8) e (1.9), $g_i(n)$ é reescrito como

$$g_i(n) = \frac{1}{c(n)} \max[a(n), |w_i(n)|] \quad (2.1)$$

com

$$c(n) = \frac{1}{N} \sum_{j=1}^N \phi_j(n), \quad j = 1, 2, \dots, N \quad (2.2)$$

sendo uma variável comum a todos os ganhos definidos em (2.1). Agora, analisando (2.1), algumas definições importantes são apresentadas:

- i) **Ganho dos coeficientes inativos** $g^{\text{inativo}}(n)$. Se $a(n) > |w_i(n)|$, o i -ésimo coeficiente $w_i(n)$ é inativo³ e seu ganho associado, obtido a partir de (2.1), é dado por

$$g^{\text{inativo}}(n) = \frac{1}{c(n)} a(n). \quad (2.3)$$

Visto que o fator de ativação $a(n)$ é comum a todos os coeficientes do filtro, um ganho mínimo e comum $g^{\text{inativo}}(n)$ é atribuído a todos os coeficientes inativos. Essa é uma característica indesejável do algoritmo, pois não está em concordância com a filosofia proporcional.

- ii) **Ganho dos coeficientes ativos** $g_i^{\text{ativo}}(n)$. Se $a(n) \leq |w_i(n)|$, o i -ésimo coeficiente $w_i(n)$ é ativo e seu ganho associado, obtido a partir de (2.1), é dado por

$$g_i^{\text{ativo}}(n) = \frac{1}{c(n)} |w_i(n)|. \quad (2.4)$$

Aqui, cada ganho $g_i^{\text{ativo}}(n)$ depende da magnitude do correspondente coeficiente ativo $w_i(n)$. Essa característica representa a essência da filosofia proporcional.

³ Qualitativamente, é considerado coeficiente inativo aquele que tem magnitude nula ou próxima de zero. Por outro lado, aquele coeficiente que tem magnitude significativa é considerado ativo [1], [23], [43].

- iii) **Ganho total distribuído entre os coeficientes adaptativos em cada iteração** $g_{\text{total}}(n)$. A partir de (1.7), note que o ganho total em cada iteração é expresso pelo traço da matriz $\mathbf{G}(n)$. Agora, usando (1.8), tal ganho é escrito como

$$\begin{aligned}
 g_{\text{total}}(n) &= \text{tr}[\mathbf{G}(n)] \\
 &= g_1(n) + g_2(n) + \dots + g_N(n) \\
 &= \frac{\phi_1(n) + \phi_2(n) + \dots + \phi_N(n)}{\frac{1}{N} \sum_{j=1}^N \phi_j(n)} \\
 &= N.
 \end{aligned} \tag{2.5}$$

Note que, por construção, o ganho $g_{\text{total}}(n)$ é constante no tempo. Alternativamente, considerando as definições i) e ii), esse ganho pode ser reescrito como

$$g_{\text{total}}(n) = \frac{N - N_{\text{ativo}}}{c(n)} a(n) + \sum_{i \in A} g_i(n) \tag{2.6}$$

onde N_{ativo} é o número de coeficientes ativos e A é o conjunto de índices associados às suas posições.

Em (2.6), o primeiro termo do lado direito representa o ganho total distribuído entre os coeficientes inativos e o segundo, o ganho total distribuído entre os coeficientes ativos. Note que um decréscimo em $a(n)$ implica um aumento no segundo termo do lado direito de (2.6), visto que $g_{\text{total}}(n)$ é sempre constante e igual a N . Isso significa que os ganhos associados aos coeficientes ativos também são dependentes de $a(n)$. Portanto, conclui-se que o fator de ativação $a(n)$ afeta os ganhos atribuídos aos coeficientes ativos bem como aos inativos. Logo, $a(n)$ tem um papel fundamental na distribuição de ganhos do algoritmo.

2.3. Estudo do Comportamento do Algoritmo

Nesta seção, o comportamento do algoritmo PNLMS padrão em relação ao fator de ativação é estudado em detalhes. Para tal, considerando um problema de identificação de sistemas, simulações de Monte Carlo (MC) são realizadas (média de 100 realizações independentes). O cenário para todas as simulações consiste de uma resposta ao impulso esparsa com $N=100$ coeficientes, dada em [7]. Seus coeficientes ativos têm valores iguais a $\{0,1, 1,0, -0,5, 0,1\}$, os quais estão localizados nas posições $\{1, 30, 35, 85\}$, respectivamente. Utilizando (1.1), o grau de esparsidade dessa resposta ao impulso é $S_D(\mathbf{w}^o) = 0,9435$, onde $\mathbf{w}^o = [w_1^o \ w_2^o \ \dots \ w_N^o]^T$ representa a resposta ao impulso a ser identificada. O sinal de entrada é correlacionado, com média zero e variância unitária, obtido através de um processo AR(2) dado por

$$x(n) = b_1 x(n-1) + b_2 x(n-2) + v(n) \quad (2.7)$$

onde $v(n)$ é um ruído branco com variância $\sigma_v^2 = 0,77$ ($b_1 = 0,40$ e $b_2 = -0,40$) e a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é $\chi = 10$. O ruído de medição $z(n)$ é branco com variância $\sigma_z^2 = 10^{-3}$ (SNR = 30 dB).

2.3.1. Avaliação do Comportamento

O comportamento do algoritmo PNLMS padrão é avaliado utilizando o desalinhamento normalizado (em dB), dado por [1]

$$\kappa(n) = 10 \log_{10} \frac{\|\mathbf{w}^o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}^o\|_2^2}. \quad (2.8)$$

Para mostrar a dependência do algoritmo com respeito ao fator de ativação, curvas de desalinhamento são mostradas na Figura 2.1, considerando $a(n)$ dado por (1.10), $\mu = 0,5$ e $\delta = 0,01$ com o parâmetro ρ assumindo os seguintes valores: 0,01, 0,05 e 0,50. Além

disso, uma outra simulação numérica, agora usando um fator de ativação constante $a(n)=10^{-3}$, manualmente ajustado para maximizar a velocidade de convergência, é também incluída na Figura 2.1. Essa figura mostra que a convergência do algoritmo é impactada pelo fator de ativação bem como pelo parâmetro ρ . Note que usando $a(n)$ dado em (1.10) e reduzindo o valor de ρ , a velocidade de convergência aumenta.

Para o caso em que $a(n)=10^{-3}$, uma velocidade de convergência maior é obtida, visto que a distribuição de ganhos do algoritmo atende melhor a filosofia proporcional, isto é, um menor ganho por iteração é atribuído aos coeficientes inativos (veja Seção 2.3.2). Dessa forma, a partir das curvas mostradas na Figura 2.1, um fato interessante pode ser inferido: o fator de ativação dado por (1.10) não é o melhor caminho para se obter uma distribuição de ganhos adequada.

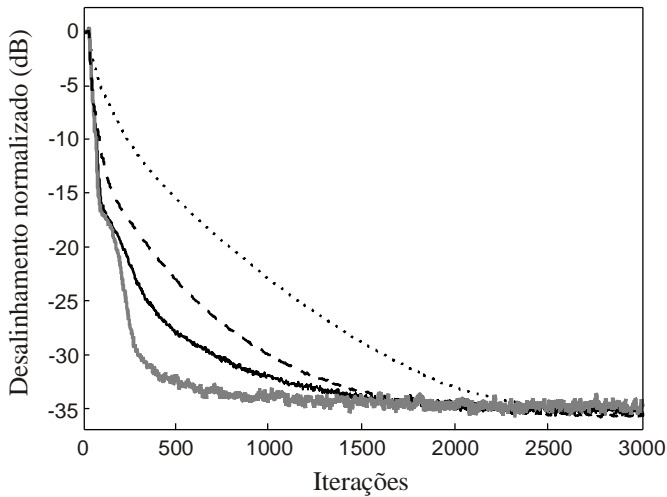


Figura 2.1. Desalinhamento normalizado do algoritmo PNLMS padrão usando $\mu = 0,5$ e $\delta = 0,01$. (Linha pontilhada) $\rho = 0,50$ e $a(n)$ dado por (1.10). (Linha tracejada) $\rho = 0,05$ e $a(n)$ dado por (1.10). (Linha escura sólida) $\rho = 0,01$ e $a(n)$ dado por (1.10). (Linha cinza sólida) $a(n) = 10^{-3}$.

2.3.2. Distribuição de Ganhos

Para avaliar a distribuição de ganhos do algoritmo, as seguintes figuras de mérito são definidas:

Distribuição de ganho total para L iterações

$$\theta_i = \sum_{n=0}^{L-1} g_i(n) \quad (2.9)$$

Média dos ganhos totais θ_i atribuídos aos coeficientes inativos

$$\theta_{\text{med}}^{\text{inativo}} = \frac{1}{N - N_{\text{ativo}}} \sum_{i \notin A} \theta_i. \quad (2.10)$$

Na Tabela 2.1, as distribuições de ganhos totais obtidas para os casos considerados na Figura 2.1 são apresentadas. Os valores de ganhos dessa tabela são obtidos através de (2.9) e (2.10) com $N_{\text{ativo}} = 4$ e $A = \{1, 30, 35, 85\}$. Note da tabela que, no caso do fator de ativação constante [$a(n) = 10^{-3}$], uma maior parcela de ganho dos coeficientes inativos é transferida para os ativos, resultando em uma melhor distribuição de ganhos de adaptação.

TABELA 2.1
DISTRIBUIÇÃO DE GANHOS TOTAIS DO ALGORITMO
PNLMS PADRÃO PARA 5000 ITERAÇÕES

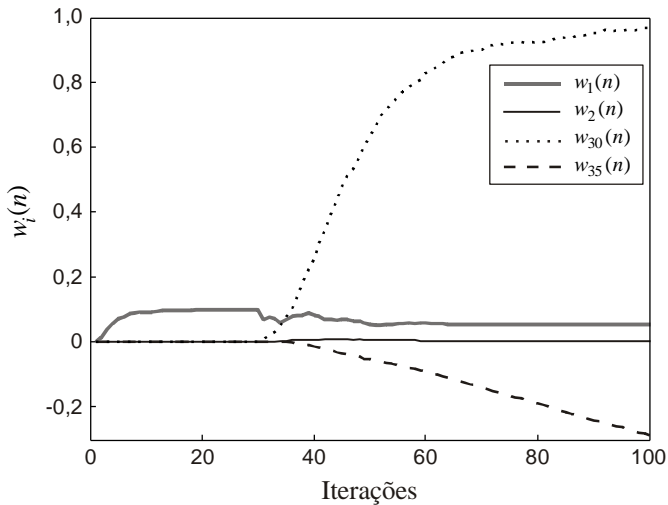
Parâmetros	θ_1	θ_{30}	θ_{35}	θ_{85}	$\theta_{\text{med}}^{\text{inativo}}$
$\rho = 0,01$	19800	186850	92450	17680	1910
$\rho = 0,05$	8050	76530	37620	7210	3860
$\rho = 0,50$	4990	9860	4950	4950	4950
$a(n) = 10^{-3}$	28390	274930	136400	26220	350

2.3.3. Comportamento de Coeficientes e Ganhos

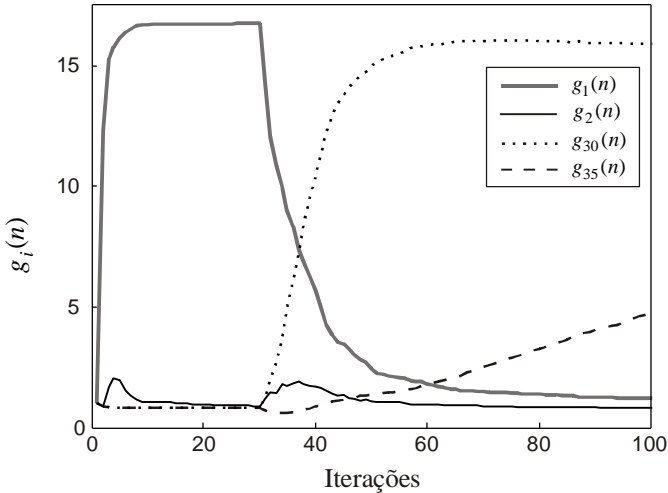
Aqui, mostra-se que os coeficientes e ganhos do algoritmo PNLMS padrão podem apresentar um comportamento indesejável no início da fase de aprendizagem.

A Figura 2.2 mostra o comportamento inicial dos coeficientes ativos $w_1(n)$, $w_{30}(n)$ e $w_{35}(n)$, do coeficiente inativo $w_2(n)$ e dos correspondentes ganhos $g_1(n)$, $g_{30}(n)$, $g_{35}(n)$ e $g_2(n)$. A partir da Figura 2.2(a), observa-se que $|w_{30}(n)| < |w_1(n)|$ no início da fase de aprendizagem ($0 \leq n < 30$), resultando em $g_{30}(n) < g_1(n)$ [veja Figura 2.2(b)]. Esse comportamento não é desejável, visto que $g_1(n)$ e $g_{30}(n)$ não são proporcionais às magnitudes dos correspondentes coeficientes da planta $|w_1^o| = 0,1$ e $|w_{30}^o| = 1,0$, em contraste à condição desejada

$g_{30}(n) > g_1(n)$. Esse fato mostra que o algoritmo PNLMS padrão pode apresentar um comportamento inadequado no início do processo de adaptação. Tal problema é considerado no Capítulo 4 para a formulação de uma versão melhorada do algoritmo.



(a)



(b)

Figura 2.2. Curvas de $w_i(n)$ e $g_i(n)$ para $\mu=0,5$, $\rho=0,05$ e $\delta=0,01$. (a) Coeficientes adaptativos $w_1(n)$, $w_2(n)$, $w_{30}(n)$ e $w_{35}(n)$. (b) Ganhos de adaptação $g_1(n)$, $g_2(n)$, $g_{30}(n)$ e $g_{35}(n)$.

2.4. Conclusões

Neste capítulo, foi apresentado um estudo do comportamento do algoritmo PNLMS padrão considerando os efeitos do fator de ativação. Deste estudo, as seguintes conclusões são obtidas:

- a) O fator de ativação tem um impacto significativo na distribuição de ganhos do algoritmo PNLMS padrão, afetando não somente os ganhos atribuídos aos coeficientes inativos, mas também aqueles atribuídos aos ativos.
- b) A forma como o fator de ativação é determinado e os valores selecionados para os parâmetros ρ e δ alteram substancialmente a velocidade de convergência do algoritmo. Assim, (1.10) não é a expressão mais adequada para se obter uma distribuição de ganhos satisfatória.

- c) Os coeficientes e ganhos do algoritmo PNLMS padrão podem apresentar um comportamento indesejável no início da fase de aprendizagem.

Considerando a), b) e c), uma nova abordagem para a determinação do fator de ativação é proposta no Capítulo 4.

Variantes do Algoritmo PNLMS Padrão

3.1. Introdução

As vantagens do algoritmo PNLMS padrão em relação ao algoritmo NLMS, quando usado em aplicações cujas respostas ao impulso são esparsas, estão bem documentadas na literatura [1]-[3], [5]-[6]. Durante a fase inicial do processo de adaptação, a velocidade de convergência do algoritmo PNLMS padrão é alta; contudo, após a fase inicial, essa velocidade reduz-se drasticamente [5], [9]—[10]. Além disso, o desempenho de tal algoritmo sofre degradação à medida que o grau de esparsidade da resposta ao impulso é reduzido, podendo apresentar um desempenho inferior àquele proporcionado pelo algoritmo NLMS. Para contornar tais desvantagens, algumas variantes do algoritmo PNLMS padrão têm sido propostas na literatura, tais como o PNLMS++, o PNLMS melhorado (IPNLMS), o PNLMS com lei μ (MPNLM), dentre outras. Neste capítulo, os algoritmos IPNLMS e MPNLMS (as mais importantes variantes do algoritmo PNLMS padrão) são revisitados.

3.2. Algoritmo IPNLMS

O algoritmo IPNLMS, introduzido em [8], utiliza uma regra para determinar os ganhos de adaptação que leva em conta a esparsidade da resposta ao impulso. A seguir, a derivação deste algoritmo é discutida.

3.2.1. Derivação do Algoritmo IPNLMS

O algoritmo IPNLMS é obtido a partir das equações do algoritmo PNLMS padrão, fazendo as seguintes modificações [8]:

- i) Em (1.10), substitui-se a norma infinita do vetor de coeficientes do filtro adaptativo por $\|\mathbf{w}(n)\|_1$ e faz-se $\rho = 1/N$. Dessa forma, considerando que o parâmetro δ é utilizado somente no processo de partida do algoritmo ($n=0$), quando o vetor de coeficientes é nulo, o fator de ativação pode ser expresso como

$$a(n) = \frac{1}{N} \max \left\{ \delta, \|\mathbf{w}(n)\|_1 \right\} \quad (3.1)$$

$$= \frac{\|\mathbf{w}(n)\|_1}{N}, \quad n \geq 1.$$

A partir de (3.1), observa-se que a regra para o cálculo do fator de ativação do algoritmo IPNLMS considera a magnitude de todos os coeficientes do filtro. Essa expressão representa a média aritmética das magnitudes dos elementos do vetor de coeficientes, contrastando com a regra usada no algoritmo PNLMS padrão, em que o fator de ativação é função da norma infinita do vetor de coeficientes. Assim, a função de proporcionalidade do algoritmo IPNLMS é escrita como

$$\phi_i(n) = \max \left\{ \underbrace{\frac{\|\mathbf{w}(n)\|_1}{N}}_{\text{fator de ativação}}, |w_i(n)| \right\}, \quad i = 1, 2, \dots, N. \quad (3.2)$$

- ii) Agora, (3.2) é modificada, considerando a média ponderada entre o fator de ativação e a magnitude $|w_i(n)|$ para calcular a função de proporcionalidade $\phi_i(n)$. Logo,

$$\phi_i(n) = (1 - \alpha) \frac{\|\mathbf{w}(n)\|_1}{N} + (1 + \alpha) |w_i(n)|, \quad -1 \leq \alpha < 1 \quad (3.3)$$

onde α é um parâmetro de ponderação (ou de proporcionalidade) que controla a importância atribuída a cada

uma das parcelas do lado direito de (3.3). Note que o primeiro termo do lado direito de (3.3) é comum a todos os coeficientes do filtro, correspondendo a uma parcela não proporcional; já o segundo termo corresponde à parcela proporcional.

3.2.2. Forma Fechada para o Cálculo do Ganho Individual

A partir de (1.8), o ganho individual pode ser expresso por

$$g_i(n) = \frac{\phi_i(n)}{\frac{1}{N} \|\boldsymbol{\phi}(n)\|_1}, \quad i = 1, 2, \dots, N \quad (3.4)$$

com $\boldsymbol{\phi}(n) = [\phi_1(n) \ \phi_2(n) \ \dots \ \phi_N(n)]^T$ denotando o vetor função de proporcionalidade. Considerando (3.3), a norma 1 de $\boldsymbol{\phi}(n)$ é

$$\begin{aligned} \|\boldsymbol{\phi}(n)\|_1 &\triangleq \sum_{i=1}^N |\phi_i(n)| \\ &= \sum_{i=1}^N \left[(1-\alpha) \frac{\|\mathbf{w}(n)\|_1}{N} + (1+\alpha) |w_i(n)| \right] \\ &= \left[\frac{1-\alpha}{N} \sum_{i=1}^N \|\mathbf{w}(n)\|_1 \right] + \left[(1+\alpha) \sum_{i=1}^N |w_i(n)| \right] \\ &= (1-\alpha) \|\mathbf{w}(n)\|_1 + (1+\alpha) \|\mathbf{w}(n)\|_1 \\ &= 2 \|\mathbf{w}(n)\|_1. \end{aligned} \quad (3.5)$$

Então, substituindo (3.3) e (3.5) em (3.4), obtém-se o ganho individual do algoritmo IPNLMS como

$$g_i(n) = \frac{(1-\alpha)}{2} + \frac{N(1+\alpha)|w_i(n)|}{\|\mathbf{w}(n)\|_1}, \quad i = 1, 2, \dots, N. \quad (3.6)$$

Para evitar divisão por zero, especialmente na inicialização quando $\mathbf{w}(0) = \mathbf{0}$, (3.6) é modificada para

$$g_i(n) = \frac{(1-\alpha)}{2} + \frac{N(1+\alpha)|w_i(n)|}{\|\mathbf{w}(n)\|_1 + \varsigma} \quad (3.7)$$

onde ς é uma pequena constante positiva [8]. Na Tabela 3.1, um resumo desse algoritmo é apresentado.

TABELA 3.1
ALGORITMO IPNLMS

Sinal de erro

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

Ganho individual

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

Matriz de ganhos ($N \times N$)

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

Equação de atualização ($N \times 1$)

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

Note de (3.7) que o i -ésimo ganho de adaptação do algoritmo IPNLMS exibe as seguintes características:

- i) Cada ganho individual $g_i(n)$ tem um termo constante dado por $2^{-1}(1-\alpha)$, representando a parcela não proporcional do ganho. Observe que essa parcela não depende de $|w_i(n)|$. Dessa forma, quando a magnitude do i -ésimo coeficiente é nula ou muito pequena, o primeiro termo do lado direito de (3.7) é preponderante em relação ao segundo termo. Portanto, o termo constante do ganho individual é responsável pelos ganhos atribuídos aos coeficientes inativos.
- ii) A função do parâmetro α é controlar as parcelas proporcional e não proporcional do ganho individual $g_i(n)$. Note, a partir de (3.7), que

$$\lim_{\alpha \rightarrow 1} g_i(n) = \frac{2N|w_i(n)|}{\|\mathbf{w}(n)\|_1 + \zeta} \quad (3.8)$$

Nessa condição, o algoritmo IPNLMS tem um comportamento similar ao do algoritmo PNLMS padrão, sendo apropriado para aplicações nas quais o grau de esparsidade da planta é próximo de 1. Por outro lado, a condição

$$\lim_{\alpha \rightarrow -1} g_i(n) = 1 \quad (3.9)$$

leva o algoritmo IPNLMS a se comportar como o algoritmo NLMS, sendo agora apropriado para aplicações nas quais a planta não é esparsa. À medida que a esparsidade da planta aumenta, deve-se aumentar o valor de α [conforme indica (3.7)], o que leva o algoritmo a ter um comportamento apropriado para uma condição de maior esparsidade. Os valores de α que levam a resultados satisfatórios para plantas com graus de esparsidade médio e alto estão na faixa $-0,5 \leq \alpha \leq 0$ [8]. Para

tais valores, o algoritmo IPNLMS, em geral, apresenta desempenho superior àquele apresentado pelos algoritmos PNLMS padrão e NLMS [8].

3.3. Algoritmo MPNLMS

Como é bem conhecido na literatura, o algoritmo PNLMS padrão apresenta uma maior velocidade de convergência apenas na fase inicial do processo de adaptação. Visando contornar essa desvantagem, o algoritmo PNLMS com lei μ (MPNLMS) é proposto em [9]-[10], o qual fornece uma velocidade de convergência uniforme durante todo o processo de adaptação.

Nesta seção, o algoritmo MPNLMS é revisitado. Primeiramente, medidas quantitativas da velocidade de convergência do algoritmo *steepest descent* (SD) são definidas; em seguida, a matriz de ganhos de um algoritmo SD proporcional que leva à máxima velocidade de convergência é determinada e, finalmente, o algoritmo MPNLMS é derivado a partir do algoritmo SD proporcional.

3.3.1. Algoritmo *Steepest Descent*

A equação de atualização dos coeficientes do algoritmo SD é dada por [9], [29]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \quad (3.10)$$

onde $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ é a matriz de autocorrelação do vetor de entrada e o vetor $\mathbf{p} = E[d(n)\mathbf{x}(n)]$ denota a correlação cruzada entre o sinal desejado e o vetor de sinal de entrada. Agora, considerando que o vetor de inicialização dos coeficientes do filtro adaptativo é $\mathbf{w}(0)$, o vetor de coeficientes do filtro adaptativo na n -ésima iteração é obtido a partir de (3.10) como

$$\mathbf{w}(n) = \mathbf{w}^o + (\mathbf{I} - \mu\mathbf{R})^n[\mathbf{w}(0) - \mathbf{w}^o] \quad (3.11)$$

onde $\mathbf{w}^o = \mathbf{R}^{-1}\mathbf{p}$ é a solução ótima (solução de Wiener) e \mathbf{I} denota a matriz identidade. Usando $\mathbf{w}(0) = \mathbf{0}$, uma escolha razoável para aplicações nas quais a planta é esparsa, visto que a maioria dos seus coeficientes é nulo, e assumindo em (3.11) que $\mathbf{R} = \sigma_x^2 \mathbf{I}$, ou seja, sinal de entrada branco, obtém-se [9]

$$w_i(n) = \left[1 - (1 - \mu\sigma_x^2)^n \right] w_i^o. \quad (3.12)$$

A partir de (3.12), importantes parâmetros de convergência são obtidos como segue.

A. Medidas Quantitativas da Velocidade de Convergência

Visando avaliar a convergência do algoritmo SD, as seguintes quantidades associadas à velocidade de convergência são definidas [9]-[10]:

- i) **Tempo de convergência do i -ésimo coeficiente.** É o número de iterações τ_i necessário para que o i -ésimo coeficiente do filtro adaptativo atinja a vizinhança ε ($\varepsilon > 0$) do valor ótimo w_i^o . Essa vizinhança pode ser definida como

$$w_i^o - \varepsilon < V_\varepsilon < w_i^o + \varepsilon. \quad (3.13)$$

Assim, $w_i(n)|_{n=\tau_i} \in V_\varepsilon$.

- ii) **Tempo de convergência global.** É o número de iterações τ_{global} necessário para que todos os coeficientes atinjam o intervalo V_ε . Dessa forma, τ_{global} é dado por

$$\tau_{\text{global}} = \max \{ \tau_1, \tau_2, \dots, \tau_N \}. \quad (3.14)$$

A partir de (3.12) e (3.13), pode-se calcular o tempo de convergência do i -ésimo coeficiente como

$$\tau_i = \frac{1}{\iota} \cdot \ln \frac{|w_i^o|}{\varepsilon} \quad (3.15)$$

com

$$\iota = \ln \frac{1}{|1 - \mu \sigma_x^2|} \quad (3.16)$$

onde é assumido que $|w_i^o| > \varepsilon$ e $0 < |1 - \mu \sigma_x^2| < 1$, garantindo que $\tau_i > 0$.

Analisando (3.15), verifica-se que quanto maior for magnitude do i -ésimo coeficiente w_i^o , maior será o tempo de convergência do i -ésimo coeficiente do filtro adaptativo. Dessa forma, visando reduzir o tempo de convergência dos coeficientes, passos de adaptação maiores devem ser atribuídos aos coeficientes de maior magnitude (princípio dos algoritmos PNLMS) [9]-[10]. Para tal, (3.15) é modificada para

$$\tau_i = \frac{1}{\iota_i} \cdot \ln \frac{|w_i^o|}{\varepsilon} \quad (3.17)$$

com

$$\iota_i = \ln \frac{1}{|1 - \mu g_i \sigma_x^2|} \quad (3.18)$$

onde g_i é um fator de controle de passo a ser determinado (ganho individual). Assim, o passo de adaptação atribuído ao i -ésimo coeficiente é μg_i . Por conseqüência, o tempo de convergência do i -ésimo coeficiente do filtro adaptativo depende agora do i -ésimo ganho g_i , o qual é utilizado para reduzir o tempo de convergência τ_i .

B. Algoritmo Steepest Descent Proporcional

Agora, considerando que é possível reduzir o tempo de convergência dos coeficientes do filtro através dos ganhos g_i , a equação de atualização (3.10) pode ser reescrita como

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{G}[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \quad (3.19)$$

onde $\mathbf{G} = \text{diag}[g_1 \ g_2 \ \dots \ g_N]$ é uma matriz de ganhos individuais, cujo objetivo é controlar os passos de adaptação dos coeficientes do filtro adaptativo. A expressão (3.19) caracteriza o algoritmo SD proporcional, sendo a matriz \mathbf{G} obtida como uma função do vetor de coeficientes ótimos \mathbf{w}^o [9]-[10], [20]-[21].

Em [10], é mostrado que, para o algoritmo SD proporcional com matriz \mathbf{G} variante no tempo, a máxima velocidade de convergência global é atingida se todos os coeficientes convergirem simultaneamente ($\tau_1 = \tau_2 = \dots = \tau_N$) para dentro do intervalo V_ϵ . Dessa forma, utilizando a condição $0 < \mu \sigma_x^2 g_i \ll 1$ e a restrição $\sum_{i=1}^N g_i = N$, pode-se mostrar que [9]-[10]

$$g_i \cong \frac{\ln(|w_i^o|/\epsilon)}{\frac{1}{N} \sum_{j=1}^N \ln(|w_j^o|/\epsilon)} \quad (3.20)$$

é o ganho a ser atribuído ao i -ésimo coeficiente para que a sua velocidade de convergência seja máxima. A expressão (3.20) é ainda modificada para evitar instabilidade numérica, a qual ocorre quando $|w_i^o| = 0$. Assim, para o cálculo de g_i , considera-se

$$g_i \cong \frac{\ln(1 + |w_i^o|/\epsilon)}{\frac{1}{N} \sum_{j=1}^N \ln(1 + |w_j^o|/\epsilon)}. \quad (3.21)$$

Em geral, o vetor \mathbf{w}^o não é conhecido *a priori*. Então, o uso de (3.21) para determinar a matriz de ganhos não é prático. Entretanto, os resultados obtidos para o algoritmo SD proporcional podem ser estendidos para o algoritmo PNLMS padrão, o qual não requer o conhecimento *a priori* do vetor \mathbf{w}^o .

3.3.2. Algoritmo PNLMS com Lei μ

Os resultados obtidos para algoritmo SD proporcional são agora usados para determinar a matriz de ganhos do algoritmo PNLMS padrão, originando o algoritmo PNLMS com lei⁴ μ (MPNLMS). Esse algoritmo tende a exibir características de convergência similares àquelas apresentadas pelo algoritmo SD proporcional [10].

Levando em conta (3.21) e usando $|w_i(n)|$ como uma estimativa de $|w_i^o|$, o ganho de adaptação individual do algoritmo MPNLMS é escrito como

$$g_i(n) \cong \frac{\max[a(n), \ln(1 + |w_i(n)|/\varepsilon)]}{\frac{1}{N} \sum_{j=1}^N \max[a(n), \ln(1 + |w_j(n)|/\varepsilon)]} \quad (3.22)$$

onde $a(n)$ é o fator de ativação usado para evitar congelamento dos coeficientes do filtro adaptativo quando $|w_i(n)| = 0$. Note que o numerador de (3.22) representa a função de proporcionalidade $\phi_i(n)$ e o denominador, a norma $\|\phi(n)\|_1$ multiplicada por $1/N$. Um resumo do algoritmo MPNLMS é apresentado na Tabela 3.1.

⁴ A terminologia lei μ usada para denominar o algoritmo MPNLMS é uma alusão a um método de quantização não uniforme utilizado em aplicações de telecomunicações, o qual usa a função logarítmica para implementar a compressão de dados.

TABELA 3.2
ALGORITMO MPNLMS

Sinal de erro

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

Fator de ativação

$$a(n) = \rho \max \left[\delta, \ln(1 + |w_1(n)|/\varepsilon), \ln(1 + |w_2(n)|/\varepsilon), \dots, \ln(1 + |w_N(n)|/\varepsilon) \right]$$

Função de proporcionalidade

$$\phi_i(n) = \max \left[a(n), \ln(1 + |w_i(n)|/\varepsilon) \right]$$

Ganho individual

$$g_i(n) = \frac{\phi_i(n)}{\frac{1}{N} \|\boldsymbol{\Phi}(n)\|_1}, \quad i = 1, 2, \dots, N$$

Matriz de ganhos ($N \times N$)

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

Equação de atualização ($N \times 1$)

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

3.4. Conclusões

Neste capítulo, os algoritmos IPNLMS e MPNLMS foram revisitados. O algoritmo IPNLMS apresenta uma derivação *ad hoc* a partir do algoritmo PNLMS padrão, visando introduzir um parâmetro de controle de esparsidade. Com tal estratégia, é possível ajustar o comportamento desse algoritmo de acordo com a esparsidade da planta do sistema. No que concerne ao algoritmo MPNLMS, também com derivação *ad hoc*, pode-se considerá-lo uma aproximação estocástica do algoritmo SD proporcional. Em geral, é esperado que o algoritmo MPNLMS herde as características de convergência do algoritmo SD proporcional como, por exemplo, uma velocidade convergência uniforme durante todo o processo de adaptação. Uma desvantagem do algoritmo MPNLMS é a sua elevada complexidade computacional. Além do mais, o desempenho desse algoritmo degrada à medida que a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada aumenta (veja Capítulo 7).

No Capítulo 4, as características de convergência dos algoritmos IPNLMS e MPNLMS, considerando diferentes condições de operação, são avaliadas.

Algoritmo PNLMS com Fatores de Ativação Individuais

4.1. Introdução

O fator de ativação $a(n)$ do algoritmo PNLMS padrão [veja (1.10)] é comum a todos os coeficientes do filtro adaptativo. Além disso, $a(n)$ é uma função da norma $\|\mathbf{w}(n)\|_\infty$, levando a uma distribuição de ganhos que não atende completamente a filosofia proporcional. Como consequência, o desempenho do algoritmo em termos de velocidade de convergência e resposta a perturbações é consideravelmente prejudicado [33]-[34].

Neste capítulo, é proposta uma versão melhorada do algoritmo PNLMS padrão, considerando uma forma alternativa para calcular o fator de ativação. Nessa proposta, são considerados fatores de ativação individuais para cada coeficiente do filtro adaptativo, determinados em função da magnitude dos correspondentes coeficientes. A nova versão do algoritmo é denominada PNLMS com fatores de ativação individuais (IAF-PNLMS), visto que agora existe um fator de ativação para cada coeficiente. Para respostas ao impulso com elevada esparsidade, resultados de simulação mostram que o algoritmo proposto apresenta maior velocidade de convergência e resposta mais rápida a perturbações na planta em relação aos algoritmos PNLMS padrão e IPNLMS.

4.2. Derivação do Algoritmo IAF-PNLMS

Nesta seção, uma nova estratégia para calcular o fator de ativação é proposta. Tal estratégia é baseada na análise do impacto desse fator no desempenho do algoritmo PNLMS padrão apresentada no Capítulo 2.

Agora, concentramos a nossa discussão nas seguintes características:

- i) Quando $w_i(n)$ é um coeficiente ativo, observa-se de (2.4) que seu ganho é sempre proporcional a $|w_i(n)|$.
- ii) Nota-se de (2.3) que, quando $w_i(n)$ é inativo, o ganho não é proporcional a $|w_i(n)|$.

A característica i) é desejável visto que está de acordo com a filosofia proporcional. Por outro lado, ii) não está em concordância com o conceito de proporcionalidade. Aqui, o objetivo é contornar tal desvantagem, fazendo o ganho $g_i(n)$ tender a ser proporcional a $|w_i(n)|$ mesmo que $w_i(n)$ seja inativo.

Com o propósito de obter um algoritmo que atenda melhor a filosofia proporcional, cada ganho atribuído a um coeficiente inativo deve estar associado a um fator de ativação individual $a_i(n)$, no lugar de se usar um fator comum para todos os coeficientes inativos. Assim, substituindo $a(n)$ por $a_i(n)$ em (1.9), a função de proporcionalidade é reescrita como

$$\phi_i(n) = \max[a_i(n), |w_i(n)|] \quad (4.1)$$

e, por consequência, quando $w_i(n)$ é inativo, a partir de (1.8) e (4.1), a expressão (2.3) é modificada para

$$g_i^{\text{inativo}}(n) = \frac{1}{c(n)} a_i(n). \quad (4.2)$$

4.2.1. Condições Requeridas para o Novo Fator de Ativação

Agora, as condições requeridas para cada fator de ativação $a_i(n)$ são estabelecidas como segue:

C1) $a_i(n)$ deve convergir para a correspondente magnitude do coeficiente $|w_i(n)|$, isto é,

$$\lim_{n \rightarrow \infty} [a_i(n) - |w_i(n)|] = 0, \quad i = 1, 2, \dots, N.$$

C2) $a_i(n)$ deve sempre ser maior do que zero, isto é,

$$a_i(n) > 0, \quad i = 1, 2, \dots, N.$$

Assim, se C1 for satisfeita, note a partir de (4.2) que $g_i^{\text{inativo}}(n)$ tende a ser proporcional a $|w_i(n)|$ quando $n \rightarrow \infty$. Além disso, a condição C2 garante que $g_i(n) > 0$ quando $|w_i(n)| = 0$, evitando o congelamento de $w_i(n)$.

4.2.2. Abordagem Proposta para Determinar $a_i(n)$

Nesta seção, o objetivo é fazer o fator de ativação individual $a_i(n)$ tender à magnitude $|w_i(n)|$. Note que no instante $n = 1$, os únicos dados disponíveis são a estimação do i -ésimo coeficiente $w_i(1)$ e a função de proporcionalidade $\phi_i(0)$. De (4.1), verifica-se que $\phi_i(0) = a_i(0)$, visto que o filtro adaptativo é inicializado com $\mathbf{w}(0) = \mathbf{0}$. Então, é razoável a seguinte formulação para calcular $a_i(1)$:

$$a_i(1) = \gamma |w_i(1)| + (1 - \gamma)\phi_i(0) \tag{4.3}$$

onde $0 < \gamma < 1$. Aqui, os fatores de ativação são inicializados considerando um pequeno valor positivo [tipicamente, $a_i(0) = 10^{-2} / N$], tal que $a_i(0) > 0$. Em (4.3), o primeiro termo do lado direito é devido ao objetivo pretendido, isto é, fazer $a_i(n)$ convergir para $|w_i(n)|$. O segundo termo garante que $a_i(1)$ seja sempre maior do que zero. Agora, generalizando essa abordagem para todo n , obtém-se

$$a_i(n) = \gamma |w_i(n)| + (1 - \gamma)\phi_i(n - 1). \tag{4.4}$$

Pode ser mostrado que o fator de ativação $a_i(n)$, dado por (4.4), contém informação passada (memória) da magnitude do i -ésimo coeficiente do filtro adaptativo [veja (I.1) e (I.2) no Apêndice I], onde, neste caso, γ pode ser considerado como um fator de esquecimento. Levando em conta que não se tem conhecimento *a priori* da planta (posição dos coeficientes ativos e tipo de perturbação a que a planta será submetida), usualmente considera-se $\gamma = 1/2$, ponderando igualmente $|w_i(n)|$ e $\phi_i(n-1)$. Assim, obtém-se

$$a_i(n) = \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1). \quad (4.5)$$

Note que $a_i(n)$, dado por (4.5), satisfaz concomitantemente às condições C1 e C2 (veja Apêndice I). Além disso, observe de (4.5) que agora $a_i(n)$ depende de $|w_i(n)|$. Então, para uma operação apropriada do algoritmo, requer-se que as magnitudes instantâneas dos coeficientes estimados sejam proporcionais às magnitudes dos correspondentes coeficientes da planta. Contudo, $|w_i(n)|$ pode não ser proporcional a $|w_i^0(n)|$ no início do processo de adaptação [veja Figura 2.2(a)], o que pode ser contornado fazendo a atualização periódica de $a_i(n)$ após um período de aprendizagem de N amostras (igual ao comprimento do filtro adaptativo). Portanto, (4.5) é reescrita como

$$a_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN, \quad m = 1, 2, 3, \dots \\ a_i(n-1), & \text{caso contrário.} \end{cases} \quad (4.6)$$

As expressões (4.1) e (4.6) caracterizam o algoritmo IAF-PNLMS proposto. Agora, cada coeficiente, seja ativo ou inativo, tem um fator de ativação associado $a_i(n)$ calculado por (4.6). Também, pode-se verificar que o fator de ativação $a_i(n)$ preserva a memória inerente associada ao i -ésimo coeficiente do filtro adaptativo [veja (II.4) e (II.8) no Apêndice II]. Um resumo do algoritmo IAF-PNLMS é apresentado na Tabela 4.1.

4.2.3. Propriedades das Variáveis $a_i(n)$ e $\phi_i(n)$ do Algoritmo IAF-PNLMS

Assumindo a convergência do algoritmo IAF-PNLMS, o fator de ativação $a_i(n)$ e a função de proporcionalidade $\phi_i(n)$, dados, respectivamente, por (4.1) e (4.6), apresentam as seguintes propriedades (veja provas das propriedades no Apêndice II):

P1) Se o i -ésimo coeficiente é ativo, no instante $n = mN$, o fator de ativação $a_i(n)$ é obtido como

$$a_i(mN) = \frac{1}{2} |w_i(mN - 1)| + \frac{1}{2} |w_i(mN)| \quad (4.7)$$

e a função de proporcionalidade $\phi_i(n)$ é

$$\phi_i(mN) = \max \left\{ \frac{1}{2} [|w_i(mN - 1)| + |w_i(mN)|], |w_i(mN)| \right\}. \quad (4.8)$$

P2) Se o i -ésimo coeficiente é inativo, $a_i(n)$ e $\phi_i(n)$, no instante $n = mN$, são, respectivamente, dados por

$$a_i(mN) = \frac{1}{2^m} a_i(0) + \frac{1}{2^m} |w_i(N)| + \frac{1}{2^{m-1}} |w_i(2N)| \\ + \dots + \frac{1}{2^2} |w_i[(m-1)N]| + \frac{1}{2} |w_i(mN)| \quad (4.9)$$

e

$$\phi_i(mN) = \max \left\{ \frac{1}{2^m} a_i(0) + \frac{1}{2^m} |w_i(N)| + \frac{1}{2^{m-1}} |w_i(2N)| \\ + \dots + \frac{1}{2^2} |w_i[(m-1)N]| + \frac{1}{2} |w_i(mN)|, |w_i(mN)| \right\}. \quad (4.10)$$

TABELA 4.1
ALGORITMO IAF-PNLMS

Sinal de erro

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

Fator de ativação

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

Função de proporcionalidade

$$\phi_i(n) = \max[a_i(n), |w_i(n)|]$$

Ganho individual

$$g_i(n) = \frac{\phi_i(n)}{\frac{1}{N}\|\boldsymbol{\phi}(n)\|_1}, \quad i = 1, 2, \dots, N$$

Matriz de ganhos ($N \times N$)

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

Equação de atualização ($N \times 1$)

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

Portanto, quando m aumenta e $|w_i(mN-1)| \rightarrow |w_i(mN)|$, verifica-se que $\phi_i(mN) \rightarrow |w_i(mN)|$ tanto para coeficientes ativos quanto inativos, o que é o objetivo pretendido. Dessa forma, de (1.8), P1 e P2 conclui-se que $g_i(mN)$ tende a ser proporcional a $|w_i(mN)|$ tanto para os coeficientes ativos quanto inativos. Além disso, $\phi_i(mN)$ nunca é zero, visto que o primeiro termo do lado direito de (4.9) é sempre maior do que zero, evitando o congelamento dos coeficientes do filtro adaptativo.

4.2.4. Complexidade Computacional e Memória Requerida

O algoritmo proposto requer uma memória adicional de tamanho $2N$ para armazenar ambos $\phi_i(n-1)$ e $a_i(n-1)$. Por outro lado, a determinação de $\rho \max[\delta, \|\mathbf{w}(n)\|_\infty]$ (exigido pelo algoritmo PNLMS padrão) não é mais necessária para o algoritmo IAF-PNLMS, economizando-se assim N operações de comparação e uma operação de multiplicação. No que concerne à determinação dos fatores de ativação, requer-se uma adição e uma multiplicação por iteração, visto que tais fatores são calculados somente a cada N amostras. Na Tabela 4.2, a complexidade computacional e a memória requerida pelos algoritmos NLMS, PNLMS padrão e IAF-PNLMS são indicadas. Visto que $\mathbf{G}(n)$ é uma matriz diagonal, o termo $\mathbf{G}(n)\mathbf{x}(n)$ em (1.5) pode ser obtido como segue:

$$\mathbf{G}(n)\mathbf{x}(n) = [g_1(n)x(n) \quad g_2(n)x(n-1) \quad \cdots \quad g_N(n)x(n-N+1)]^T. \quad (4.11)$$

Assim, na Tabela 4.2, consideramos que N multiplicações são requeridas para obter $\mathbf{G}(n)\mathbf{x}(n)$.

TABELA 4.2
COMPARAÇÃO DE COMPLEXIDADE COMPUTACIONAL E
MEMÓRIA REQUERIDA

Algoritmos	Multiplicações	Adições	Divisões	Comparações	Memória
NLMS	$3N + 1$	$3N$	1	0	N
PNLMS padrão	$4N + 3$	$4N - 1$	1	$2N$	N
IAF-PNLMS	$4N + 3$	$4N$	1	N	$3N$

4.3. Resultados de Simulação

Nesta seção, simulações de Monte Carlo (média de 100 realizações independentes) dos algoritmos NLMS, PNLMS padrão, IPNLMS, MPNLMS e IAF-PNLMS são realizadas, considerando um problema de identificação de sistemas. Os objetivos dessas simulações são:

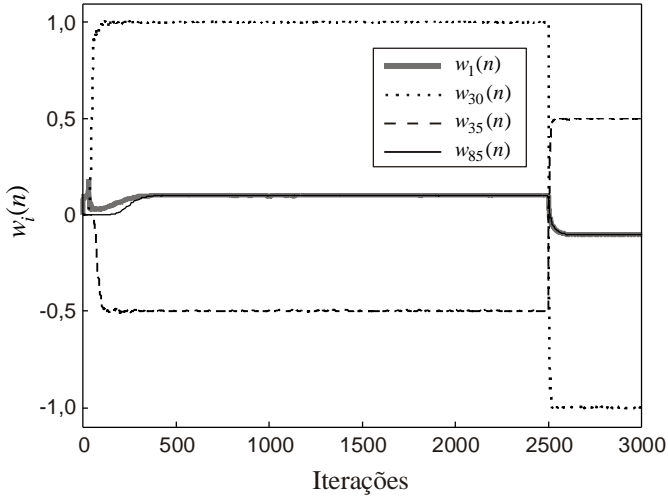
- i) Verificar as propriedades do algoritmo IAF-PNLMS.
- ii) Comparar os algoritmos NLMS, PNLMS padrão, IPNLMS e MPNLMS com o algoritmo IAF-PNLMS em termos de velocidade de convergência, resposta a perturbações na planta e habilidade de rastreamento.
- iii) Avaliar as distribuições de ganhos dos algoritmos PNLMS padrão, IPNLMS, MPNLMS e IAF-PNLMS.
- iv) Verificar o desempenho do algoritmo IAF-PNLMS para plantas esparsas com muitos coeficientes inativos de pequena magnitude.
- v) Avaliar o desempenho do algoritmo proposto considerando como planta uma resposta de caminho de eco típica.

Para tal, cinco exemplos são apresentados. Nos primeiros quatro, usamos o mesmo cenário de simulação especificado no Capítulo 2 (veja Seção 2.3), consistindo de um sinal de entrada correlacionado $x(n)$ (com $\chi = 10$) e de uma planta \mathbf{w}^o com seus coeficientes ativos iguais a $\{0,1, 1,0, -0,5, 0,1\}$, localizados nas posições $\{1, 30, 35, 85\}$, respectivamente. Para o último exemplo, um sinal de entrada correlacionado com $\chi = 191$ e uma resposta de caminho de eco típica são usados.

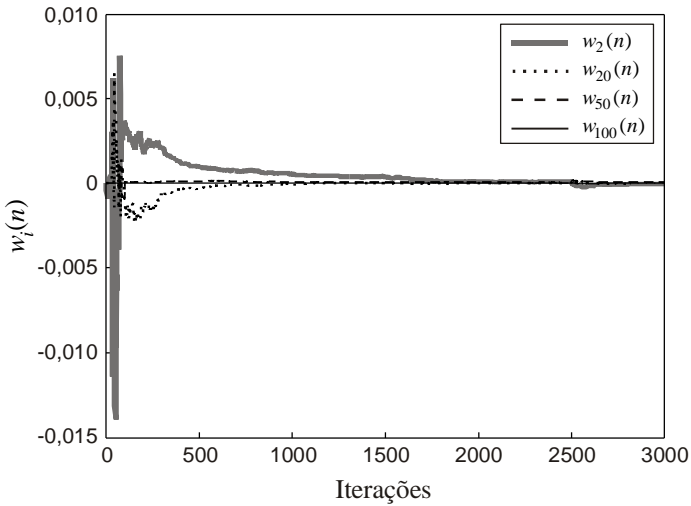
4.3.1. Exemplo 4.1

Este exemplo ilustra o comportamento das seguintes variáveis do algoritmo IAF-PNLMS: coeficientes do filtro adaptativo, fatores de ativação e funções de proporcionalidade. Além disso, a velocidade de convergência e a resposta a perturbações na planta dos algoritmos NLMS, PNLMS padrão, IPNLMS e IAF-PNLMS são comparadas. Para tal, considere que uma perturbação na planta ocorre no instante $n = 2500$, quando \mathbf{w}^o é trocado para $-\mathbf{w}^o$.

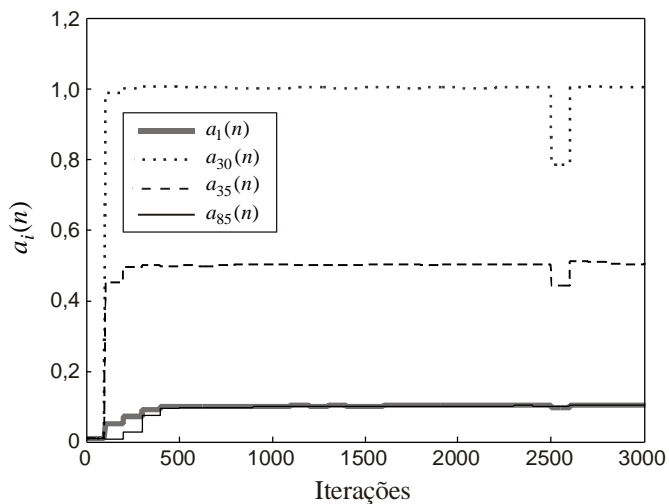
A Figura 4.1 mostra o comportamento das seguintes variáveis do algoritmo IAF-PNLMS: $w_i(n)$, $a_i(n)$ e $\phi_i(n)$. Note que os fatores de ativação tendem para as magnitudes dos correspondentes coeficientes do filtro adaptativo, isto é, $a_i(mN) \rightarrow |w_i(mN)|$. Esse comportamento está em concordância com as propriedades P1 e P2 [veja (4.7) e (4.9)]. Também, observe que $\phi_i(mN) \rightarrow |w_i(mN)|$ para os coeficientes ativos e inativos, de acordo com P1 e P2 [veja (4.8) e (4.10)], significando que $g_i(mN)$ tende a ser proporcional a $|w_i(mN)|$ como esperado para o algoritmo IAF-PNLMS.



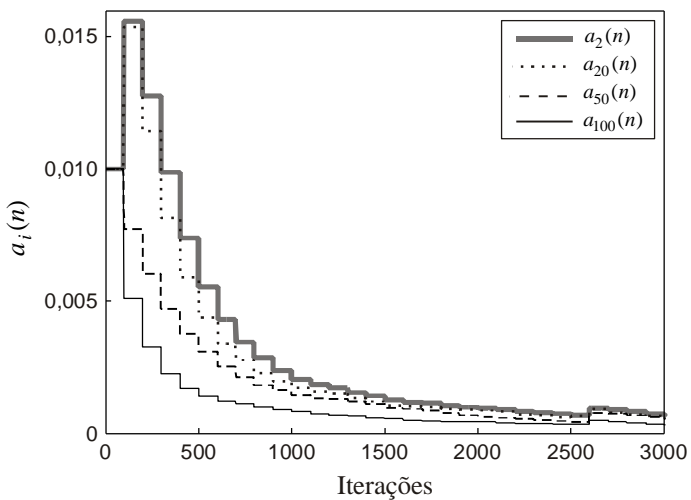
(a)



(b)



(c)



(d)

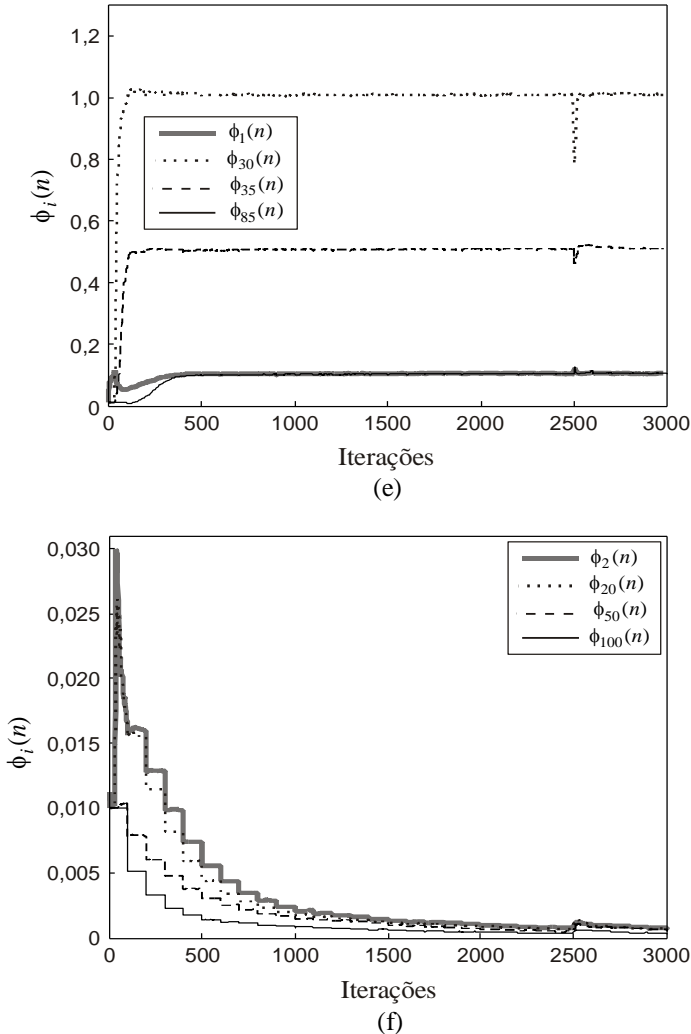


Figura 4.1. Exemplo 4.1. Curvas das variáveis do algoritmo IAF-PNLMS considerando uma perturbação na planta em $n = 2500$ (planta \mathbf{w}^o é mudada para $-\mathbf{w}^o$) com $\mu = 0,5$ e $a_i(0) = 10^{-4}$. (a) Coeficientes ativos. (b) Coeficientes inativos. (c) Fatores de ativação associados aos coeficientes ativos. (d) Fatores de ativação associados aos coeficientes inativos. (e) Funções de proporcionalidade associadas aos coeficientes ativos. (f) Funções de proporcionalidade associadas aos coeficientes inativos.

A Figura 4.2 ilustra as curvas de desalinhamento normalizado dos algoritmos NLMS, PNLMS padrão (com $\rho=0,05$ e $\delta=0,01$), IPNLMS (com $\alpha=0$) e IAF-PNLMS [com $a_i(0)=10^{-4}$]. Comparando as curvas na Figura 4.2, observa-se que o algoritmo IAF-PNLMS atinge a maior velocidade de convergência dentre os algoritmos avaliados.

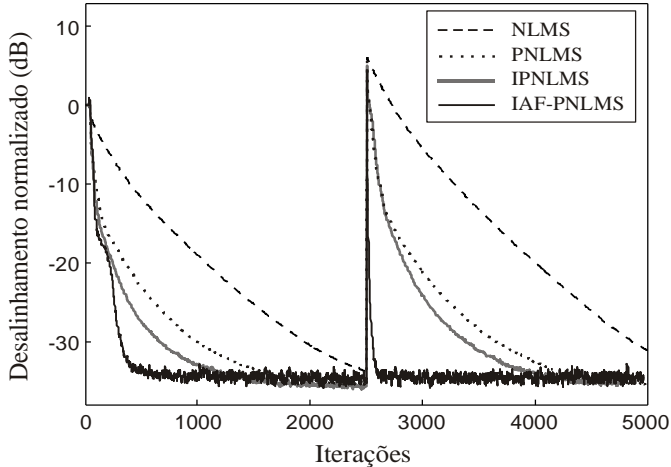


Figura 4.2. Exemplo 4.1. Curvas de desalinhamento normalizado considerando a inversão do sinal dos coeficientes da planta no instante $n=2500$ com $\mu=0,5$, $\rho=0,05$, $\delta=0,01$, $a_i(0)=10^{-4}$ e $\alpha=0$.

TABELA 4.3

EXEMPLO 4.1. NÚMERO NECESSÁRIO DE ITERAÇÕES PARA O DESALINHAMENTO NORMALIZADO $\kappa(n)$ ATINGIR -33dB APÓS A PERTURBAÇÃO NA ITERAÇÃO $n=2500$

Algoritmos	Valores dos Parâmetros		Número de Iterações
PNLMS padrão	ρ	0,05	1570
		0,01	1290
IPNLMS	α	-0,5	1713
		0,0	1228
		0,5	925
IAF-PNLMS	$a_i(0)$	10^{-4}	81
		10^{-2}	103

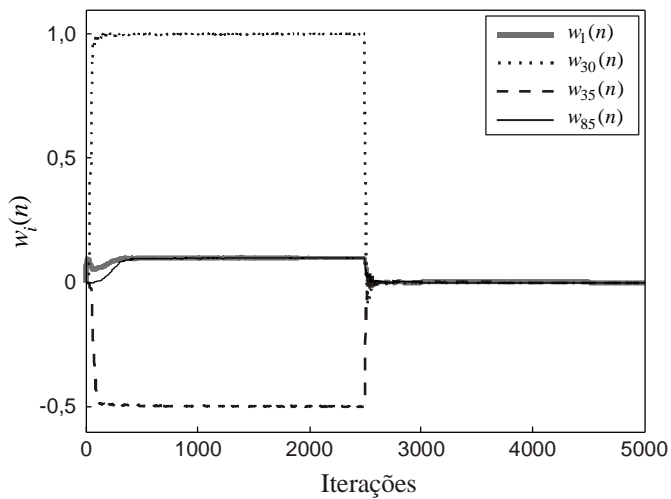
A Tabela 4.3 mostra a velocidade de convergência em termos do número necessário de iterações para o desalinhamento normalizado $\kappa(n)$ atingir -33dB após a ocorrência da perturbação na iteração $n = 2500$, considerando diferentes valores de parâmetros para os algoritmos PNLMS padrão, IPNLMS e IAF-PNLMS. Nessa tabela, usamos $\rho = \{0,05, 0,01\}$ para o algoritmo PNLMS padrão, $\alpha = \{-0,5, 0,0, 0,5\}$ para o algoritmo IPNLMS e $a_i(0) = \{10^{-4}, 10^{-2}\}$ para o algoritmo IAF-PNLMS. Note que para ambos os valores iniciais de $a_i(n)$, o algoritmo proposto, em comparação aos outros algoritmos avaliados, exige um menor número de iterações para atingir os objetivos definidos.

4.3.2. Exemplo 4.2

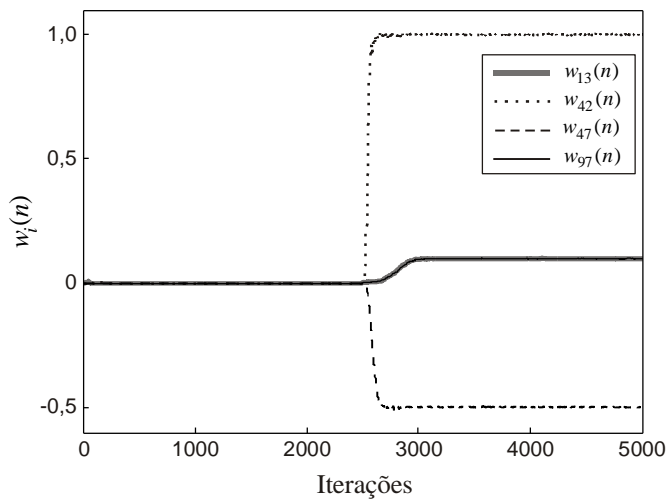
Neste exemplo, novamente é mostrado o comportamento das variáveis do algoritmo IAF-PNLMS e, além disso, os algoritmos NLMS, PNLMS padrão, IPNLMS e IAF-PNLMS são comparados, em termos da velocidade de convergência e da resposta a perturbações na planta. Os mesmos valores dos parâmetros do Exemplo 1 são considerados. Agora, no instante $n = 2500$, o vetor de coeficientes da planta \mathbf{w}^o é deslocado de 12 amostras para a direita, alterando a posição de todos os coeficientes ativos. Dessa maneira, os valores dos coeficientes ativos da planta iguais a $\{0,1, 1,0, -0,5, 0,1\}$, localizados nas posições $\{1, 30, 35, 85\}$, são movidos para as posições $\{13, 42, 47, 97\}$, após $n = 2500$.

A Figura 4.3 ilustra a habilidade do algoritmo IAF-PNLMS no rastreamento dos coeficientes ativos antes e após uma perturbação. Da mesma forma que no Exemplo 1, mostra-se que o comportamento do algoritmo IAF-PNLMS está em concordância com as propriedades P1 e P2.

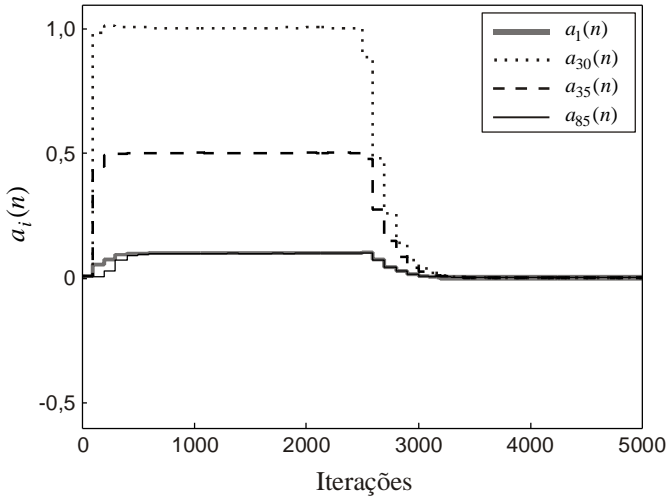
A Figura 4.4 mostra as curvas de desalinhamento normalizado dos algoritmos NLMS, PNLMS padrão (com $\rho = 0,05$ e $\delta = 0,01$), IPNLMS (com $\alpha = 0$) e IAF-PNLMS [com $a_i(0) = 10^{-4}$]. Note, dessa figura, que o algoritmo IAF-PNLMS novamente atinge a maior velocidade de convergência para as condições consideradas.



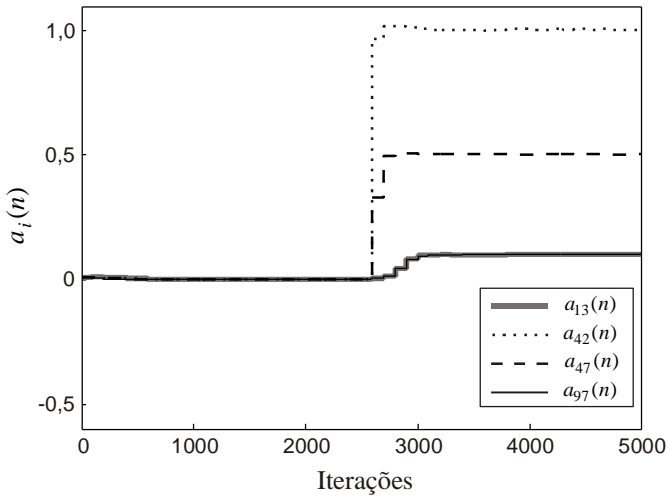
(a)



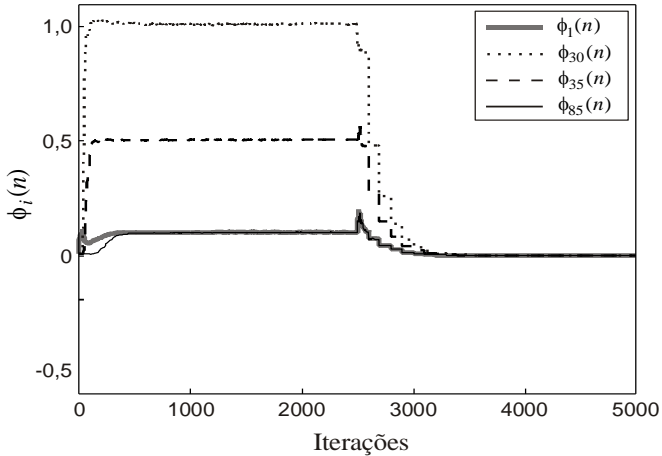
(b)



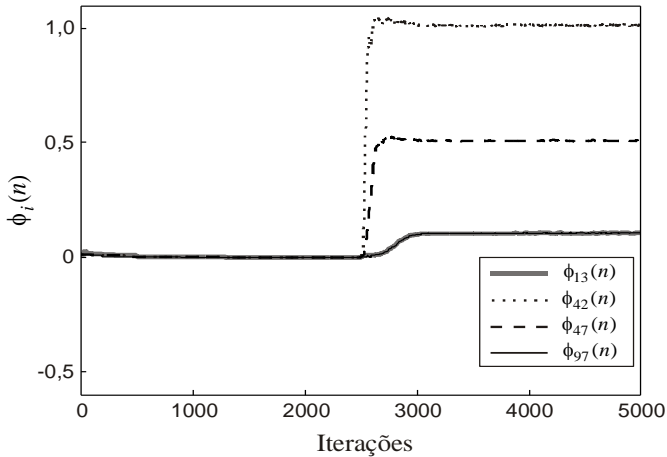
(c)



(d)



(e)



(f)

Figura 4.3. Exemplo 4.2. Curvas das variáveis do algoritmo IAF-PNLMS com um deslocamento nos coeficientes da planta no instante $n = 2500$, usando $\mu = 0,5$ e $a_i(0) = 10^{-4}$. (a) Coeficientes ativos antes da perturbação. (b) Coeficientes ativos após a perturbação. (c) Fatores de ativação associados aos coeficientes ativos antes da perturbação. (d) Fatores de ativação associados aos coeficientes ativos após a perturbação. (e) Funções de proporcionalidade associadas aos coeficientes ativos antes da perturbação. (f) Funções de proporcionalidade associadas aos coeficientes ativos após a perturbação.

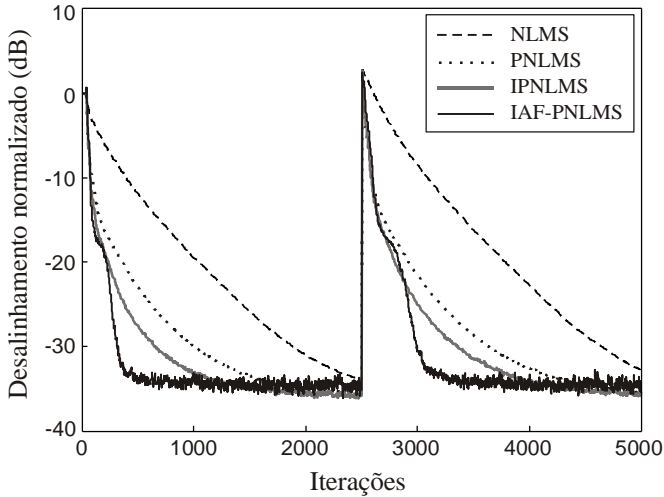


Figura 4.4. Exemplo 4.2. Curvas de desalinhamento normalizado considerando um deslocamento nos coeficientes da planta no instante $n = 2500$ com $\mu = 0,5$, $\rho = 0,05$, $\delta = 0,01$, $a_i(0) = 10^{-4}$ e $\alpha = 0$.

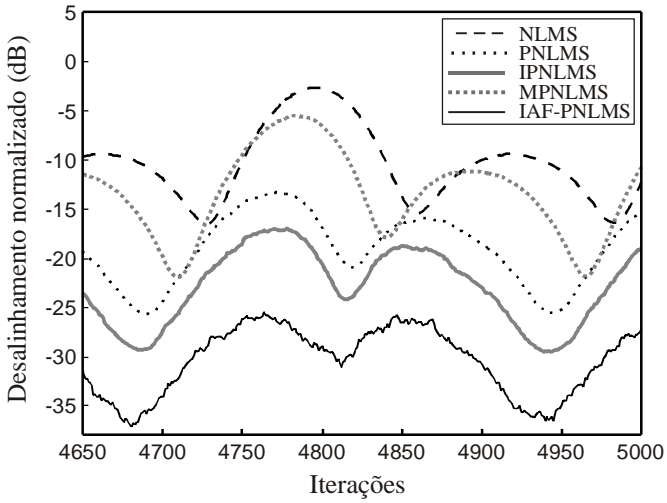
4.3.3. Exemplo 4.3

Neste exemplo, a habilidade de rastreamento dos algoritmos NLMS, PNLMS padrão, IPNLMS, MPNLMS e IAF-PNLMS são comparadas. Aqui, a simulação parte com a planta $\mathbf{w}^o = [w_1^o \ w_2^o \ \dots \ w_{100}^o]^T$, mas agora dois dos seus coeficientes são modulados [7], segundo a expressão:

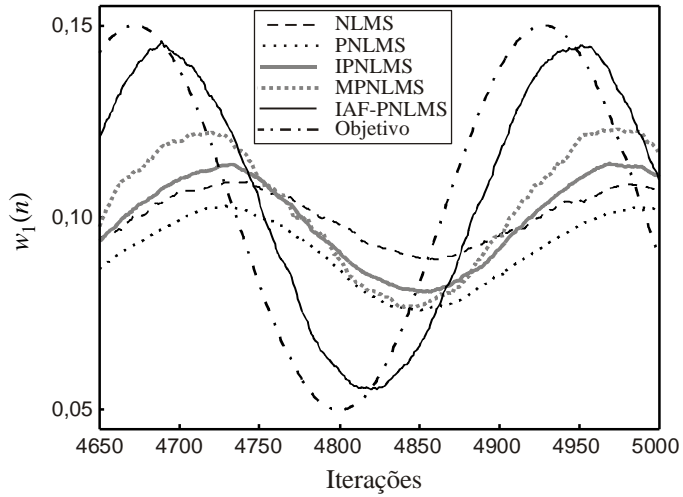
$$q_i(n) = \begin{cases} w_i^o \left[1 + 0,5 \operatorname{sen} \left(\frac{2\pi n}{256} \right) \right], & i = 1, 30 \\ w_i^o, & i = 25, 85 \\ 0, & \text{caso contrário} \end{cases} \quad (4.12)$$

onde $q_i(n)$ é o i -ésimo coeficiente da planta (variante no tempo) a ser rastreado.

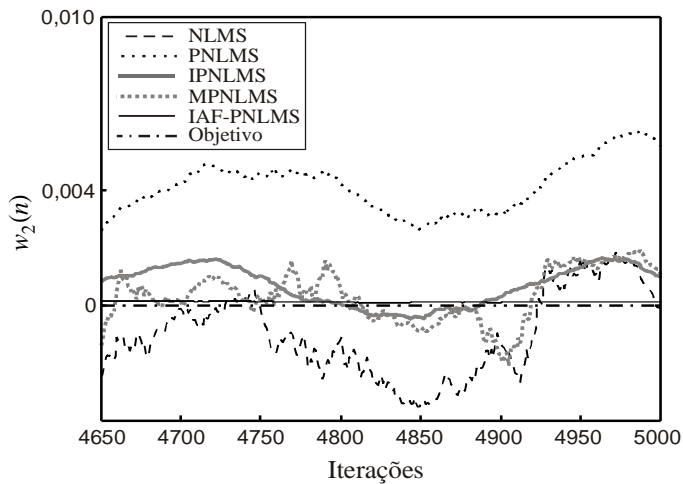
A Figura 4.5 ilustra a habilidade de rastreamento em regime estacionário dos algoritmos NLMS, PNLMS padrão (com $\rho = 0,05$ e $\delta = 0,01$), IPNLMS (com $\alpha = 0$), MPNLMS (com $\rho = 0,05$ e $\delta = 0,01$) e IAF-PNLMS [com $a_i(0) = 0,01$]. No algoritmo MPNLMS, $\varepsilon = 0,001$ (um parâmetro desse algoritmo) é usado [9], [10]. Da Figura 4.5(a), observa-se que o algoritmo IAF-PNLMS atinge o menor desalinhamento, superando os outros algoritmos avaliados. Nas Figuras 4.5(b), 4.5(c) e 4.5(d), verifica-se que os coeficientes ativos estimados pelo algoritmo IAF-PNLMS são os que possuem o melhor rastreamento dos correspondentes coeficientes da planta $q_1(n)$, $q_2(n)$ e $q_{30}(n)$. Ou seja, os coeficientes $w_1(n)$ e $w_{30}(n)$ rastreiam, respectivamente, os coeficientes ativos variantes no tempo $q_1(n)$ e $q_{30}(n)$; e $w_2(n) \cong q_2(n) = 0$. Também, note na Figura 4.5(c) que o coeficiente inativo $w_2(n)$ do algoritmo IAF-PNLMS não sofre o efeito das flutuações dos coeficientes $w_1(n)$ e $w_{30}(n)$ que é observado nos outros algoritmos. Tal comportamento é devido ao fato de que os fatores de ativação do algoritmo IAF-PNLMS são individuais, dependendo somente da magnitude do correspondente coeficiente (no algoritmo IAF-PNLMS não há acoplamento entre coeficientes) em contraste ao fator de ativação $a(n)$ dos algoritmos PNLMS padrão e MPNLMS, os quais dependem de todos os elementos do vetor de coeficientes do filtro adaptativo.



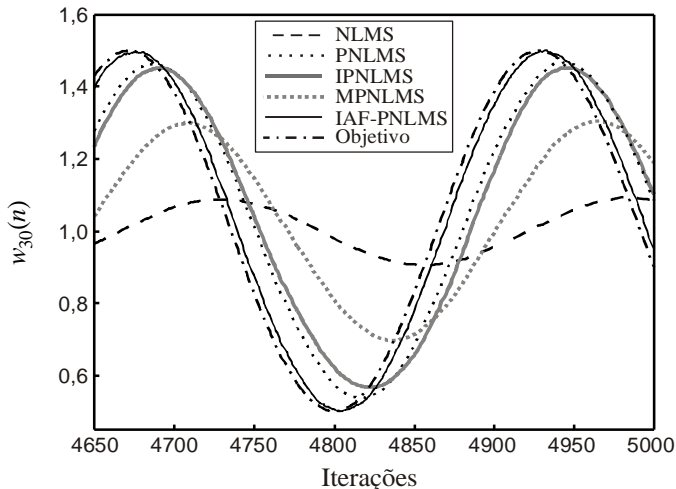
(a)



(b)



(c)



(d)

Figura 4.5. Exemplo 4.3. Comportamento em regime estacionário do desalinamento e dos coeficientes do filtro adaptativo com coeficientes da planta variantes no tempo, considerando $\mu = 0,5$, $\rho = 0,05$, $\delta = 0,01$, $\varepsilon = 0,001$, $a_i(0) = 0,01$ e $\alpha = 0$. (a) Curvas de desalinamento normalizado. (b) Curvas dos coeficientes adaptativos $w_1(n)$. (c) $w_2(n)$. (d) $w_{30}(n)$.

4.3.4. Exemplo 4.4

Neste exemplo, é considerada uma planta esparsa com muitos coeficientes inativos de pequena magnitude. Para esse fim, adiciona-se uma seqüência gaussiana branca $s(i)$ com variância σ_s^2 aos coeficientes da planta \mathbf{w}^0 , formando uma planta modificada \mathbf{h} . Então, as distribuições de ganhos dos algoritmos PNLMS padrão, IPNLMS, MPNLMS e IAF-PNLMS são avaliadas para 5000 iterações. Além disso, a resposta a uma perturbação na planta e a velocidade de convergência dos algoritmos MPNLMS e IAF-PNLMS são comparadas, considerando que o vetor de coeficientes da planta \mathbf{h} é mudado para $-\mathbf{h}$ em $n = 2500$.

Para avaliar a distribuição de ganhos totais do algoritmo IAF-PNLMS, define-se um fator de ativação ideal $a_i^{\text{ideal}}(n)$ como

$$a_i^{\text{ideal}}(n) = |h_i| \tag{4.13}$$

onde $|h_i|$ é a magnitude do i -ésimo coeficiente da planta. Essa definição é razoável, visto que (assumindo a convergência do algoritmo) $a_i(\infty) \rightarrow |w_i(\infty)|$ de acordo com as propriedades P1 e P2 (veja Figuras 4.1 e 4.3).

A Tabela 4.4 apresenta as distribuições de ganhos totais dos algoritmos PNLMS padrão, IPNLMS (com $\alpha = 0$), MPNLMS (com $\varepsilon = 0,001$) e IAF-PNLMS para 5000 iterações. Nessa tabela, são utilizados $\sigma_s^2 = 10^{-8}$, $\rho = 0,05$ e $\delta = 0,01$. O parâmetro $\mu = 0,3$ é usado para o algoritmo IAF-PNLMS, enquanto $\mu = 0,5$ é considerado para os outros algoritmos. Aqui, para realizar uma comparação justa, são selecionados valores de μ que levam os algoritmos avaliados ao mesmo erro quadrático médio (EQM) em regime estacionário. Note que as distribuições de ganhos obtidas para os dois casos do algoritmo IAF-PNLMS considerados são numericamente muito similares, sugerindo que a nova regra (4.6) usada para obter $a_i(n)$ é razoável. Note também que o algoritmo proposto transfere ganhos dos coeficientes inativos para os ativos. Por outro lado, o algoritmo MPNLMS modifica a distribuição de ganhos dos coeficientes ativos, mantendo inalterados os ganhos atribuídos aos coeficientes inativos. Esse algoritmo aplica maior ganho aos coeficientes ativos de magnitude pequena em comparação ao algoritmo PNLMS padrão. As distribuições de ganhos dos algoritmos MPNLMS e IAF-PNLMS levam a um aumento da velocidade de convergência. Note também que, em princípio, é possível combinar a idéia central dos algoritmos MPNLMS e IAF-PNLMS (redistribuição de ganhos dos coeficientes ativos através da lei μ e transferência de ganhos dos coeficientes inativos para os ativos, respectivamente) para aumentar ainda mais a velocidade de convergência do algoritmo resultante. Aqui, é importante ressaltar que o uso de $a_i^{\text{ideal}}(n)$ não é uma solução prática, visto que em geral a planta não é conhecida *a priori*. O fator de ativação ideal é considerado somente para propósito de comparação.

TABELA 4.4

EXEMPLO 4.4. DISTRIBUIÇÕES DE GANHOS TOTAIS DOS ALGORITMOS PNLMS PADRÃO, IPNLMS, MPNLMS E IAF-PNLMS PARA 5000 ITERAÇÕES

Algoritmos	θ_1	θ_{30}	θ_{35}	θ_{85}	$\theta_{med}^{inativo}$
PNLMS padrão	7770	76220	36890	6820	3880
IPNLMS	15960	129520	64980	14190	2860
MPNLMS	28010	41550	37290	27330	3810
IAF-PNLMS, $a_i(0) = 10^{-4}$	31620	287120	142840	27480	110
IAF-PNLMS, $a_i^{ideal}(n)$	29690	291710	146310	29620	30

A Figura 4.6 ilustra as curvas de erro quadrático médio (EQM) dos algoritmos MPNLMS (com $\rho = 0,05$, $\delta = 0,01$ e $\varepsilon = 0,001$) e IAF-PNLMS [com $a_i(0) = 10^{-4}$], comparando seus desempenhos. Aqui, o desempenho do algoritmo IAF-PNLMS é verificado para uma planta com muitos coeficientes inativos de pequena magnitude, considerando dois valores diferentes de σ_s^2 $\{10^{-8}, 5 \times 10^{-6}\}$. Note que os desempenhos dos algoritmos avaliados são muito similares antes da perturbação; contudo, após a perturbação, o algoritmo IAF-PNLMS supera o algoritmo MPNLMS. Para $\sigma_s^2 = 5 \times 10^{-6}$, observa-se da Figura 4.6(b) que, antes da perturbação, ocorre uma pequena redução na velocidade de convergência do algoritmo IAF-PNLMS. Esse comportamento do algoritmo IAF-PNLMS é esperado, visto que um aumento em σ_s^2 implica uma redução do grau de esparsidade da planta.

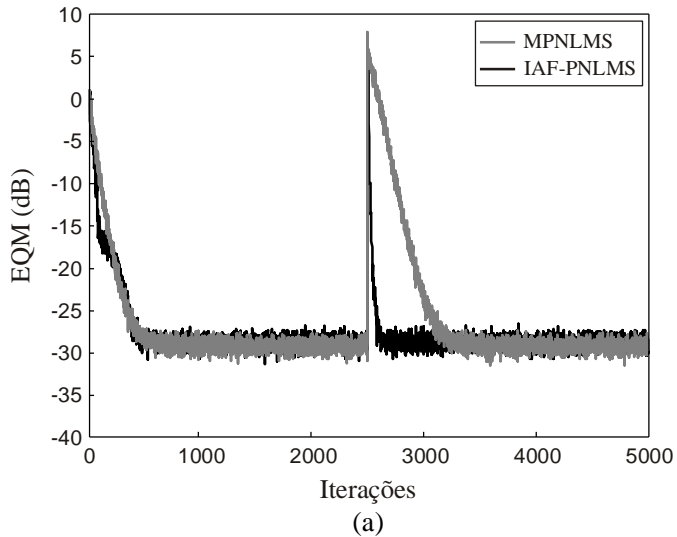
4.3.5. Exemplo 4.5

Neste exemplo, compara-se a habilidade de rastreamento dos algoritmos MPNLMS e IAF-PNLMS sob uma condição de planta mais realista. Para tal, considera-se uma resposta ao impulso esparsa \mathbf{r} com $N = 512$ coeficientes, exibindo as seguintes características:

- i) Uma região ativa em consequência da resposta ao impulso de

uma híbrida (modelo #1 do caminho de eco para teste de voz de canceladores de eco definido nas Recomendações ITU-T G.168 [22]), representada por 64 coeficientes centrais.

- ii) Duas regiões inativas em virtude do *bulk delay* (veja nota de rodapé no Capítulo 1, Seção 1.1.1) e da cauda final, correspondendo a 448 coeficientes [11], [20], [23], [24].



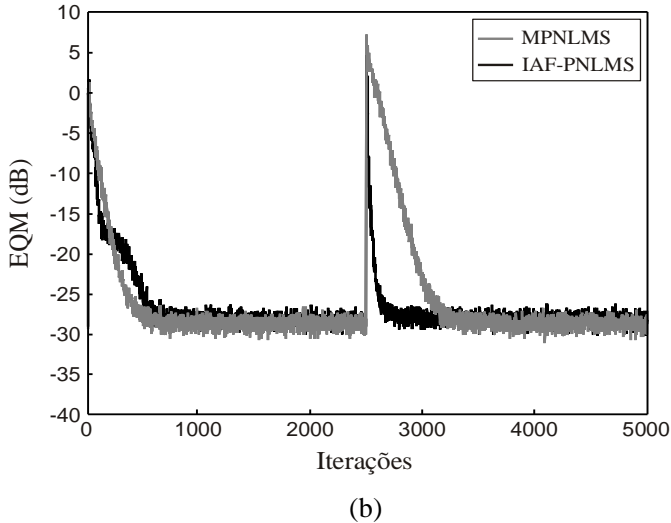


Figura 4.6. Exemplo 4.4. Curvas de aprendizagem (EQM) usando $\rho=0,05$, $\delta=0,01$, $\varepsilon=0,001$ e $a_i(0)=10^{-4}$. (a) Planta \mathbf{h} [com $\sigma_s^2=10^{-8}$ e $S_D(\mathbf{h})=0,9428$] considerando $\mu=0,5$ para o algoritmo MPNLMS e $\mu=0,30$ para o algoritmo IAF-PNLMS. (b) Planta \mathbf{h} [com $\sigma_s^2=5\times 10^{-6}$ e $S_D(\mathbf{h})=0,9286$] com $\mu=0,5$ para o algoritmo MPNLMS e $\mu=0,2$ para o IAF-PNLMS.

Desta maneira, o cenário de simulação para o referido exemplo consiste de uma planta esparsa que subitamente muda a cada 1000 amostras, alternando entre \mathbf{r} e $-\mathbf{r}$. Aqui, o sinal de entrada usado é um processo AR(2) de variância unitária com $\chi=191$, obtido de (2.7) usando $b_1=0,73$, $b_2=-0,80$ e $\sigma_v^2=0,30$.

A Figura 4.7 ilustra as curvas de EQM dos algoritmos MPNLMS (usando $\rho=5/N$, $\delta=0,01$, $\varepsilon=0,001$ e $\mu=0,5$) e IAF-PNLMS [com $a_i(0)=10^{-2}/N$ e $\mu=0,1$]. Aqui, o parâmetro de passo μ é selecionado tal que os algoritmos tenham o mesmo valor de EQM nos instantes em que as perturbações na planta ocorrem. Comparando as

curvas da Figura 4.7, observa-se que o algoritmo IAF-PNLMS apresenta melhor habilidade de rastreamento do que o algoritmo MPNLMS. Tal desempenho do algoritmo proposto é devido ao fato de que os fatores de ativação individuais retêm alguma memória das magnitudes dos coeficientes do filtro adaptativo (veja Propriedades P1 e P2), resultando em um processo de aprendizagem do algoritmo IAF-PNLMS muito mais rápido do que aquele apresentado pelo algoritmo MPNLMS.

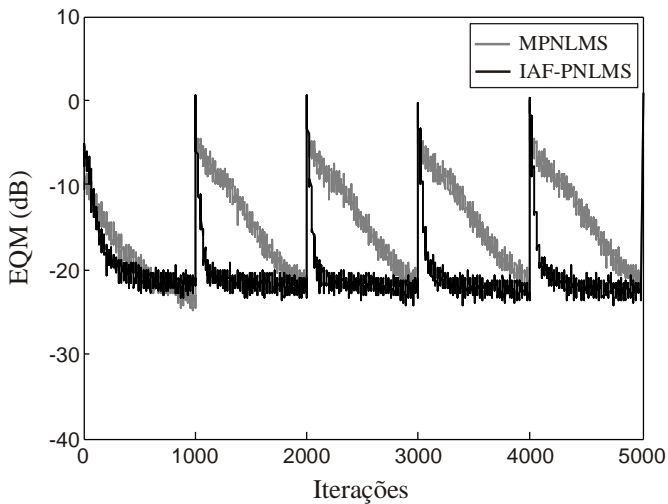


Figura 4.7. Exemplo 4.5. Curvas de aprendizagem (EQM) usando $\rho = 5 / N$, $\delta = 0,01$, $\varepsilon = 0,001$, $a_i(0) = 10^{-2} / N$, $\mu = 0,5$ para o algoritmo MPNLMS e $\mu = 0,1$ para o algoritmo IAF-PNLMS [com grau de esparsidade da planta $S_D(\mathbf{r}) = 0,8970$].

4.4. Conclusões

Neste capítulo, uma versão melhorada do algoritmo PNLMS padrão, denominada IAF-PNLMS, é proposta, usando um fator de ativação individual para cada coeficiente do filtro adaptativo. Cada fator de ativação é recursivamente calculado em termos da magnitude de seu correspondente coeficiente e não mais depende dos parâmetros de proporcionalidade e inicialização considerados no algoritmo PNLMS padrão. O algoritmo IAF-PNLMS apresenta uma melhor distribuição de ganhos do que os algoritmos PNLMS padrão e IPNLMS, conduzindo a uma maior velocidade de convergência, superando os algoritmos PNLMS padrão e IPNLMS para respostas ao impulso com grau de esparsidade elevado. No que concerne ao algoritmo MPNLMS, simulações numéricas mostram que sua distribuição de ganhos é bastante diferente daquela apresentada pelo algoritmo IAF-PNLMS, embora as características de convergência sejam similares quando a planta apresenta alto grau de esparsidade. No que concerne à habilidade de rastreamento bem como às perturbações na planta, o algoritmo IAF-PNLMS supera o algoritmo MPNLMS, considerando respostas ao impulso com grau de esparsidade elevado.

Modelo Estocástico da Evolução dos Coeficientes do Algoritmo IAF-PNLMS

5.1. Introdução

A modelagem estocástica é uma ferramenta importante em processamento adaptativo de sinais. Seu propósito é prever o desempenho do algoritmo sob diferentes condições de operação (evitando o uso de simulações exaustivas para avaliação do algoritmo), bem como obter parâmetros apropriados do algoritmo, tais como o valor do passo de adaptação, o desajuste, o comprimento do filtro, dentre outros [35]-[38]. Além do mais, o modelo pode ser usado para identificar e estudar algum comportamento indesejado do algoritmo; após isso, fazendo modificações adequadas, pode-se contornar tal comportamento, resultando em um algoritmo melhorado [37].

Os algoritmos da classe LMS normalizados proporcionais (PNLMS), tais como o PNLMS padrão, o PNLMS melhorado (IPNLMS), o PNLMS com lei μ (MPNLMS) e o PNLMS com fatores de ativação individuais (IAF-PNLMS), projetados para lidar com respostas ao impulso esparsas, vêm sendo amplamente discutidos na literatura [5]-[14], [33]-[34]. Entretanto, há somente alguns trabalhos de pesquisa que tratam da modelagem desses algoritmos devido à complexidade matemática envolvida. Os modelos existentes são baseados em simplificações severas (ou até mesmo grosseiras); por consequência, os modelos, muitas vezes, não prevêem precisamente o comportamento do algoritmo. Por exemplo, um modelo simplificado do algoritmo PNLMS padrão é apresentado em [6]. Tal modelo é derivado considerando um sinal de entrada branco; além disso, a matriz de ganhos é assumida invariante no tempo, o que não é uma condição de operação real dos algoritmos da classe PNLMS. Esta última condição, também usada em [39], permite derivar expressões que prevêem somente o comportamento do algoritmo em regime estacionário. Um modelo mais realístico é discutido em [40], no qual agora a matriz de

ganhos é assumida variante no tempo, levando a um modelo que prediz satisfatoriamente o comportamento transitório do algoritmo; contudo, tal modelo é derivado para condições restritas, isto é, uma versão simplificada do algoritmo é usada e o sinal de entrada é assumido branco. Portanto, no nosso conhecimento, a modelagem dos algoritmos da classe PNLMS é ainda uma questão aberta. Neste capítulo, focando o algoritmo IAF-PNLMS discutido in [33], [34] e [41], os seguintes objetivos são considerados:

- a) Derivar um modelo estocástico para a evolução dos coeficientes do algoritmo IAF-PNLMS.
- b) Usar o mínimo de simplificações, obtendo um modelo preciso tanto na fase transitória quanto estacionária.
- c) Conceber uma metodologia de análise que possa ser aplicada a outros algoritmos da classe PNLMS.

Baseado nos objetivos mencionados anteriormente, um modelo estocástico para a evolução dos coeficientes do algoritmo IAF-PNLMS é derivado. Comparações entre simulações de Monte Carlo e previsões teóricas obtidas comprovam a precisão do modelo proposto, considerando dados de entrada gaussianos.

5.2. Metodologia de Modelagem

Nesta seção, as hipóteses simplificativas usadas são apresentadas e, em seguida, aplicadas para obter o modelo estocástico da evolução dos coeficientes do algoritmo.

5.2.1. Definição do Problema

O maior obstáculo encontrado na obtenção de um modelo preciso dos algoritmos da classe PNLMS é a matemática complexa envolvida na determinação dos seguintes valores esperados:

- i) $E \left[\frac{\mathbf{x}(n)\mathbf{x}^T(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right].$
- ii) $E \left[\frac{\mathbf{x}(n)d(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right].$
- iii) $E[\mathbf{G}(n)].$

Para calcular os valores esperados i) e ii), uma abordagem similar àquela apresentada em [36] é usada. Com respeito ao valor esperado iii), a literatura contorna a complexidade matemática envolvida usando hipóteses simplificativas severas [39]-[40]. Visando obter aqui um modelo mais preciso, $E[\mathbf{G}(n)]$ é obtido considerando poucas hipóteses simplificativas.

Então, substituindo (1.6) em (1.5), assumindo o valor do parâmetro de regularização muito pequeno, isto é, $\xi \ll \mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)$, e tomando o valor esperado de ambos os lados da expressão resultante, obtém-se

$$\begin{aligned}
 E[\mathbf{w}(n+1)] &= E[\mathbf{w}(n)] \\
 &\quad -\mu E \left[\frac{\mathbf{G}(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] \\
 &\quad +\mu E \left[\frac{\mathbf{G}(n)\mathbf{x}(n)d(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] \\
 &\quad +\mu E \left[\frac{\mathbf{G}(n)\mathbf{x}(n)z(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right].
 \end{aligned} \tag{5.1}$$

Os valores esperados em (5.1) são determinados segundo algumas hipóteses e aproximações estabelecidas a seguir.

5.2.2. Hipóteses e Aproximações Gerais

As hipóteses gerais utilizadas no desenvolvimento do modelo estocástico para a evolução dos coeficientes da classe de algoritmos PNLMS são:

H1) O sinal de entrada é obtido a partir de um processo gaussiano com média zero e o sinal $z(n)$ é um ruído de medição independente e identicamente distribuído (i.i.d.) com média zero, variância σ_z^2 e descorrelacionado de qualquer outro sinal do sistema. Na literatura, essa hipótese é típica no desenvolvimento de modelos estocásticos [29].

H2) Considerando condições de adaptação lenta, $\mathbf{w}(n)$ e $\mathbf{x}(n)$ são estatisticamente independentes [29]. A Hipótese H2, também comum na literatura, permite separar os valores esperados envolvendo as funções $f_1[\mathbf{w}(n)]$ e $f_2[\mathbf{x}(n)]$. Como consequência, $E\{f_1[\mathbf{w}(n)]f_2[\mathbf{x}(n)]\} \cong E\{f_1[\mathbf{w}(n)]\}E\{f_2[\mathbf{x}(n)]\}$.

H3) A matriz $\mathbf{G}(n)$ e o vetor $\mathbf{x}(n)$ são independentes sob condições de adaptação lenta. Essa hipótese é uma consequência direta de H2, visto que a matriz $\mathbf{G}(n)$ é uma função do vetor de coeficientes $\mathbf{w}(n)$.

H4) A matriz $\mathbf{G}(n)$ varia lentamente em relação ao vetor de coeficientes $\mathbf{w}(n)$. Essa hipótese implica que $E[\mathbf{G}(n)\mathbf{w}(n)] \cong E[\mathbf{G}(n)]E[\mathbf{w}(n)]$ (veja o teste da Hipótese H4, Seção 5.4).

H5) Para valores de N grande, o i -ésimo ganho $g_i(n)$ e o fator $f_N(n) = \mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)$ são fracamente correlacionados. Segundo a Hipótese H5, pode ser assumindo que $E[g_i(n)f_N^{-1}(n)] \cong E[g_i(n)]E[f_N^{-1}(n)]$.

Baseado nas Hipóteses H2-H5, as seguintes aproximações gerais são estabelecidas (veja teste de aproximações, Seção 5.4):

Aproximação E1

$$\begin{aligned} \mathbb{E} \left[\frac{\mathbf{G}(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] &\cong \mathbb{E}[\mathbf{G}(n)]\mathbb{E} \\ &\times \mathbb{E} \left[\frac{\mathbf{x}(n)\mathbf{x}^T(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] \mathbb{E}[\mathbf{w}(n)]. \end{aligned} \quad (5.2)$$

Aproximação E2

$$\begin{aligned} \mathbb{E} \left[\frac{\mathbf{G}(n)\mathbf{x}(n)d(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] &\cong \mathbb{E}[\mathbf{G}(n)] \\ &\times \mathbb{E} \left[\frac{\mathbf{x}(n)d(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right]. \end{aligned} \quad (5.3)$$

Agora, usando a Hipótese H1, e as Aproximações E1 e E2, (5.1) pode ser reescrita como

$$\begin{aligned} \mathbb{E}[\mathbf{w}(n+1)] &\cong \mathbb{E}[\mathbf{w}(n)] \\ &\quad -\mu\bar{\mathbf{G}}(n)\mathbf{R}_{\mathbf{G}}(n)\mathbb{E}[\mathbf{w}(n)] + \mu\bar{\mathbf{G}}(n)\mathbf{p}_{\mathbf{G}}(n) \end{aligned} \quad (5.4)$$

onde

$$\bar{\mathbf{G}}(n) \triangleq \text{diag}[\bar{g}_1(n) \quad \bar{g}_2(n) \quad \cdots \quad \bar{g}_N(n)] \quad (5.5)$$

com $\bar{g}_i(n) \triangleq \mathbb{E}[g_i(n)]$. As variáveis $\mathbf{R}_{\mathbf{G}}(n)$ e $\mathbf{p}_{\mathbf{G}}(n)$ representam, respectivamente, uma matriz de autocorrelação normalizada e um vetor de correlação cruzada normalizada entre $d(n)$ e $\mathbf{x}(n)$, dadas pelas seguintes expressões:

$$\mathbf{R}_{\mathbf{G}}(n) = \mathbb{E} \left[\frac{\mathbf{x}(n)\mathbf{x}^T(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right] \quad (5.6)$$

e

$$\mathbf{p}_{\mathbf{G}}(n) = \mathbb{E} \left[\frac{d(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} \right]. \quad (5.7)$$

As hipóteses e aproximações estabelecidas nesta seção são, em geral, aplicáveis na modelagem de todos os algoritmos da classe PNLMS. Na próxima seção, uma metodologia para obtenção de um modelo estocástico para a evolução dos coeficientes do algoritmo IAF-PNLMS é proposta.

5.3. Modelo da Evolução dos Coeficientes do Filtro Adaptativo: Caso do Algoritmo IAF-PNLMS

Nesta seção, o valor esperado associado às características da planta $\bar{\mathbf{G}}(n)$ e os valores esperados associados aos dados de entrada $[\mathbf{R}_{\mathbf{G}}(n)$ e $\mathbf{p}_{\mathbf{G}}(n)$] são determinados.

5.3.1. Determinação de $\bar{\mathbf{G}}(n)$

De (1.8), o i -ésimo elemento da diagonal de $\bar{\mathbf{G}}(n)$ é dado por

$$\bar{g}_i(n) \cong \frac{\mathbb{E}[\phi_i(n)]}{\frac{1}{N} \sum_{j=1}^N \mathbb{E}[\phi_j(n)]}. \quad (5.8)$$

Visando determinar o numerador de (5.8), a seguinte aproximação (específica para o algoritmo IAF-PNLMS) é utilizada (veja o teste de aproximações, Seção 5.4):

Aproximação E3

$$\mathbb{E}[\phi_i(n)] \cong \max \{ \mathbb{E}[a_i(n)], |\mathbb{E}[w_i(n)]| \} \quad (5.9)$$

com

$$E[a_i(n)] = \begin{cases} \frac{1}{2} |E[w_i(n)]| + \frac{1}{2} E[\phi_i(n-1)], & n = mN, m = 1, 2, \dots \\ E[a_i(n-1)], & \text{caso contrário.} \end{cases} \quad (5.10)$$

O valor esperado (5.9) é baseado no fato de que $a_i(n) \rightarrow |w_i(n)|$ quando n aumenta (veja Seção 4.3.1).

5.3.2. Determinação de $\mathbf{R}_G(n)$

Assumindo que os elementos $x_i(n)$ e $x_k(n)$ do vetor de entrada $\mathbf{x}(n)$ sejam variáveis aleatórias conjuntamente gaussianas, um elemento genérico da matriz de autocorrelação normalizada $\mathbf{R}_G(n)$ pode ser expresso como

$$[\mathbf{R}_G(n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} e^{-\frac{\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x} \quad (5.11)$$

onde $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ é a matriz de autocorrelação do vetor de entrada e $x_i(n) = x(n-i+1)$ é o i -ésimo elemento do vetor $\mathbf{x}(n)$. Visando determinar (5.11), é utilizada uma abordagem similar à usada em [35]. Dessa forma, uma função auxiliar $\mathbf{J}(\omega, n)$ é definida, cujos elementos são dados por

$$\begin{aligned}
[\mathbf{J}(\omega, n)]_{i,k} &= \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \\
&\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} e^{-\omega[\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)]} e^{\frac{-\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x}.
\end{aligned} \tag{5.12}$$

Note que para $\omega = 0$, $\mathbf{J}(\omega, n)$ é o valor esperado desejado, isto é,

$$\mathbf{R}_{\mathbf{G}}(n) = \mathbf{J}(\omega, n)|_{\omega=0} = \mathbf{J}(0, n). \tag{5.13}$$

Agora, resolvendo $\mathbf{J}(0, n)$, a matriz de autocorrelação normalizada é

$$\mathbf{R}_{\mathbf{G}}(n) = \mathbf{R}\mathbf{Q}(n)\mathbf{H}(n)\mathbf{Q}^{-1}(n) \tag{5.14}$$

onde $\mathbf{H}(n)$ é uma matriz diagonal (definida no Apêndice III), $\mathbf{Q}(n)$ representa a matriz de autovetores da autodecomposição $\mathbf{G}(n)\mathbf{R} = \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{Q}^{-1}(n)$ (veja Apêndice III) com a matriz diagonal $\mathbf{\Lambda}(n)$ contendo os autovalores $\lambda_i(n)$ dessa autodecomposição.

5.3.3. Determinação de $\mathbf{p}_{\mathbf{G}}(n)$

O vetor de correlação cruzada normalizada $\mathbf{p}_{\mathbf{G}}(n)$ é determinado através da solução da seguinte integral múltipla:

$$\begin{aligned}
[\mathbf{p}_{\mathbf{G}}(n)]_i &= \frac{1}{(2\pi)^{\frac{N+1}{2}} \sqrt{\det(\mathbf{\Gamma})}} \\
&\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{u}_{\mathbf{A}}^T(n)\mathbf{\Gamma}^{-1}\mathbf{u}_{\mathbf{A}}(n)}{2}} d\mathbf{u}_{\mathbf{A}}
\end{aligned} \tag{5.15}$$

com $\mathbf{u}_A(n) = [\mathbf{x}^T(n) \ d(n)]^T$ e

$$\Gamma = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{p}^T & \sigma_d^2 \end{bmatrix} \quad (5.16)$$

onde $\mathbf{p} = E[d(n)\mathbf{x}(n)]$, enquanto $\sigma_d^2 = E[d^2(n)]$ é a variância do sinal desejado. Para obter (5.15), uma função auxiliar $\mathbf{j}(\omega, n)$ é definida, com elementos dados por [35]

$$\begin{aligned} [\mathbf{j}(\omega, n)]_i &= \frac{1}{(2\pi)^{N+1} \sqrt{\det(\Gamma)}} \\ &\times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} e^{-\omega[\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)]} e^{\frac{-\mathbf{u}_A^T(n) \Gamma^{-1} \mathbf{u}_A(n)}{2}} d\mathbf{u}_A. \end{aligned} \quad (5.17)$$

Note que para $\omega=0$, $\mathbf{j}(\omega, n)$ é o vetor de correlação cruzada normalizada, i.e.,

$$\mathbf{p}_G(n) = \mathbf{j}(\omega, n)|_{\omega=0} = \mathbf{j}(0, n). \quad (5.18)$$

Resolvendo (5.17), mostra-se que (veja Apêndice IV)

$$\mathbf{p}_G(n) = \mathbf{T}(n) \mathbf{H}(n) \mathbf{T}^{-1}(n) \mathbf{p} \quad (5.19)$$

onde $\mathbf{T}(n)$ é a matriz de autovetores da decomposição $\mathbf{R}\mathbf{G}(n) = \mathbf{T}(n) \mathbf{\Lambda}(n) \mathbf{T}^{-1}(n)$ e $\mathbf{H}(n)$ é uma matriz diagonal definida no Apêndice III.

5.3.4. Expressão do Modelo

Substituindo (5.14) e (5.19) em (5.4), a expressão final para o vetor de coeficientes médio do algoritmo IAF-PNLMS é dada por

$$\begin{aligned} E[\mathbf{w}(n+1)] &= [\mathbf{I} - \mu \bar{\mathbf{G}}(n) \mathbf{R} \mathbf{Q}(n) \mathbf{H}(n) \mathbf{Q}^{-1}(n)] \\ &\times E[\mathbf{w}(n)] + \mu \bar{\mathbf{G}}(n) \mathbf{T}(n) \mathbf{H}(n) \mathbf{T}^{-1}(n) \mathbf{p} \end{aligned} \quad (5.20)$$

onde \mathbf{I} denota a matriz identidade.

Note que a abordagem desenvolvida nesta seção pode também ser aplicada aos outros algoritmos da classe PNLMS. Para tal, é necessário somente modificar a forma na qual o valor esperado da matriz de ganhos é calculada, levando em conta o algoritmo PNLMS particular a ser modelado. Na próxima seção, a precisão do modelo proposto é avaliada.

5.4. Simulações Numéricas

Nesta seção, visando avaliar a eficácia do modelo proposto, simulações de Monte Carlo (média de 100 realizações independentes) do algoritmo IAF-PNLMS são realizadas considerando um problema de identificação de sistemas. Além disso, a Hipótese H4 bem como as Aproximações E1, E2 e E3 são também verificadas.

O cenário para todas as simulações numéricas consiste de uma resposta ao impulso esparsa $\mathbf{w}^o = [w_1^o \ w_2^o \ \cdots \ w_N^o]^T$ com $N = 32$ coeficientes e grau de esparsidade $S_D(\mathbf{w}^o) = 0,7972$, cujos coeficientes são dados por

$$w_i^o = \begin{cases} 1, & i = 17 \\ 0,8, & i = 20 \\ -0,5, & i = 22 \\ -0,7, & i = 24 \\ 0, & \text{caso contrário.} \end{cases} \quad (5.21)$$

O sinal de entrada é correlacionado, com média zero e variância unitária, obtido a partir de um processo AR(2) expresso por

$$x(n) = 0,73x(n-1) - 0,80x(n-2) + v(n) \quad (5.22)$$

onde $v(n)$ é um ruído branco com variância $\sigma_v^2 = 0,30$ e a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é $\chi = 144$. O ruído de medição $z(n)$ é branco com variância $\sigma_z^2 = 10^{-3}$ (SNR = 30dB). Os parâmetros do algoritmo IAF-PNLMS são $a_i(0) = 10^{-2}$ e $\mu = 5 \times 10^{-3}$.

A Figura 5.1 ilustra as curvas do comportamento médio dos coeficientes localizados nas posições {10, 17, 20, 22, 24}, obtidas usando simulações de Monte Carlo e o modelo proposto (5.20). Dessa figura, a precisão do modelo proposto é verificada para os coeficientes ativos e inativos.

Na Figura 5.2, curvas com os testes da Hipótese H4 e das Aproximações E1 e E2 são apresentadas. A Figura 5.2(a) mostra um casamento muito bom entre o valor esperado $\{E[\mathbf{G}(n)\mathbf{w}(n)]\}_{20}$ e sua aproximação $\{E[\mathbf{G}(n)]E[\mathbf{w}(n)]\}_{20}$, indicando que H4 é uma hipótese razoável. Nas Figuras 5.2(b) e 5.2(c), as curvas correspondentes aos lados esquerdo e direito de (5.2) e (5.3), respectivamente, são apresentadas, verificando a precisão das Aproximações E1 e E2. Note, dessas curvas, o casamento muito bom entre as curvas que representam ambos os lados de (5.2) e (5.3), mostrando a consistência dessas aproximações. Salienta-se aqui que a consistência das Aproximações E1 e E2 sustentam as Hipóteses H2, H3 e H5, visto que essas aproximações são baseadas em tais hipóteses.

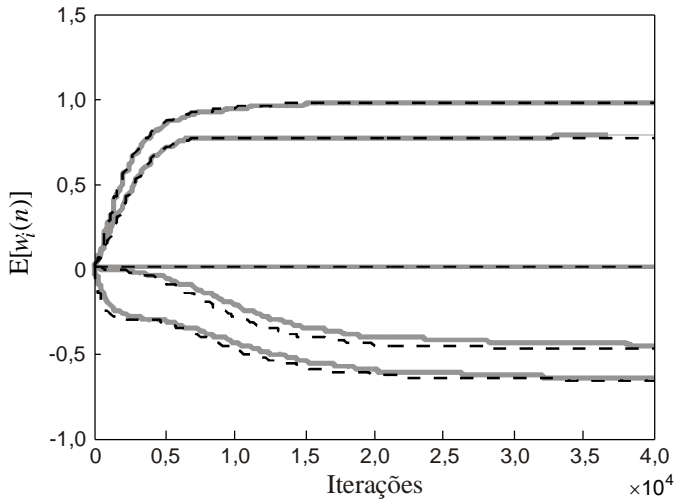
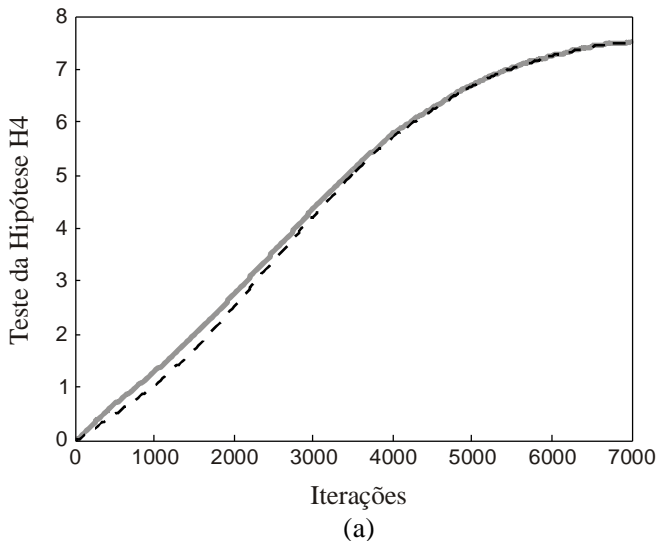
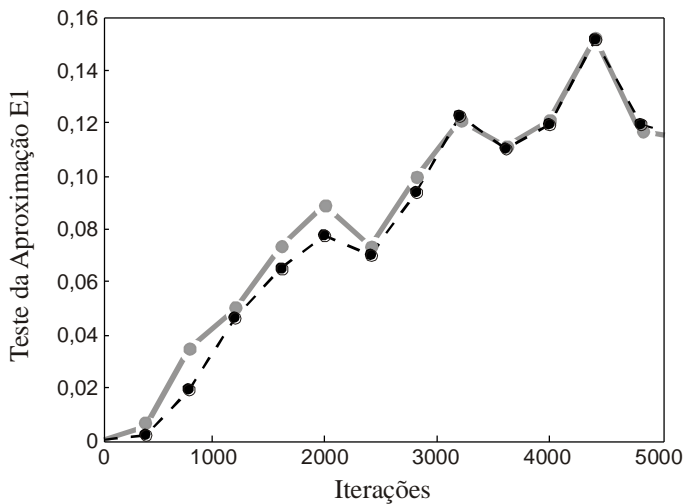
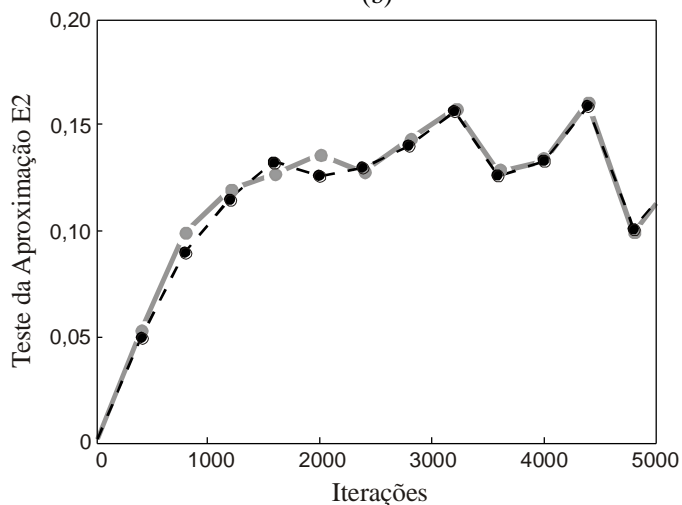


Figura 5.1. Curvas do comportamento médio dos coeficientes, considerando um sinal de entrada correlacionado com $\chi = 144$, obtidas usando simulações de Monte Carlo (linhas tracejadas pretas) e o modelo proposto (5.20) (linhas contínuas cinzas). De cima para baixo: $w_{17}(n)$, $w_{20}(n)$, $w_{10}(n)$, $w_{22}(n)$ e $w_{24}(n)$.





(b)

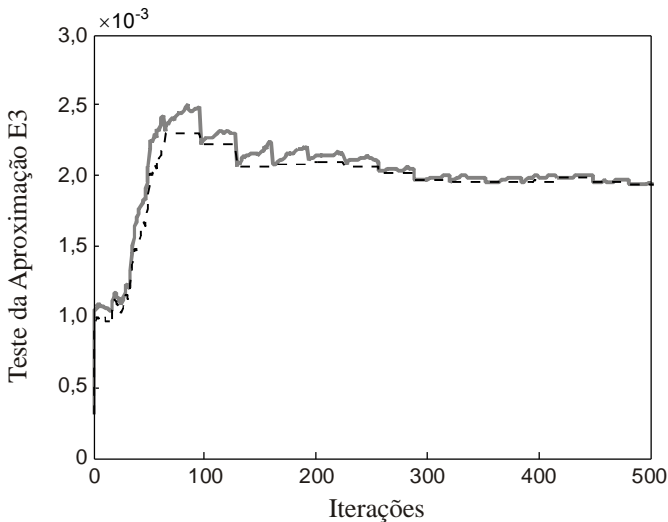


(c)

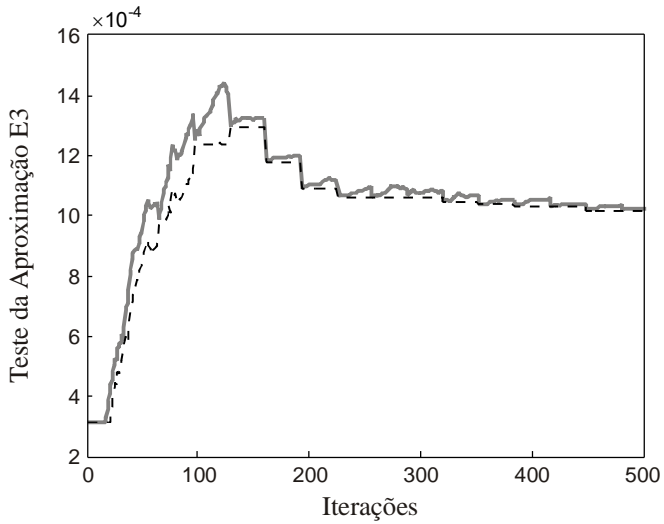
Figura 5.2. Testes da Hipótese H4 e das Aproximações E1 e E2. (a) Evoluções no tempo de $\{E[\mathbf{G}(n)\mathbf{w}(n)]\}_{20}$ (linha sólida cinza) e $\{E[\mathbf{G}(n)]E[\mathbf{w}(n)]\}_{20}$ (linha tracejada preta). (b) Lados esquerdo (linha sólida cinza) e direito (linha tracejada preta) do 17º elemento de (5.2). (c) Lados esquerdo (linha sólida cinza) e direito (linha tracejada preta) do 17º elemento de (5.3).

Os testes da Aproximação E3 são apresentados nas Figuras 5.3(a), 5.3(b), 5.3(c) e 5.3(d). Em tais figuras, a correspondente aproximação é verificada tanto para os casos em que os coeficientes são inativos quanto para aqueles em que os coeficientes são ativos. As Figuras 5.3(a) e 5.3(b) ilustram a evolução no tempo dos valores esperados das funções de proporcionalidade associadas aos coeficientes inativos $w_1(n)$ e $w_{10}(n)$ bem como as suas aproximações. Note, dessas figuras, o bom casamento entre os valores esperados avaliados; também, observe que a partir da iteração $n = 300$ já não há mais distinção entre tais valores esperados. Os testes da Aproximação E3 associados aos coeficientes ativos são ilustrados nas Figuras 5.3(c) e 5.3(d). Nesse caso, observe que, praticamente, não há distinção entre os valores esperados avaliados, mostrando, portanto, um excelente casamento entre eles.

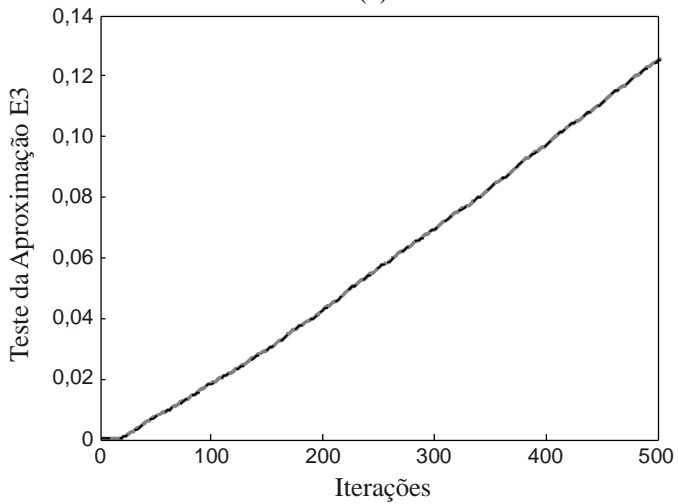
Analisando a Figura 5.3, note a maior precisão do valor esperado $E[\phi_i(n)]$ aproximado quando esse está associado a um coeficiente ativo em relação àquele associado a um coeficiente inativo. Tal comportamento é devido ao fato de que $a_i(n) \rightarrow |w_i(n)|$ mais rapidamente para os coeficientes ativos do que para os inativos (veja Propriedades P1 e P2, Seção 4.2.3).



(a)



(b)



(c)

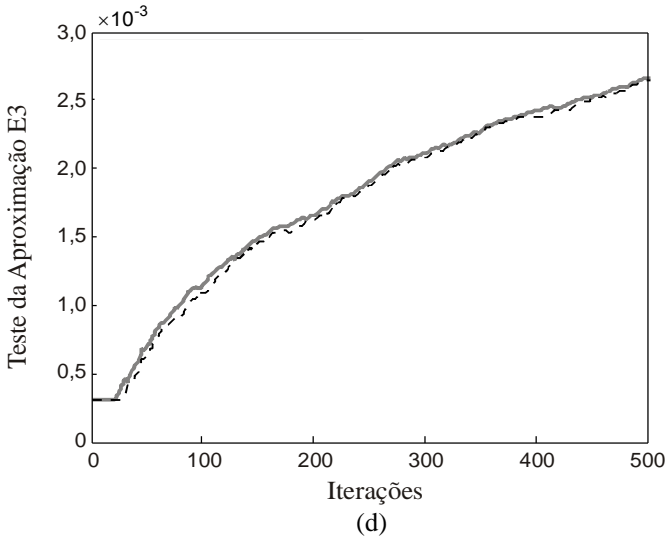


Figura 5.3. Teste da Aproximação E3. (a) Evoluções no tempo de $E[\phi_1(n)]$ (linha sólida cinza) e $\max\{E[a_1(n)], |E[w_1(n)]|\}$ (linha tracejada preta). (b) $E[\phi_{10}(n)]$ (linha sólida cinza) e $\max\{E[a_{10}(n)], |E[w_{10}(n)]|\}$ (linha tracejada preta). (c) $E[\phi_{17}(n)]$ (linha sólida cinza) e $\max\{E[a_{17}(n)], |E[w_{17}(n)]|\}$ (linha tracejada preta). (d) $E[\phi_{22}(n)]$ (linha sólida cinza) e $\max\{E[a_{22}(n)], |E[w_{22}(n)]|\}$ (linha tracejada preta).

5.5. Conclusões

Neste capítulo, uma metodologia para obter um modelo estocástico dos coeficientes do algoritmo IAF-PNLMS foi apresentada. Para tal, a matriz de ganhos é assumida variante no tempo e o sinal de entrada é correlacionado. Por consequência, um modelo de primeira ordem é obtido, predizendo as fases transitória e estacionária do comportamento médio do vetor de coeficientes do algoritmo IAF-PNLMS. Através de simulações numéricas, a precisão do modelo proposto assim como as hipóteses e aproximações principais foram verificadas. A metodologia aqui apresentada pode ser aplicada a outros algoritmos da classe PNLMS.

Modelo Estocástico para o Momento de Segunda Ordem do Vetor de Erro dos Coeficientes do Algoritmo IAF-PNLMS

6.1. Introdução

Neste capítulo, expressões de um modelo estocástico para o momento de segunda ordem, bem como para a curva de aprendizagem do algoritmo IAF-PNLMS são derivadas. Para tal, é considerado aqui um novo procedimento de autodecomposição das matrizes surgidas no cálculo das expectâncias necessárias ao desenvolvimento do modelo. Além do mais, é obtida uma nova expressão descrevendo a evolução dos coeficientes do algoritmo IAF-PNLMS. Resultados de simulações numéricas atestam a eficácia do modelo de segunda ordem proposto.

6.2. Equação Geral do Momento de Segunda Ordem do Vetor de Erro dos Coeficientes

Nesta seção, a equação geral do momento de segunda ordem do vetor de erro dos coeficientes é apresentada. Para tal, o vetor de erro dos coeficientes é definido como

$$\mathbf{v}(n) \triangleq \mathbf{w}(n) - \mathbf{w}^0 \quad (6.1)$$

onde \mathbf{w}^0 é o vetor de coeficientes ótimo. Então, subtraindo-se \mathbf{w}^0 de ambos os lados de (1.5), desprezando ξ , e usando (1.6) e (6.1), pode-se reescrever a equação de atualização do filtro adaptativo em termos do vetor de erro dos coeficientes $\mathbf{v}(n)$ como

$$\mathbf{v}(n+1) = \mathbf{C}(n)\mathbf{v}(n) + \frac{\mu \mathbf{G}(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} z(n) \quad (6.2)$$

com

$$\mathbf{C}(n) = \mathbf{I} - \frac{\mu \mathbf{G}(n) \mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)}. \quad (6.3)$$

A expressão (6.2) é uma equação a diferenças com matriz característica $\mathbf{C}(n)$ [29]. Agora, usando (6.2), uma expressão recursiva para a matriz de covariância do vetor de erro dos coeficientes $\mathbf{K}(n) = E[\mathbf{v}(n) \mathbf{v}^T(n)]$ é obtida. Para esse propósito, visando reduzir a complexidade matemática envolvida, o método de expectância direta de Kushner (*KDAM – Kushner’s direct averaging method*) é invocado [29], [42]-[43]. Dessa forma, usando as Hipóteses H3 e H5, e considerando condição de adaptação lenta, a matriz característica $\mathbf{C}(n)$ pode ser aproximada por

$$\mathbf{C}(n) \cong \mathbf{I} - \mu \bar{\mathbf{G}}(n) \mathbf{R}_G(n). \quad (6.4)$$

Então, calculando $\mathbf{v}(n) \mathbf{v}^T(n)$ usando (6.2) e considerando (6.4) como matriz característica, a matriz de covariância do vetor de erro dos coeficientes, é obtida como

$$\begin{aligned} \mathbf{K}(n+1) = & \left[\mathbf{I} - \mu \bar{\mathbf{G}}(n) \mathbf{R}_G(n) \right] \mathbf{K}(n) \left[\mathbf{I} - \mu \mathbf{R}_G(n) \bar{\mathbf{G}}(n) \right] \\ & + \mu^2 \bar{\mathbf{G}}(n) \mathbf{R}_2(n) \bar{\mathbf{G}}(n) \sigma_z^2 \end{aligned} \quad (6.5)$$

onde

$$\mathbf{R}_2(n) = E \left\{ \frac{\mathbf{x}(n) \mathbf{x}^T(n)}{[\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)]^2} \right\}. \quad (6.6)$$

6.3. Determinação dos Valores Esperados Associados aos Dados de Entrada

A partir de (5.6), (5.7) e (6.6), verifica-se que a determinação dos valores esperados associados aos dados de entrada requer o conhecimento da matriz $\mathbf{G}(n)$; entretanto, tal matriz não está

prontamente disponível para ser utilizada no modelo. Note que a matriz $\mathbf{G}(n)$ é uma variável do algoritmo [veja (1.5)], podendo somente ser obtida através de simulações. Agora, considerando que, em cada iteração n , o valor esperado da matriz de distribuição de ganhos $\bar{\mathbf{G}}(n)$ está prontamente disponível [veja (5.4) e (5.8)-(5.10)], usa-se $\bar{\mathbf{G}}(n)$ no lugar de $\mathbf{G}(n)$ para determinar $\mathbf{R}_G(n)$, $\mathbf{p}_G(n)$ e $\mathbf{R}_2(n)$, em contraste ao procedimento considerado no Capítulo 5. Assim, assumindo que a entrada $\mathbf{x}(n)$ é um vetor aleatório real com distribuição gaussiana multivariável [44], obtém-se $\mathbf{R}_G(n)$, $\mathbf{p}_G(n)$ e $\mathbf{R}_2(n)$ como segue.

6.3.1 Determinação de $\mathbf{R}_G(n)$

Um elemento genérico da matriz de autocorrelação normalizada $\mathbf{R}_G(n)$ pode ser expresso como [45], [46]

$$[\mathbf{R}_G(n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)} e^{-\frac{\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x} \quad (6.7)$$

onde $\bar{\mathbf{G}}(n)$ é a matriz de distribuição de ganhos média obtida a partir do modelo. Visando determinar (6.7), é definida uma função auxiliar $\mathbf{J}_G(\omega, n)$ com seus elementos obtidos por

$$[\mathbf{J}_G(\omega, n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)} e^{-\omega[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]} e^{-\frac{\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x}. \quad (6.8)$$

Note que para $\omega = 0$, $\mathbf{J}_G(0, n) = \mathbf{R}_G(n)$ é o valor esperado desejado. Aqui, para determinar $\mathbf{J}_G(0, n)$, considera-se a autodecomposição alternativa $\bar{\mathbf{G}}(n)\mathbf{R} \cong \mathbf{Q}_A(n)\mathbf{\Lambda}_A(n)\mathbf{P}_A(n)$ derivada no Apêndice V. Dessa forma, o uso de um parâmetro de regularização para estabilizar o cálculo das integrais hiperelípticas de alta ordem não é mais necessário. Assim, reescreve-se (6.8) como [35]

$$[\mathbf{J}_G(\omega, n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{x}^T(n)\mathbf{\Psi}_1^{-1}(\omega, n)\mathbf{x}(n)}{2}} d\mathbf{x} \quad (6.9)$$

com $\mathbf{\Psi}_1(\omega, n) = \mathbf{R}[\mathbf{I} + 2\omega\bar{\mathbf{G}}(n)\mathbf{R}]^{-1}$. Agora, diferenciando (6.9) com respeito a ω , o termo $\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)$ no denominador é eliminado, resultando em [35], [36]

$$\frac{\partial[\mathbf{J}_G(\omega, n)]_{i,k}}{\partial\omega} = \frac{-1}{\sqrt{\det[\mathbf{I} + 2\omega\bar{\mathbf{G}}(n)\mathbf{R}]}} F_{i,k}(\omega, n) \quad (6.10)$$

com

$$F_{i,k}(\omega, n) = \frac{1}{(2\pi)^{N/2} \sqrt{\det[\mathbf{\Psi}_1(\omega, n)]}} \times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} x_i(n)x_k(n) e^{\frac{-\mathbf{x}^T(n)\mathbf{\Psi}_1^{-1}(\omega, n)\mathbf{x}(n)}{2}} d\mathbf{x} \quad (6.11)$$

onde $F_{i,k}(\omega, n)$ denota a correlação cruzada entre os elementos $x_i(n)$ e $x_k(n)$ quando eles são variáveis aleatória conjuntamente gaussianas com matriz de covariância $\mathbf{\Psi}_1(\omega, n)$ [35], [36]. Visto que

$F_{i,k}(\omega, n) = [\Psi_1(\omega, n)]_{i,k}$, integrando (6.10) em relação a ω e usando a autodecomposição $\bar{\mathbf{G}}(n)\mathbf{R} \cong \mathbf{Q}_A(n)\mathbf{\Lambda}_A(n)\mathbf{P}_A(n)$, tem-se

$$\begin{aligned} \mathbf{R}_G(n) &= \mathbf{J}_G(\omega, n) \Big|_{\omega=0} \\ &= \mathbf{R}\mathbf{Q}_A(n)\mathbf{H}_G(\omega, n)\mathbf{P}_A(n) \Big|_{\omega=0} \\ &= \mathbf{R}\mathbf{Q}_A(n)\mathbf{H}_G(n)\mathbf{P}_A(n) \end{aligned} \quad (6.12)$$

onde $\mathbf{Q}_A(n)$ e $\mathbf{P}_A(n)$ são as matrizes de autovetores da autodecomposição $\bar{\mathbf{G}}(n)\mathbf{R} \cong \mathbf{Q}_A(n)\mathbf{\Lambda}_A(n)\mathbf{P}_A(n)$ (veja Apêndice V), $\mathbf{\Lambda}_A(n)$ é uma matriz diagonal contendo os autovalores mais significativos da autodecomposição e $\mathbf{H}_G(n)$, uma matriz diagonal cujos elementos são dados por

$$\begin{aligned} [h_G(0, n)]_i &\cong \frac{-1}{2\lambda_i(n)\sqrt{a_M(n)}} \\ &\times \left\{ B_i(n)\ln[-\omega_i(n)] + \sum_{q=1}^{M/2} A_{i,q}(n)\ln[-\omega'_q(n)] \right\} \end{aligned} \quad (6.13)$$

com

$$B_i(n) = \frac{1}{\prod_{q=1}^{M/2} [\omega_i(n) - \omega'_q(n)]} \quad (6.14)$$

e

$$A_{i,q}(n) = \frac{1}{[\omega'_q(n) - \omega_i(n)] \prod_{\substack{j=1 \\ j \neq q}}^{M/2} [\omega'_q(n) - \omega'_j(n)]} \quad (6.15)$$

onde, sem perda de generalidade, os autovalores $\lambda_i(n)$ são assumidos distintos. As variáveis $a_M(n)$, $\omega_i(n)$ e $\omega'_q(n)$ são obtidas, respectivamente, por

$$a_M(n) = 2^M \prod_{i=1}^M \lambda_i(n) \quad (6.16)$$

$$\omega_i(n) = \frac{-1}{2\lambda_i(n)} \quad (6.17)$$

e

$$\omega'_q(n) = -\sqrt{\omega_{2q-1}(n)\omega_{2q}(n)}, \quad q = 1, 2, \dots, \frac{M}{2} \quad (6.18)$$

com M definida no Apêndice V. Ressalta-se que para determinar (6.12), o procedimento de autodecomposição alternativo de $\bar{\mathbf{G}}(n)\mathbf{R}$ e $\mathbf{R}\bar{\mathbf{G}}(n)$ [i.e., $\lambda_i(n) > 0$] é considerado, dessa forma, garantindo a estabilidade do cálculo das integrais hiperelípticas.

6.3.2 Determinação de $\mathbf{p}_G(n)$

Assumindo que $d(n)$ e $x_i(n)$ são conjuntamente gaussianos, o vetor de correlação cruzada $\mathbf{p}_G(n)$ é determinado resolvendo a seguinte integral múltipla:

$$\begin{aligned} [\mathbf{p}_G(n)]_i &= \frac{1}{(2\pi)^{\frac{N+1}{2}} \sqrt{\det(\Gamma)}} \\ &\times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{u}_A^T(n)\Gamma^{-1}\mathbf{u}_A(n)}{2}} d\mathbf{u}_A \end{aligned} \quad (6.19)$$

com $\mathbf{u}_A(n) = [\mathbf{x}^T(n) \ d(n)]^T$ e

$$\Gamma = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{p}^T & \sigma_d^2 \end{bmatrix} \quad (6.20)$$

onde $\mathbf{p} = E[d(n)\mathbf{x}(n)]$ é o vetor de correlação cruzada entre o sinal desejado $d(n)$ e o vetor de entrada $\mathbf{x}(n)$, e $\sigma_d^2 = E[d^2(n)]$ é a variância do sinal desejado. Para resolver (6.19), define-se a função auxiliar $\mathbf{j}(\omega, n)$ com seus elementos dados por [36], [45]

$$[\mathbf{j}(\omega, n)]_i = \frac{1}{(2\pi)^{\frac{N+1}{2}} \sqrt{\det(\Gamma)}}$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n) \bar{\mathbf{G}}(n) \mathbf{x}(n)} e^{-\omega[\mathbf{x}^T(n) \bar{\mathbf{G}}(n) \mathbf{x}(n)]} e^{\frac{-\mathbf{u}_A^T(n) \Gamma^{-1} \mathbf{u}_A(n)}{2}} d\mathbf{u}_A. \quad (6.21)$$

Note que para $\omega = 0$, $\mathbf{j}(0, n) = \mathbf{p}_G(n)$ é o vetor de correlação cruzada normalizada. Aqui, para determinar $\mathbf{j}(0, n)$, considera-se usar a autodecomposição $\mathbf{R}\bar{\mathbf{G}}(n) \cong \mathbf{T}_A(n)\mathbf{\Lambda}_A(n)\mathbf{U}_A(n)$, introduzida no Apêndice V, no lugar da autodecomposição $\mathbf{R}\mathbf{G}(n) = \mathbf{T}(n)\mathbf{\Lambda}(n)\mathbf{T}^{-1}(n)$ utilizada no Capítulo 5. Agora, a função $\mathbf{j}(\omega, n)$ é reescrita então como [35]

$$[\mathbf{j}(\omega, n)]_i = \frac{1}{(2\pi)^{\frac{N+1}{2}} \sqrt{\det(\mathbf{\Gamma})}} \quad (6.22)$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{u}_A^T(n)\Psi_2^{-1}(\omega, n)\mathbf{u}_A(n)}{2}} d\mathbf{u}_A$$

com

$$\Psi_2^{-1}(\omega, n) = \mathbf{\Gamma}^{-1} + 2\omega \begin{bmatrix} \bar{\mathbf{G}}(n) & 0 \\ 0 & 0 \end{bmatrix}. \quad (6.23)$$

Diferenciando (6.22) com respeito a ω , o termo $\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)$ no denominador é eliminado. Assim, obtém-se

$$\frac{\partial [\mathbf{j}(\omega, n)]_i}{\partial \omega} = \frac{-\sqrt{\det[\Psi_2(\omega, n)]}}{\sqrt{\det(\mathbf{\Gamma})}} S_i(\omega, n) \quad (6.24)$$

onde

$$S_i(\omega, n) = \frac{(2\pi)^{\frac{N+1}{2}}}{\sqrt{\det[\Psi_2(\omega, n)]}} \quad (6.25)$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} d(n)x_i(n) e^{\frac{-\mathbf{u}_A^T(n)\Psi_2^{-1}(\omega, n)\mathbf{u}_A(n)}{2}} d\mathbf{u}_A$$

corresponde à correlação cruzada entre $d(n)$ e $x_i(n)$ quando \mathbf{u}_A é um vetor gaussiano [35], [36]. Agora, de (6.23), a matriz de covariância $\Psi_2(\omega, n)$ é dada por

$$\Psi_2(\omega, n) = \begin{bmatrix} \mathbf{B}(\omega, n)\mathbf{R} & \mathbf{B}(\omega, n)\mathbf{p} \\ \mathbf{p}^T\mathbf{B}^T(\omega, n) & \sigma_d^2 - 2\omega\mathbf{p}^T\bar{\mathbf{G}}(n)\mathbf{B}(\omega, n)\mathbf{p} \end{bmatrix} \quad (6.26)$$

com $\mathbf{B}(\omega, n) = [\mathbf{I} + 2\omega\mathbf{R}\bar{\mathbf{G}}(n)]^{-1}$. Dessa forma, por inspeção, obtém-se a partir de (6.26)

$$S_i(\omega, n) = \{[\mathbf{I} + 2\omega\mathbf{R}\bar{\mathbf{G}}(n)]^{-1}\mathbf{p}\}_i. \quad (6.27)$$

Além do mais, de (6.23), obtém-se

$$\det[\Psi_2(\omega, n)] = \frac{\det(\Gamma)}{\det[\mathbf{I} + 2\omega\mathbf{R}\bar{\mathbf{G}}(n)]}. \quad (6.28)$$

Assim, usando a autodecomposição $\mathbf{R}\bar{\mathbf{G}}(n) \cong \mathbf{T}_A(n)\Lambda_A(n)\mathbf{U}_A(n)$, substituindo (6.27) e (6.28) em (6.24) e integrando em relação a ω , obtém-se

$$\begin{aligned} \mathbf{p}_G(n) &= \mathbf{j}(\omega, n)|_{\omega=0} \\ &= \mathbf{T}_A(n)\mathbf{H}_G(\omega, n)\mathbf{U}_A(n)\mathbf{p}|_{\omega=0} \\ &= \mathbf{T}_A(n)\mathbf{H}_G(n)\mathbf{U}_A(n)\mathbf{p}. \end{aligned} \quad (6.29)$$

onde $\mathbf{T}_A(n)$ e $\mathbf{U}_A(n)$ são as matrizes de autovetores da autodecomposição $\mathbf{R}\bar{\mathbf{G}}(n) \cong \mathbf{T}_A(n)\Lambda_A(n)\mathbf{U}_A(n)$.

6.3.3 Determinação de $\mathbf{R}_2(n)$

Os elementos da matriz $\mathbf{R}_2(n)$ são dados por

$$[\mathbf{R}_2(n)]_{i,k} \cong \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]^2} e^{\frac{-\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x}. \quad (6.30)$$

Esses elementos, definidos em (6.30), podem ser determinados usando uma função matriz auxiliar $\mathbf{J}_2(\omega, n)$, cujos elementos são dados por

$$[\mathbf{J}_2(\omega, n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]^2} e^{-\omega[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]} e^{\frac{-\mathbf{x}^T(n)\mathbf{R}^{-1}\mathbf{x}(n)}{2}} d\mathbf{x}. \quad (6.31)$$

Dessa forma, para $\omega=0$, $\mathbf{J}_2(\omega, n)$ é o valor esperado requerido, isto é, $\mathbf{R}_2(n) = \mathbf{J}_2(\omega, n)|_{\omega=0} = \mathbf{J}_2(0, n)$. Para determinar a matriz $\mathbf{J}_2(\omega, n)$, reescreve-se (6.31) como [35]

$$[\mathbf{J}_2(\omega, n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]^2} e^{\frac{-\mathbf{x}^T(n)\Psi_1^{-1}(\omega, n)\mathbf{x}(n)}{2}} d\mathbf{x}. \quad (6.32)$$

Agora, diferenciando (6.32) duas vezes com respeito a ω , o termo $[\mathbf{x}^T(n)\bar{\mathbf{G}}(n)\mathbf{x}(n)]^2$ no denominador é eliminado, resultando em [35], [36]

$$\frac{\partial^2 [\mathbf{J}_2(\omega, n)]_{i,k}}{\partial^2 \omega} = \frac{1}{\sqrt{\det[\mathbf{I} + 2\omega \bar{\mathbf{G}}(n) \mathbf{R}]}} F_{i,k}(\omega, n). \quad (6.33)$$

Assim, integrando (6.33) duas vezes em relação a ω , usando a decomposição $\bar{\mathbf{G}}(n) \mathbf{R} \cong \mathbf{Q}_A(n) \Lambda_A(n) \mathbf{P}_A(n)$ e fazendo $\omega = 0$, obtém-se

$$\begin{aligned} \mathbf{R}_2(n) &= \mathbf{J}_2(\omega, n) \Big|_{\omega=0} \\ &= \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_2(\omega, n) \mathbf{P}_A(n) \Big|_{\omega=0} \\ &= \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_2(n) \mathbf{P}_A(n). \end{aligned} \quad (6.34)$$

onde $\mathbf{H}_2(n)$ é uma matriz diagonal com elementos dados por

$$\begin{aligned} [h_2(0, n)]_i &\cong \frac{0.5}{\lambda_i(n) \sqrt{a_N(n)}} \\ &\times \left(\begin{aligned} &B_i(n) \{-\omega_i(n) \ln[-\omega_i(n)] + \omega_i(n)\} \\ &+ \sum_{q=1}^{M/2} A_{i,q}(n) \{-\omega'_q(n) \ln[-\omega'_q(n)] + \omega'_q(n)\} \end{aligned} \right). \end{aligned} \quad (6.35)$$

6.4. Expressões do Modelo

Aqui, levando em conta os valores esperados (6.12), (6.29) e (6.34), bem como as autodecomposições $\bar{\mathbf{G}}(n) \mathbf{R} \cong \mathbf{Q}_A(n) \Lambda_A(n) \mathbf{P}_A(n)$ e $\mathbf{R} \bar{\mathbf{G}}(n) \cong \mathbf{T}_A(n) \Lambda_A(n) \mathbf{U}_A(n)$, as expressões finais do modelo estocástico para os momentos de primeira e segunda ordens do vetor de

erro dos coeficientes são determinadas. Dessa forma, como consequência, obtém-se as variáveis determinísticas $E[\mathbf{w}(n)]$ e $E[e^2(n)]$ que caracterizam o comportamento do algoritmo nas fases transitória e estacionária.

6.4.1 Comportamento Médio do Vetor de Coeficientes

Substituindo (6.12) e (6.29) em (5.4), a expressão final para o vetor de coeficientes médio do algoritmo IAF-PNLMS é reescrita como

$$E[\mathbf{w}(n+1)] = [\mathbf{I} - \mu \bar{\mathbf{G}}(n) \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_G(n) \mathbf{P}_A(n)] E[\mathbf{w}(n)] + \mu \bar{\mathbf{G}}(n) \mathbf{T}_A(n) \mathbf{H}_G(n) \mathbf{U}_A(n) \mathbf{p} \quad (6.36)$$

onde \mathbf{I} denota a matriz identidade $N \times N$.

6.4.2 Curva de Aprendizagem

Elevando ao quadrado ambos os lados de (1.6) e calculando o valor esperado da expressão resultante, obtém-se [29]

$$E[e^2(n)] = \sigma_z^2 + \text{tr}[\mathbf{R} \mathbf{K}(n)]. \quad (6.37)$$

Note que o erro quadrático médio (EQM), dado por (6.37), é completamente determinado se a matriz de covariância $\mathbf{K}(n)$ é conhecida.

Para determinar a expressão final da matriz $\mathbf{K}(n)$, substitui-se (6.12) e (6.34) em (6.5), resultando em

$$\begin{aligned}
 \mathbf{K}(n+1) = & \left[\mathbf{I} - \mu \bar{\mathbf{G}}(n) \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_G(n) \mathbf{P}_A(n) \right] \\
 & \times \mathbf{K}(n) \left[\mathbf{I} - \mu \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_G(n) \mathbf{P}_A(n) \bar{\mathbf{G}}(n) \right] \quad (6.38) \\
 & + \mu^2 \bar{\mathbf{G}}(n) \mathbf{R} \mathbf{Q}_A(n) \mathbf{H}_2(n) \mathbf{P}_A(n) \bar{\mathbf{G}}(n) \sigma_z^2.
 \end{aligned}$$

Agora, substitui-se (6.38) em (6.37) visando obter a expressão final da curva de aprendizagem. Assim, obtém-se

$$\begin{aligned}
 E[e^2(n)] = & \sigma_z^2 + \text{tr} \left\{ \mathbf{R} \left[\mathbf{I} - \mu \bar{\mathbf{G}}(n-1) \mathbf{R} \mathbf{Q}_A(n-1) \mathbf{H}_G(n-1) \mathbf{P}_A(n-1) \right] \right. \\
 & \times \mathbf{K}(n-1) \left[\mathbf{I} - \mu \mathbf{R} \mathbf{Q}_A(n-1) \mathbf{H}_G(n-1) \mathbf{P}_A(n-1) \bar{\mathbf{G}}(n-1) \right] \\
 & \left. + \mu^2 \mathbf{R} \bar{\mathbf{G}}(n-1) \mathbf{R} \mathbf{Q}_A(n-1) \mathbf{H}_2(n-1) \mathbf{P}_A(n-1) \bar{\mathbf{G}}(n-1) \sigma_z^2 \right\}. \quad (6.39)
 \end{aligned}$$

Comentário: A abordagem desenvolvida neste capítulo para derivar o modelo estocástico do algoritmo IAF-PNLMS pode também ser aplicada a outros algoritmos da classe PNLMS; para tal, é necessário modificar a maneira na qual o valor esperado da matriz de ganhos é calculado, levando em conta o algoritmo particular a ser modelado.

6.5. Resultados de Simulação

Nesta seção, visando verificar a eficácia do modelo proposto, simulações de Monte Carlo (media de 100 realizações independentes) do algoritmo IAF-PNLMS são realizadas, considerando um problema de identificação de sistemas. Para tal, dois exemplos são apresentados: o primeiro exemplo considera uma planta esparsa com $N = 32$ coeficientes e o segundo, uma planta esparsa com $N = 128$ coeficientes. O sinal de entrada é correlacionado, com média zero e variância unitária, obtido a partir de um processo AR(2), expresso por

$$x(n) = 1,29x(n-1) - 0,75x(n-2) + v(n) \quad (6.40)$$

onde $v(n)$ é um ruído branco com variância $\sigma_v^2 = 0,20$ e o sinal desejado é dado por

$$d(n) = \mathbf{x}^T(n)\mathbf{w}^o. \quad (6.41)$$

Em todas as simulações, os parâmetros do algoritmo IAF-PNLMS são $a_i(0) = 10^{-3}$ e $\mu = 7,25 \times 10^{-3}$. A condição inicial considerada na simulação de Monte Carlo é $\mathbf{w}(0) = \mathbf{0}$ e no modelo proposto, $E[\mathbf{w}(0)] = \mathbf{0}$.

6.5.1 Exemplo 6.1

Neste exemplo, visando avaliar as expressões do modelo proposto (6.36) e (6.39), é considerada uma resposta ao impulso esparsa \mathbf{w}_1^o com $N = 32$ e grau de esparsidade $S_D(\mathbf{w}_1^o) = 0,7889$, cujos elementos não nulos são dados por

$$(\mathbf{w}_1^o)_i = \begin{cases} 0.8, & i = 1 \\ 1.0, & i = 10 \\ -0.9, & i = 15 \\ -0.7, & i = 24. \end{cases} \quad (6.42)$$

Para este exemplo, a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é $\chi = 270.98$.

Nas Figuras 6.1 e 6.2, a precisão do modelo proposto é verificada considerando um filtro de comprimento curto com 32 coeficientes. A Figura 6.1 mostra as curvas do comportamento médio dos coeficientes do filtro adaptativo, obtidas através de simulações de Monte Carlo e do modelo de predição (6.36), enquanto a Figura 6.2 ilustra as curvas de aprendizagem obtidas através de simulações de Monte Carlo e do modelo de predição (6.39). Aqui, observa-se a precisão das expressões do modelo

para dois valores de SNR: 20 dB [Figuras 6.1 e 6.2(a)] e 30 dB [Figura 6.2(b)].

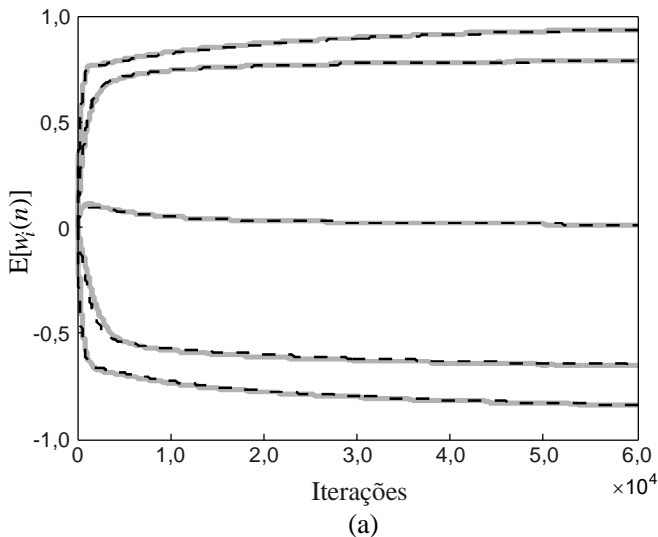
6.5.2 Exemplo 6.2

Neste exemplo, o modelo proposto é avaliado considerando uma resposta ao impulso esparsa \mathbf{w}_2^0 com $N = 128$ e grau de esparsidade $S_D(\mathbf{w}_2^0) = 0.9047$, cujos elementos não nulos são dados por

$$(\mathbf{w}_2^0)_i = \begin{cases} 0.8, & i = 1 \\ 1.0, & i = 30 \\ -0.9, & i = 35 \\ -0.7, & i = 85. \end{cases} \quad (6.43)$$

Neste exemplo, a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é $\chi = 324$.

A Figura 6.3 ilustra as curvas de comportamento médio dos coeficientes obtidas usando simulações de Monte Carlo e o modelo proposto (6.36). Na Figura 6.4, são apresentadas as curva de aprendizagem, obtidas usando simulações de Monte Carlo e o modelo de predição (6.39). Dessas figuras, a precisão das expressões do modelo é verificada considerando SNR = 20 dB.



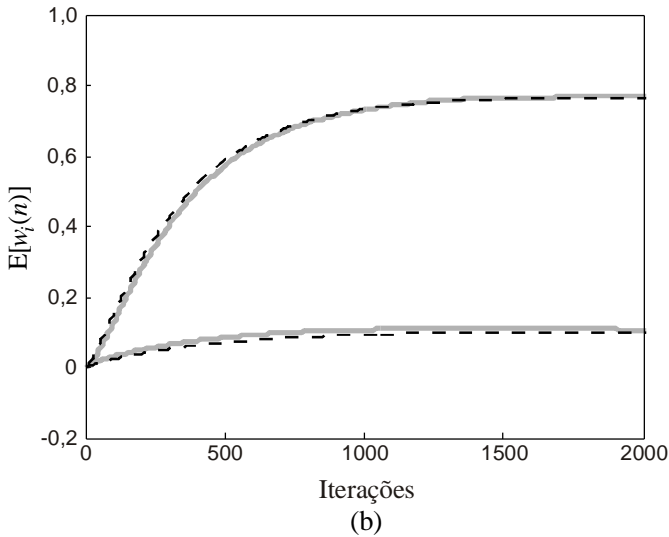


Figura 6.1. Exemplo 6.1. Curvas do comportamento médio dos coeficientes, obtidas usando simulações de Monte Carlo (linhas contínuas cinzas) e o modelo proposto (6.36) (linhas tracejadas pretas) considerando $\text{SNR} = 20$ dB. (a) De cima para baixo: $E[w_{10}(n)]$, $E[w_1(n)]$, $E[w_2(n)]$, $E[w_{24}(n)]$ e $E[w_{15}(n)]$. (b) Detalhe da fase transitória de $E[w_{10}(n)]$ (linhas superiores) e $E[w_2(n)]$ (linhas inferiores).

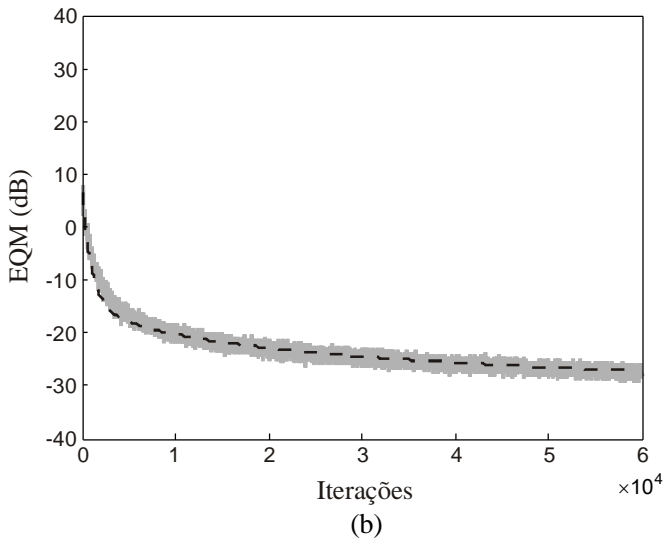
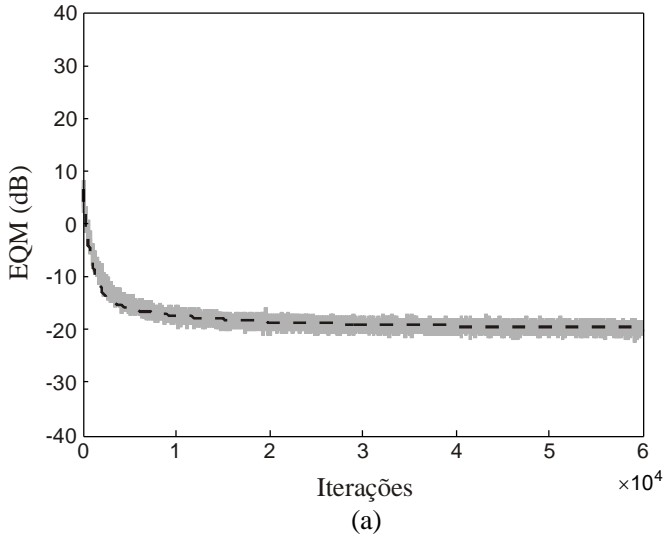


Figura 6.2. Exemplo 6.1. Curvas de aprendizagem (EQM) obtidas usando simulações de Monte Carlo (linha sólida cinza) e o modelo proposto (6.39) (linha tracejada preta). (a) SNR = 20 dB. (b) SNR = 30 dB.

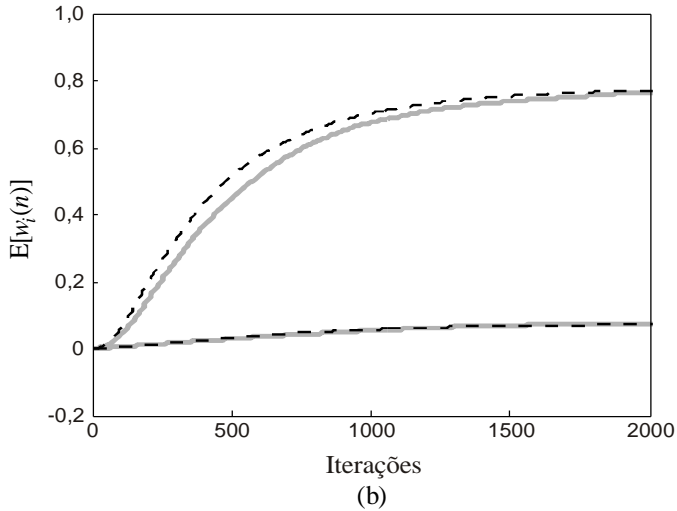
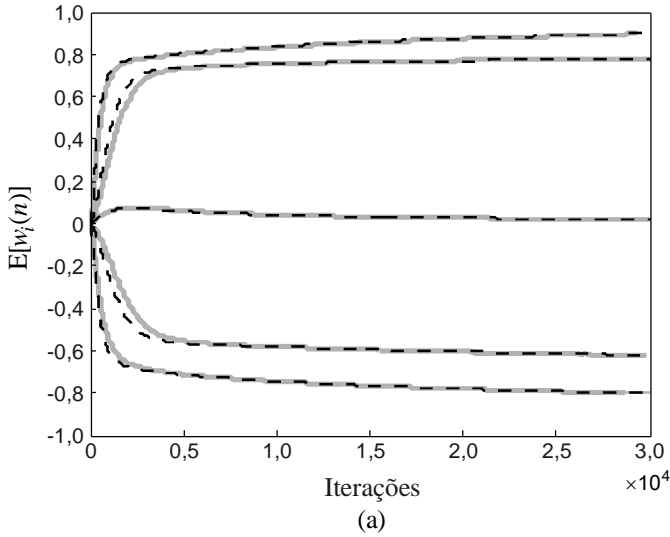


Figura 6.3. Exemplo 6.2. Curvas do comportamento médio dos coeficientes obtidas usando simulações de Monte Carlo (linha sólida cinza) e modelo proposto (6.36) (linha sólida preta) considerando $SNR = 20$ dB. (a) De cima para baixo: $E[w_{30}(n)]$, $E[w_1(n)]$, $E[w_2(n)]$, $E[w_{85}(n)]$ e $E[w_{35}(n)]$. (b) Detalhe da fase transitória $E[w_{30}(n)]$ (linhas superiores) e $E[w_2(n)]$ (linhas inferiores).

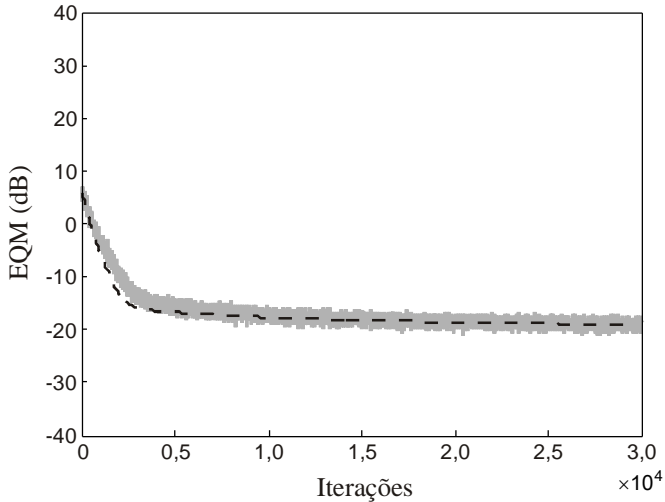


Figura 6.4. Exemplo 6.2. Curvas de aprendizagem (EQM) obtidas usando simulações de Monte Carlos (linha sólida cinza) e o modelo proposto (6.39) (linha tracejada preta) considerando SNR = 20 dB.

6.6. Conclusões

Neste capítulo, foi apresentado um modelo estocástico para o segundo momento do vetor de erro dos coeficientes bem como para a curva de aprendizagem do algoritmo adaptativo IAF-PNLMS. Além disso, uma nova expressão de modelo descrevendo o comportamento médio do vetor de coeficientes foi obtida. No desenvolvimento das expressões do modelo estocástico, considera-se que a matriz de ganhos é variante no tempo e os dados de entrada são sinais gaussianos correlacionados. O modelo proposto foi obtido levando em conta um novo procedimento de autodecomposição das matrizes surgidas no cálculo dos valores esperados envolvidos. Tal procedimento de autodecomposição é baseado na ordenação dos autovalores e descarte dos menos significativos. Através de simulações numéricas, a precisão das expressões de modelo propostas foi confirmada. A metodologia derivada aqui pode também ser usada para modelar outros algoritmos da classe PNLMS.

Uma Nova Estratégia de Distribuição de Ganhos para o Algoritmo IAF-PNLMS

7.1. Introdução

Neste capítulo, é discutido um algoritmo IAF-PNLMS que (durante o processo adaptativo) utiliza uma nova estratégia de distribuição de ganhos para atualização dos coeficientes do filtro adaptativo. Essa estratégia consiste em aumentar o ganho atribuído aos coeficientes inativos à medida que os coeficientes ativos aproximam-se da convergência. Para tal, sempre que um pré-definido limiar é atingido durante o processo de aprendizagem, uma nova distribuição de ganhos é utilizada no lugar de atribuir ganhos proporcionais às magnitudes dos coeficientes como o algoritmo IAF-PNLMS faz. Essa nova versão do algoritmo IAF-PNLMS leva a uma melhor distribuição da energia de adaptação durante todo o processo de aprendizagem. Como consequência, para respostas ao impulso exibindo elevada esparsidade, o algoritmo proposto aqui atinge uma convergência mais rápida, superando em termos de velocidade de convergência o IAF-PNLMS e outros algoritmos bem conhecidos da classe PNLMS.

7.2. Formulação do Algoritmo

A distribuição de ganhos dos algoritmos da classe PNLMS tem um impacto importante na velocidade de convergência desses algoritmos. Por exemplo, como mostrado em [34], o algoritmo IAF-PNLMS transfere ganhos dos coeficientes inativos para os ativos, levando a uma mais rápida convergência dos últimos. Como resultado, a velocidade de convergência global do algoritmo aumenta significativamente visto que a convergência dos coeficientes ativos é o principal responsável pela redução do desalinhamento durante a fase transitória. Por outro lado, quando a convergência dos coeficientes ativos é quase atingida, a contribuição dos coeficientes inativos para

reduzir o desalinhamento torna-se tão relevante quanto a dos ativos; assim nessa fase de convergência, é vantajoso atribuir maior ganho aos coeficientes inativos visando melhorar a convergência global do algoritmo.

Na próxima seção, as expressões gerais dos algoritmos da classe PNLMS são primeiramente revisitadas, e então uma nova estratégia para o cálculo dos ganhos do algoritmo IAF-PNLMS é introduzida.

7.2.1. Expressões Gerais dos Algoritmos da Classe PNLMS

Conforme mostrado na Seção 1.1.2., a equação de atualização dos coeficientes dos algoritmos da classe PNLMS é dada por [8], [34], [47]-[49]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n) + \xi} \quad (7.1)$$

onde $\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \cdots \ w_N(n)]^T$ é o vetor de coeficientes adaptativos, $\mathbf{x}(n) = [x(n) \ x(n-1) \ \cdots \ x(n-N+1)]^T$ denota o vetor de entrada, μ é o parâmetro de passo e $\xi > 0$, uma constante de regularização que previne divisão por zero em (7.1), estabilizando a solução. A matriz

$$\mathbf{G}(n) = \text{diag}[g_1(n) \ g_2(n) \ \cdots \ g_N(n)] \quad (7.2)$$

distribui os ganhos individuais entre os coeficientes do filtro adaptativo de acordo com uma regra específica levando em conta o tipo de algoritmo da classe PNLMS considerado. O sinal de erro é dado por

$$e(n) = d(n) - \mathbf{w}^T(n) \mathbf{x}(n) + z(n) \quad (7.3)$$

onde a variável $d(n)$ denota o sinal desejado e $z(n)$ é um ruído de medição de média zero, variância σ_z^2 , independente e identicamente distribuído (i.i.d.).

7.2.2. Algoritmo PNLMS com Fatores de Ativação Individuais

O ganho individual que governa o ajuste do passo de adaptação do i -ésimo coeficiente do filtro adaptativo do algoritmo IAF-PNLMS é obtido por [33], [34]

$$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^N \phi_j(n)}, \quad i = 1, 2, \dots, N \quad (7.4)$$

onde

$$\phi_i(n) = \max[a_i(n), |w_i(n)|] \quad (7.5)$$

com fator de ativação individual

$$a_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN, \quad m = 1, 2, 3, \dots \\ a_i(n-1), & \text{caso contrário.} \end{cases} \quad (7.6)$$

Em contraste aos outros algoritmos da classe PNLMS, tais como PNLMS padrão e o IPNLMS, o algoritmo IAF-PNLMS transfere parte dos ganhos dos coeficientes inativos para os coeficientes ativos via $a_i(n)$ [34] e, como consequência, apresenta as seguintes principais características:

- i) Fornece melhor distribuição de ganhos do que os algoritmos PNLMS e IPNLMS na fase transitória.
- ii) Reduz a velocidade de convergência dos coeficientes inativos quando o algoritmo aproxima-se da fase estacionária.

O primeiro ponto i) leva a uma melhoria das características de convergência bem como da habilidade de rastreamento, permitindo ao algoritmo IAF-PNLMS superar ambos os algoritmos PNLMS e IPNLMS para respostas ao impulso com elevada esparsidade. Por outro lado, ii) é indesejável, surgindo do fato de que o algoritmo IAF-PNLMS transfere ganhos dos coeficientes inativos para os ativos durante todo o processo de adaptação.

Na próxima seção, propõe-se uma estratégia para melhorar o algoritmo IAF-PNLMS (EIAF-PNLMS - *Enhanced Individual-Activation-Factor PNLMS*), visando contornar o segundo ponto ii). Para tal, uma nova abordagem é considerada para o cálculo dos ganhos dos coeficientes do filtro adaptativo.

7.2.3. Algoritmo PNLMS com Fatores de Ativação Individuais Melhorado

Nesta seção, uma nova estratégia para distribuir ganhos para os coeficientes do filtro adaptativo IAF-PNLMS é desenvolvida. Essa estratégia é baseada no fato de que a filosofia proporcional, a qual é a essência dos algoritmos da classe PNLMS, não precisa ser aplicada durante todo o processo de adaptação, visto que atribuir ganhos maiores aos coeficientes do filtro adaptativo que já praticamente convergiram não tem efeito adicional sobre a convergência global do algoritmo.

A. Nova Estratégia de Distribuição de Ganhos

Aqui, o objetivo é conceber uma estratégia para redistribuir os ganhos de adaptação dos coeficientes do filtro adaptativo que atingiram a vizinhança K_o de seus valores ótimos, definida como

$$\left| w_i^o - w_i(L_i) \right| \leq K_o, \quad i = 1, 2, \dots, N \quad (7.7)$$

onde w_i^o é o i -ésimo coeficiente ótimo da planta e L_i é o número de iterações necessário para o i -ésimo coeficiente do filtro adaptativo atingir a vizinhança K_o . Assim, no instante L_i , o i -ésimo coeficiente adaptativo $w_i(n)$ quase convergiu; portanto, esse coeficiente não precisa mais receber ganho proporcional à sua magnitude. Dessa forma, parte do ganho previamente atribuído a tal coeficiente pode ser agora redistribuída aos coeficientes que ainda não atingiram a vizinhança K_o , conseqüentemente acelerando as suas convergências. Então, levando em conta a condição de limiar (7.7), o ganho individual de cada coeficiente é reescrito como

$$g_i(n) = \begin{cases} \gamma_i(n), & |w_i^o - w_i(L_i)| \leq K_o \\ \frac{\phi_i(n)}{\sum_{j=1}^N \phi_j(n)}, & \text{caso contrário} \end{cases} \quad (7.8)$$

onde $\gamma_i(n)$ é o ganho mínimo atribuído aos coeficientes que atingiram a vizinhança K_o . A partir de (7.8), nota-se que é requerido o conhecimento *a priori* do i -ésimo coeficiente da planta w_i^o para utilizar essa estratégia de distribuição de ganhos. Assim, é um desafio usar (7.8) em aplicações reais, visto que em geral a planta do sistema não é conhecida *a priori*. A seguir, visando contornar o problema previamente mencionado, é proposta uma abordagem que não requer o conhecimento de w_i^o .

B. Abordagem Proposta

O algoritmo IAF-PNLMS atribui ganhos proporcionais aos coeficientes do filtro adaptativo, baseado nas suas magnitudes instantâneas. Visto que as magnitudes dos coeficientes inativos são muito pequenas, os ganhos atribuídos a tais coeficientes são também pequenos. Assim, em cada iteração, a maior parte do ganho total $g_{\text{total}}(n)$ [veja definição do ganho total em (2.5)] é atribuída aos coeficientes ativos, mesmo durante a fase de adaptação na qual tais coeficientes já atingiram a vizinhança K_o . Durante essa fase, é desejável atribuir maior ganho aos coeficientes inativos, visto que tais coeficientes tornam-se cada vez mais importantes para o processo de convergência global do algoritmo. Portanto, é proposta aqui uma abordagem para usar a nova estratégia de distribuição de ganhos dada por (7.8), visando aumentar a velocidade de convergência global do algoritmo. Para tal, procede-se como segue:

- a) A partir de (7.3) e considerando que $d(n) = \mathbf{x}^T(n)\mathbf{w}^o$, o erro quadrático instantâneo é reescrito como

$$e^2(n) = \{ [w_1^0 - w_1(n)]x_1(n) + [w_2^0 - w_2(n)]x_2(n) + \dots + [w_N^0 - w_N(n)]x_N(n) + z(n) \}^2 \quad (7.9)$$

onde $x_i(n) \triangleq x(n-i+1)$ é o i -ésimo elemento do vetor $\mathbf{x}(n)$. Agora comparando (7.7) e (7.9), nota-se que o erro quadrático instantâneo pode ser visto como um indicador global de convergência, levando em conta o conhecimento da convergência de todos os coeficientes adaptativos, em contraste à condição de limiar (7.7), a qual é um indicador de convergência individual do i -ésimo coeficiente. Mais importante ainda é o fato de que se pode calcular $e^2(n)$ usando (7.3), visto que em geral o sinal $[d(n) + z(n)]$ está disponível, não requerendo o conhecimento de w_i^0 . Portanto, define-se a seguir uma condição de limiar baseada em $e^2(n)$, no lugar de usar (7.7).

- b) Agora, baseado no erro quadrático instantâneo $e^2(n)$, define-se um limiar no processo convergência para indicar o ponto no qual a distribuição de ganhos deve ser modificada. Assim, considere

$$\psi_e(n) \leq K \quad (7.10)$$

onde $\psi_e(n)$ é uma versão suavizada do erro quadrático instantâneo, dada por

$$\psi_e(n) = \beta\psi_e(n-1) + (1-\beta)e^2(n) \quad (7.11)$$

com β ($0 < \beta < 1$) sendo o fator de suavização. Note que o valor de limiar K , juntamente com L_i determinam o instante no qual um ganho mínimo deve ser atribuído para cada coeficiente do filtro adaptativo, enquanto um limiar do tipo global K juntamente com $\psi_e(n)$ definem o instante no qual um ganho uniforme deve ser aplicado a todos os coeficientes

do filtro adaptativo. Portanto, levando em conta a condição de limiar global (7.10), o ganho individual atribuído ao i -ésimo coeficiente é revisado para

$$g_i(n) = \begin{cases} \gamma_i(n), & \psi_e(n) \leq K \\ \frac{\phi_i(n)}{\sum_{j=1}^N \phi_j(n)}, & \text{caso contrário.} \end{cases} \quad (7.12)$$

A seguir, determina-se o ganho $\gamma_i(n)$ a ser usado sempre que a condição de limiar (7.10) ocorra.

- c) Verifica-se de (7.2) e (7.4) que em cada iteração, o ganho total atribuído a todos os coeficientes é dado por [34]

$$\begin{aligned} g_{\text{tot}}(n) &= \text{tr}[\mathbf{G}(n)] \\ &= N. \end{aligned} \quad (7.13)$$

Agora, sempre que a condição (7.10) ocorra e assumindo que todos os coeficientes atingiram a vizinhança K_0 , isto é, $L_1 = L_2 = \dots = L_N$, um ganho comum mínimo dado por

$$\gamma_i(n) = \frac{1}{g_{\text{tot}}(n)} = \frac{1}{N} \quad (7.14)$$

é atribuído a todos os coeficientes do filtro adaptativo.

Aqui, enfatiza-se que é sempre requerido calcular (7.5) e (7.6), mesmo se $\psi_e(n) \leq K$. Isso é devido ao fato de que i -ésimo fator de ativação do algoritmo IAF-PNLMS contém informação passada (alguma memória) da magnitude do i -ésimo coeficiente. Como consequência, a habilidade de rastreamento do algoritmo é melhorada (para mais detalhes, veja [34]). Assim, visando manter essa propriedade na versão melhorada do IAF-PNLMS (algoritmo proposto), as variáveis $\phi_i(n)$ e

$a_i(n)$ devem ser calculadas e armazenadas para serem usadas na próxima iteração [veja (7.6)].

7.3. Resultados de Simulação

Nesta seção, simulações de Monte Carlo (MC) (media de 100 realizações independentes) do algoritmo EIAF-PNLMS são realizadas, considerando um problema de identificação de sistemas. O objetivo dessas simulações é comparar a velocidade de convergência dos algoritmos NLMS, IPNLMS, MPNLMS, IAF-PNLMS e EIAF-PNLMS. Para tal, considera-se uma resposta ao impulso esparsa sintética \mathbf{w}^o com N coeficientes, gerada a partir de uma seqüência gaussiana i.i.d. Para obter tal resposta ao impulso, considere primeiro o seguinte vetor (de dimensão N_u) com decaimento exponencial [19], [22]:

$$\mathbf{u} = [1 \ e^{-1/\tau} \ e^{-2/\tau} \ \dots \ e^{-(N_u-1)/\tau}]^T \quad (7.15)$$

onde $\tau > 0$ é a constante de decaimento. Agora, usando (7.15), a resposta ao impulso é obtida como

$$\mathbf{w}^o = [\mathbf{s}^T \ \mathbf{u}^T \mathbf{L}]^T \quad (7.16)$$

onde $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_{N-N_u}]^T$ é uma seqüência gaussiana i.i.d. com media zero e variância σ_s^2 , e \mathbf{L} é uma matriz diagonal de dimensão $N_u \times N_u$, cujos elementos são obtidos a partir de uma seqüência gaussiana i.i.d. com media zero e variância σ_ℓ^2 . Para quantificar a esparsidade da resposta ao impulso \mathbf{w}^o , usa-se a medida definida em (1.1), dada por [1], [23], [48]

$$S_D(\mathbf{w}^o) \triangleq \frac{N}{N - \sqrt{N}} \left(1 - \frac{\|\mathbf{w}^o\|_1}{\sqrt{N} \|\mathbf{w}^o\|_2} \right) \quad (7.17)$$

onde o vetor \mathbf{w}^o de dimensão N representa a resposta ao impulso da planta, e $\|\mathbf{w}^o\|_1$ e $\|\mathbf{w}^o\|_2$ são, respectivamente, as normas ℓ_1 e ℓ_2 de \mathbf{w}^o . A medida $S_D(\mathbf{w}^o)$ varia de 0 (grau de esparsidade de um filtro uniforme) a 1 (grau de esparsidade do filtro de Dirac) [1], [48].

Em todas as simulações, o sinal de entrada $x(n)$ is correlacionado, com média zero e variância $\sigma_x^2 = 1$, obtido a partir do processo AR(2)

$$x(n) = b_1 x(n-1) + b_2 x(n-2) + \eta(n) \quad (7.18)$$

onde b_1 e b_2 são os coeficientes do processo AR(2), e $\eta(n)$ é um ruído branco com variância σ_η^2 . Nas simulações, a razão sinal-ruído (SNR) é definida como

$$\text{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_z^2} \quad (7.19)$$

onde σ_z^2 é a variância do ruído de medição $z(n)$ [veja (7.3)]. O sinal desejado é dado por

$$d(n) = \mathbf{x}^T(n) \mathbf{w}^o \quad (7.20)$$

e a condição inicial escolhida para o vetor de coeficientes adaptativos é $\mathbf{w}(0) = \mathbf{0}$. A inicialização do fator de ativação para ambos os algoritmos IAF-PNLMS e EIAF-PNLMS é $a_i(0) = 10^{-4}$. Aqui, para propósito de comparações, usa-se $\mu = 0.5$ como valor do parâmetro de passo a menos que seja afirmado o contrário.

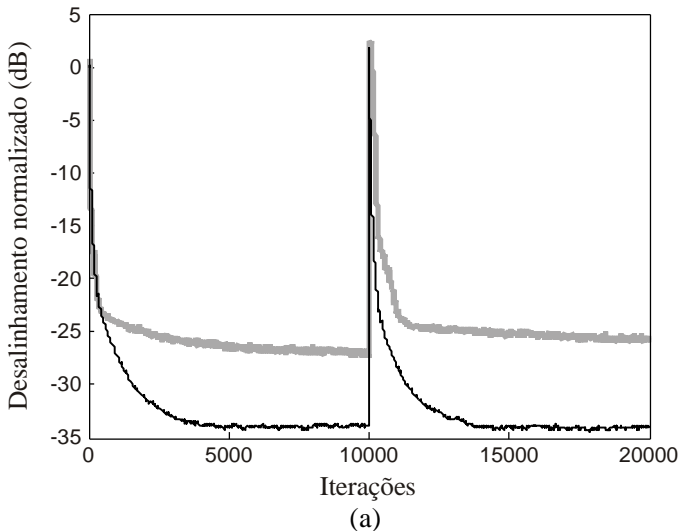
Para avaliar o desempenho dos algoritmos aqui considerados, usa-se o desalinhamento normalizado (em dB), isto é [1], [48]

$$\kappa(n) = 10 \log_{10} \frac{\|\mathbf{w}^o - \mathbf{w}(n)\|_2^2}{\|\mathbf{w}^o\|_2^2}. \quad (7.21)$$

7.3.1. Exemplo 7.1

Este exemplo compara as características de convergência dos algoritmos IAF-PNLMS e EIAF-PNLMS. Para esse exemplo, uma resposta ao impulso \mathbf{w}^o com $N=32$ e grau de esparsidade $S_D(\mathbf{w}^o)=0,9402$ é usada, a qual é obtida de (7.16) considerando $\tau=1,3$, $\sigma_s^2=1,055 \times 10^{-4}$, $\sigma_u^2=0,9146$ e $N_u=16$. O sinal de entrada é obtido de (7.18) com $b_1=1,29$, $b_2=-0,75$ e $\sigma_{\eta}^2=0,20$. Assim, a dispersão dos autovalores da matriz de autocorrelação do vetor de entrada é $\chi=271$.

Na Figura 7.1, a convergência dos algoritmos IAF-PNLMS e EIAF-PNLMS é avaliada, considerando uma perturbação na resposta ao impulso da planta na iteração $n=10000$, quando a resposta ao impulso \mathbf{w}^o é deslocada para a direita de 12 amostras, mudando a posição de todos os coeficientes. Observe que o algoritmo EIAF-PNLMS supera em desempenho o IAF-PNLMS para SNR = 30dB bem como para SNR = 40dB.



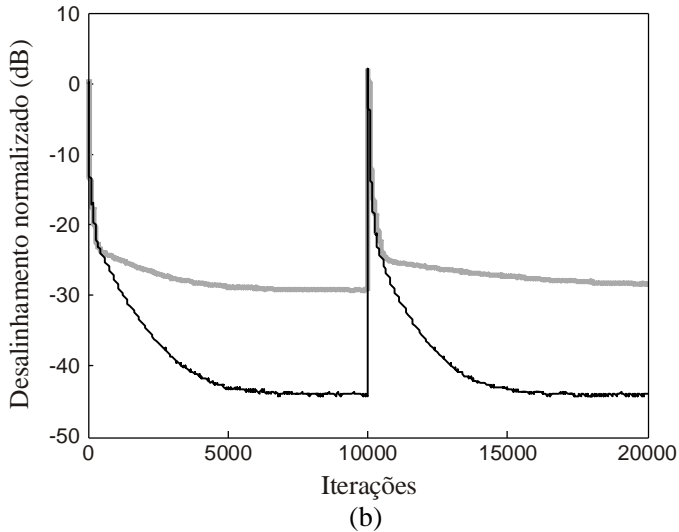


Figura 7.1. Exemplo 7.1. Curvas de desalinhamento normalizado dos algoritmos IAF-PNLMS (linhas cinza) e EIAF-PNLMS (linhas pretas) considerando um deslocamento dos coeficientes da planta com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. (a) SNR = 30 dB. (b) SNR = 40 dB.

Na Figura 7.2, avalia-se as características de convergência do algoritmo EIAF-PNLMS com respeito ao limiar K , considerando dois valores de SNR (20 dB e 40 dB). Para tal, a Figura 2(a) mostra o desalinhamento normalizado usando $K = 2 \times 10^{-2}$ (linha preta) e $K = 3 \times 10^{-2}$ (linha cinza), considerando SNR = 20 dB. Na Figura 2(b), as curvas de desalinhamento normalizado usando $K = 3 \times 10^{-3}$ (linha preta) e $K = 7 \times 10^{-3}$ (linha cinza) são mostrada, usando SNR = 40 dB. Salienta-se que a escolha do valor desse parâmetro não é crítica visto que tal valor pode ser escolhido dentro de uma faixa de valores sem praticamente alterar o comportamento do algoritmo. Por exemplo, para SNR = 20 dB, uma faixa típica de valores de limiar que levam a uma operação adequada do algoritmo é $2 \times 10^{-2} \leq K \leq 3 \times 10^{-2}$, enquanto para SNR = 40 dB, uma faixa típica é $3 \times 10^{-3} \leq K \leq 7 \times 10^{-3}$.

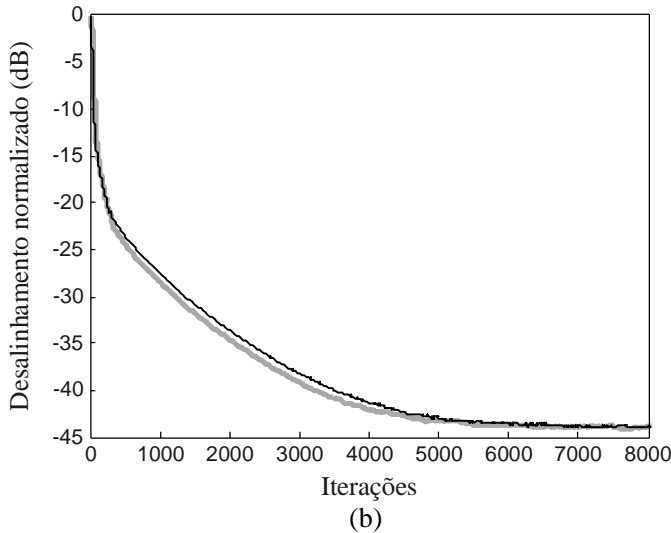
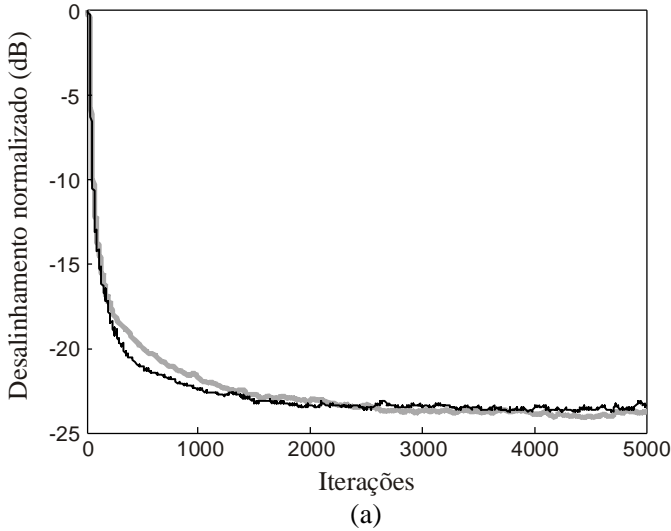
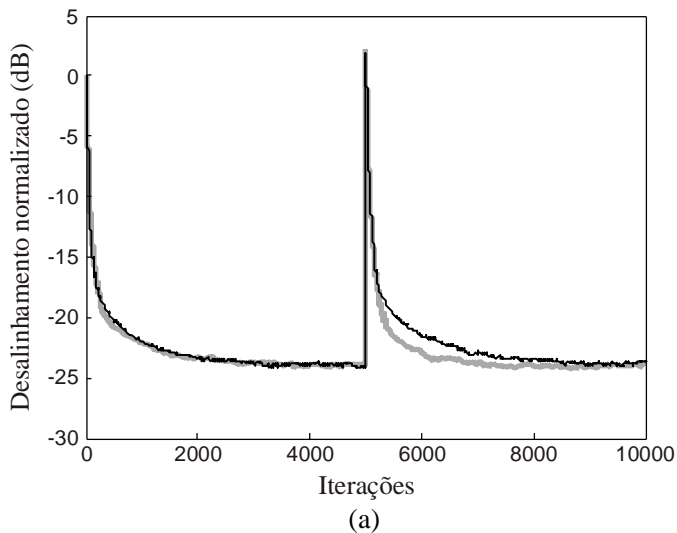
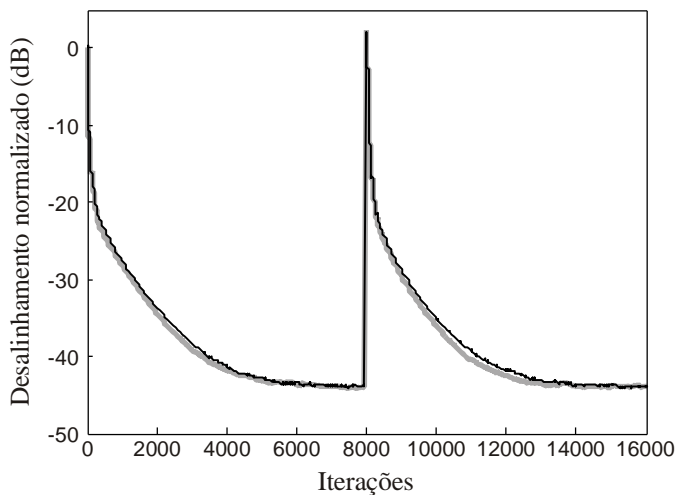


Figura 7.2. Exemplo 7.1. Curvas de desalinhamento normalizado do algoritmo EIAF-PNLMS usando $\beta = 0,98$. (a) $K = 2 \times 10^{-2}$ (linha preta) e $K = 3 \times 10^{-2}$ (linha cinza), considerando SNR = 20 dB. (b) $K = 3 \times 10^{-3}$ (linha preta) e $K = 7 \times 10^{-3}$ (linha cinza), considerando SNR = 40 dB.

A Figura 7.3 ilustra o impacto do fator de suavização β no comportamento do algoritmo EIAF-PNLMS, considerando também dois valores de SNR: 20 dB e 40 dB. Além disso, o impacto do fator β sobre a habilidade de rastreamento do algoritmo EIAF-PNLMS é avaliada levando em conta uma perturbação na resposta ao impulso da planta (resposta ao impulso \mathbf{w}^o é deslocada de 12 amostras para a direita). Aqui, mostra-se o desalinhamento normalizado usando dois valores de β . Note dessas figuras que a faixa típica de valores apropriados para o fator de suavização é $0,97 \leq \beta \leq 0,99$.





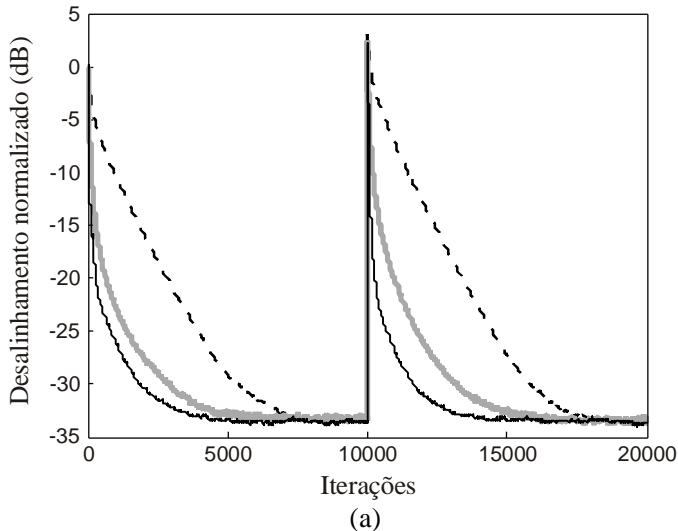
(b)

Figura 7.3. Exemplo 7.1. Desalinhamento normalizado do algoritmo EIAF-PNLMS considerando um deslocamento dos coeficientes da planta. Fator de suavização $\beta = 0,97$ (linhas pretas) e $\beta = 0,99$ (linhas cinzas). (a) SNR = 20 dB com $K = 2,5 \times 10^{-2}$. (b) SNR = 40 dB com $K = 5 \times 10^{-3}$.

7.3.2. Exemplo 7.2

Neste exemplo, os algoritmos NLMS, IPNLMS e EIAF-PNLMS são comparados tanto em termos de velocidade de convergência quanto em termos de resposta a perturbações na planta. Para tal, usa-se uma resposta ao impulso da planta \mathbf{w}^o com $N=32$ e $S_D(\mathbf{w}^o)=0,9246$, obtida de (7.16) considerando $\tau=1,5$, $\sigma_s^2=1,055 \times 10^{-4}$, $\sigma_u^2=0,9246$ e $N_u=16$. O sinal de entrada tem dispersão $\chi=396$, obtido de (7.18) com $b_1=1,09$ e $b_2=-0,85$, usando $\sigma_\eta^2=0,18$.

A Figura 7.4 ilustra o desalinhamento normalizado dos algoritmos NLMS, IPNLMS e EIAF-PNLMS para SNR = 30 dB [Figura 7.4(a)] e SNR = 40 dB [Figura 7.4(b)], usando o mesmo padrão de perturbação de planta considerado para a obtenção dos resultados apresentados na Figura 7.1. Para o IPNLMS, $\alpha=0$ (um parâmetro desse algoritmo) é usado [8], [34]; ressalta-se aqui que para uma comparação de desempenho justa, são selecionados valores de passo μ que levem os algoritmos avaliados ao mesmo desalinhamento em regime permanente. Note que o algoritmo EIAF-PNLMS atinge a mais rápida convergência dentre os algoritmos avaliados. Além disso, verifica-se também dessas figuras que o algoritmo EIAF-PNLMS é aquele que exibe a melhor resposta a perturbações na planta.



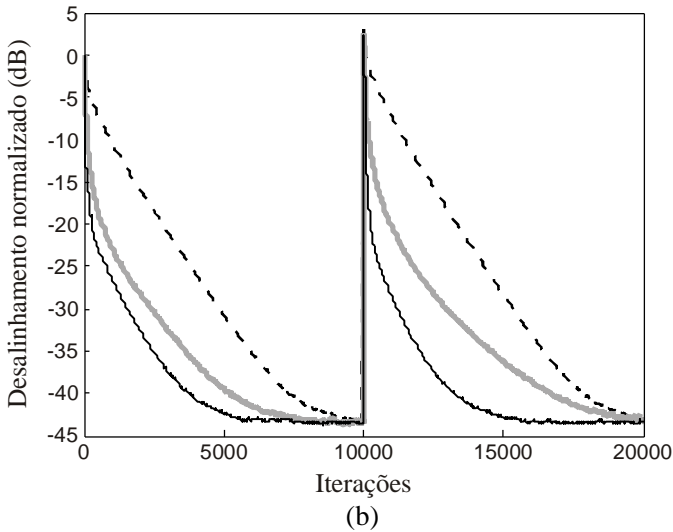


Figura 7.4. Exemplo 7.2. Desalinhamento normalizado dos algoritmos NLMS (linhas pretas tracejadas), IPNLMS com $\mu = 0,45$ (linhas cinzas sólidas) e EIAF-PNLMS (linhas pretas sólidas), considerando um deslocamento dos coeficientes da planta com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. (a) SNR = 30 dB. (b) SNR = 40 dB.

7.3.3. Exemplo 7.3

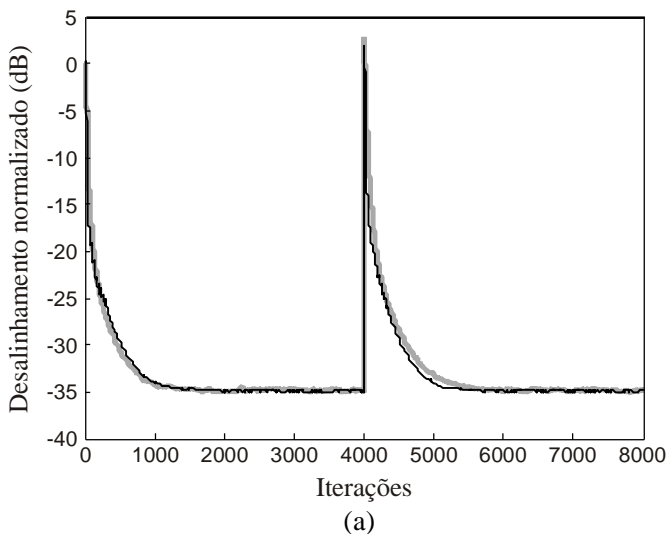
Este exemplo compara os algoritmos MPNLMS e EIAF-PNLMS tanto em termos de velocidade de convergência quanto de resposta a perturbações na planta sob diferentes condições de entrada. Para tal, considera-se dados de entrada com diferentes dispersões de autovalores da matriz de autocorrelação do vetor de entrada, usando os parâmetros do processo AR(2) apresentados na Tabela 7.1. Aqui, a resposta ao impulso da planta \mathbf{w}^o com $N = 64$ e $S_D(\mathbf{w}^o) = 0,9602$ é obtida de (7.16) usando $\tau = 1,3$, $\sigma_s^2 = 1,055 \times 10^{-4}$, $\sigma_u^2 = 0,9146$ e $N_u = 48$.

TABELA 7.1

EXEMPLO 7.3. PARÂMETROS DO PROCESSO AR(2) USADOS PARA OBTER DADOS DE ENTRADA COM DIFERENTES DISPERSÕES DE AUTOVALORES

Dispersão de Autovalores χ	b_1	b_2	σ_η^2
10	0,62	-0,25	0,70
232	1,17	-0,75	0,24

A Figura 7.5 mostra o desalinhamento normalizado dos algoritmos MPNLMS e EIAF-PNLMS para SNR = 30 dB, considerando dispersões de autovalores $\chi=10$ e $\chi=232$. No MPNLMS, $\varepsilon=0,0316$ é usado [10]; os parâmetros de proporcionalidade e inicialização são, respectivamente, $5/N$ e 0,01. O parâmetro de passo μ é selecionado para levar os algoritmos MPNLMS e EIAF-PNLMS ao mesmo nível de desalinhamento normalizado em regime permanente. Observa-se que os algoritmos avaliados apresentam desempenhos similares para valor de χ pequeno [Fig. 5(a)]; contudo, para valor de χ grande [Fig. 5(b)], o algoritmo EIAF-PNLMS atinge convergência mais rápida, superando o algoritmo MPNLMS.



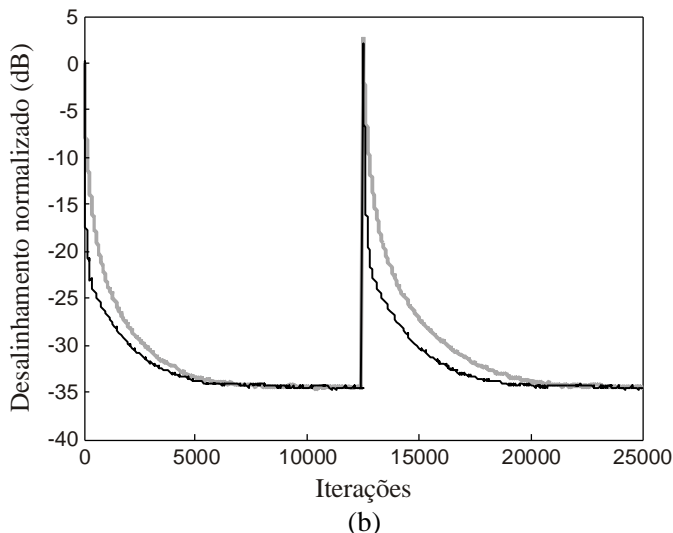


Figura 7.5. Exemplo 7.3. Desalinhamento normalizado dos algoritmos MPNLMS (linhas cinzas) e EIAF-PNLMS (linhas pretas) com $K = 5 \times 10^{-3}$ e $\beta = 0,98$. Os coeficientes da resposta ao impulso são deslocados de 12 amostras para a direita. (a) $\chi = 10$ usando $\mu = 0,5$ para o MPNLMS. (b) $\chi = 232$ usando $\mu = 0,43$ para o MPNLMS.

7.3.4. Discussão

Aqui, baseado nos resultados de simulações apresentados neste capítulo, são apresentadas algumas observações importantes sobre os parâmetros do algoritmo EIAF-PNLMS. Além disso, alguns aspectos filosóficos dos algoritmos IPNLMS e EIAF-PNLMS são também destacados.

A. Parâmetro de Limiar K

Esse parâmetro define o ponto no qual o algoritmo EIAF-PNLMS muda a sua política de distribuição de ganhos. Conforme os resultados de simulações, nota-se que determinar o valor apropriado de K não é uma tarefa trivial; de fato, tal valor depende das características da

planta, tais como grau de esparsidade e comprimento da resposta ao impulso. Uma regra prática para selecionar o valor apropriado para o parâmetro de limiar é considerar $K > \sigma_z^2$, visto que a mudança da política de distribuição de ganhos deve ocorrer antes que o algoritmo atinja seu regime permanente. Assim, de acordo com o cenário de aplicação, pode-se selecionar uma faixa apropriada de valores de K para uma operação segura do algoritmo.

B. Fator de suavização β

Visando evitar que flutuações rápidas do erro quadrático instantâneo $e^2(n)$ possam levar a mudanças indesejáveis na política de distribuição de ganhos, degradando o desempenho do algoritmo, a versão suavizada do erro quadrático instantâneo $\psi_e(n)$ é usada no lugar de $e^2(n)$ para estabelecer a condição de limiar (7.10). A variável $\psi_e(n)$ é calculada recursivamente em termos do fator de suavização β . Conforme mostrado nos resultados de simulações, valores na faixa $0.97 \leq \beta \leq 0.98$ levam a uma operação apropriada do algoritmo proposto. Ressalta-se que a versão suavizada do erro quadrático instantâneo aqui utilizada, corresponde a uma filtragem passa-baixa de $e^2(n)$ e tem sido largamente considerada na literatura [50]-[52].

C. Filosofia do Algoritmo EIAF-PNLMS

O uso da técnica de limiar em conjunto com uma medida suavizada de $e^2(n)$ tem sido também aplicado em [51] para decidir o ponto no qual o algoritmo deve mudar a sua condição de operação, contudo lá, para um propósito diferente do usado aqui. Na literatura, alguns algoritmos da classe PNLMS também têm usado uma versão suavizada do erro quadrático instantâneo. Por exemplo, em [50] e [52], é utilizada uma medida suavizada de $e^2(n)$ visando gradualmente e recursivamente ajustar o parâmetro μ do algoritmo MPNLMS; contudo, nesse caso, a técnica de limiar não é considerada para mudar a política de distribuição de ganhos do algoritmo durante o processo de

adaptação. Concernente ao algoritmo IPNLMS, há uma significativa diferença entre tal algoritmo e o EIAF-PNLMS. O ganho do algoritmo IPNLMS é formado por duas parcelas: uma é individual e proporcional à correspondente magnitude do coeficiente do filtro adaptativo e a outra, comum a todos os coeficientes do filtro sendo um fator de ganho não proporcional; essa estratégia para determinar o ganho é aplicada durante todo o processo de adaptação. Por outro lado, a abordagem utilizada no algoritmo EIAF-PNLMS altera a política de distribuição de ganhos para calcular o ganho durante o processo de adaptação sempre que um nível de limiar predefinido é atingido.

7.4. Conclusões

Neste capítulo, uma versão melhorada do algoritmo IAF-PNLMS, denominada EIAF-PNLMS, é proposta, a qual (durante o processo de aprendizagem) usa uma estratégia de redistribuição de ganhos visando aumentar os ganhos atribuídos aos coeficientes inativos quando os coeficientes ativos aproximam-se da convergência. Para realizar tal redistribuição, define-se primeiro um limiar para indicar o ponto no qual a redistribuição de ganhos deve ocorrer. Então, quando esse limiar é atingido, a redistribuição é realizada atribuindo ganho uniforme a todos os coeficientes do filtro adaptativo. Resultados de simulação mostram que as características de convergência do algoritmo são significativamente melhoradas, tal que o EIAF-PNLMS supera em desempenho os algoritmos IPNLMS e IAF-PNLMS para respostas ao impulso com elevado grau de esparsidade. Além disso, embora os algoritmos MPNLMS e EIAF-PNLMS exibam desempenhos similares para sinais de entrada fracamente correlacionados, o algoritmo proposto supera o MPNLMS para dados de entrada fortemente correlacionados.

Comentários e Conclusões Finais

A concepção de filtros adaptativos explorando a natureza esparsa da resposta ao impulso da planta vem se tornando um tópico de pesquisa de grande relevância, visto que essa classe de respostas é encontrada em muitas aplicações, tais como cancelamento de eco, localização de fontes, sistemas de transmissão de televisão digital, processos sísmicos, dentre outras. Para essa classe de respostas ao impulso, os algoritmos adaptativos clássicos, usando o mesmo passo de adaptação para todos os coeficientes do filtro, tal como o algoritmo LMS normalizado (NLMS – *normalized least-mean-square*), são superados tanto em termos de velocidade de convergência quanto em termos de habilidade de rastreamento por aqueles algoritmos especialmente projetados para explorar a natureza esparsa da planta do sistema. Por exemplo, os algoritmos da classe NLMS proporcional (PNLMS), tais como o PNLMS padrão, o PNLMS melhorado (IPNLMS) e o PNLMS com lei μ (MPNLMS) são especialmente projetados para lidar com plantas esparsas. Nesse contexto, duas novas versões melhoradas para a classe de algoritmos PNLMS foram concebidas neste trabalho de pesquisa.

Considerando a ampla gama de aplicações dos algoritmos da classe PNLMS, a sua modelagem estocástica é de grande utilidade prática. Ela é usada como uma ferramenta para prever o desempenho do algoritmo adaptativo sob diferentes condições de operação (evitando o uso de simulações exaustivas para a avaliação do algoritmo) bem como para obter valores apropriados de parâmetros do algoritmo, tais como o passo de adaptação e o comprimento do filtro, dentre outros. Além do mais, o modelo pode ser usado para entender melhor o funcionamento do algoritmo. Contudo, a modelagem dos algoritmos da classe PNLMS ainda é um tema em aberto na literatura especializada, devido principalmente à complexidade matemática envolvida. Assim, no presente trabalho, uma metodologia de análise que pode ser aplicada à classe de algoritmos PNLMS é desenvolvida. Em particular, essa metodologia foi aplicada na modelagem estocástica do algoritmo IAF-PNLMS.

No Capítulo 1, os conceitos gerais concernentes aos algoritmos da classe PNLMS (como filtragem adaptativa proporcional e grau de esparsidade), a motivação para o desenvolvimento de tais algoritmos, bem como os objetivos gerais e específicos deste trabalho de pesquisa foram apresentados. Também, os aspectos históricos e filosóficos da filtragem adaptativa proporcional foram brevemente discutidos.

Um estudo do comportamento do algoritmo PNLMS padrão considerando os efeitos do fator de ativação foi apresentado no Capítulo 2. De tal estudo, as seguintes conclusões foram obtidas:

- O fator de ativação tem um impacto significativo na distribuição de ganhos do algoritmo PNLMS padrão, afetando não somente os ganhos atribuídos aos coeficientes inativos, mas também aqueles atribuídos aos ativos.
- A forma como o fator de ativação é determinado e os valores selecionados para os parâmetros de proporcionalidade e inicialização alteram substancialmente a velocidade de convergência do algoritmo.
- Os coeficientes e ganhos do algoritmo PNLMS padrão podem apresentar um comportamento indesejável no início da fase de aprendizagem.

No Capítulo 3, dois dos principais algoritmos da classe PNLMS, isto é, os algoritmos IPNLMS e MPNLMS foram revisitados. Mostrou-se que o algoritmo IPNLMS apresenta uma derivação *ad hoc* a partir do algoritmo PNLMS padrão, visando introduzir um parâmetro de controle de esparsidade. Com tal estratégia, é possível ajustar o comportamento desse algoritmo de acordo com o grau de esparsidade da planta a ser identificada. No que concerne ao algoritmo MPNLMS, também com derivação *ad hoc*, pode-se considerá-lo uma aproximação estocástica do algoritmo de descida mais íngreme (*SD – Steepest Descent*) proporcional. Em geral, é esperado que o algoritmo MPNLMS herde as características de convergência do algoritmo SD proporcional como, por exemplo, uma velocidade convergência uniforme durante todo o processo de adaptação. Uma desvantagem do algoritmo MPNLMS é a sua elevada complexidade computacional.

Uma nova versão do algoritmo PNLMS padrão, denominada IAF-PNLMS, foi apresentada no Capítulo 4. Tal versão usa um fator de

ativação individual para cada coeficiente do filtro adaptativo. Cada fator de ativação é recursivamente calculado em função da magnitude de seu correspondente coeficiente e não mais depende dos parâmetros de proporcionalidade e inicialização considerados no algoritmo PNLMS padrão. O algoritmo IAF-PNLMS apresenta melhor distribuição de ganhos do que os algoritmos PNLMS padrão e IPNLMS, conduzindo a uma maior velocidade de convergência, superando tais algoritmos para respostas ao impulso com grau de esparsidade elevado. Além disso, simulações numéricas mostraram que a distribuição de ganhos do algoritmo proposto é bastante diferente daquela apresentada pelo algoritmo MPNLMS, embora as características de convergência sejam similares quando a planta exibe alto grau de esparsidade. No que concerne à habilidade de rastreamento bem como às perturbações na planta, mostrou-se que o algoritmo IAF-PNLMS supera o algoritmo MPNLMS, considerando respostas ao impulso com grau de esparsidade elevado.

No Capítulo 5, uma metodologia para obter um modelo estocástico dos coeficientes do algoritmo IAF-PNLMS foi apresentada. Em tal modelo, a matriz de ganhos é assumida variante no tempo e o sinal de entrada é correlacionado. Por conseqüência, um modelo do momento de primeira ordem é obtido, predizendo as fases transitória e estacionária do comportamento médio do vetor de coeficientes do algoritmo IAF-PNLMS. Através de simulações numéricas, a precisão do modelo proposto assim como as hipóteses e aproximações principais foram verificadas.

Um modelo estocástico para o momento de segunda ordem do vetor de erro dos coeficientes bem como a curva de aprendizagem do algoritmo adaptativo IAF-PNLMS foram propostos no Capítulo 6. Além disso, uma nova expressão de modelo descrevendo o comportamento médio do vetor de coeficientes foi obtida. No desenvolvimento das expressões de tal modelo, considera-se que a matriz de ganhos é variante no tempo e os dados de entrada são sinais gaussianos correlacionados. Esse modelo foi obtido levando em conta um novo procedimento de autodecomposição das matrizes surgidas no cálculo dos valores esperados envolvidos. Tal procedimento de autodecomposição é baseado na ordenação dos autovalores e descarte dos menos significativos. Através de simulações numéricas, a precisão do modelo proposto foi confirmada. A metodologia derivada nesse capítulo pode também ser considerada para modelar outros algoritmos da classe PNLMS.

No Capítulo 7, uma versão melhorada do algoritmo IAF-PNLMS, denominada EIAF-PNLMS, foi proposta, a qual (durante o processo de aprendizagem) usa uma estratégia de redistribuição de ganhos visando aumentar os ganhos atribuídos aos coeficientes inativos quando os coeficientes ativos aproximam-se da convergência. Para realizar tal redistribuição, define-se primeiro um limiar para indicar o ponto no qual a redistribuição de ganhos deve ocorrer. Então, quando esse limiar é atingido, a redistribuição é realizada atribuindo ganho uniforme aos coeficientes do filtro adaptativo. Resultados de simulação mostraram que as características de convergência do algoritmo são significativamente melhoradas, tal que o EIAF-PNLMS supera em desempenho os algoritmos IPNLMS e IAF-PNLMS para respostas ao impulso com elevado grau de esparsidade.

Como proposta para trabalhos futuros, visando aumentar a velocidade de convergência, sugere-se a concepção de novos algoritmos para a classe PNLMS com parâmetro de controle passo variável no tempo. Outra proposta para a continuação deste trabalho seria o desenvolvimento de modelos estocásticos para importantes algoritmos da classe PNLMS, tais como o IPNLMS e MPNLMS, usando a metodologia aqui arquitetada. Por último, porém não menos importante, um trabalho de pesquisa futuro é modelagem estocástica do algoritmo IAF-PNLMS para o caso de dados de entrada complexo.

Verificação das Condições C1 e C2

Neste apêndice, mostramos que (4.5) satisfaz as condições C1 e C2 apresentadas no Capítulo 4. Primeiramente, considere o caso em que o i -ésimo coeficiente $w_i(n)$ é ativo. Então, $\phi_i(n) = |w_i(n)|$ visto que, por definição, $a_i(n) \leq |w_i(n)|$. Assim, usando (4.1) e (4.5), a partir do instante 1 até o instante n e considerando que $\phi_i(0) = a_i(0)$, obtém-se

$$a_i(n) = \frac{1}{2} |w_i(n)| + \frac{1}{2} |w_i(n-1)|. \quad (\text{I.1})$$

Agora, assumindo que $w_i(n-1) \rightarrow w_i(n)$ quando $n \rightarrow \infty$, de (I.1) verifica-se que (4.5) satisfaz C1. Além disso, quando $w_i(n)$ é ativo, C2 é diretamente verificada de (I.1), visto que neste caso $|w_i(n)| > 0$.

A seguir, considera-se o caso em que o i -ésimo coeficiente $w_i(n)$ é inativo. Note que $\phi_i(n) = a_i(n)$ visto que, por definição, $a_i(n) > |w_i(n)|$. Então, iterando (4.1) e (4.5), a partir do instante 1 até o instante n e considerando que $\phi_i(0) = a_i(0)$, tem-se

$$\begin{aligned} a_i(n) &= \frac{1}{2^n} a_i(0) \\ &+ \left[\frac{1}{2} |w_i(n)| + \frac{1}{2^2} |w_i(n-1)| + \frac{1}{2^3} |w_i(n-2)| + \cdots + \frac{1}{2^{n-1}} |w_i(2)| + \frac{1}{2^n} |w_i(1)| \right] \\ &= \frac{1}{2^n} a_i(0) + \sum_{k=0}^{n-1} \frac{1}{2^{k+1}} |w_i(n-k)|. \end{aligned} \quad (\text{I.2})$$

Agora, assumindo que $w_i(n-1) \rightarrow w_i(n)$ quando $n \rightarrow \infty$, verifica-se que o segundo termo no lado direito de (I.2) converge para $w_i(n)$, ou seja,

$$\lim_{n \rightarrow \infty} \left[|w_i(n)| - \sum_{k=0}^{n-1} \frac{1}{2^{k+1}} |w_i(n-k)| \right] = 0. \quad (\text{I.3})$$

Já o primeiro termo de (I.2) torna-se insignificante. Dessa forma, tem-se

$$\lim_{n \rightarrow \infty} [a_i(n) - |w_i(n)|] = 0, \quad i = 1, 2, \dots, N. \quad (\text{I.4})$$

Portanto, verifica-se que, para o caso em que o coeficiente $w_i(n)$ é inativo, (4.5) satisfaz C1. Além disso, como $a_i(0) > 0$, note de (I.2) que $a_i(n) > 0$, para todo n , mostrando que (4.5) também satisfaz C2 quando $w_i(n)$ é inativo.

Provas das Propriedades P1 e P2

Neste apêndice, as propriedades P1 e P2 apresentadas no Capítulo 4 são verificadas. Para esse fim, a partir do instante $n=1$ até o instante $n=mN$, (4.1) e (4.6) são aplicadas recursivamente, atualizando o fator de ativação $a_i(n)$ dado por (4.6) somente a cada N amostras.

Primeiramente, verifica-se a propriedade P1 correspondente aos coeficientes ativos. Nesse caso, $a_i(n) \leq |w_i(n)|$; assim, obtemos $\phi_i(n)$ no instante $n=N-1$ como

$$\begin{aligned} \phi_i(N-1) &= \max \left[a_i(N-1), |w_i(N-1)| \right] \\ &= \max \left[a_i(0), |w_i(N-1)| \right] = |w_i(N-1)|. \end{aligned} \tag{II.1}$$

Agora, para $m=1$, substituindo (II.1) em (4.6), o fator de ativação $a_i(n)$ é atualizado, resultando

$$\begin{aligned} a_i(N) &= \frac{1}{2} \phi_i(N-1) + \frac{1}{2} |w_i(N)| \\ &= \frac{1}{2} |w_i(N-1)| + \frac{1}{2} |w_i(N)|. \end{aligned} \tag{II.2}$$

Substituindo-se (II.2) em (4.1), obtemos a função de proporcionalidade $\phi_i(n)$ no instante $n=N$ como

$$\begin{aligned}
\phi_i(N) &= \max [a_i(N), |w_i(N)|] \\
&= \max \left\{ \frac{1}{2} |w_i(N-1)| + \frac{1}{2} |w_i(N)|, |w_i(N)| \right\} \quad (\text{II.3}) \\
&= |w_i(N)|.
\end{aligned}$$

Repetindo o procedimento anterior até o instante $n = mN$, tem-se

$$\begin{aligned}
a_i(mN) &= \frac{1}{2} \phi_i(mN-1) + \frac{1}{2} |w_i(mN)| \\
&= \frac{1}{2} |w_i(mN-1)| + \frac{1}{2} |w_i(mN)|. \quad (\text{II.4})
\end{aligned}$$

Então, substituindo (II.4) em (4.1), obtém-se (4.8), ou seja, a propriedade P1.

Agora, verificamos a propriedade P2 correspondente aos coeficientes inativos. Nesse caso, $a_i(n) > |w_i(n)|$; assim, obtemos $\phi_i(n)$ no instante $n = N-1$ como

$$\begin{aligned}
\phi_i(N-1) &= \max [a_i(N-1), |w_i(N-1)|] \\
&= \max [a_i(0), |w_i(N-1)|] \quad (\text{II.5}) \\
&= a_i(0).
\end{aligned}$$

Assim, para $m = 1$, o fator de ativação $a_i(n)$ é atualizado, resultando

$$\begin{aligned}
 a_i(N) &= \frac{1}{2} \phi_i(N-1) + \frac{1}{2} |w_i(N)| \\
 &= \frac{1}{2} a_i(0) + \frac{1}{2} |w_i(N)|.
 \end{aligned}
 \tag{II.6}$$

Substituindo (II.6) em (4.1), obtemos a função de proporcionalidade no instante $n = N$ como

$$\begin{aligned}
 \phi_i(N) &= \max [a_i(N), |w_i(N)|] \\
 &= \max \left[\frac{1}{2} a_i(0) + \frac{1}{2} |w_i(N)|, |w_i(N)| \right] \\
 &= \frac{1}{2} a_i(0) + \frac{1}{2} |w_i(N)|.
 \end{aligned}
 \tag{II.7}$$

Repetindo o procedimento acima até o instante $n = mN$, obtemos

$$\begin{aligned}
 a_i(mN) &= \frac{1}{2} \phi_i(mN-1) + \frac{1}{2} |w_i(mN)| \\
 &= \frac{1}{2^m} a_i(0) + \frac{1}{2^m} |w_i(N)| + \frac{1}{2^{m-1}} |w_i(2N)| \\
 &\quad + \dots + \frac{1}{2^2} |w_i[(m-1)N]| + \frac{1}{2} |w_i(mN)|
 \end{aligned}
 \tag{II.8}$$

da qual segue a propriedade P2 em (4.10).

APÊNDICE III

Determinação de $\mathbf{J}(0, n)$

Neste apêndice, o método utilizado para determinar os elementos da matriz $\mathbf{J}(0, n)$ é apresentado. Primeiramente, (5.12) é reescrita como [35]

$$[\mathbf{J}(\omega, n)]_{i,k} = \frac{1}{(2\pi)^{N/2} \sqrt{\det(\mathbf{R})}} \quad (\text{III.1})$$

$$\times \underbrace{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}}_{N \text{ vezes}} \frac{x_i(n)x_k(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{x}^T(n)\Psi_1^{-1}(\omega, n)\mathbf{x}(n)}{2}} d\mathbf{x}$$

com $\Psi_1(\omega, n) = \mathbf{R}[\mathbf{I} + 2\omega\mathbf{G}(n)\mathbf{R}]^{-1}$. Agora, diferenciando (III.1) com respeito a ω , o termo $\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)$ no denominador é eliminado, resultando em

$$\frac{\partial [\mathbf{J}(\omega, n)]_{i,k}}{\partial \omega} = \frac{-1}{\sqrt{\det[\mathbf{I} + 2\omega\mathbf{G}(n)\mathbf{R}]}} A_1(\omega, n) \quad (\text{III.2})$$

onde

$$A_1(\omega, n) = \frac{1}{(2\pi)^{N/2} \sqrt{\det[\Psi_1(\omega, n)]}} \quad (\text{III.3})$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} x_i(n) x_k(n) e^{\frac{-\mathbf{x}^T(n) \Psi_1^{-1}(\omega, n) \mathbf{x}(n)}{2}} d\mathbf{x}$$

corresponde à correlação cruzada entre $x_i(n)$ e $x_k(n)$ quando essas variáveis aleatórias são conjuntamente gaussianas com matriz de covariância $\Psi_1(\omega, n)$. Visto que $A_1(\omega, n) = [\Psi_1(\omega, n)]_{i,k}$, integrando (III.2) com respeito a ω e usando a decomposição $\mathbf{G}(n)\mathbf{R} = \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{Q}^{-1}(n)$, tem-se

$$\mathbf{J}(\omega, n) = \mathbf{RQ}(n)\mathbf{H}(\omega, n)\mathbf{Q}^{-1}(n) \quad (\text{III.4})$$

onde $\mathbf{H}(\omega, n)$ é uma matriz diagonal cujos elementos são integrais hiperelípticas dadas por

$$h_i(\omega, n) = \int_{\omega} \frac{-1}{[1 + 2\omega\lambda_i(n)]\sqrt{P(\omega, n)}} d\omega + C. \quad (\text{III.5})$$

A constante C é obtida da condição $\lim_{\omega \rightarrow \infty} [\mathbf{J}(\omega, n)]_{i,k} = 0$ [veja (5.12)] e o polinômio $P(\omega, n)$ é dado por

$$P(\omega, n) = \prod_{i=1}^N [1 + 2\omega\lambda_i(n)] = a_N(n) \prod_{i=1}^N [\omega - \omega_i(n)] \quad (\text{III.6})$$

onde o coeficiente $a_N(n)$ do polinômio $P(\omega, n)$, correspondente ao termo de maior grau ω^N , é dado por $a_N(n) = 2^N \prod_{i=1}^N \lambda_i(n)$ e as raízes desse polinômio são obtidas a partir de $\omega_i(n) = -1/[2\lambda_i(n)]$ para $i = 1, 2, \dots, N$.

Para determinar a integral (III.5), uma abordagem similar àquela apresentada em [36] é usada. Assim, considere a seguinte aproximação:

$$\sqrt{P(\omega, n)} \cong \sqrt{a_N(n)} \times \prod_{q=1}^{N/2} [\omega - \omega'_q(n)], \quad q = 1, 2, \dots, \frac{N}{2} \quad \text{com } N \text{ par} \quad (\text{III.7})$$

onde $\omega'_q(n) = -\sqrt{\omega_{2q-1}(n)\omega_{2q}(n)}$ é média geométrica dos pares de raízes adjacentes de $P(\omega, n)$. Então, substituindo (III.7) em (III.5), pode-se mostrar que

$$h_i(\omega, n) \cong \frac{-1}{2[\lambda_i(n) + \zeta]\sqrt{a_N(n)}} \times \left\{ B_i(n) \ln[\omega - \omega_i(n)] + \sum_{q=1}^{N/2} A_{i,q}(n) \ln[\omega - \omega'_q(n)] \right\} \quad (\text{III.8})$$

onde $\zeta > 0$ é um parâmetro de regularização, prevenindo divisão por autovalores muito pequenos e estabilizando a solução, e $B_i(n)$ e $A_{i,q}(n)$ são, respectivamente, dados por

$$B_i(n) = \frac{1}{\prod_{q=1}^{N/2} [\omega_i(n) - \omega'_q(n)]} \quad (\text{III.9})$$

e

$$A_{i,q}(n) = \frac{1}{[\omega'_q(n) - \omega_i(n)] \prod_{\substack{j=1 \\ j \neq q}}^{N/2} [\omega'_q(n) - \omega'_j(n)]}. \quad (\text{III.10})$$

Agora, fazendo $\omega = 0$ em (III.8), $\mathbf{J}(0, n)$ é obtido como

$$\mathbf{J}(0, n) = \mathbf{R}\mathbf{Q}(n)\mathbf{H}(n)\mathbf{Q}^{-1}(n) \quad (\text{III.11})$$

com $\mathbf{H}(n) = \text{diag}[h_1(\omega, n) \ h_2(\omega, n) \ \cdots \ h_N(\omega, n)]|_{\omega=0}$.

APÊNDICE IV

Determinação de $\mathbf{j}(0, n)$

Neste apêndice, a variável auxiliar $\mathbf{j}(0, n)$ é determinada. Para tal, (5.17) é reescrita como

$$[\mathbf{j}(\omega, n)]_i = \frac{1}{(2\pi)^{N+1} \sqrt{\det(\Gamma)}} \quad (IV.1)$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{(N+1) \text{ vezes}} \frac{d(n)x_i(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)} e^{\frac{-\mathbf{u}_A^T(n)\Psi_2^{-1}(\omega, n)\mathbf{u}_A(n)}{2}} d\mathbf{u}_A$$

com

$$\Psi_2^{-1}(\omega, n) = \Gamma^{-1} + 2\omega \begin{bmatrix} \mathbf{G}(n) & 0 \\ 0 & 0 \end{bmatrix}. \quad (IV.2)$$

Diferenciando (IV.1) com respeito a ω , o termo $\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n)$ no denominador é eliminado, resultando em

$$\frac{\partial [\mathbf{j}(\omega, n)]_i}{\partial \omega} = \frac{-\sqrt{\det[\Psi_2(\omega, n)]}}{\sqrt{\det(\Gamma)}} A_2(\omega, n) \quad (IV.3)$$

onde

$$A_2(\omega, n) = \frac{(2\pi)^{\frac{N+1}{2}}}{\sqrt{\det[\Psi_2(\omega, n)]}} \quad (IV.4)$$

$$\times \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{N \text{ vezes}} d(n) x_i(n) e^{\frac{-\mathbf{u}_A^T(n) \Psi_2^{-1}(\omega, n) \mathbf{u}_A(n)}{2}} d\mathbf{u}_A$$

corresponde à correlação cruzada entre $d(n)$ e $x_i(n)$ quando $\mathbf{u}_A(n)$ é vetor gaussiano. Agora, de (IV.2), a matriz de covariância $\Psi_2(\omega, n)$ é obtida como

$$\Psi_2(\omega, n) = \begin{bmatrix} \mathbf{B}(\omega, n) \mathbf{R} & \mathbf{B}(\omega, n) \mathbf{p} \\ \mathbf{p}^T \mathbf{B}^T(\omega, n) & \sigma_d^2 - 2\omega \mathbf{p}^T \mathbf{G}(n) \mathbf{B}(\omega, n) \mathbf{p} \end{bmatrix} \quad (IV.5)$$

com $\mathbf{B}(\omega, n) = [\mathbf{I} + 2\omega \mathbf{R} \mathbf{G}(n)]^{-1}$. Dessa forma, por inspeção, obtém-se de (IV.5)

$$A_2(\omega, n) = [\mathbf{I} + 2\omega \mathbf{R} \mathbf{G}(n)]^{-1} \mathbf{p}. \quad (IV.6)$$

Além disso, de (IV.2), obtém-se

$$\det[\Psi_2(\omega, n)] = \frac{\det(\mathbf{I})}{\det[\mathbf{I} + 2\omega \mathbf{R} \mathbf{G}(n)]}. \quad (IV.7)$$

Assim, usando a decomposição $\mathbf{R} \mathbf{G}(n) = \mathbf{T}(n) \mathbf{\Lambda}(n) \mathbf{T}^{-1}(n)$, substituindo (IV.6) e (IV.7) em (IV.3), considerando que $\lim_{\omega \rightarrow \infty} [\mathbf{j}(\omega, n)]_i = 0$ [veja (5.17)] e integrando com respeito a ω , tem-se

$$\mathbf{j}(0, n) = \mathbf{T}(n) \mathbf{H}(n) \mathbf{T}^{-1}(n) \mathbf{p}. \quad (IV.8)$$

Autodecomposição das Matrizes

$$\bar{\mathbf{G}}(n)\mathbf{R} \text{ e } \mathbf{R}\bar{\mathbf{G}}(n)$$

A abordagem usada em [21] para obter o modelo da evolução dos coeficientes médios do algoritmo IAF-PNLMS leva em conta um parâmetro de regularização, prevenindo divisão por autovalores muito pequenos surgidos da autodecomposição da matriz $\bar{\mathbf{G}}(n)\mathbf{R}$. Tal abordagem estabiliza o cálculo das integrais hiperelípticas de alta ordem (HOHIs). Aqui, para assegurar o cálculo dessas integrais, usa-se uma abordagem alternativa baseada na ordenação dos autovalores de $\bar{\mathbf{G}}(n)\mathbf{R}$, descartando aqueles menos significativos. Para esse fim, primeiro ordena-se todos os autovalores da matriz $\bar{\mathbf{G}}(n)\mathbf{R}$ (em ordem decrescente). Assim, seja a autodecomposição expressa como

$$\bar{\mathbf{G}}(n)\mathbf{R} = \mathbf{Q}(n)\mathbf{\Lambda}(n)\mathbf{Q}^{-1}(n) \quad (\text{V.1})$$

com

$$\mathbf{\Lambda}(n) = \begin{bmatrix} \mathbf{\Lambda}_A(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_B(n) \end{bmatrix}. \quad (\text{V.2})$$

As matrizes $\mathbf{\Lambda}_A(n)$ e $\mathbf{\Lambda}_B(n)$ são, respectivamente, dadas por

$$\mathbf{\Lambda}_A(n) = \text{diag}[\lambda_1(n) \quad \lambda_2(n) \quad \cdots \quad \lambda_M(n)] \quad (\text{V.3})$$

e

$$\mathbf{\Lambda}_B(n) = \text{diag}[\lambda_{M+1}(n) \quad \lambda_{M+2}(n) \quad \cdots \quad \lambda_N(n)] \quad (\text{V.4})$$

com

$$\lambda_1(n) > \cdots > \lambda_M(n) > \lambda_{M+1}(n) > \cdots > \lambda_N(n). \quad (\text{V.5})$$

Em (V.3), $\lambda_M(n) \geq \gamma_L$ define o último elemento da diagonal de $\Lambda_A(n)$, onde γ_L é um limiar de magnitude para o menor autovalor significativo de $\Lambda(n)$. A partir de (V.1) e (V.2), a matriz de autovetores $\mathbf{Q}(n)$ pode ser reescrita como

$$\begin{aligned} \mathbf{Q}(n) &\triangleq [\mathbf{Q}_A(n) \quad \mathbf{Q}_B(n)] \\ &= [\mathbf{q}_1(n) \quad \cdots \quad \mathbf{q}_M(n) \quad \mathbf{q}_{M+1}(n) \quad \cdots \quad \mathbf{q}_N(n)] \end{aligned} \quad (\text{V.6})$$

onde os vetores N -dimensionais $\mathbf{q}_i(n)$, para $i=1, 2, \dots, N$, são os autovetores associados aos autovalores $\lambda_i(n)$. Agora, substituindo (V.2) e (V.6) em (V.1), obtém-se

$$\bar{\mathbf{G}}(n)\mathbf{R} = [\mathbf{Q}_A(n) \quad \mathbf{Q}_B(n)] \begin{bmatrix} \Lambda_A(n) & \mathbf{0} \\ \mathbf{0} & \Lambda_B(n) \end{bmatrix} \begin{bmatrix} \mathbf{P}_A(n) \\ \mathbf{P}_B(n) \end{bmatrix} \quad (\text{V.7})$$

com $[\mathbf{P}_A^T(n) \quad \mathbf{P}_B^T(n)]^T \triangleq \mathbf{Q}^{-1}(n)$. Então, assumindo que $\gamma_L \ll 1$, resulta em

$$\bar{\mathbf{G}}(n)\mathbf{R} \cong \mathbf{Q}_A(n)\Lambda_A(n)\mathbf{P}_A(n) \quad (\text{V.8})$$

onde a matriz $\mathbf{P}_A(n)$ é formada pelas primeiras M linhas de $\mathbf{Q}^{-1}(n)$.

Agora, utilizando o mesmo procedimento descrito anteriormente para obter a autodecomposição da matriz $\mathbf{R}\bar{\mathbf{G}}(n)$, tem-se

$$\mathbf{R}\bar{\mathbf{G}}(n) = \mathbf{T}(n)\Lambda(n)\mathbf{T}^{-1}(n) \quad (\text{V.9})$$

$$= [\mathbf{T}_A(n) \quad \mathbf{T}_B(n)] \begin{bmatrix} \Lambda_A(n) & \mathbf{0} \\ \mathbf{0} & \Lambda_B(n) \end{bmatrix} \begin{bmatrix} \mathbf{U}_A(n) \\ \mathbf{U}_B(n) \end{bmatrix}$$

com

$$\begin{aligned} \mathbf{T}(n) &\triangleq [\mathbf{T}_A(n) \quad \mathbf{T}_B(n)] \\ &= [\mathbf{t}_1(n) \quad \cdots \quad \mathbf{t}_M(n) \quad \mathbf{t}_{M+1}(n) \quad \cdots \quad \mathbf{t}_N(n)] \end{aligned} \tag{V.10}$$

e $[\mathbf{U}_A^T(n) \quad \mathbf{U}_B^T(n)]^T \triangleq \mathbf{T}^{-1}(n)$. Dessa forma, obtém-se

$$\mathbf{R}\bar{\mathbf{G}}(n) \cong \mathbf{T}_A(n)\Lambda_A(n)\mathbf{U}_A(n) \tag{V.11}$$

onde a matriz $\mathbf{U}_A(n)$ representa as primeiras M linhas de $\mathbf{T}^{-1}(n)$. Salienta-se aqui que a utilização do limiar γ_L (definido nesse apêndice) não degrada a eficácia do modelo proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. New York: Springer-Verlag, 2006.
- [2] 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.
- [3] 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.
- [4] Y. Gu, J. Jin, and S. Mei, " l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Process. Lett.*, vol. 16, no. 3, pp. 774-777, Sep. 2009.
- [5] 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.
- [6] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancellers," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508-518, Sep. 2000.
- [7] 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.

- [8] 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.
- [9] 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.
- [10] 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.
- [11] P. A. Naylor, J. Cui, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Process.*, vol. 86, no. 6, pp. 1182-1192, Jun. 2006.
- [12] 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.
- [13] J. Cui, P. A. Naylor, and D. T. Brown, "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.
- [14] 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.

- [15] S. Amari, “Natural gradient works efficiently in learning,” *Neural Comput.*, vol. 10, no. 2, pp. 251–276, Feb. 1998.
- [16] 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.
- [17] 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. 1405-1408.
- [18] 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.
- [19] 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.
- [20] K. T. Wagner, M. I. 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.
- [21] K. T. Wagner, M. I. 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.

- [22] A. W. H. Khong, P. A. Naylor, and J. Benesty, "A low delay and fast converging improved proportionate algorithm for sparse system identification," *EURASIP J. Audio, Speech, Music Process.*, vol. 2007, no. 1, Article ID 84376, pp. 1-8, 2007.
- [23] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learning Research*, vol. 5, pp. 1457-1469, Nov. 2004.
- [24] ITU-T Recommendation G.168, *Digital Network Echo Cancellers*, Int. Telecomm. Union, Geneva, Switzerland, 2007.
- [25] 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.
- [26] 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.
- [27] J. H. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," *IEEE Comm. Magazine*, pp. 36-41, Jul. 2004.
- [28] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [29] S. Haykin, *Adaptive Filter Theory*, 3^a ed. Englewood Cliffs, NJ. Ed. Prentice-Hall, 1996.

- [30] B. Widrow, "Thinking about thinking: The discovery of the LMS algorithm," *IEEE Signal Process. Magazine*, vol. 22, pp. 100-106, Jan. 2005.
- [31] I. S. Reed, "Brief history of adaptive arrays," *IEEE Military Comm. Conf.*, Boston, MA, Oct. 1985, pp. 515-518.
- [32] M. M. Sondhi, "The history of echo cancellation," *IEEE Signal Process. Magazine*, vol. 23, pp. 95-102, Sept. 2006.
- [33] F. 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.
- [34] F. 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.
- [35] M. Rupp, "The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1149-1160, Mar. 1993.
- [36] E. M. Lobato, O. J. Tobias, and R. Seara, "Stochastic modeling of the transform-domain ϵ LMS algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1840-1852, May 2008.
- [37] J. E. Kolodziej, O. J. Tobias, R. Seara, and D. R. Morgan, "On the constrained stochastic gradient algorithm: Model, performance, and improved version," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1304-1315, Apr. 2009.

- [38] J. E. Kolodziej, O. J. Tobias, and R. Seara, "Stochastic analysis of the transform domain LMS algorithm for a non-stationary environment," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Glasgow, U.K., Aug. 2009, pp. 1730-1734.
- [39] M. I. Doroslovacki and H. Deng, "On convergence of proportionate-type NLMS adaptive algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Toulouse, France, May 2006, vol. 3, pp. 105-108.
- [40] K. T. Wagner and M. I. Doroslovacki, "Towards analytical convergence analysis of proportionate-type NLMS algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Las Vegas, NV, Mar. 2008, pp. 3825-3828.
- [41] F. C. de Souza, O. J. Tobias, and R. Seara, "Considerações sobre o algoritmo PNLMS com fatores de ativação individuais," in *XXVII Simpósio Brasileiro de Telecomunicações (SBrT 2009)*, Blumenau-SC, Brazil, Oct. 2009.
- [42] S. Haykin, "Statistical learning theory of the LMS algorithm under slowly varying conditions, using the Langevin equation," in: *Proc. 40th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 2006, pp. 229-232.
- [43] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, Hoboken, 2009.
- [44] C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, 1992

- [45] F. C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, "Stochastic model for the mean weight evolution of the IAF-PNLMS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5895-5901, Nov. 2010.
- [46] F. C. de Souza, R. Seara, "Considerações sobre a modelagem estocástica dos coeficientes do algoritmo adaptativo IAF-PNLMS," in *XXIX Simpósio Brasileiro de Telecomunicações (SBrT 2011)*, Curitiba-PR, Brazil, Oct. 2011.
- [47] C. Paleologu, S. Ciochină, 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.
- [48] C. Paleologu, J. Benesty, and S. Ciochina, *Sparse Adaptive Filters for Echo Cancellation*. San Rafael, CA: Morgan and Claypool Publishers, 2010.
- [49] F. C. de Souza, R. Seara, and D. R. Morgan, "An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3301-3307, Jun. 2012.
- [50] 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.

- [51] 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. 3233-3236.
- [52] K. T. Wagner and M. I. Doroslovacki, "On proportionate-type NLMS algorithms for fast decay of output error at all times," in *Proc. 43rd Asilomar Conf. Signals, Syst. and Comput.*, Pacific Grove, CA, Nov. 2009, pp. 186-190.