

UNIVERSIDADE FEDERAL DO MARANHÃO
PRÓ-REITORIA DE PÓS-GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
AUTOMAÇÃO E CONTROLE

ALZENEIDE DUTRA PESSOA

**Algoritmo ZA-PNLMS com fatores de ativação
individuais e ganhos dos coeficientes limitados
superiormente**

SÃO LUÍS, MA
2023

ALZENEIDE DUTRA PESSOA

**Algoritmo ZA-PNLMS com fatores de ativação
individuais e ganhos dos coeficientes limitados
superiormente**

Dissertação submetida ao Programa
de Pós-Graduação em Engenharia
Elétrica da Universidade Federal do
Maranhão para a obtenção do grau
de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Francisco das Chagas de Souza

SÃO LUÍS, MA
2023

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Dutra Pesoa, Alzeneide.

Algoritmo ZA-PNLMS com fatores de ativação individuais
e ganhos dos coeficientes limitados superiormente /
Alzeneide Dutra Pesoa. - 2023.

90 p.

Orientador(a): Francisco das Chagas de Souza.

Dissertação (Mestrado) - Programa de Pós-graduação em
Engenharia Elétrica/ccet, Universidade Federal do
Maranhão, São Luís - MA, 2023.

1. Algoritmo adaptativo. 2. Atrator para zero. 3.
Fatores de ativação. 4. Ganhos limitados superiormente.
5. Plantas esparsas. I. das Chagas de Souza, Francisco.
II. Título.

**Algoritmo ZA-PNLMS com fatores de ativação
individuais e ganhos dos coeficientes limitados
superiormente**

ALZENEIDE DUTRA PESSOA

Dissertação aprovada em 30/08/2023.

Prof. Francisco das Chagas de Souza, Dr.

(Orientador – UFMA)

Prof. Ewaldo Eder Carvalho Santana, Dr.

(Membro da Banca Examinadora – UEMA)

Profa. Patrícia Helena Moraes Rêgo, Dra.

(Membro da Banca Examinadora – UEMA)

DEDICATÓRIA

Dedico este trabalho a minha família, em especial, a minha mãe, Francisca Dutra, por todo o incentivo durante esta caminhada.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter me sustentado até aqui e por ter me dado forças para vencer cada obstáculo que surgiu durante esta jornada.

Agradeço a minha mãe, Francisca Dutra, que mesmo com pouca instrução me mostrou que a Educação é o melhor caminho. Sempre esteve comigo me incentivando a estudar e a nunca desistir dos meus sonhos. Agradeço também aos meus pais, João Pessoa e José Santana, por todo o amor, cuidado, e sacrifícios que fizeram de bom grado para que eu pudesse chegar até aqui.

Aos meus irmãos que sempre me apoiaram. As minhas irmãs, Adriana Dutra pelo cuidado e preocupação e, a Alcineide Dutra por todo o incentivo, companheirismo e ajuda durante todo este trabalho. Deixo também um agradecimento especial ao meu esposo Ayrton Santos que soube me compreender nos momentos em que eu estive ausente e que me incentivou a concluir essa etapa importante da minha vida.

Aos meus filhos, Nicolas e Marcos, que são a minha motivação e inspiração diária. Aos meus amigos, minha família, que estiveram presente me motivando a continuar, em especial, Priscila, Abigail e Gleison.

Meus sinceros agradecimentos ao meu orientador, o Prof. Dr. Francisco das Chagas de Souza, por toda dedicação, paciência, compreensão e ensinamentos valiosos. Não teria conseguido sem seus conselhos, dicas, opiniões e explicações. Meu muito obrigado!

Agradeço também à Universidade Federal do Maranhão, em especial ao Programa de Pós-Graduação em Engenharia de Eletricidade pela oportunidade. À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), à Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

RESUMO

Algoritmos de filtragem adaptativa têm sido bastante difundidos no meio acadêmico. Um desafio importante nessa área é a identificação de plantas esparsas. Com o objetivo de identificar plantas com diferentes graus de esparsidade é que neste trabalho propõe-se um algoritmo adaptativo LMS normalizado proporcional com atrator para zero (ZAPNLMS – zero-attracting proportionate normalized LMS algorithm), o qual combina fatores de ativação individuais com ganhos limitados superiormente. A proposta aqui apresentada limita superiormente os ganhos dos coeficientes do algoritmo para identificação de plantas com alto grau de esparsidade, levando a uma melhor distribuição da energia de adaptação entre os coeficientes do algoritmo. Simulações computacionais, considerando sistemas esparsos com perturbação e rastreamento, atestam que o algoritmo proposto é capaz de aglutinar as características de boa convergência em estado transiente e baixo erro em estado permanente.

Palavras-chave: Fatores de ativação; algoritmo adaptativo; plantas esparsas; ganhos limitados superiormente; atrator para zero.

ABSTRACT

Adaptive filtering algorithms have been widespread in academia. An important challenge in this area is the identification of sparse plants. With the aim of identifying plants with different sparsity degrees, this work proposes an adaptive LMS proportional normalized algorithm with zero attractor (ZA-PNLMS – zero-attracting proportionate normalized LMS algorithm), which combines individual activation factors with earnings limited from above. The proposal presented here superiorly limits the gains of the algorithm's coefficients for identifying plants with a high degree of sparsity, leading to a better distribution of the adaptation energy between the algorithm's coefficients. Computational simulations, considering sparse systems with perturbation and tracking, attest that the proposed algorithm is capable of agglutinating the characteristics of good convergence in transient state and low error in steady state.

Keywords: Activation factors; adaptive algorithm; sparse plants; upper limited gains; zero-attracted.

LISTA DE FIGURAS

Figura 1.1 - Filtro adaptativo aplicado ao problema de identificação de sistemas.....	3
Figura 1.2 - Exemplo de planta esparsa com $n = 100$ e $S(\mathbf{p}) = 0,9343$	4
Figura 2.1 - Esquema de filtragem adaptativa.....	8
Figura 2.2 - Representação em diagrama de blocos de um sistema de tempo discreto...	9
Figura 2.3 - Realização do filtro FIR (forma direta)..	14
Figura 2.4 - Realização do sistema de todos os polos (forma direta).....	15
Figura 2.5 - Realização do sistema polo zero (forma direta).....	15
Figura 2.6 - Sistema dinâmico desconhecido (a). Gráfico de fluxo de sinal do modelo adaptativo para o sistema (b). O gráfico incorpora um ciclo de feedback..	17
Figura 2.7 - Trajetória do método de descida mais íngreme em um espaço bidimensional para dois valores diferentes referentes ao tamanho do passo: (a) μ pequeno (b) μ grande. As coordenadas w_1 e w_2 são elementos do vetor de peso \mathbf{w} ; ambos estão no plano W	22
Figura 2.8 - Gráfico de fluxo de sinal representação do algoritmo LMS. O gráfico incorpora <i>feedback</i> representado na parte hachurada.....	31
Figura 4.1 - Curva teórica do erro indicando os regimes transientes e permanente.....	52
Figura 5.1 - Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando os seguintes valores de parametro $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0,95$, e $\beta_1 = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$	54
Figura 5.2 - Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0,95$, $\beta_1 = 1$, $\beta_2 = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$	55
Figura 5.3 - Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$, $\lambda_e = 0,95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$	56
Figura 5.4 - Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0,95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = 0,01$, $\epsilon = 0.01$ e $f_i(0) = \{10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ (para o algoritmo	

- proposto) e $\rho_g = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$ (para UT-ZA-PNLMS).. 57
- Figura 5.5 - Curvas de desalinhamento normalizado, $\eta(n)$, para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$ 59
- Figura 5.6 - Estimaco dos coeficientes da planta **p** que logo aps $n=2500$ sofre uma perturbao e passa a ser **-p**. (a) coeficientes ativos $\{1, 0.1\}$ nas posies $\{30, 85\}$. (b) coeficientes ativos $\{-1, -0.1\}$ nas posies $\{30, 85\}$ 60
- Figura 5.7 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 60
- Figura 5.8 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 61
- Figura 5.9 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 62
- Figura 5.10 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 63
- Figura 5.11 - Estimaco dos coeficientes da planta **p** que logo aps $n=2500$ sofre uma perturbao e passa a ser **-p**. (a) coeficientes ativos $\{0.1, 1.0, -0.5, 0.1\}$ nas posies $\{1, 30, 35, 85\}$. (b) coeficientes ativos $\{-0.1, -1.0, 0.5, -0.1\}$ nas posies $\{1, 30, 35, 85\}$ 63
- Figura 5.12 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 64
- Figura 5.13 - Estimaco dos coeficientes da planta **p** que logo aps $n=2500$ sofre uma perturbao e passa a ser **-p**. (a) coeficientes ativos $\{0.1, 1.0, -0.5, 0.1, -0.1\}$ nas posies $\{1, 30, 35, 85, 95\}$. (b) coeficientes ativos $\{-0.1, -1.0, 0.5, -0.1, 0.1\}$ nas posies $\{1, 30, 35, 85, 95\}$ 65
- Figura 5.14 - Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$,

$\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 65

Figura 5.15 - Comportamento de $w_{30}(n)$ para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\varepsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 66

Figura 5.16 - Comportamento de $w_{30}(n)$ para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\varepsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 67

Figura 5.17 - Comportamento de $w_{30}(n)$ para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\varepsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 67

Figura 5.18 - Curvas de desalinhameno normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\varepsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$ 68

LISTA DE TABELAS

Tabela 2.1 - Sumário do algoritmo adaptativo LMS.	30
Tabela 2.2 - Sumário do algoritmo adaptativo NLMS.	33
Tabela 3.1 - Sumário do algoritmo adaptativo PNLMS padrão.	35
Tabela 3.2 - Sumário do algoritmo adaptativo IPNLMS.	38
Tabela 4.1 - Sumário do algoritmo adaptativo UT-ZA-PNLMS.....	48
Tabela 4.2 - Sumário do algoritmo proposto IAF-UT-ZA-PNLMS.....	50
Tabela 5.1 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro β_1 sofre variação.....	54
Tabela 5.2 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro β_2 sofre variação.....	55
Tabela 5.3 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro λ_d sofre variação..	56
Tabela 5.4 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro $f_i(0)$ sofre variação..	58
Tabela 5.5 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro ρ_g sofre variação..	58
Tabela 5.6 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro ρ_g sofre variação..	68

LISTA DE ALGORITMOS

LMS - <i>least-mean-square</i>	16
NLMS - <i>normalized least-mean-square</i>	31
PNLMS - <i>proportionate normalized least-mean-square</i>	35
IPNLMS - <i>improved proportionate normalized least-mean-square</i>	36
IAF-PNLMS - <i>individual activation factors proportionate normalized least-mean-square</i>	39
ZA-LMS - <i>zero-attracting LMS</i>	40
RZA-LMS - <i>reweighted zero-attracting LMS</i>	41
ZA-PNLMS - <i>zero-attracting proportionate normalized least-mean-square</i>	42
UT-ZA-PNLMS - <i>upper threshold based ZA-PNLMS</i>	46
IAF-UT-ZA-PNLMS - <i>algoritmo proposto</i>	49

SUMÁRIO

Capítulo 1 Introdução	1
1.1 IDENTIFICAÇÃO DE SISTEMAS	2
1.2 PLANTAS ESPARSAS	3
1.3 MOTIVAÇÃO	4
1.4 OBJETIVOS DA PESQUISA	5
1.4.1 Objetivo Geral	5
1.4.2 Objetivos Específicos	5
1.5 TRABALHOS PUBLICADOS	6
1.6 ORGANIZAÇÃO DA DISSERTAÇÃO	6
Capítulo 2 Filtros Adaptativos LMS e LMS Normalizado	8
2.1 FILTROS DISCRETOS NO TEMPO	9
2.1.1 Sistemas Lineares Invariantes no Tempo	9
2.1.2 Sistemas discretos por equações lineares de diferença de coeficiente constante	12
2.1.3 Resposta a Entradas Periódicas	15
2.2 ALGORITMO LMS	16
2.3 ESTRUTURA DE FILTRAGEM DO ALGORITMO LMS	16
2.4 OTIMIZAÇÃO SEM RESTRIÇÕES: UMA REVISÃO	19
2.4.1 Método da descida mais íngreme	20
2.4.2 Método de Newton	22
2.4.3 Método Gauss-Newton	24
2.5 O FILTRO WIENER	26
2.5.1 Filtro Wiener: Forma Limitante do Filtro de Mínimos Quadrados para um Ambiente Ergódico	27
2.6 O ALGORITMO DE MÍNIMOS QUADRADOS MÉDIO – LMS	28
2.6.1 Representação gráfica de fluxo de sinal do algoritmo LMS	30
2.7 ALGORITMO LMS NORMALIZADO - NLMS	31
Capítulo 3 Algoritmos Proporcionais e <i>Zero-Attracting</i>	34

3.1 ALGORITMOS PROPORCIONAIS	34
3.1.1 Algoritmo PNLMS	35
3.1.2 Algoritmo IPNLMS	36
3.1.3 Algoritmo IAF-PNLMS	38
3.2 ALGORITMOS DA FAMÍLIA <i>ZERO- ATTRACTING</i>	40
3.2.1 Algoritmo ZA-LMS	40
3.2.2 Algoritmo RZA-LMS	41
3.2.3 Algoritmo ZA-PNLMS	42
Capítulo 4 Metodologia Proposta e Métricas de Avaliação	44
4.1 MÉTODO DE ESTIMAÇÃO DE PARÂMETROS MÁXIMA VEROSSIMILHANÇA	44
4.2 ALGORITMO UT-ZA-PNLMS	46
4.3 METODOLOGIA PROPOSTA	49
4.4 MÉTRICAS DE AVALIAÇÃO	51
Capítulo 5 Simulações Computacionais	53
5.1 AVALIAÇÃO COMPARATIVA ENTRE OS ALGORITMOS UT-ZA-PNLMS E IAF-UT-ZA-PNLMS.....	53
5.2 AVALIAÇÃO DOS ALGORITMOS NA IDENTIFICAÇÃO DE DIFERENTES PLANTAS ESPARSAS.....	59
5.2.1 Exemplo 1	59
5.2.2 Exemplo 2	63
5.2.3 Exemplo 3	64
5.4.1 Exemplo 4	66
5.3 AVALIAÇÃO DOS ALGORITMOS MEDIANTE A MUDANÇAS NA VARIÂNCIA DO RUÍDO.....	68
Capítulo 6 Considerações Finais	70
6.1 CONCLUSÃO	70
6.2 TRABALHOS FUTUROS	71
REFERÊNCIAS	72

Capítulo 1

Introdução

Entre as técnicas de processamento digital de sinais, os filtros adaptativos têm se destacado, principalmente porque apresentam bom desempenho na solução de diversos problemas (Müller et al., 2006; Bakri et al., 2020). O termo “filtro” pode ser empregado para apresentar um dispositivo físico ou computacional que colete uma série de dados aleatórios e processa-os segundo um conjunto de regras pré-definidas (Haykin, 2014; Fonseca, 2022).

Filtros adaptativos, são aqueles capazes de alterarem os valores de seus respectivos parâmetros com o tempo (Farhang-Boroujeny, 2013). Atualmente estes vêm sendo utilizados em diversas aplicações, das quais se pode destacar a identificação de sistemas, a equalização de canais, a predição de sinais, o realçamento de sinais e o cancelamento de eco (Müller et al., 2006).

No que tange à aplicação de identificação de sistemas, em especial à sistemas com alto grau de esparsidade, os algoritmos adaptativos clássicos como o LMS (*least-mean-square*) e LMS normalizado (NLMS – *normalized least-mean-square*) apresentam desempenho pobre em termo de velocidade de convergência. Por outro lado, o algoritmo LMS normalizado proporcional (PNLMS – *proportionate normalized least-mean-square*) tem demonstrado desempenho superior ao LMS e ao NLMS (Gay, 1998; Gay, 2002; Huang, 2006; Paleologu et al., 2010)

A fim de aprimorar o desempenho do PNLMS, vários algoritmos adaptativos foram propostos, surgindo então uma classe de algoritmos PNLMS, tais como o PNLMS aprimorado (IPNLMS – *improved proportionate normalized least-mean-square*) (Benesty et al., 2002) e o PNLMS com fatores de ativação individuais (IAF-PNLMS – *individual-activation-factor proportionate normalized least-mean-square*) (Souza et al., 2010). No PNLMS, o fator de ativação é o mesmo para todos os coeficientes e depende da norma infinita instantânea do vetor coeficientes do filtro adaptativo. Já no IAF-PNLMS, é utilizados um fator de ativação para cada coeficiente do filtro (Souza et al., 2010).

Outro algoritmo da classe PNLMS foi introduzido por (Das, 2016) e denominado PNLMS com atração para zero (ZA-PNLMS – *zero-attracting proportionate normalized least-mean-square*). A proposta do ZA-PNLMS é que uma norma l_1 do vetor de coeficientes do filtro seja introduzida como um termo de penalidade na função de custo do algoritmo. Das (2020), além da introdução de um termo de penalidade, propôs que a matriz de ganhos utilizada na adaptação dos coeficientes seja calculada considerando uma função de limite superior adaptativa, concebendo assim o algoritmo denominado ZA-PNLMS com limite superior (UT-ZA-PNLMS – *upper threshold based ZA-PNLMS*).

Entre os algoritmos da classe PNLMS destaca-se, para efeito deste estudo, o IAF-PNLMS e o UT-ZA-PNLMS, pois segundo Souza *et al.* (2010) o primeiro (IAF-PNLMS) destaca-se pela boa convergência em estado transitório. Já o segundo (UT-ZA-PNLMS) tem como característica principal o erro alcançado em estado estacionário (Das, 2020). A ideia central do desenvolvimento desse trabalho consiste em propor um algoritmo que reúna as características citadas do algoritmo IAF-PNLMS e do UT-ZA-PNLMS.

Nesse trabalho então, é proposto um algoritmo que combina as melhores características dos filtros IAF-PNLMS e UT-ZA-PNLMS, a fim de se obter uma melhoria em termos de convergência e um menor erro em estado estacionário. A análise do algoritmo proposto é realizada com base em simulações numéricas variando os diversos parâmetros inerentes à proposta do algoritmo.

1.1 IDENTIFICAÇÃO DE SISTEMAS

Entre os problemas nos quais se utiliza filtros adaptativos está a identificação de sistemas (Müller *et al.*, 2006). O desafio de identificar um sistema torna-se ainda mais desafiador quando este possui alto grau de esparsidade, ou seja, poucos coeficientes diferentes de zero (Widrow *et al.*, 1985).

A Figura 1 mostra a aplicação de um filtro adaptativo na resolução de um problema de identificação de sistemas. Nesta figura, $x(n)$ é o sinal de entrada da planta, $d(n)$ é a saída da planta (ou o sinal desejado), $y(n)$ é a saída do filtro adaptativo, $e(n)$ é o sinal de erro e $v(n)$ é o ruído de medição. Os vetores $\mathbf{p}(n)$ e $\mathbf{w}(n)$ representam, respectivamente, as respostas impulsivas da planta e do filtro adaptativo. Este último, por sua vez, usa o algoritmo adaptativo A, o qual possui como entradas $e(n)$ e $x(n)$, para modificar, a cada instante de

tempo, n , a resposta impulsiva do filtro, $\mathbf{w}(n)$. O objetivo destas modificações iterativas é aproximar $e(n)$ de zero e, conseqüentemente, tornar $y(n)$ uma boa estimativa de $d(n)$.

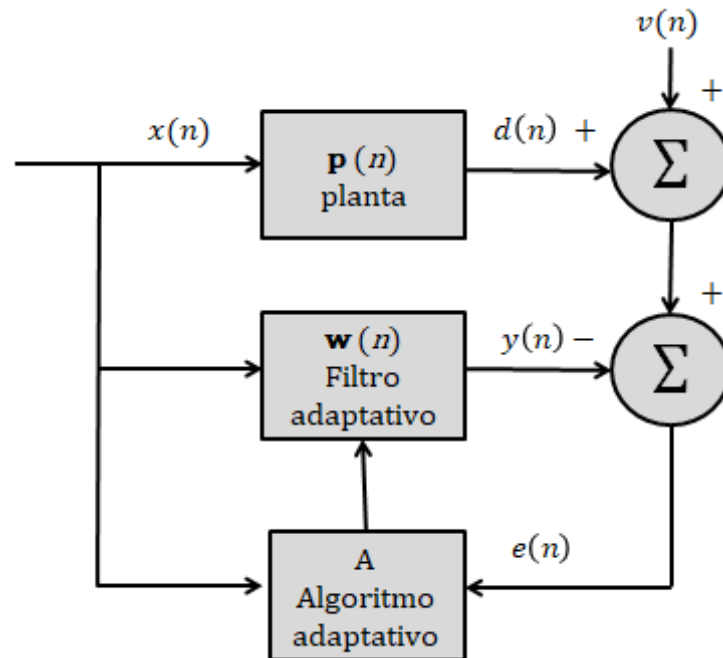


Figura 1.1 – Filtro adaptativo aplicado ao problema de identificação de sistemas.

Neste tipo de aplicação escolhe-se, com o apoio de algum conhecimento a priori da planta a ser identificada $\mathbf{p}(n)$, um filtro adaptativo $\mathbf{w}(n)$, com uma dada quantidade de parâmetros que podem ser ajustados pelo algoritmo adaptativo, A, de forma que a diferença entre a saída da planta, $d(n)$, e a do filtro adaptativo, $y(n)$, seja minimizada.

1.2 PLANTAS ESPARSAS

Algumas aplicações, principalmente, as que envolvem matrizes de alta ordem, percebe-se que a proporção de elementos não nulos tende a ser muito pequena em relação à ordem da matriz (ou vetor), ou seja, a quantidade de elementos iguais ou próximos a zero é relativamente maior do que os não nulos. Quando uma matriz (ou vetor) apresenta essa característica define-se a mesma como matriz esparsa.

Entre as aplicações nas quais se observa problemas com plantas esparsas estão à estimação de componentes harmônicos em sistemas elétricos de potência, o cancelamento de eco em telecomunicações, a estimação de canais de comunicação subaquáticos e identificação de eventos sísmicos (Branco, 2016).

Dado um vetor $\mathbf{p} = [p_1, p_2, p_3, \dots, p_n]^T$, com o subscrito T denotando transposição, o grau de esparsidade $S(\mathbf{p})$ de \mathbf{p} é calculado da seguinte forma (Paleologu; Benesty; Ciochina, 2010):

$$S(\mathbf{p}) = \frac{n}{n - \sqrt{n}} \left(1 - \frac{\|\mathbf{p}\|_1}{\sqrt{n} \|\mathbf{p}\|_2} \right), \quad (1.1)$$

em que $\|\mathbf{p}\|_1$ e $\|\mathbf{p}\|_2$ são a norma-1 e norma-2 respectivamente. Quanto maior for a quantidade de termos não nulos em \mathbf{p} mais próximo de 1 estará $S(\mathbf{p})$ e mais esparsa será o vetor. Na Figura 1.2 está exemplificada uma planta esparsa com grau de esparsidade igual a 0,9343, cujos coeficientes não nulos $\{0,1; 1,0; -0,5; 0,1; -0,1\}$ se encontram respectivamente nas posições $\{1, 30, 35, 85, 95\}$.

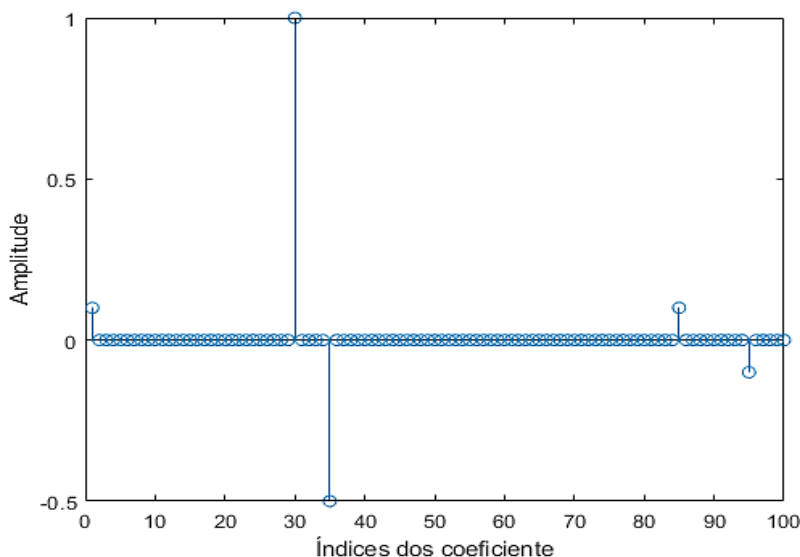


Figura 1.2 – Exemplo de planta esparsa com $n = 100$ e $S(\mathbf{p}) = 0,9343$.

Algoritmos utilizados para identificar plantas como a ilustrada na Figura 1.2 é que serão objetos dessa pesquisa. A motivação para a realização desses estudos está descrita no subtópico a seguir.

1.3 MOTIVAÇÃO

O algoritmo adaptativo mais básico é o LMS (Haykin, 1996). Entretanto, apesar de simples e eficaz, o mesmo possui uma convergência lenta quando a resposta ao impulso é de natureza esparsa (Duttweiler, 2000). A fim de superar essa desvantagem, novos algoritmos foram sendo criados como, por exemplo, o algoritmo PNLMS.

O algoritmo PNLMS é conhecido por produzir uma taxa de convergência inicial mais rápida, fornecendo um maior ganho para coeficientes ativos em comparação com os inativos (Duttweiler, 2000). Embora o PNLMS apresente uma convergência inicial mais rápida, tem sido observado que sua taxa de convergência reduz-se conforme avançam as iterações, podendo até ser menor do que a do NLMS na fase posterior do processo de adaptação (Das, 2020).

Durante o processo de revisão da literatura percebeu-se que o algoritmo UT-ZA-PNLMS proposto por Das (2020) melhora o desempenho, em estado estacionário, do algoritmo ZA-PNLMS através de um limiar superior para os ganhos dos coeficientes, enquanto que o algoritmo baseado em fatores de ativação individuais, IAF-PNLMS, possui uma convergência melhor do que o PNLMS. Essas duas características observadas no desempenho dos filtros mencionados motivaram a proposta desse trabalho de combinar os algoritmos UT-ZA-PNLMS e IAF-PNLMS a fim de se obter um novo filtro que apresentasse as melhores características daqueles que o compõe.

1.4 OBJETIVOS DA PESQUISA

Os objetivos dessa pesquisa estão divididos em gerais e específicos, sendo apresentados a seguir.

1.4.1 Objetivo Geral

Combinar em um único algoritmo as filosofias dos algoritmos UT-ZA-PNLMS e IAF-PNLMS, utilizando um limite superior adaptativo para o cálculo dos ganhos do filtro e introduzindo o uso dos fatores de ativação determinados de maneira individual para cada coeficiente.

1.4.2 Objetivos Específicos

- Propor uma função de custo para cálculo dos melhores fatores de ativação individuais e utilizá-la no algoritmo UT-ZA-PNLMS;
- Realizar testes com diferentes valores para os parâmetros do filtro a fim de observar como o novo algoritmo se comporta;
- Analisar o desempenho do algoritmo proposto antes e depois de perturbações ocorridas no vetor de coeficientes e variações na esparsidade da planta;

- Analisar, em termos de erro e velocidade de convergência, o desempenho do algoritmo proposto considerando modificações na variância do ruído.

A seguir, são mostrados os trabalhos publicados a partir desta pesquisa.

1.5 TRABALHOS PUBLICADOS

Durante a execução dessa pesquisa foi publicado e apresentado o seguinte artigo no XXIV Congresso Brasileiro de Automática - CBA 2022

- 1) Pessoa, A. D.; Souza, F. C. (2022) Algoritmo ZA-PNLMS com Fatores de Ativação Individuais e Ganhos dos Coeficientes Limitados Superiormente. XXIV Congresso Brasileiro de Automática - CBA 2022, Fortaleza - CE, Brasil, 16 a 19 de outubro de 2022.

1.6 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho de pesquisa está organizado da seguinte forma:

- Capítulo 2: apresenta alguns aspectos fundamentais da teoria de filtragem adaptativa com foco no algoritmo LMS e LMS normalizado. Este capítulo descreve um filtro linear de tempo discreto de resposta ao impulso de duração finita, revisa duas técnicas de otimização irrestrita: o método da descida mais íngreme e o método de Newton e apresenta a derivação do algoritmo LMS.
- Capítulo 3: tem como foco a apresentação dos algoritmos proporcionais e os com atração para zero. É nesse capítulo que um dos algoritmos que dão sustentação teórica para essa proposta é apresentados, a saber, o IAF-PNLMS. Primeiramente, são mostradas as principais definições e, em seguida, as respectivas construções matemáticas do mesmo. Para fins didáticos este capítulo foi dividido em dois tópicos importantes. No primeiro, são apresentados os conceitos pertinentes aos algoritmos proporcionais e no segundo tópico está à teoria dos algoritmos com atração para zero.
- Capítulo 4: aborda a metodologia proposta. Inicialmente o algoritmo UT-ZA-PNLMS é apresentado, pois o mesmo está diretamente ligado à proposta. Para tal, uma introdução teórica sobre estimação de parâmetros é apresentado, dando ênfase ao método da máxima verossimilhança, pois este algoritmo tem sua construção baseada

no referido método. Em seguida é exposta a formulação teórico-matemática do algoritmo proposto e, as métricas de avaliação são apresentadas.

- Capítulo 5: os testes computacionais são apresentados a fim de avaliar o algoritmo apresentado. As simulações aqui apresentadas estão organizadas em três grupos. No primeiro avaliou-se o impacto que a variação dos principais parâmetros causa nos algoritmos. No segundo grupo, avaliou-se a capacidade do algoritmo proposto em identificar plantas com diferentes graus de esparsidade. Por fim, no terceiro grupo verificou-se a resposta do algoritmo quanto a variação do ruído adicionado ao sinal de entrada.
- Capítulo 6: apresenta as conclusões deste trabalho de pesquisa, além de indicar possíveis sugestões de trabalhos futuros.

Capítulo 2

Filtros Adaptativos LMS e LMS

Normalizado

Um filtro adaptativo é um sistema que depende, para sua operação, de um algoritmo recursivo, o que possibilita ao mesmo funcionar satisfatoriamente em um ambiente onde o conhecimento completo das características relevantes do sinal não está disponível. Em um ambiente estacionário, após sucessivos ciclos de adaptação do algoritmo, ele converge para a solução ótima. Em um ambiente não estacionário, o algoritmo oferece uma capacidade de rastreamento (Haykin, 2014). A Figura 2.1 ilustra um esquema de filtragem adaptativa.

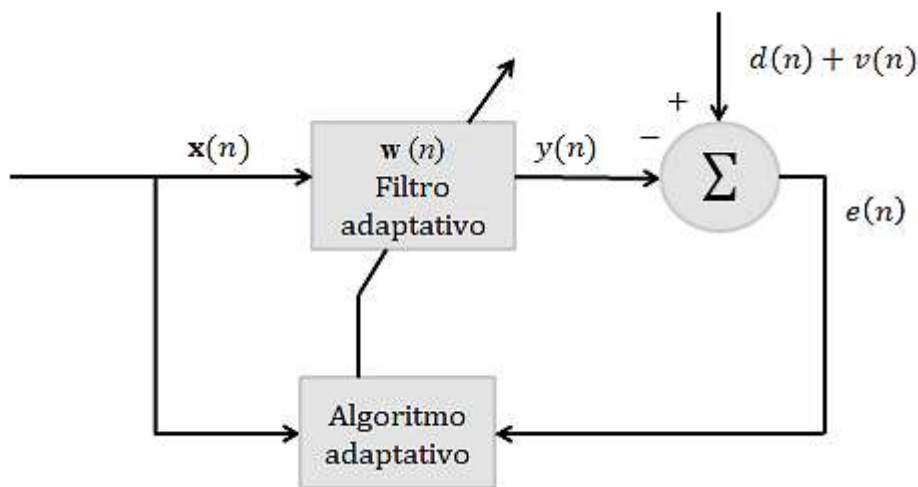


Figura 2.1 – Esquema de filtragem adaptativa.

No esquema apresentado na Figura 2.1 $x(n)$ é o sinal de entrada, $w(n)$ é o vetor com os coeficientes do filtro adaptativo, $v(n)$ é um ruído, $y(n)$ consiste na saída do filtro, $d(n)$ é o sinal desejado e $e(n)$ é definido como a diferença entre a saída do filtro e a soma entre o sinal desejado e o ruído.

O algoritmo adaptativo mais básico é o LMS (Haykin, 1996). Devido à simplicidade computacional o LMS é bastante utilizado em aplicações de filtragem adaptativa (Diniz,

2008). A fim de conhecer um pouco mais sobre o LMS será abordado nos tópicos a seguir à construção matemática desse algoritmo.

2.1 FILTROS DISCRETOS NO TEMPO

Um sistema é definido como sendo qualquer dispositivo físico ou algoritmo que transforme um sinal, chamado de entrada, em outro sinal, chamado de saída ou resposta. Quando o sistema é simplesmente um algoritmo, pode ser realizado em hardware ou software. As relações matemáticas entre os sinais de entrada e saída de um sistema serão citadas como um modelo (sistema) (Manolakis et al., 2005, p. 47). No caso de um sistema de tempo discreto, o modelo é simplesmente uma transformação que mapeia exclusivamente o sinal de entrada $\mathbf{x}(n)$ para um sinal de saída $y(n)$. Isso é denotado por.

$$y(n) = \mathbf{H}[\mathbf{x}(n)] \quad -\infty < n < \infty \quad (2.1)$$

e é representado graficamente como na Figura 2.2

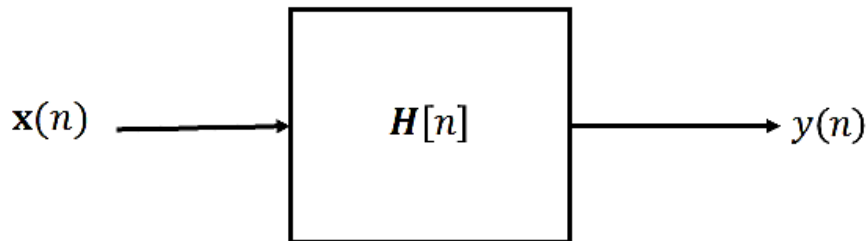


Figura 2.2 – Representação em diagrama de blocos de um sistema de tempo discreto.

2.1.1 Sistemas Lineares Invariantes no Tempo

Os sistemas tratados nesta pesquisa são lineares e invariantes no tempo. Um sistema é dito linear quando possui a propriedade de superposição, ou seja, um sistema é linear se:

- 1) A resposta a $x_1(t) + x_2(t)$ é $y_1(t) + y_2(t)$.
- 2) A resposta a $ax_1(t)$ é $ay_1(t)$.

Sobre a invariância no tempo, um sistema é invariante no tempo se o comportamento e as características do sistema são fixos ao longo do tempo.

A saída de um sistema linear, invariante no tempo, sempre pode ser expressa como a convolução entre a sequência de entrada $\mathbf{x}(n)$ e a resposta ao impulso ou amostra unitária $h(n) \triangleq \mathbf{H}[\delta(n)]$ do sistema, ou seja,

$$y(n) = \mathbf{x}(n) * h(n) \triangleq \sum_{k=-\infty}^{\infty} \mathbf{x}(k)h(n-k) \quad (2.2)$$

onde $*$ denota a operação de convolução. Pode-se demonstrar que uma expressão equivalente a Equação (2.2) é:

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)\mathbf{x}(n-k) = h(n) * \mathbf{x}(n) \quad (2.3)$$

Assim, dada a entrada $\mathbf{x}(n)$ para um sistema linear, invariante no tempo, a saída $y(n)$ pode ser calculada utilizando a resposta de impulso $h(n)$ do sistema e a fórmula (2.2) ou (2.3). Se $\mathbf{x}(n)$ e $h(n)$ são sequências arbitrárias de duração finita, então a convolução acima também pode ser calculada usando uma operação de multiplicação matricial-vetorial.

Considerando que $\mathbf{x}(n)$, $0 \leq n \leq N-1$, e $h(n)$, $0 \leq n \leq M-1$, sejam duas sequências de duração finita de comprimentos N e $M (< N)$ respectivamente. Então de (2.3), a sequência $y(n)$ também é uma sequência de duração finita, com $0 \leq n \leq L-1$ e $L \triangleq N + M - 1$ amostras.

Se as amostras de $y(n)$ e $h(n)$ estão dispostos nos vetores de coluna \mathbf{y} e \mathbf{h} , respectivamente, então de (2.3) pode ser escrita na forma matricial

$$\begin{bmatrix} y(0) \\ \vdots \\ y(M-1) \\ \vdots \\ y(N-1) \\ \vdots \\ y(L-1) \end{bmatrix} = \begin{bmatrix} x(0) & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ x(M-1) & \dots & \dots & x(0) \\ \vdots & \vdots & \ddots & \vdots \\ x(N-1) & \dots & \dots & x(N-M) \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & x(N-1) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{bmatrix} \quad (2.4)$$

ou

$$\mathbf{y} = \mathbf{Xh}, \quad (2.5)$$

onde a matriz \mathbf{X} de ordem $L \times M$ contém deslocamentos lineares em $\mathbf{x}(n - k)$ para $n = 0, \dots, N - 1$ que são dispostos como linhas. A matriz \mathbf{X} é denominada matriz de dados de entrada.

Uma propriedade interessante sobre a matriz \mathbf{X} é que todos os elementos ao longo de qualquer diagonal são iguais. Essa matriz é chamada de matriz de *Toeplitz*, e, portanto, \mathbf{X} tem uma estrutura *Toeplitz*. Vale ressaltar ainda que a primeira e a última $M - 1$ linha de \mathbf{X} contêm valores zero (ou limite). Portanto, a primeira e a última $M - 1$ amostra de $y(n)$ contêm efeitos de contorno transitórios. Nota-se que o vetor \mathbf{y} também pode ser obtido como

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \quad (2.6)$$

em que \mathbf{H} é uma matriz de *Toeplitz* obtida a partir de (2.2).

Outra propriedade presente (ou ausente) em sistemas é a causalidade. Um sistema é chamado causal se o valor presente do sinal de saída depende apenas de os valores presentes e/ou passados do sinal de entrada. Embora a causalidade seja necessária para a implementação, em tempo real, de sistemas de tempo discreto, não é realmente um problema de aplicações *off-line*, onde o sinal de entrada já foi gravado. A condição para que um sistema linear, invariante no tempo, seja causal é que a resposta de impulso $h(n) = 0$ para $n < 0$.

A estabilidade é outra propriedade importante do sistema. Existem vários critérios de estabilidade. Um sistema é chamado de BIBO (*bounded-input bounded-output*) estável ou simplesmente estável, se e somente se, cada entrada limitada, a saber, $|\mathbf{x}(n)| \leq M_x < \infty$ para todo n , produz uma saída limitada, ou seja, $|y(n)| \leq M_y < \infty$ para todos n .

Sistemas instáveis geram ilimitados sinais de saída e, portanto, não são úteis em aplicações práticas, porque resultarão em um estouro na saída. Pode-se mostrar que um sistema linear invariante no tempo é BIBO estável, se e somente se,

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \quad (2.7)$$

Além da abordagem de convolução no domínio do tempo, a saída de um sistema linear, invariante no tempo, pode ser determinada usando técnicas de transformação. De fato, usando a propriedade de convolução da transformada z ,

$$Y(z) = H(z)X(z) \quad (2.8)$$

onde $X(z)$, $Y(z)$ e $H(z)$ são as transformadas z da entrada, saída e resposta de impulso, respectivamente. A transformada z , $H(z) = \mathcal{Z}[h(n)]$ da resposta de impulso desempenha um papel muito importante na análise e caracterização de sistemas lineares e invariantes no tempo. Se o círculo unitário estiver dentro da ROC de $H(z)$, o sistema é estável e $H(e^{j\omega})$ fornece sua resposta de frequência.

Avaliando (2.8) no círculo unitário dá-se:

$$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega}) \quad (2.9)$$

onde $H(e^{j\omega})$ é a função de resposta de frequência do sistema. Como, em geral, $H(e^{j\omega})$ é de valor complexo, temos:

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j\angle H(e^{j\omega})} \quad (2.10)$$

$|H(e^{j\omega})|$ e $\angle H(e^{j\omega})$ é a resposta de magnitude e a resposta de fase do sistema. Para um sistema com uma resposta de impulso real, $|H(e^{j\omega})|$ tem simetria par e $\angle H(e^{j\omega})$ tem simetria ímpar. A resposta de atraso de grupo de um sistema com resposta de frequência $H(e^{j\omega})$ é definido como:

$$\tau(e^{j\omega}) = -\frac{d}{d\omega} \angle H(e^{j\omega}) \quad (2.11)$$

e fornece uma medida do atraso médio do sistema em função da frequência.

2.1.2 Sistemas discretos por equações lineares de diferença de coeficiente constante

Um sistema de tempo discreto é chamado de praticamente realizável se satisfizer as seguintes condições:

- 1) Requer uma quantidade finita de memória;
- 2) Requer a quantidade de operações aritméticas necessárias, pois o cálculo de cada amostra de saída é finito.

Se, além de linear e invariante no tempo, exigimos que um sistema seja causal e praticamente realizável, então a descrição mais geral de entrada/saída de tal sistema leva a forma de uma equação de diferença linear de coeficiente constante

$$y(n) = -\sum_{k=1}^P a_k y(n-k) + \sum_{k=0}^Q d_k x(n-k) \quad (2.12)$$

Caso os parâmetros do sistema $\{a_k, d_k\}$ dependam do tempo, o sistema é linear e varia no tempo. Se, no entanto, os parâmetros do sistema dependerem dos sinais de entrada ou saída, então o sistema torna-se não linear.

Considerando que os parâmetros são constantes e avaliando a transformada z de ambos os lados de (2.12), obtém-se:

$$\mathbf{H}(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^Q d_k z^{-k}}{1 + \sum_{k=1}^P a_k z^{-k}} \triangleq \frac{D(z)}{A(z)} \quad (2.13)$$

Um sistema com uma função de sistema racional pode ser descrito, dentro de um fator de ganho, por a localização de seus polos e zeros no plano z :

$$\mathbf{H}(z) = \frac{D(z)}{A(z)} = G \frac{\prod_{k=1}^Q (1 - z_k z^{-1})}{\prod_{k=1}^P (1 - p_k z^{-1})} \quad (2.14)$$

O sistema descrito em (2.12) ou equivalentemente por (2.13) ou (2.14) é estável se os seus polos, isto é, as raízes do polinômio denominador $A(z)$ estiverem todas dentro do círculo unitário.

Se os coeficientes a_k em (2.12) forem zero, teremos:

$$y(n) = \sum_{k=0}^Q d_k x(n-k) \quad (2.15)$$

que em comparação com (2.3) produz

$$h(n) = \begin{cases} d_n & 0 \leq n \leq Q \\ 0 & \text{caso contrário} \end{cases} \quad (2.16)$$

isto é, o sistema em (2.15) tem uma resposta de impulso com duração finita e é chamado de sistema de resposta a impulsos finitos (FIR).

A partir de (2.13), conclui-se que a função do sistema de um sistema FIR é um polinômio em z^{-1} , e assim $H(z)$ tem polos triviais Q em $z = 0$ e Q zeros. Por esta razão, os sistemas FIR também são referidos como sistemas *all-zero* (AZ). A Figura 2.3 mostra uma realização simples do diagrama de blocos do sistema FIR (2.15) em termos de atrasos unitários, aditivos e multiplicadores.

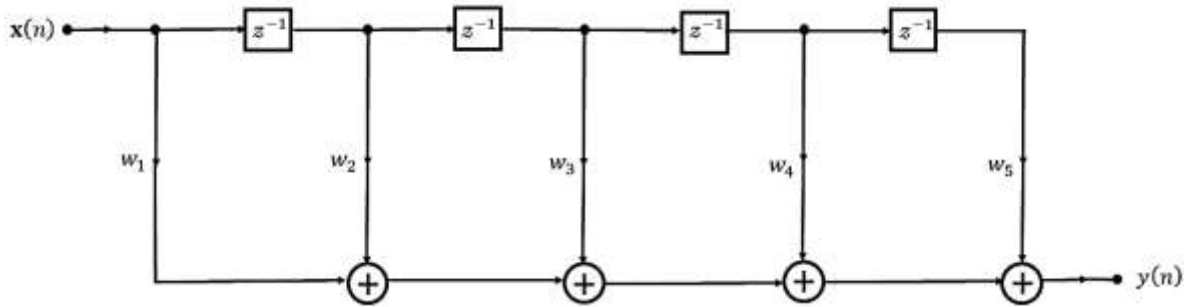


Figura 2.3 – Realização do filtro FIR (forma direta).

Quando um sistema tem polos e zeros, $H(z)$ pode ser expresso usando a forma de expansão da fração parcial da seguinte forma:

$$H(z) = \sum_{k=1}^P \frac{A_k}{1 - p_k z^{-k}} \quad (2.17)$$

se os polos são distintos e $Q < P$. A resposta de impulso correspondente é então dada por

$$h(n) = \sum_{k=1}^P A_k (p_k)^n x(n) \quad (2.18)$$

ou seja, cada polo contribui com um modo exponencial de duração infinita para a resposta do impulso. Com isso, conclui-se que a presença de qualquer polo não trivial em um sistema implica na duração da resposta ao impulso. Classificam-se tais sistemas como sistemas de resposta infinita a impulsos (IIR). Se $Q = 0$, o sistema tem apenas polos, com zeros em $z = 0$, e é chamado de sistema de polo total (AP).

Deve-se ressaltar ainda que, embora os sistemas de todos os polos e polos zero sejam IIR, nem todos os sistemas IIR são sistemas polo zero (PZ). De fato, existem muitos sistemas úteis, por exemplo, um filtro passa-baixa ideal, que não pode ser descrito pelas funções do sistema racional de ordem finita. As figuras 2.4 e 2.5 mostram realizações de forma direta de um sistema polo total e um sistema polo zero.

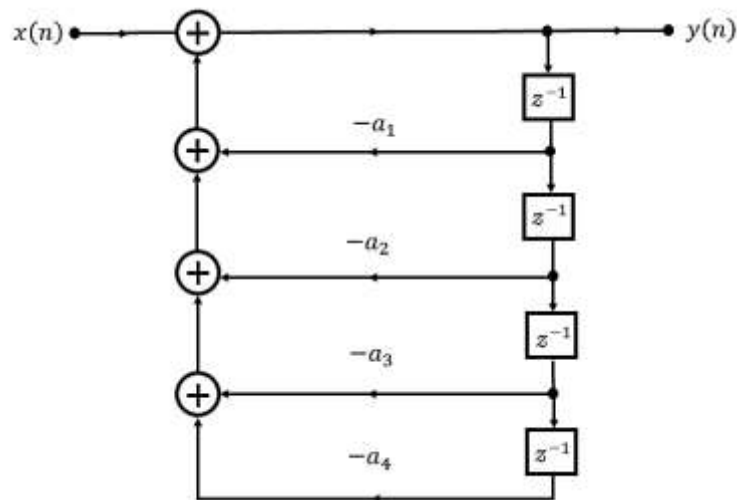


Figura 2.4 – Realização do sistema de todos os polos (forma direta)

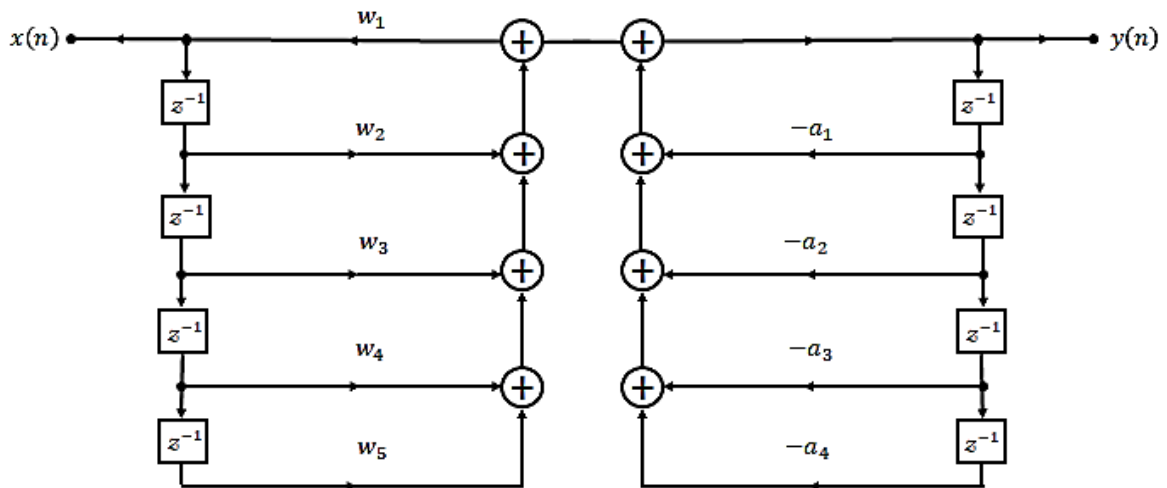


Figura 2.5 – Realização do sistema polo zero (forma direta).

2.1.3 Resposta a Entradas Periódicas

Embora a fórmula de soma de convolução possa ser usada para calcular a resposta de um sistema estável a qualquer sinal de entrada, (2.8) não pode ser usada com entradas periódicas porque sinais periódicos não possuem uma transformada z . No entanto, uma fórmula no domínio da frequência semelhante para (2.9) pode ser desenvolvido para entradas periódicas.

Seja $\mathbf{x}(n)$ um sinal periódico com período fundamental N . Este sinal pode ser expandido em uma série de Fourier como:

$$\mathbf{x}(n) = \sum_{k=0}^{N-1} X_k e^{\frac{j2\pi kn}{N}} \quad n = 0, 1, \dots, N-1 \quad (2.19)$$

2.2 ALGORITMO LMS

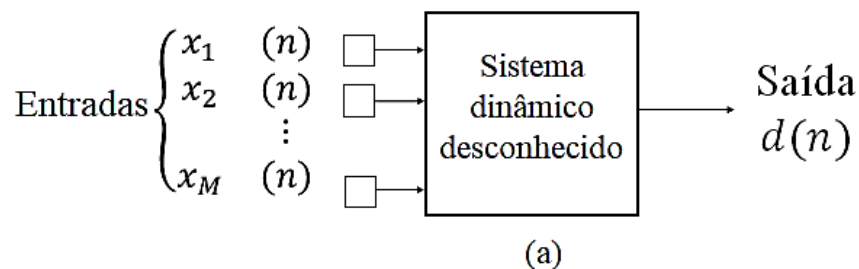
Inspirado no *perceptron*, o algoritmo de mínimos quadrados médio (LMS) foi o primeiro algoritmo de filtragem adaptativa linear para resolver problemas como previsão e equalização de canais de comunicação. Esses dois algoritmos, embora diferentes em aplicações, compartilham uma característica comum, ambos envolvem o uso de um combinador linear, daí a designação “linear”. (Haykin, 2008, p. 91)

Bastante utilizado em filtragem adaptativa, o LMS tornou-se referência para os demais algoritmos de filtragem. Conforme Haykin (2008), as razões por trás deste histórico são múltiplas como:

- A complexidade do algoritmo LMS é linear em relação aos parâmetros ajustáveis, o que torna o algoritmo computacionalmente eficiente;
- É eficaz no desempenho;
- O algoritmo é simples de codificar e, portanto, fácil de construir;
- Acima de tudo, o algoritmo é robusto no que diz respeito a perturbações externas.

2.3 ESTRUTURA DE FILTRAGEM DO ALGORITMO LMS

A Figura 2.6 (a) mostra o diagrama de blocos de um sistema dinâmico desconhecido que é estimulado por um vetor de entrada composto pelos elementos $x_1(n), x_2(n), \dots, x_M(n)$ onde n , denota o instante de tempo em que o estímulo (excitação) é aplicado ao sistema. O índice de tempo $n = 1, 2, \dots, M$.



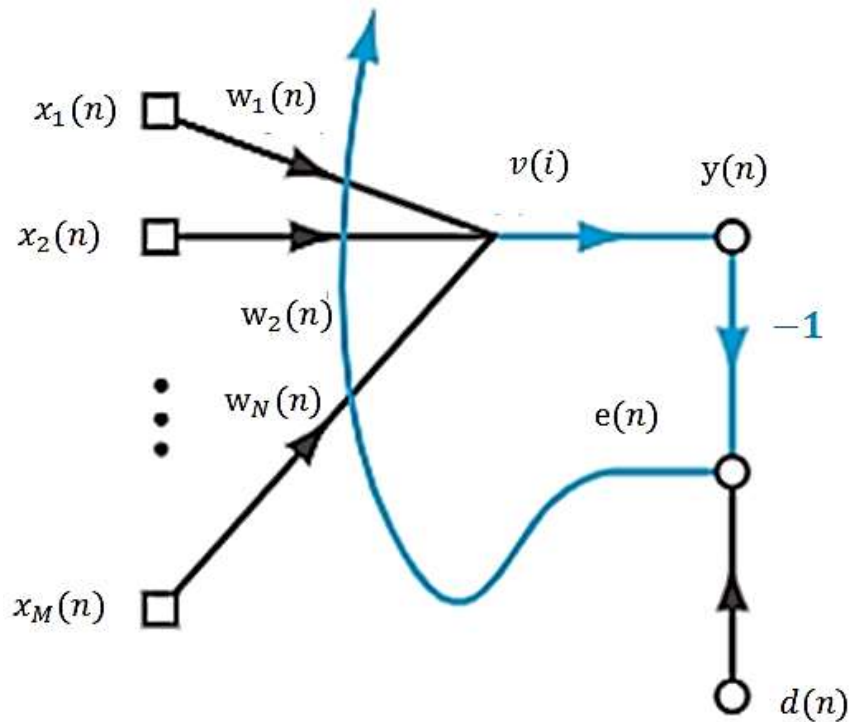


Figura 2.6 - Sistema dinâmico desconhecido (a). Gráfico de fluxo de sinal do modelo adaptativo para o sistema (b). O gráfico incorpora um ciclo de *feedback*.

Em resposta a este estímulo, o sistema produz uma saída denotada por $y(n)$. Assim, o comportamento externo do sistema é descrito pelo conjunto de dados

$$\mathcal{T}: \{\mathbf{x}(n), d(n); n = 1, 2, \dots, M\} \quad (2.20)$$

onde

$$\mathbf{x}(n) = [x_1(n), x_2(n), \dots, x_M(n)]^T \quad (2.21)$$

Os pares de amostras que compõem \mathcal{T} são distribuídos de forma idêntica de acordo com uma lei de probabilidade desconhecida. A dimensão M que pertence ao vetor de entrada $\mathbf{x}(n)$ é referida como a dimensionalidade do espaço de entrada, ou simplesmente como a dimensionalidade de entrada.

O vetor de estímulo $\mathbf{x}(n)$ pode surgir de duas maneiras fundamentalmente diferentes, uma espacial e o outro temporal:

- Os elementos M de $\mathbf{x}(n)$ originam-se em diferentes pontos do espaço; neste caso, falamos de $\mathbf{x}(n)$ como dados instantâneos.

- Os elementos M de $\mathbf{x}(n)$ representam o conjunto de valores presentes e $(M - 1)$ valores passados de alguma excitação uniformemente espaçada no tempo.

A Figura 2.6 (b) mostra um gráfico de fluxo de sinal de um filtro adaptativo. Sua operação consiste em dois processos contínuos:

- 1) Processo de filtragem, que envolve o cálculo de dois sinais:
 - Uma saída, denotada por $y(n)$, que é produzida em resposta aos elementos M do vetor de estímulo $\mathbf{x}(n)$, a saber, $x_1(n), x_2(n), \dots, x_N(n)$;
 - Um sinal de erro, denotado por $e(n)$, que é obtido comparando a saída $y(n)$ com a saída correspondente $d(n)$ produzida pelo sistema desconhecido. Com efeito, $d(n)$ atua como uma resposta desejada.
- 2) Processo adaptativo, que envolve o ajuste automático dos coeficientes do filtro de acordo com o sinal de erro $e(n)$.

Assim, a combinação desses dois processos trabalhando juntos constitui um circuito de realimentação como mostrado na Figura 2.6 (b). Desse modo, a saída do filtro $y(n)$ é exatamente igual ao campo local induzido $v(n)$; isso é,

$$y(n) = v(n) = \sum_{k=1}^N w_k(n)x_k(n) \quad (2.22)$$

onde $w_1(n), w_2(n), \dots, w_N(n)$ são os coeficientes do filtro, medidos no tempo n . Na forma de matriz, podemos expressar $y(n)$ como um produto interno dos vetores $\mathbf{x}(n)$ e $\mathbf{w}(n)$ como:

$$y(n) = \mathbf{x}^T(n)\mathbf{w}(n) \quad (2.23)$$

onde

$$\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_N(n)]^T \quad (2.24)$$

A saída do filtro $y(n)$ é comparada com a saída correspondente $d(n)$ recebida do sistema desconhecido no tempo n . Normalmente, $y(n)$ é diferente de $d(n)$, portanto, sua comparação resulta no sinal de erro

$$e(n) = d(n) - y(n) \quad (2.25)$$

A maneira pela qual o sinal de erro $e(n)$ é usado para controlar os coeficientes do filtro é determinada pela função de custo usada para derivar o algoritmo de filtragem

adaptativa de interesse. Esta questão está intimamente relacionada com processos de otimização. Neste contexto, apresenta-se a seguir uma revisão dos métodos de otimização irrestrita.

2.4 OTIMIZAÇÃO SEM RESTRIÇÕES: UMA REVISÃO

Considere uma função de custo $J(\mathbf{w})$ que é uma função diferenciável de algum vetor de peso desconhecido (\mathbf{w}). A função $J(\mathbf{w})$ mapeia os elementos de \mathbf{w} em números reais. É uma medida de como escolher o vetor de peso \mathbf{w} de um algoritmo de filtragem adaptativa para que ele se comporte de maneira ideal. Deseja-se encontrar uma solução ótima \mathbf{w}^* que satisfaça a seguinte condição

$$J(\mathbf{w}^*) \leq J(\mathbf{w}), \quad (2.26)$$

ou seja, é preciso resolver um problema de otimização irrestrita, expresso da seguinte forma:

Minimize a função de custo $J(\mathbf{w})$ em relação ao vetor de peso \mathbf{w} .

A condição necessária para otimalidade é

$$\nabla J(\mathbf{w}^*) = \mathbf{0} \quad (2.27)$$

onde $\nabla J(\mathbf{w})$ é o operador gradiente,

$$\nabla = \left[\frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_N} \right]^T \quad (2.28)$$

E $\nabla J(\mathbf{w})$ é o vetor gradiente da função de custo,

$$\nabla J(\mathbf{w}) = \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}, \dots, \frac{\partial J}{\partial w_N} \right]^T \quad (2.29)$$

Uma classe de algoritmos de otimização irrestrita que é particularmente adequada para o projeto de filtros adaptativos é baseada na ideia de descida iterativa local, na qual se inicia com uma estimativa denotada por $\mathbf{w}(0)$, em seguida, gera-se uma sequência de vetores de peso $w(1), w(2) \dots$, tal que a função de custo $J(\mathbf{w})$ seja reduzida a cada iteração do algoritmo, como mostrado por

$$J(\mathbf{w}(n+1)) < J(\mathbf{w}(n)) \quad (2.30)$$

onde $\mathbf{w}(n)$ é o valor antigo do vetor de peso e $\mathbf{w}(n+1)$ é seu valor atualizado.

Esperamos que o algoritmo eventualmente convirja para a solução ótima \mathbf{w}^* . Dizemos "esperança" porque existe uma possibilidade distinta de que o algoritmo torne-se instável, a menos que sejam tomadas precauções especiais.

2.4.1 Método da descida mais íngreme

No método da descida mais íngreme, os ajustes sucessivos aplicados ao vetor de peso \mathbf{w} são na direção da descida mais íngreme, ou seja, em direção oposta ao vetor gradiente $\nabla J(\mathbf{w})$. Por conveniência de apresentação, escrevemos

$$\mathbf{g} = \nabla J(\mathbf{w}) \quad (2.31)$$

Assim, o algoritmo de descida mais íngreme é formalmente descrito por

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{g}(n) \quad (2.32)$$

onde μ é o tamanho do passo e $\mathbf{g}(n)$ é o vetor gradiente avaliado no ponto $\mathbf{w}(n)$. Ao passar da iteração n para $n+1$, o algoritmo aplica a correção

$$\begin{aligned} \Delta \mathbf{w}(n) &= \mathbf{w}(n+1) - \mathbf{w}(n) \\ &= -\mu \mathbf{g}(n) \end{aligned} \quad (2.33)$$

Para mostrar que a formulação do algoritmo de descida mais íngreme satisfaz a condição da Equação (2.30) para descida iterativa, usamos uma expansão em série de Taylor de primeira ordem em torno de $\mathbf{w}(n)$ para aproximar $J(\mathbf{w}(n+1))$ como:

$$J(\mathbf{w}(n+1)) \approx J(\mathbf{w}(n)) + \mathbf{g}^T(n) \Delta \mathbf{w}(n) \quad (2.34)$$

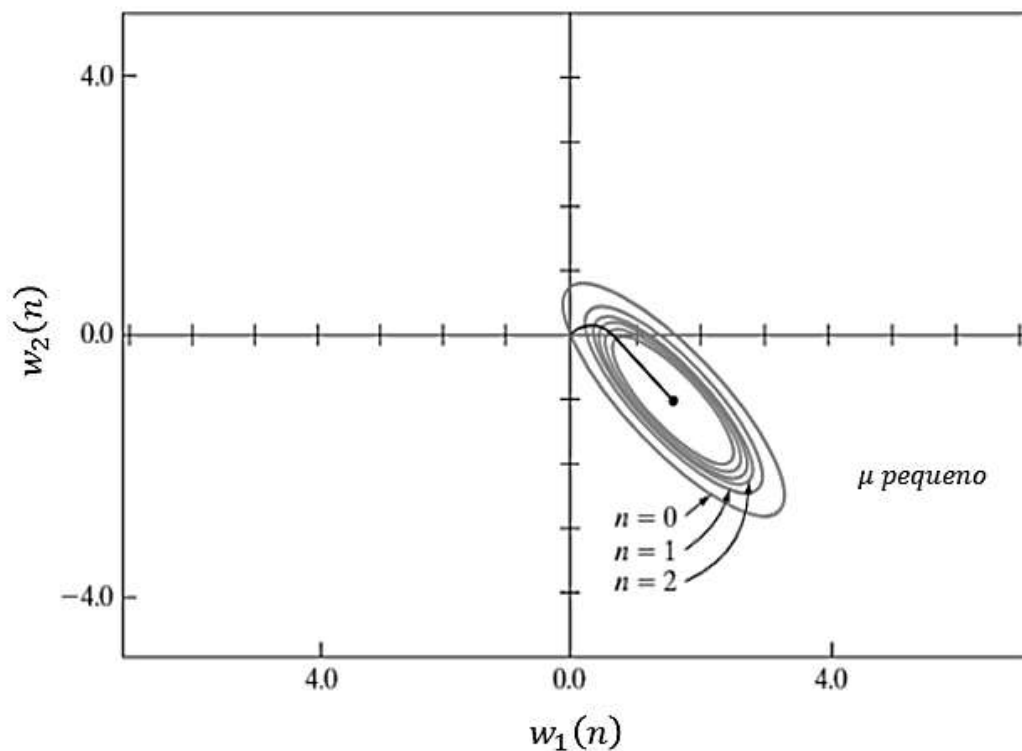
O uso é justificado para μ pequeno. Substituindo a Equação (2.33) nesta relação aproximada resulta

$$\begin{aligned} J(\mathbf{w}(n+1)) &\approx J(\mathbf{w}(n)) - \mu \mathbf{g}^T(n) \mathbf{g}(n) \\ &= J(\mathbf{w}(n)) - \mu \|\mathbf{g}(n)\|^2 \end{aligned} \quad (2.35)$$

O que mostra que, para um tamanho de passo positivo μ , a função de custo diminui à medida que o algoritmo avança de uma iteração para a próxima. O raciocínio apresentado aqui é aproximado na medida em que este resultado final é verdadeiro apenas para tamanhos de passo suficientemente pequenos.

O método da descida mais íngreme converge lentamente para a solução ótima \mathbf{w}^* . Além disso, o tamanho do passo μ apresenta uma influência profunda no seu comportamento de convergência:

- Quando μ é pequeno, a resposta transitória do algoritmo é superamortecida, na medida em que a trajetória traçada por $\mathbf{w}(n)$ segue um caminho suave no plano \mathcal{W} , como ilustrado na Figura 2.7 (a);
- Quando μ é grande, a resposta transitória do algoritmo é subamortecida, pois a trajetória de $\mathbf{w}(n)$ segue um caminho em zigue-zague (oscilatório), como ilustrado na Figura 2.7 (b);
- Quando μ excede certo valor crítico, o algoritmo torna-se instável (ou seja, diverge).



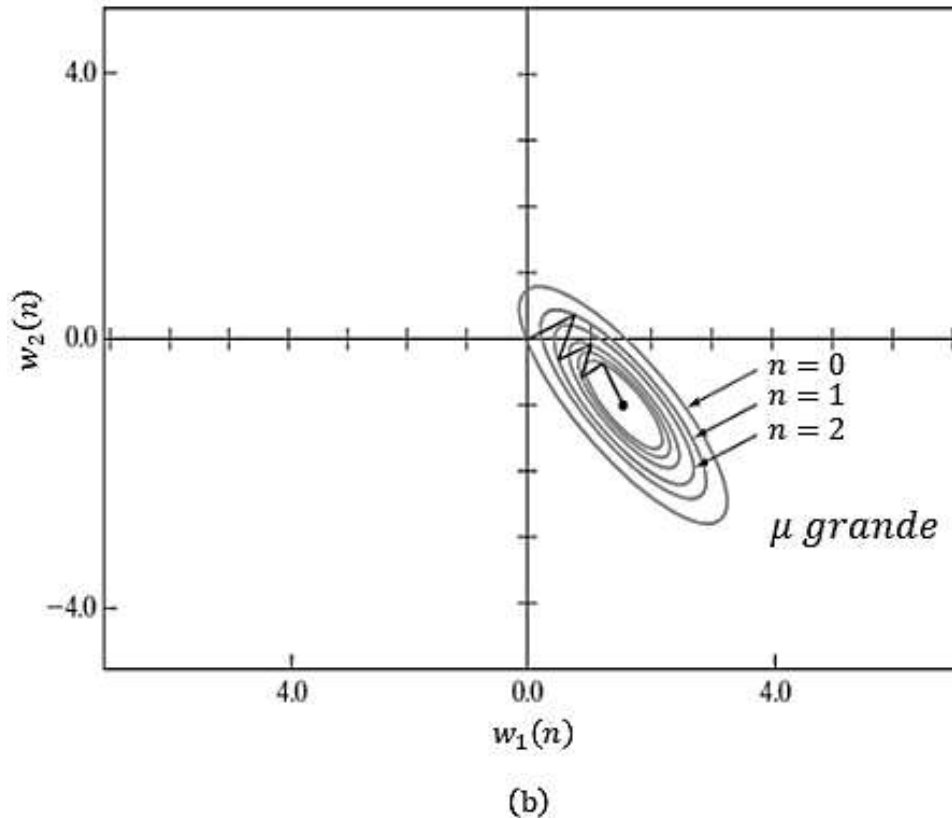


Figura 2.7 – Trajetória do método de descida mais íngreme em um espaço bidimensional para dois valores diferentes referentes ao tamanho do passo: (a) μ pequeno (b) μ grande. As coordenadas w_1 e w_2 são elementos do vetor de peso \mathbf{w} ; ambos estão no plano W .

2.4.2 Método de Newton

Para uma técnica de otimização mais elaborada, podemos recorrer ao método de Newton, cuja ideia básica é minimizar a aproximação quadrática da função de custo $J(\mathbf{w})$ em torno do ponto atual $\mathbf{w}(n)$; esta minimização é realizada a cada iteração do algoritmo. Especificamente, usando uma expansão em série de Taylor de segunda ordem da função custo em torno do ponto $\mathbf{w}(n)$, podemos escrever:

$$\begin{aligned} \Delta J(\mathbf{w}(n)) &= J(\mathbf{w}(n+1)) - J(\mathbf{w}(n)) \\ &\approx \mathbf{g}^T(n)\Delta\mathbf{w}(n) + \frac{1}{2}\Delta\mathbf{w}^T(n)\mathbf{H}(n)\Delta\mathbf{w}(n) \end{aligned} \quad (2.36)$$

Como antes, $\mathbf{g}(n)$ é o vetor gradiente $N \times 1$ por da função de custo $J(\mathbf{w})$ avaliada no ponto $\mathbf{w}(n)$. A matriz $\mathbf{H}(n)$ é a Hessiana m por m de $J(\mathbf{w})$, também avaliada em $\mathbf{w}(n)$. O Hessiano de $J(\mathbf{w})$ é definido por

$$\mathbf{H} = \nabla^2 J(\mathbf{w}) \quad (2.37)$$

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 J}{\partial w_1^2} & \frac{\partial^2 J}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_1 \partial w_N} \\ \frac{\partial^2 J}{\partial w_2 \partial w_1} & \frac{\partial^2 J}{\partial w_2^2} & \cdots & \frac{\partial^2 J}{\partial w_2 \partial w_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial w_N \partial w_1} & \frac{\partial^2 J}{\partial w_N \partial w_2} & \cdots & \frac{\partial^2 J}{\partial w_N^2} \end{bmatrix} \quad (2.38)$$

A equação (2.38) exige que a função de custo $J(\mathbf{w})$ seja duas vezes continuamente diferenciável em relação aos elementos de \mathbf{w} . Diferenciando a Equação (2.36) em relação à $\Delta \mathbf{w}$, minimizamos a mudança resultante $\Delta J(\mathbf{w})$ quando

$$\mathbf{g}(n) + \mathbf{H}(n)\Delta \mathbf{w}(n) = \mathbf{0} \quad (2.39)$$

Resolvendo esta equação para rendimentos de $\Delta \mathbf{w}(n)$

$$\Delta \mathbf{w}(n) = -\mathbf{H}^{-1}(n)\mathbf{g}(n) \quad (2.40)$$

isso é,

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \Delta \mathbf{w}(n) \\ &= \mathbf{w}(n) - \mathbf{H}^{-1}(n)\mathbf{g}(n) \end{aligned} \quad (2.41)$$

onde $\mathbf{H}^{-1}(n)$ é o inverso do Hessiano de $J(\mathbf{w})$.

De um modo geral, o método de Newton converge rapidamente assintoticamente e não apresenta o comportamento em zigue-zague que por vezes caracteriza o método de descida mais íngreme. No entanto, para que o método de Newton funcione, o Hessiano $\mathbf{H}(n)$ tem que ser uma matriz definida positiva para todo n . Infelizmente, em geral, não há garantia de que $\mathbf{H}(n)$ seja definido positivamente em cada iteração do algoritmo.

2.4.3 Método *Gauss-Newton*

Para lidar com a complexidade computacional do método de Newton sem comprometer seriamente o seu comportamento de convergência, podemos utilizar o método de *Gauss-Newton*. Para aplicar este método, adotamos uma função de custo que é expressa como a soma dos quadrados dos erros.

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^n e^2(i) \quad (2.42)$$

onde o fator de escala é incluído para simplificar as questões na análise subsequente. Todos os termos de erro nesta fórmula são calculados com base em um vetor de peso \mathbf{w} que é fixado durante todo o intervalo de observação $1 \leq i \leq n$.

O sinal de erro $e(n)$ é uma função do vetor de peso ajustável \mathbf{w} . Dado um ponto operacional $\mathbf{w}(n)$, linearizamos a dependência de $e(n)$ em \mathbf{w} introduzindo o novo termo

$$\mathbf{e}'(n, \mathbf{w}) = e(n) + \left[\frac{\partial e(n)}{\partial \mathbf{w}} \right]_{\mathbf{w}=\mathbf{w}(n)}^T \times (\mathbf{w} - \mathbf{w}(n)), \quad n = 1, 2, \dots, M \quad (2.43)$$

Equivalentemente, usando notação matricial, podemos escrever

$$\mathbf{e}'(n, \mathbf{w}) = \mathbf{e}(n) + \mathbf{J}(n) (\mathbf{w} - \mathbf{w}(n)) \quad (2.44)$$

onde $\mathbf{e}(n)$ é o vetor de erro

$$\mathbf{e}(n) = [e(1), e(2), \dots, e(n)]^T \quad (2.45)$$

e $\mathbf{J}(n)$ é o Jacobiano n por m de $\mathbf{e}(n)$:

$$\mathbf{J}(n) = \left[\begin{array}{cccc} \frac{\partial e(1)}{\partial w_1} & \frac{\partial e(1)}{\partial w_2} & \dots & \frac{\partial e(1)}{\partial w_N} \\ \frac{\partial e(2)}{\partial w_1} & \frac{\partial e(2)}{\partial w_2} & \dots & \frac{\partial e(2)}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e(n)}{\partial w_1} & \frac{\partial e(n)}{\partial w_2} & \dots & \frac{\partial e(n)}{\partial w_N} \end{array} \right]_{\mathbf{w}=\mathbf{w}(n)} \quad (2.46)$$

O Jacobiano $\mathbf{J}(n)$ é a transposição da matriz gradiente m por n $\nabla \mathbf{e}(n)$, onde

$$\nabla \mathbf{e}(n) = [\nabla e(1), \nabla e(2), \dots, \nabla e(n)] \quad (2.47)$$

o vetor de peso atualizado $\mathbf{w}(n + 1)$ agora é definido por $\nabla e(n)$, onde

$$\nabla e(n) = [\nabla e(1), \nabla e(2), \dots, \nabla e(n)] \quad (2.48)$$

o vetor de peso atualizado $\mathbf{w}(n + 1)$ agora é definido por

$$\mathbf{w}(n + 1) = \arg \min_{\mathbf{w}} \left\{ \frac{1}{2} \|e'(n, \mathbf{w})\|^2 \right\} \quad (2.49)$$

Usando Equação (2.44) para avaliar a norma euclidiana quadrada de $e'(n, \mathbf{w})$, obtemos

$$\begin{aligned} \frac{1}{2} \|e'(n, \mathbf{w})\|^2 &= \frac{1}{2} \|\mathbf{e}(n)\|^2 + e^T(n) \mathbf{J}(n) (\mathbf{w} - \mathbf{w}(n)) \\ &\quad + \frac{1}{2} (\mathbf{w} - \mathbf{w}(n))^T \mathbf{J}^T(n) \mathbf{J}(n) (\mathbf{w} - \mathbf{w}(n)) \end{aligned} \quad (2.50)$$

Assim, diferenciando essa expressão em relação à \mathbf{w} e fixando o resultado igual a zero, obtém-se

$$\mathbf{J}^T(n) \mathbf{e}(n) + \mathbf{J}^T(n) \mathbf{J}(n) (\mathbf{w} - \mathbf{w}(n)) = \mathbf{0} \quad (2.51)$$

Resolvendo esta equação para \mathbf{w} escrevemos assim, em parte da Equação 2.49, que descreve a forma pura do método Gauss-Newton.

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - \mathbf{J}^T(n) \mathbf{J}(n)^{-1} \mathbf{J}^T(n) \mathbf{e}(n) \quad (2.52)$$

Ao contrário do método de Newton, que requer o conhecimento do Hessiano da função de custo $J(n)$, o método de Gauss-Newton requer apenas o Jacobiano do vetor de erro $e(n)$. Entretanto, para que a iteração de Gauss-Newton seja computável, o produto da matriz $\mathbf{J}^T(n) \mathbf{J}(n)$ deve ser não singular.

Com relação ao último ponto, reconhecemos que $\mathbf{J}^T(n) \mathbf{J}(n)$ é sempre definido não negativo. Para garantir que não seja singular, o Jacobiano $\mathbf{J}(n)$ deve ter classificação de linha n ; isto é, as linhas de $\mathbf{J}(n)$ na Equação (2.46) deve ser linearmente independente. Infelizmente, não há garantia de que esta condição será sempre mantida. Para evitar a possibilidade de $\mathbf{J}(n)$ ser deficiente em classificação, a prática habitual é adicionar a matriz diagonal $\delta \mathbf{I}$ à matriz $\mathbf{J}^T(n) \mathbf{J}(n)$, onde \mathbf{I} é a matriz identidade. O parâmetro δ é uma pequena constante positiva escolhida para garantir que $\mathbf{J}^T(n) \mathbf{J}(n) + \delta \mathbf{I}$ é definida positiva para todo n .

O método Gauss-Newton pode ser implementado na forma ligeiramente modificada

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - (\mathbf{J}^T(n) \mathbf{J}(n) + \delta \mathbf{I})^{-1} \mathbf{J}^T(n) \mathbf{e}(n) \quad (2.53)$$

onde $\mathbf{w}(n)$ é o valor atual do vetor de peso.

2.5 O FILTRO WIENER

A abordagem escolhida para apresentar o filtro de Wiener nesta sessão considera inicialmente o filtro de mínimos quadrados. Para tal, é apresentada a seguir a derivação da fórmula para este filtro usando o método de *Gauss-Newton*. Para prosseguir então, define-se o vetor de erro como:

$$\begin{aligned} e(n) &= \mathbf{d}(n) - [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(n)]^T \mathbf{w}(n) \\ &= d(n) - \mathbf{x}(n)\mathbf{w}(n) \end{aligned} \quad (2.54)$$

onde $\mathbf{d}(n)$ é o vetor de resposta desejado de dimensão n por 1 .

$$\mathbf{d}(n) = [d(1), d(2), \dots, d(n)]^T \quad (2.55)$$

e $\mathbf{X}(n)$ é a matriz de dados N por M .

$$\mathbf{X}(n) = [\mathbf{x}(1) \ \mathbf{x}(2) \ \dots \ \mathbf{x}(n)]^T \quad (2.56)$$

Diferenciar o vetor de erro $e(n)$ em relação a $\mathbf{w}(n)$ produz a matriz gradiente

$$\nabla e(n) = -\mathbf{X}^T(n) \quad (2.57)$$

Correspondentemente, o Jacobiano de $e(n)$ é

$$\mathbf{J}(n) = -\mathbf{X}(n) \quad (2.58)$$

Como a equação de erro (2.44) já é linear no vetor de peso $\mathbf{w}(n)$, o método de *Gauss-Newton* converge em uma única iteração. Substituindo (2.54) e (2.58) em (2.52) temos:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}\mathbf{X}^T(n) (\mathbf{d}(n) - \mathbf{X}(n)\mathbf{w}(n)) \\ &= (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}\mathbf{X}^T(n)\mathbf{d}(n) \end{aligned} \quad (2.59)$$

O termo $(\mathbf{X}^T(n)\mathbf{X}(n))^{-1}\mathbf{X}^T(n)$ é chamado de pseudoinverso da matriz de dados $\mathbf{X}(n)$; aquilo é,²

$$\mathbf{X}^+(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}\mathbf{X}^T(n) \quad (2.60)$$

Portanto, podemos reescrever a Equação (2.59) na forma compacta

$$\mathbf{w}(n + 1) = \mathbf{X}^+(n) \mathbf{d}(n) \quad (2.61)$$

Esta fórmula representa uma maneira conveniente de afirmar o seguinte: o vetor de pesos $\mathbf{w}(n + 1)$ resolve o problema linear dos mínimos quadrados, definido ao longo de um intervalo de observação de duração n , como o produto de dois termos o pseudoinverso $\mathbf{X}^+(n)$ e o vetor de resposta desejado $\mathbf{d}(n)$.

2.5.1 Filtro Wiener: Forma Limitante do Filtro de Mínimos Quadrados para um Ambiente Ergódico

Considerando que \mathbf{w}_0 , pode denotar a forma limite do filtro de mínimos quadrados como o número de observações quando n tende ao infinito, então podemos então usar a Equação (2.59) escrever

$$\begin{aligned} \mathbf{w}_0 &= \lim_{n \rightarrow \infty} \mathbf{w}(n + 1) \\ &= \lim_{n \rightarrow \infty} \mathbf{X}^T(n) \mathbf{X}(n)^{-1} \mathbf{X}^T(n) \mathbf{d}(n) \\ &= \lim_{n \rightarrow \infty} \left(\frac{1}{n} \mathbf{X}^T(n) \mathbf{X}(n) \right)^{-1} \\ &\quad \times \lim_{n \rightarrow \infty} \frac{1}{n} \mathbf{X}^T(n) \mathbf{d}(n) \end{aligned} \quad (2.62)$$

Supondo agora que o vetor de entrada $\mathbf{x}(n)$ e a resposta desejada correspondente $d(n)$ sejam extraídos de um ambiente ergódico que também é estacionário, podemos então substituir médias de tempo por médias de conjuntos. Por definição, a forma média do conjunto da matriz de correlação do vetor de entrada $\mathbf{x}(n)$ é

$$\mathbf{R}_{xx} = E[\mathbf{x}(n) \mathbf{x}^T(n)] \quad (2.63)$$

e, correspondentemente, a forma média do conjunto do vetor de correlação cruzada entre o vetor de entrada $\mathbf{x}(n)$ e o vetor de resposta desejado $d(n)$ é

$$\mathbf{r}_{dx} = E[\mathbf{x}(n) d(n)] \quad (2.64)$$

onde E é o operador esperança. Portanto, sob o pressuposto da ergodicidade, agora pode escrever

$$\mathbf{R}_{xx} = \lim_{n \rightarrow \infty} \frac{1}{n} \mathbf{X}(n) \mathbf{X}^T(n) \quad (2.65)$$

e

$$\mathbf{r}_{dx} = \lim_{n \rightarrow \infty} \frac{1}{n} \mathbf{X}^T(n) \mathbf{d}(n) \quad (2.66)$$

Assim, podemos reformular a Equação (2.62) em termos de parâmetros de correlação com média de conjunto como:

$$\mathbf{w}_0 = \mathbf{R}_{xx}^{-1} \mathbf{r}_{dx} \quad (2.67)$$

onde \mathbf{R}_{xx}^{-1} é o inverso da matriz de correlação \mathbf{R}_{xx} .

O vetor de peso \mathbf{w}_0 é chamado de *solução de Wiener* para o problema de filtragem linear ótima. Assim, podemos fazer a afirmação: para um processo ergódico, o filtro de mínimos quadrados se aproxima assintoticamente do filtro de Wiener quando o número de observações se aproxima do infinito

Projetar o filtro de Wiener requer conhecimento da estatística de segunda ordem: a matriz de correlação \mathbf{R}_{xx} do vetor de entrada $\mathbf{x}(n)$, e o vetor de correlação cruzada \mathbf{r}_{dx} entre $\mathbf{x}(n)$ e a resposta desejada $d(n)$. Contudo, esta informação não está disponível quando o ambiente em que o filtro opera é desconhecido. Podemos lidar com tal ambiente usando um filtro adaptativo linear, adaptativo no sentido de que o filtro é capaz de ajustar seus parâmetros livres, em resposta a variações estatísticas no ambiente. Um algoritmo altamente popular para fazer esse tipo de ajuste de forma contínua é o algoritmo *Least-Mean-Square* (LMS), onde será discutido na sessão a seguir.

2.6 O ALGORITMO DE MÍNIMOS QUADRADOS MÉDIO – LMS

O algoritmo de mínimos quadrados médio (*Least-Mean-Square* -LMS) é configurado para minimizar o valor instantâneo da função de custo,

$$J(\hat{\mathbf{w}}) = \frac{1}{2} e^2(n) \quad (2.68)$$

onde $e(n)$ é o sinal de erro medido no tempo n . Diferenciar $J(\hat{\mathbf{w}})$ em relação ao vetor de peso $\hat{\mathbf{w}}$ produz

$$\frac{\partial J(\hat{\mathbf{w}})}{\partial(\hat{\mathbf{w}})} = e(n) \frac{\partial e(n)}{\partial \mathbf{w}} \quad (2.69)$$

Assim como acontece com o filtro de mínimos quadrados, no algoritmo LMS, pode-se expressar o sinal de erro como

$$e(n) = d(n) - \mathbf{x}^T(n) \hat{\mathbf{w}}(n) \quad (2.70)$$

Por isso,

$$\frac{\partial e(\hat{\mathbf{w}})}{\partial(\hat{\mathbf{w}})(n)} = -\mathbf{x}(n) \quad (2.71)$$

e

$$\frac{\partial J(\hat{\mathbf{w}})}{\partial(\hat{\mathbf{w}})(n)} = -\mathbf{x}(n)e(n) \quad (2.72)$$

Usando este último resultado como estimativa instantânea do vetor gradiente, podemos escrever

$$\hat{\mathbf{g}}(n) = -\mathbf{x}(n)e(n) \quad (2.73)$$

Finalmente, usando (2.73) para o vetor gradiente em (2.32) para o método da descida mais íngreme, podemos formular o algoritmo LMS da seguinte forma:

$$\hat{\mathbf{w}}(n + 1) = \hat{\mathbf{w}}(n) + \mu\mathbf{x}(n)e(n) \quad (2.74)$$

É também digno de nota que o inverso do parâmetro de passo μ atua como uma medida da memória do algoritmo LMS: quanto menor for o valor de μ , maior será o intervalo de memória durante o qual o algoritmo LMS lembra dados passados. Conseqüentemente, quando μ é pequeno, o algoritmo LMS funciona com precisão, mas a taxa de convergência do algoritmo é lenta.

Ao derivar a Equação (2.74), usamos $\hat{\mathbf{w}}(n)$ no lugar de $\mathbf{w}(n)$ para enfatizar o fato de que o algoritmo LMS produz uma estimativa instantânea do vetor de peso que resultaria do uso do método da descida mais íngreme.

No algoritmo de descida mais íngreme, o vetor de peso $\mathbf{w}(n)$ segue uma trajetória bem definida no espaço de peso \mathcal{W} para um μ prescrito. Em contraste, no algoritmo LMS, o vetor de peso $\hat{\mathbf{w}}(n)$ traça uma trajetória aleatória. Por esse motivo, o algoritmo LMS às vezes é chamado de "algoritmo de gradiente estocástico". À medida que o número de iterações no algoritmo LMS se aproxima do infinito, $\hat{\mathbf{w}}(n)$ realiza um passeio aleatório (movimento browniano) em torno da solução de Wiener \mathbf{w}_0 .

É importante a notar que, ao contrário do método de descida mais íngreme, o algoritmo LMS não requer conhecimento das estatísticas do ambiente. Esta característica do algoritmo LMS é importante do ponto de vista prático.

Um resumo do algoritmo LMS, baseado em (2.70) e (2.74), é apresentado na Tabela 2.1, que ilustra claramente a simplicidade do algoritmo. Conforme indicado nesta tabela, a inicialização do algoritmo é feita simplesmente definindo o valor do peso vetor $\hat{\mathbf{w}}(0) = \mathbf{0}$.

Tabela 2.1 - Sumário do algoritmo adaptativo LMS.

Algoritmo 2.1 – LMS

1. Inicialização e parâmetros

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

$$0 < \mu < \frac{1}{\lambda_{\max}}$$

2. Dados de entrada e saída da planta e do filtro adaptativo

$$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$$

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

3. Cálculo do sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Atualização dos coeficientes do filtro

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

2.6.1 Representação gráfica de fluxo de sinal do algoritmo LMS

Combinando (2.70) e (2.74), podemos expressar a evolução do vetor de peso no algoritmo LMS como

$$\begin{aligned}\hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \mu\mathbf{x}(n)[d(n) - \mathbf{x}^T(n)\hat{\mathbf{w}}(n)] \\ &= [\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n)]\hat{\mathbf{w}}(n) + \mu\mathbf{x}(n)d(n)\end{aligned}\quad (2.75)$$

onde \mathbf{I} é a matriz identidade. Ao usar o algoritmo LMS, reconhecemos que

$$\hat{\mathbf{w}}(n) = z^{-1}[\hat{\mathbf{w}}(n+1)] \quad (2.76)$$

onde z^{-1} é o operador de atraso de tempo unitário, implicando armazenamento. Usando (2.75) e (2.76) podemos assim representar o algoritmo LMS pelo gráfico de fluxo de sinal mostrado na Figura 2.8.

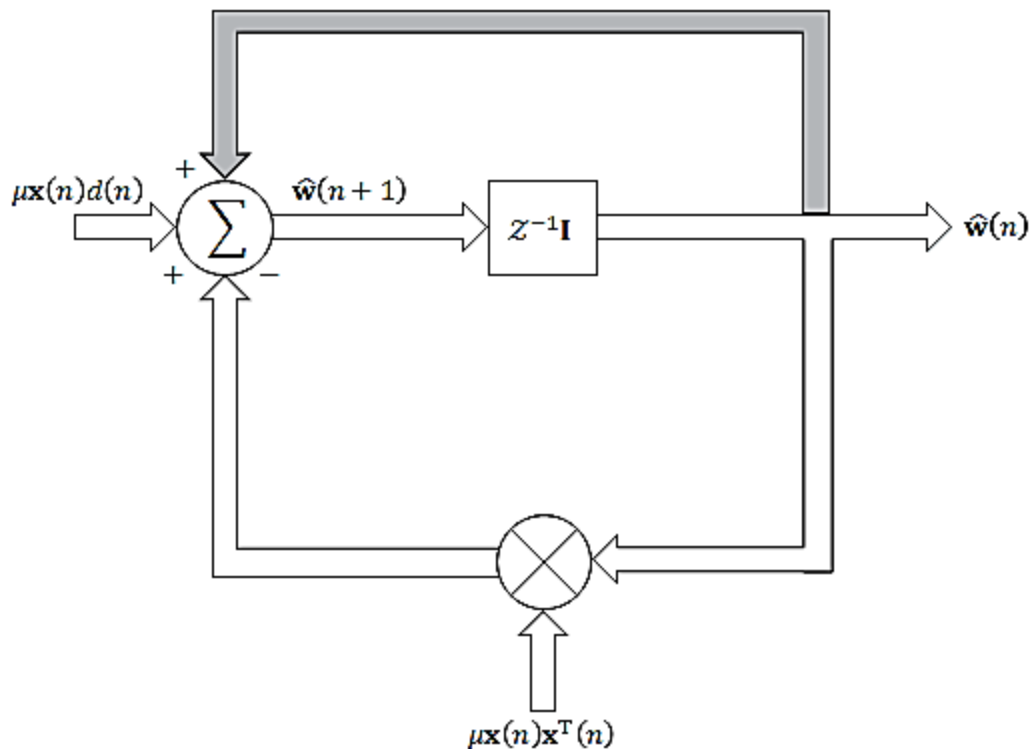


Figura 2.8 – Gráfico de fluxo de sinal representação do algoritmo LMS. O gráfico incorpora *feedback* representado na parte hachurada.

Este gráfico de fluxo de sinal revela que o algoritmo LMS é um exemplo de sistema de *feedback* estocástico. A presença de *feedback* tem um impacto profundo no comportamento de convergência do algoritmo LMS.

2.7 ALGORITMO LMS NORMALIZADO - NLMS

A velocidade de convergência do algoritmo LMS pode ser melhorada, uma opção para que isso ocorra é a realização de modificações no fator que determina o passo de convergência, ou seja, na constante μ (Martins, 2020). O algoritmo LMS Normalizado, que impõe modificação no fator de convergência, em geral, converge mais rápido que o algoritmo LMS (Diniz, 2008, p. 114).

A regra de atualização do algoritmo NLMS pode ser obtida considerando $\mathbf{w}(n+1)$ que minimiza a seguinte função objetivo $\varphi(n)$,

$$\varphi(n) = [\mathbf{w}(n+1) - \mathbf{w}(n)]^T [\mathbf{w}(n+1) - \mathbf{w}(n)], \quad (2.77)$$

restrita a condição

$$\mathbf{w}(n+1)^T \mathbf{x}(n) = d(n). \quad (2.78)$$

Para solucionar esse problema de otimização com restrição, utiliza-se os multiplicadores de Lagrange de tal forma que o problema com restrição se torna o seguinte problema sem restrição:

$$J(n) = [\mathbf{w}(n+1) - \mathbf{w}(n)]^T [\mathbf{w}(n+1) - \mathbf{w}(n)] - \lambda [d(n) - \mathbf{w}(n+1)^T \mathbf{x}(n)]. \quad (2.79)$$

Desse modo, para encontrar $\mathbf{w}(n+1)$ que minimiza $J(n)$ basta resolver

$$\frac{\partial J(n)}{\partial \mathbf{w}(n+1)} = 0. \quad (2.80)$$

Resolvendo a Equação (2.80), tem-se:

$$2[\mathbf{w}(n+1) - \mathbf{w}(n)] - \lambda \mathbf{x}(n) = 0, \quad (2.81)$$

e multiplicando (à esquerda) toda a Equação (2.81) por $\mathbf{x}^T(n)$ obtém-se:

$$2[\mathbf{x}^T(n) \mathbf{w}(n+1) - \mathbf{x}^T(n) \mathbf{w}(n)] - \lambda \mathbf{x}^T(n) \mathbf{x}(n) = 0. \quad (2.82)$$

Isolando λ em (2.82) e lembrando que $\mathbf{x}^T(n) \mathbf{x}(n)$ é a norma-2 ao quadrado de $\mathbf{x}(n)$, chega-se a:

$$\lambda = \frac{2}{|\mathbf{x}(n)|_2^2} [\mathbf{x}^T(n) \mathbf{w}(n+1) - \mathbf{x}^T(n) \mathbf{w}(n)]. \quad (2.83)$$

Aplicando as propriedades de transposição de matrizes, a Equação (2.83) torna-se:

$$\lambda = \frac{2}{|\mathbf{x}(n)|_2^2} [\mathbf{w}(n+1)^T \mathbf{x}(n) - \mathbf{w}^T(n) \mathbf{x}(n)], \quad (2.84)$$

e lembrando da restrição da função objetivo $\mathbf{w}(n+1)^T \mathbf{x}(n) = d(n)$, chega-se a seguinte expressão para λ :

$$\lambda = \frac{2}{|\mathbf{x}(n)|_2^2} [d(n) - y(n)], \quad (2.85)$$

e conseqüentemente,

$$\lambda = \frac{2}{|\mathbf{x}(n)|_2^2} e(n). \quad (2.86)$$

Substituindo λ dado em (2.86) na Equação (2.82), e isolando $\mathbf{w}(n+1)$ tem-se:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{2}{\|\mathbf{x}(n)\|_2^2} e(n)\mathbf{x}(n) \quad (2.87)$$

Para realizar o controle sobre o ajuste dos coeficientes de vetor $\mathbf{w}(n)$, introduz-se um parâmetro de passo μ , e para evitar divisão por zero o parâmetro $\varepsilon > 0$ é introduzido. Portanto, a expressão de atualização dos coeficientes do filtro NLMS tornar-se da seguinte forma:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{2\mu}{\mathbf{x}^T(n)\mathbf{x}(n) + \varepsilon} e(n)\mathbf{x}(n). \quad (2.88)$$

A estabilidade e convergência do algoritmo NLMS é garantida para $0 < \mu < 2$ (Haykin, 1996). O algoritmo apresentado na Tabela 2.2, lista o passo a passo para implementação do filtro NLMS desde a sua inicialização e parâmetros, até a atualização dos coeficientes do filtro.

Tabela 2.2 - Sumário do algoritmo adaptativo NLMS.

Algoritmo 2.2 NLMS

1. Inicialização e parâmetros

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

$$0 < \mu < 2; \quad \varepsilon > 0$$

2. Dados de entrada e saída da planta e do filtro adaptativo

$$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$$

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

3. Cálculo do sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Atualização dos coeficientes do filtro

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{2\mu}{\varepsilon + \mathbf{x}^T(n)\mathbf{x}(n)} e(n)\mathbf{x}(n)$$

Comparando os algoritmos NLMS e LMS percebe-se que a diferença entre os dois consiste no fato de o passo de adaptação, agora no NLMS, ser normalizado pela norma do vetor \mathbf{x} no instante n . A partir do NLMS, diversos algoritmos têm sido propostos na literatura, entre os quais figuram os proporcionais, objeto de estudo do próximo capítulo.

Capítulo 3

Algoritmos Proporcionais e *Zero-Attracting*

Neste capítulo são apresentadas construções teóricas dos algoritmos proporcionais e dos pertencentes à família *zero-attracting* (com atração para zero). Entre os algoritmos descritos nesse capítulo destaca-se o IAF-PNLMS, pois este faz parte do centro da proposta principal dessa pesquisa.

3.1 ALGORITMOS PROPORCIONAIS

Nessa seção, são apresentados os principais algoritmos adaptativos proporcionais. Vale lembrar que algoritmo LMS, apesar de simples e eficaz, possui uma convergência lenta quando a resposta ao impulso é de natureza esparsa (Duttweiler, 2000). A fim de superar essa desvantagem, novos algoritmos foram criados, como o algoritmo PNLMS. Neste algoritmo, a regra de adaptação dos coeficientes $\mathbf{w}(n)$ é dada por (Branco, 2016):

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n) + \varepsilon}, \quad (3.1)$$

onde μ é o passo de adaptação e $\varepsilon > 0$ é um parâmetro de regularização numérica para estabilizar a solução. A matriz

$$\mathbf{G}(n) = \text{diag} [g_1(n) \quad g_2(n) \quad \dots \quad g_N(n)], \quad (3.2)$$

de ordem $N \times N$ é responsável pela distribuição de ganhos individuais $g_i(n)$, controlando o ajuste do i -ésimo coeficiente do algoritmo. O operador diag define uma matriz diagonal cujo os elementos $g_1(n), g_2(n), \dots, g_N(n)$ compõem a diagonal principal.

Comparando as regras de atualização dos coeficientes dadas nas Equações (2.88) e (3.1), algoritmos NLMS e PNLMS respectivamente, percebe-se que o algoritmo NLMS pode ser obtido a partir do PNLMS tomando $\mathbf{G}(n)$ igual a matriz identidade, ou seja, ganhos individuais iguais a 1 para todos os coeficientes do filtro.

3.1.1 Algoritmo PNLMS

Com o objetivo de melhorar o desempenho dos filtros adaptativos nas tarefas de identificação e controle de plantas esparsas foi desenvolvido o algoritmo adaptativo PNLMS, pois o passo de adaptação é proporcional a cada coeficiente do filtro (Yu; Zhao; Chen, 2015).

As Equações (3.1) e (3.2) são a base para os algoritmos da família proporcional, sendo que as diferentes versões de algoritmos desta família são obtidas modificando-se a maneira na qual o ganho individual $g_i(n)$ é calculado. No caso do algoritmo PNLMS padrão, o ganho $g_i(n)$ associado ao i -ésimo coeficiente $w_i(n)$ é obtido a partir de

$$g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}, \quad i = 1, 2, \dots, N, \quad (3.2)$$

em que a função de proporcionalidade $\phi_i(n)$ é definida como

$$\phi_i(n) = \max[f(n), |w_i(n)|], \quad (3.3)$$

sendo

$$f(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_\infty] \quad (3.4)$$

o fator de ativação, o qual influencia diretamente na adaptação dos coeficientes considerados inativos. Tal fator depende da norma infinita do vetor de coeficientes do filtro, $\|\mathbf{w}(n)\|_\infty$, e dos parâmetros de inicialização, δ , e de proporcionalidade, ρ (Souza *et al.*, 2009). O algoritmo apresentado na Tabela 3.1, lista os passos para implementação do filtro PNLMS padrão desde a sua inicialização até a atualização dos coeficientes do filtro.

Tabela 3.1 - Sumário do algoritmo adaptativo PNLMS padrão.

Algoritmo 3.1 PNLMS padrão	
1. Inicialização	$\mathbf{w}(0) = \mathbf{0}$
e parâmetros	$0 < \mu < 2; \delta > 0; \rho > 0; \varepsilon > 0$
2. Obtenção dos dados da entrada e saída da planta	$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$
e do filtro adaptativo	

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

3. Sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Fator de ativação

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

5. Função de proporcionalidade

$$\phi_j(n) = \max [f(n), |w_i(n)|], i = 1, 2, \dots, N$$

6. Ganho individual dos coeficientes do filtro adaptativo

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

7. Matriz de ganhos individuais ($N \times N$)

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

8. Atualização dos coeficientes do filtro adaptativo

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

A regra de adaptação proposta no PNLMS é menos eficaz do que a regra do NLMS para situações de plantas esparsas. Para superar essa deficiência foi proposto o algoritmo PNLMS aprimorado, o qual se mostrou eficaz independente da resposta ao impulso (Benesty; Gay, 2002). O algoritmo IPNLMS será mais bem discutido no subtópico a seguir.

3.1.2 Algoritmo IPNLMS

O algoritmo IPNLMS tem origem no PNLMS padrão, sendo que na função de ativação substitui-se $\|\mathbf{w}(n)\|_\infty$ por $\|\mathbf{w}(n)\|_1$ (norma-1 do vetor de coeficientes do filtro) e adota-se a constante de proporcionalidade $\rho = 1/N$ (Souza, 2012). Assim, tem-se um novo fator de ativação dado por:

$$f(n) = \begin{cases} \frac{1}{N} \max[\xi, \|\mathbf{w}(n)\|_1], & n = 0 \\ \frac{\|\mathbf{w}(n)\|_1}{N}, & n \geq 1 \end{cases} \quad (3.5)$$

Consequentemente, para $n \geq 1$, a Equação (3.3) pode ser reescrita da seguinte forma

$$\phi_i(n) = \max \left[\frac{\|\mathbf{w}(n)\|_1}{N}, |w_i(n)| \right], \quad i = 1, 2, \dots, N. \quad (3.6)$$

Uma modificação na função de proporcionalidade dada por (3.6) também é estabelecida, fazendo com que a mesma seja definida como uma média ponderada no lugar do operador $\max [\cdot]$. Assim, a nova função de proporcionalidade torna-se:

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

em que $-1 \leq \alpha < 1$ é um fator de ponderação.

Agora, desenvolvendo o denominador de (3.2) obtém-se:

$$\sum_{i=1}^N \phi_i(n) = \sum_{i=1}^N \left[(1 - \alpha) \frac{\|\mathbf{w}(n)\|_1}{N} + (1 + \alpha) |w_i(n)| \right] \quad (3.8)$$

$$= \left[\frac{1 - \alpha}{N} \sum_{i=1}^N \|\mathbf{w}(n)\|_1 \right] + \left[(1 + \alpha) \sum_{i=1}^N |w_i(n)| \right] \quad (3.9)$$

$$= (1 - \alpha) \|\mathbf{w}(n)\|_1 + (1 + \alpha) \|\mathbf{w}(n)\|_1 \quad (3.10)$$

$$= 2 \|\mathbf{w}(n)\|_1. \quad (3.11)$$

Então, substituindo (3.7) e (3.11) em (3.2) obtém-se o ganho individual do algoritmo IPNLMS como sendo

$$g_i(n) = (1 - \alpha) \frac{1}{2N} + (1 + \alpha) \frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \zeta} \quad (3.12)$$

onde $\zeta > 0$ é um parâmetro de regularização usado para evitar divisão por zero. O fator α é denominado parâmetro de proporcionalidade.

A regra de atualização dos coeficientes do algoritmo IPNLMS é dada por (3.1), com os elementos de $\mathbf{G}(n)$ sendo calculados a partir de (3.12). O algoritmo apresentado na Tabela 3.2, detalha os passos a ser seguido para a implementação do filtro IPNLMS padrão.

Tabela 3.2 - Sumário do algoritmo adaptativo IPNLMS.

Algoritmo 3.2 IPNLMS

1. Inicialização e parâmetros

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

e parâmetros

$$0 < \mu < 2$$

$$\varepsilon > 0$$

$$-1 \leq \alpha < 1$$

$$\varsigma > 0$$

2. Dados de entrada e saída da planta

$$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$$

e do filtro adaptativo

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

3. Sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Ganho individual dos coeficientes do filtro adaptativo

$$g_i(n) = (1 - \alpha) \frac{1}{2N} + (1 + \alpha) \frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varsigma}, \quad i = 1, 2, 3, \dots, N$$

5. Matriz de ganhos individuais ($N \times N$)

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

6. Atualização dos coeficientes do filtro adaptativo

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

Para plantas com elevada esparsidade, é possível melhorar a velocidade de convergência e as repostas a perturbações na planta dos algoritmos PNLMS e IPNLMS. Essa melhora pode ser feita atribuindo fatores de ativação individuais aos coeficientes do filtro (SOUSA, 2009). Nesse sentido, o filtro IAF-PNLMS, proposto por (SOUSA, 2009), é discutido no subtópico a seguir.

3.1.3 Algoritmo IAF-PNLMS

O algoritmo IAF-PNLMS (PNLMS com fator de ativação individual), projetado por Souza (2009), realiza uma abordagem que se contrapõe à formulação do PNLMS no que

tange a tarefa de calcular o fator de ativação. Esse algoritmo apresenta as seguintes características (Souza, 2009):

- a) Um fator de ativação individual para cada coeficiente do filtro adaptativo.
- b) Cada fator de ativação individual é calculado em função da magnitude do correspondente coeficiente.
- c) 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.

A proposta principal do IAF-PNLMS é obter um algoritmo completamente proporcional, associando cada ganho atribuído a um coeficiente inativo com um fator de ativação individual $f_i(n)$ em lugar de um fator comum para todos os coeficientes inativos. Nesse contexto, as condições requeridas para cada fator de ativação individual $f_i(n)$ são estabelecidas como segue (Souza, 2009):

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

$$\lim_{n \rightarrow \infty} [f_i(n) - |w_i(n)|] = 0, \quad i = 1, 2, \dots, N. \quad (3.13)$$

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

$$f_i(n) > 0, \quad i = 1, 2, \dots, N. \quad (3.14)$$

Desse modo, Souza et al. 2010 define o algoritmo IAF-PNLMS como sendo uma variante do PNLMS em que o fator de ativação é uma função individual definida para cada coeficiente do filtro adaptativo da seguinte forma:

$$f_i(n) = \begin{cases} \frac{1}{2} |w_i(n)| + \frac{1}{2} \phi_i(n-1), & n = mN, m = 1, 2, \dots \\ f_i(n-1) & \text{caso contrário} \end{cases} \quad (3.15)$$

De acordo com Souza *et al.* (2009), o algoritmo IAF-PNLMS fornece uma melhor distribuição de ganho de adaptação do que o PNLMS, levando a um aumento na velocidade de convergência do algoritmo em aplicações com plantas de alta esparsidade.

3.2 ALGORITMOS DA FAMÍLIA ZERO- ATTRACTING

Nessa seção, serão apresentados os principais algoritmos que incorporam o princípio do atrator para zero, ou seja, os algoritmos da família ZA. Tais algoritmos aceleram a convergência quando aplicados na identificação de plantas esparsas (Chen, 2009).

3.2.1 Algoritmo ZA-LMS

A fim de apresentar uma alternativa para identificar sistemas esparsos, Chen *et al.* (2009) introduzem, na função de custo do LMS, uma penalidade que favorece a esparsidade da resposta ao impulso, incorporando assim um termo de norma l_1 na função de custo do algoritmo. Essa modificação resulta em uma versão modificada do LMS, ou seja, um atrator para zero para todos os coeficientes. A esse algoritmo, Chen *et al.* (2009) denominam de ZA-LMS. Com essa modificação, a função de custo do ZA-LMS tornar-se

$$L_1(n) = \frac{1}{2} e^2(n) + \gamma \|\mathbf{w}(n)\|_1 \quad (3.16)$$

Agora, calculando o gradiente da função de custo (3.16), tem-se

$$\nabla_{\mathbf{w}(n)} L_1(n) = \frac{1}{2} \nabla_{\mathbf{w}(n)} e^2(n) + \gamma \nabla_{\mathbf{w}(n)} \|\mathbf{w}(n)\|_1. \quad (3.17)$$

Sabendo que

$$\nabla_{\mathbf{w}(n)} \|\mathbf{w}(n)\|_1 = \text{sgn}[\mathbf{w}(n)] \quad (3.18)$$

e

$$\nabla_{\mathbf{w}(n)} e^2(n) = -e(n)\mathbf{x}(n), \quad (3.19)$$

Substitui-se (3.18) e (3.19) em (3.17) obtendo

$$\nabla_{\mathbf{w}(n)} L_1(n) = -e(n)\mathbf{x}(n) + \gamma \text{sgn}[\mathbf{w}(n)], \quad (3.20)$$

em que a função $\text{sgn}(w_i)$ é definida como

$$\text{sgn}_{(w_i)} = \begin{cases} \frac{w_i}{|w_i|} & w_i \neq 0 \\ 0 & w_i = 0. \end{cases} \quad (3.21)$$

Logo, pelo método do Gradiente descendente, escreve-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} L_1(n) \quad (3.22)$$

da seguinte forma

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) - \rho \operatorname{sgn}[\mathbf{w}(n)], \quad (3.23)$$

em que $\rho = \mu\gamma$.

Dessa forma, obtém-se a expressão de atualização dos coeficientes do algoritmo ZA-LMS dada pela Equação (3.23).

3.2.2 Algoritmo RZA-LMS

No algoritmo ZA-LMS, todos os coeficientes são uniformemente atraídos a zero, e isso leva a uma redução de desempenho em aplicações com plantas dispersivas (menor grau de esparsidade). Chen *et al.* (2009), propõem uma abordagem chamada LMS com atrator para zero reponderado (RZA-LMS – *reweighted zero-attracting LMS*).

O RZA-LMS origina-se então de uma nova função de custo dada por

$$L_2(n) = \frac{1}{2}e^2(n) + \gamma \sum_{i=1}^N \log(1 + \varepsilon|w_i(n)|), \quad (3.24)$$

com $\varepsilon > 0$. Calculando o gradiente $\nabla_{\mathbf{w}(n)}$ de $L_2(n)$, tem-se

$$\nabla_{\mathbf{w}(n)}L_2(n) = \frac{1}{2}\nabla_{\mathbf{w}(n)}e^2(n) + \gamma\nabla_{\mathbf{w}(n)} \sum_{i=1}^N \log(1 + \varepsilon|w_i(n)|). \quad (3.25)$$

A primeira parcela do lado direito de (3.25) é dada por (3.18), restando agora determinar a segunda parcela que, para fins didáticos, é denominada

$$\mathbf{S}_w = \nabla_{\mathbf{w}(n)} \sum_{i=1}^N \log(1 + \varepsilon|w_i(n)|). \quad (3.26)$$

Abrindo o somatório em (3.26), resulta em

$$\mathbf{S}_w = \nabla_{\mathbf{w}(n)} [\log(1 + \varepsilon|w_1(n)| + \log(1 + \varepsilon|w_2(n)| \dots + \log(1 + \varepsilon|w_N(n)|)]. \quad (3.27)$$

Assim, utilizando a definição de $\nabla_{\mathbf{w}(n)}$, obtém-se

$$\mathbf{S}_w = \left[\frac{\varepsilon \operatorname{sgn}\{w_1(n)\}}{1 + \varepsilon|w_1(n)|}, \frac{\varepsilon \operatorname{sgn}\{w_2(n)\}}{1 + \varepsilon|w_2(n)|}, \dots, \frac{\varepsilon \operatorname{sgn}\{w_N(n)\}}{1 + \varepsilon|w_N(n)|} \right]^T \quad (3.28)$$

Desse modo, (3.25) pode ser escrita como

$$\nabla_{\mathbf{w}(n)}L_2(n) = \gamma\mathbf{S}_w - e(n)\mathbf{x}(n). \quad (3.29)$$

Considerando o método do gradiente descendente, tem-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} L_2(n) \quad (3.30)$$

e, substituindo (3.29) em (3.30), chega-se a expressão de atualização dos coeficientes do RZA-LMS, que é dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \rho \mathbf{S}_w + \mu e(n) \mathbf{x}(n), \quad (3.31)$$

em que $\rho = \mu\gamma$.

3.2.3 Algoritmo ZA-PNLMS

A ideia de incorporar uma penalidade também foi utilizada no algoritmo PNLMS. Em Das *et al.* (2016), visando obter uma regra de atualização dos coeficientes do PNLMS com atrator para zero, o seguinte problema de otimização com restrição é proposto:

$$\min \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_{\mathbf{G}^{-1}}^2 + \gamma \|\mathbf{G}^{-1} \mathbf{w}(n+1)\|_1 \quad (3.32)$$

sujeito a restrição

$$d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n) = 0, \quad (3.33)$$

em que $\gamma \ll 1$ e a notação $\|\mathbf{v}\|_{\mathbf{G}^{-1}}^2$ indica o produto interno generalizado $\mathbf{v}^T \mathbf{G}^{-1} \mathbf{v}$.

Utilizando (3.32) e (3.33) e o multiplicador de Lagrange λ , tem-se a seguinte função de custo a ser minimizada:

$$J(n+1) = \|\mathbf{w}(n+1) - \mathbf{w}(n)\|_{\mathbf{G}^{-1}}^2 + \gamma \|\mathbf{G}^{-1} \mathbf{w}(n+1)\|_1 + \lambda [(d(n) - \mathbf{w}^T(n+1) \mathbf{x}(n))]. \quad (3.34)$$

Para minimização de (3.34), toma-se

$$\frac{\partial J(n+1)}{\partial \mathbf{w}(n+1)} = 0 \quad (3.35)$$

e conseqüentemente

$$\mathbf{G}^{-1} [\mathbf{w}(n+1) - \mathbf{w}(n)] \mathbf{G}^{-1} + \gamma \operatorname{sgn}(\mathbf{w}(n+1)) - \lambda \mathbf{x}(n) = 0. \quad (3.36)$$

Pré-multiplicando (3.36) por \mathbf{G} e, em seguida, isolando $\mathbf{w}(n+1)$, resulta

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \gamma \operatorname{sgn}(\mathbf{w}(n+1)) + \lambda \mathbf{G}(n) \mathbf{x}(n). \quad (3.37)$$

Pré-multiplicando (3.37) por $\mathbf{x}^T(n)$, usando a restrição (3.33) e isolando λ , obtém-se

$$\lambda = \frac{e(n) + \gamma \mathbf{x}^T(n) \text{sgn}(\mathbf{w}(n+1))}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)}. \quad (3.38)$$

Agora, substituindo (3.38) em (3.37), tem-se:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n) \mathbf{G}(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} - \gamma \left[\mathbf{I} - \frac{\mathbf{G}(n) \mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} \right] \text{sgn}(\mathbf{w}(n+1)), \quad (3.39)$$

em que

$$\mathbf{B} = \frac{\mathbf{G}(n) \mathbf{x}(n) \mathbf{x}^T(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n)} \quad (3.40)$$

Das *et al.* (2016) sugerem que para se obter a equação de atualização de coeficientes, deve-se aproximar $\text{sgn}(\mathbf{w}(n+1))$ por uma estimativa, a saber, $\text{sgn}(\mathbf{w}(n))$ que é conhecido. Essa aproximação baseia-se na hipótese de que a maioria dos coeficientes não sofre mudança de sinal à medida que são atualizados de uma iteração para a outra. Ressalta-se também o fato de que qualquer elemento da matriz \mathbf{B} possui magnitude muito menor do que 1, especialmente para filtros de grande ordem e, assim, essa matriz pode ser desprezada em comparação com \mathbf{I} .

Por fim, introduzindo o tamanho do passo do algoritmo μ e um parâmetro de regularização δ_p em (3.39), obtém-se então a seguinte equação de atualização dos coeficientes do algoritmo ZA-PNLMS:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n) \mathbf{G}(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n) + \delta_p} - \rho \text{sgn}(\mathbf{w}(n)), \quad (3.41)$$

em que $\rho = \mu\gamma$.

Capítulo 4

Metodologia Proposta e Métricas de Avaliação

Neste capítulo o algoritmo UT-ZA-PNLMS é apresentado, pois este está diretamente ligado à proposta. Para tal, uma introdução teórica sobre estimação de parâmetros é apresentada, dando ênfase ao método da máxima verossimilhança, pois o algoritmo mencionado tem sua construção baseada no referido método. Em seguida é exposta a formulação teórico-matemática do algoritmo proposto, IAF-UT-ZA-PNLMS, e por fim as métricas de avaliação são apresentadas.

4.1 MÉTODO DE ESTIMAÇÃO DE PARÂMETROS MÁXIMA VEROSSIMILHANÇA

A aplicação do método da máxima verossimilhança para estimar parâmetros, inicia com a definição de um modelo de variável aleatório discreto (através de uma função de probabilidade) ou contínuo (por meio de uma função densidade de probabilidade) (Bolfarine, 2001, p. 35). Após definido o modelo, aplica-se o conceito de função de verossimilhança. Uma função de verossimilhança $L(\theta; X)$ é definida como

$$L(\theta; X) = \prod_{i=1}^n f(x_i|\theta) \quad (4.1)$$

em que $f(x|\theta)$ é uma função densidade de probabilidade (ou apenas uma função de probabilidade) da variável aleatória X , e θ é um elemento do espaço paramétrico (Bolfarine, 2001, p. 35).

Desse modo, o estimador de máxima verossimilhança de θ é o valor $\hat{\theta}$ que maximiza a função de verossimilhança $L(\theta; X)$.

Segundo (Bolfarine, 2001) o procedimento para encontrar $\hat{\theta}$ inicia com a aplicação do logaritmo natural na função de verossimilhança de θ de tal modo que

$$l(\theta; X) = \log L(\theta; X). \quad (4.2)$$

No caso uniparamétrico onde o espaço paramétrico é um intervalo da reta e $l(\theta; X)$ é derivável, o estimador de máxima verossimilhança pode ser encontrado como a raiz da equação de verossimilhança (Bolfarine, 2001, p. 35).

$$l'(\theta; X) = \frac{\partial l(\theta; X)}{\partial \theta} = 0. \quad (4.3)$$

Em alguns exemplos, a solução da equação de verossimilhança pode ser obtida explicitamente. Em situações mais complicadas, a solução da equação é, em geral, obtida por procedimentos numéricos.

Vale lembrar que para se concluir que a solução da equação é um ponto de máximo, é necessário fazer o teste da segunda derivada, ou seja, verificar que

$$l''(\hat{\theta}; X) = \frac{\partial^2 \log L(\theta; X)}{\partial \theta^2} \Big|_{\theta=\hat{\theta}} < 0. \quad (4.4)$$

Para exemplificar a aplicação do método da máxima verossimilhança consideremos X_1, \dots, X_n uma amostra aleatória da distribuição da variável aleatória $X \sim N(\mu, 1)$.

No exemplo mencionado, a função de verossimilhança é dada por

$$L(\mu; X) = \left(\frac{1}{\sqrt{2\pi}} \right)^n e^{-\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2}, \quad (4.5)$$

com o espaço paramétrico $\Theta = \{\mu; -\infty < \mu < \infty\}$. Como

$$l(\mu; X) = n \log \sqrt{2\pi} - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2, \quad (4.6)$$

então a equação de verossimilhança é dada por

$$\sum_{i=1}^n (x_i - \hat{\mu}) = 0. \quad (4.7)$$

Portanto, o estimador de máxima verossimilhança de μ é dado por

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n (X_i) = \bar{X}. \quad (4.8)$$

Então \bar{X} é, para a variável aleatória $X \sim N(\mu, 1)$, o estimador de máxima verossimilhança para μ .

4.2 ALGORITMO UT-ZA-PNLMS

No algoritmo UT-ZA-PNLMS é proposto a introdução de uma função de limite superior adaptativa $\tau_U(n)$ para os ganhos dos coeficientes do filtro (Das, 2020). A função que limita superiormente os ganhos do filtro é obtida através do método de estimação de parâmetros conhecido como “Método da Máxima Verossimilhança”, descrito anteriormente.

A fim de estabelecer uma regra matemática para estimar o limite superior dos ganhos do filtro $\tau_U(n)$, Das (2020) supôs que o vetor de pesos \mathbf{w} é uma variável aleatória com função densidade de probabilidade dada por

$$f(w|\theta_n) = \frac{1}{\sigma_w(n)\sqrt{2\pi}} e^{-\frac{(w-\mu_w(n))^2}{2\sigma_w^2(n)}} \quad (4.9)$$

e função de verossimilhança

$$L(S_n|\theta_n, D_n) = \prod_{i=0}^{N-1} f(|w_i(n)||\theta_n)^{|d_i(n)|} \quad (4.10)$$

em que $\theta_n = (\mu_w(n), \sigma_w^2(n))$ com média $\mu_w(n)$ e variância $\sigma_w^2(n)$ corresponde ao espaço paramétrico, $S_n = (|W_0(n)|, \dots, |W_n - 1(n)|)$, $D_n = (|d_0(n)|, \dots, |d_n - 1(n)|)$ e

$$d_i(n) = \lambda_d d_i(n-1) + (1 - \lambda_d) \frac{e(n)\mathbf{x}(n-i)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_N}, \quad (4.11)$$

onde d_i é um parâmetro que estima a mudança esperada no peso do i -ésimo coeficiente do filtro, λ_d sendo um fator de ponderação exponencial com $0 < \lambda_d < 1$ e δ_N uma pequena constante positiva para evitar a divisão por zero.

Seguindo com a aplicação do método da máxima verossimilhança, aplica-se o logaritmo natural em $L(S_n|\theta_n, D_n)$ e obtém-se

$$\begin{aligned} \log L(S_n|\theta_n, D_n) &= \log\left(\frac{1}{(\sigma_w(n)\sqrt{2\pi})^{\sum_{i=0}^{N-1}|d_i(n)|}} \prod_{i=0}^{N-1}|d_i(n)| - \sum_{i=0}^{N-1}|d_i(n)| \frac{(|w_i(n)| - \mu_w(n))^2}{2\sigma_w^2(n)}\right) \\ &= \log\left(\frac{1}{(\sigma_w(n)\sqrt{2\pi})^{\sum_{i=0}^{N-1}|d_i(n)|}} \prod_{i=0}^{N-1}|d_i(n)| - \frac{1}{2\sigma_w^2(n)} \sum_{i=0}^{N-1}|d_i(n)||w_i(n)|^2\right. \\ &\quad \left. - \frac{\mu_w^2(n)}{2\sigma_w^2(n)} \sum_{i=0}^{N-1}|d_i(n)| + \frac{\mu_w(n)}{\sigma_w^2(n)} \sum_{i=0}^{N-1}|d_i(n)||w_i(n)|\right). \end{aligned} \quad (4.12)$$

Agora, tomando a derivada parcial da função de custo acima em relação à $\mu_w(n)$, obtemos:

$$\frac{\partial \log L(S_n|\theta_n, D_n)}{\partial \mu_w(n)} = \frac{1}{\sigma_w^2(n)} \sum_{i=0}^{N-1} |d_i(n)| |w_i(n)| - \frac{\mu_w(n)}{\sigma_w^2(n)} \sum_{i=0}^{N-1} |d_i(n)|. \quad (4.13)$$

Para maximizar $\log L(S_n|\theta_n, D_n)$, a derivada parcial acima é igualada a zero, e, portanto, considerando a média estimada como $\hat{\mu}_w(n)$ e variância estimada como $\hat{\sigma}_w^2(n)$, nós temos:

$$\frac{1}{\hat{\sigma}_w^2(n)} \sum_{i=0}^{N-1} |d_i(n)| |w_i(n)| - \frac{\hat{\mu}_w(n)}{\hat{\sigma}_w^2(n)} \sum_{i=0}^{N-1} |d_i(n)| = 0. \quad (4.14)$$

Resolvendo a equação acima para $\hat{\mu}_w(n)$, encontramos:

$$\hat{\mu}_w(n) = \frac{\sum_{i=0}^{N-1} |d_i(n)| |w_i(n)|}{\sum_{i=0}^{N-1} |d_i(n)|}. \quad (4.15)$$

Portanto, o parâmetro de limite $\tau_U(n)$ pode ser definido como

$$\tau_U(n) = \hat{\mu}_w(n) = \frac{\sum_{i=0}^{N-1} |d_i(n)| |w_i(n)|}{\sum_{i=0}^{N-1} |d_i(n)|}. \quad (4.16)$$

É importante mencionar que, no estágio inicial do processo de adaptação, $d_i(n)$ pode não ter uma estimativa precisa do status da derivação correspondente. Portanto, é desejável que todos os coeficientes do filtro possam crescer com seus ganhos proporcionais padrão, ou seja, $\tau_U(n)$ deve ter um grande valor inicial. Para que isso ocorra um segundo termo é adicionado a $\tau_U(n)$ e o mesmo passa a assumir a forma da Equação (4.17).

$$\tau_U(n) = \frac{\sum_{i=0}^{N-1} |w_i(n)| |d_i(n)|}{\sum_{i=0}^{N-1} |d_i(n)|} + \beta_1 (1 - e^{\beta_2 \epsilon(n)}), \quad (4.17)$$

onde $\epsilon(n) = \lambda_e(n-1) + (1 - \lambda_e)e^2(n)$, que estima o quadrado do erro (ou seja, $e^2(n)$) com fator de esquecimento λ_e , e β_1 e β_2 são os parâmetros de controle. Assim como λ_d , λ_e também pode ser definido em torno de 0,9. Segundo Das (2020), o desempenho do algoritmo proposto é pouco sensível à escolha de β_1 e β_2 .

Desse modo, a função de proporcionalidade passa a ser

$$\phi_i(n) = \min\{\tau_U(n), \max[f(n), |w_i(n)|]\}, \quad (4.18)$$

em que o fator de ativação $f(n)$ é dado por

$$f(n) = \rho_g \max[\delta_g, |w_1(n)|, \dots, |w_N(n)|] \quad (4.19)$$

e cada ganho individual é

$$g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}, \quad i = 1, 2, \dots, N, \quad (4.20)$$

com a matriz de ganhos dada por

$$\mathbf{G}(n) = \text{diag}[g_1(n), g_2(n) \dots g_N(n)]. \quad (4.21)$$

O algoritmo UT-ZA-PNLMS melhora a taxa de convergência oferecendo maiores ganhos para coeficientes ativos de menor magnitude à medida que as iterações progridem, e reduz as flutuações nos coeficientes ativos de maior magnitude (Das, 2020). A Tabela 4.1 apresenta o sumário do algoritmo adaptativo UT-ZA-PNLMS.

Tabela 4.1 - Sumário do algoritmo adaptativo UT-ZA-PNLMS

Algoritmo 4.1 UT-ZA-PNLMS

1. Inicialização

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

e parâmetros

$$\begin{aligned} 0 < \mu < 2 \\ -1 \leq \alpha < 1 \\ \delta > 0 \end{aligned}$$

2. Dados de entrada e saída da planta

$$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$$

e do filtro adaptativo

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

3. Sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Calcule $d_i(n)$, $\tau_U(n)$ e $g_i(n)$ para (4.11), (4.17) e (4.18), respectivamente.

5. Calcule $g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}$, onde, $i = 1, 2, \dots, N$,

6. Monte a matriz de ganhos $\mathbf{G}(n) = \text{diag} [g_0(n), g_1(n) \dots g_{N-1}(n)]$

7. Atualize os pesos

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n) \mathbf{G}(n) \mathbf{x}(n)}{\mathbf{x}^T(n) \mathbf{G}(n) \mathbf{x}(n) + \delta \rho} - \rho \operatorname{sgn}[\mathbf{w}(n)],$$

4.3 METODOLOGIA PROPOSTA

A proposta dessa pesquisa é implementar uma regra de atualização que leve os fatores de ativação individuais $f_i(n)$ do algoritmo UT-ZA-PNLMS tenderem às respectivas magnitudes $|w_i(n)|$. Para tal, toma-se como base o procedimento adotado por Souza *et al.* (2009), o qual estabelece duas condições (C1 e C2) para obter fatores de ativação individuais em um algoritmo proporcional. Assim, considere as condições:

- C1) $f_i(n)$ deve convergir para a correspondente magnitude do coeficiente $|w_i(n)|$, ou seja,

$$[|w_i(n)| - f_i(n)] \rightarrow 0, \text{ quando } n \rightarrow \infty \quad (4.22)$$

- C2) $f_i(n)$ deve sempre ser maior do que zero.

Pode-se mostrar que a condição C1 é satisfeita se a seguinte função de custo for minimizada:

$$L(n) = \frac{1}{2} [|\mathbf{w}(n)| - \mathbf{f}(n)]^T [|\mathbf{w}(n)| - \mathbf{f}(n)], \quad (4.23)$$

em que $\mathbf{f}(n) = [f_1(n) \ f_2(n) \ \dots \ f_N(n)]^T$ é o vetor dos coeficientes de ativação individuais $f_i(n)$ e por definição:

$$|\mathbf{w}(n)| \triangleq \begin{bmatrix} |w_1(n)| \\ |w_2(n)| \\ \vdots \\ |w_N(n)| \end{bmatrix} \quad (4.24)$$

Consequentemente,

$$\nabla_{\mathbf{f}(n)} L(n) = -[|\mathbf{w}(n)| - \mathbf{f}(n)]. \quad (4.25)$$

Utilizando o método do gradiente descendente para encontrar $\mathbf{f}(n)$ que minimiza $L(n)$, tem-se:

$$\mathbf{f}(n+1) = \mathbf{f}(n) - \varepsilon \nabla_{\mathbf{f}(n)} L(n). \quad (4.26)$$

Substituindo (4.25) em (4.26), resulta em:

$$\mathbf{f}(n+1) = \mathbf{f}(n) + \varepsilon[|\mathbf{w}(n)| - \mathbf{f}(n)]. \quad (4.27)$$

Portanto,

$$\mathbf{f}(n+1) = (1 - \varepsilon)\mathbf{f}(n) + \varepsilon|\mathbf{w}(n)|. \quad (4.28)$$

Note que a expressão obtida para atualização de $\mathbf{f}(n)$ também satisfaz C2 desde que $0 < \varepsilon < 1$ e $f_i(n) > 0$, para todo $i = 1, 2, 3, \dots, N$ (N é comprimento do filtro adaptativo).

Sousa *et al.* (2010) destaca que pode ocorrer de $|w_i(n)|$ não ser proporcional a $|p_i(n)|$ no início do processo de adaptação. Para contornar essa limitação, faz-se a atualização periódica dos fatores de ativação individuais somente após períodos de aprendizagem de N amostras. Dessa forma, a expressão de atualização dada em (4.26) é reescrita como:

$$\mathbf{f}(n+1) = \begin{cases} (1 - \varepsilon)\mathbf{f}(n) + \varepsilon|\mathbf{w}(n)| & n = mN, \quad m = 1, 2, \dots \\ \mathbf{f}(n) & \text{caso contrário.} \end{cases} \quad (4.29)$$

Portanto, considerando (4.28), obtém-se um algoritmo que combina as principais características do IAF-PNLMS e UT-ZA-PNLMS, ou seja, a característica principal do IAF-PNLMS, que é a utilização de um fator de ativação para cada coeficiente do filtro adaptativo, é inserida no algoritmo UT-ZA-PNLMS. Além disso, mantém-se o cômputo da matriz de ganho $\mathbf{G}(n)$, utilizando uma função adaptativa, $\tau_U(n)$, que limita superiormente os ganhos individuais do algoritmo.

A proposta, que utiliza (4.28) no lugar de (4.19), combina as melhores características dos algoritmos IAF-PNLMS e UT-ZA-PNLMS, sendo então denominado algoritmo IAF-UT-ZA-PNLMS. O algoritmo apresentado na Tabela 4.2 lista os passos para implementação do filtro IAF-UT-ZA-PNLMS.

Tabela 4.2 - Sumário do algoritmo proposto IAF-UT-ZA-PNLMS.

Algoritmo 4.2 IAF-UT-ZA-PNLMS

1. Inicialização

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

e parâmetros

$$\begin{aligned} 0 < \mu < 2 \\ -1 \leq \alpha < 1 \end{aligned}$$

$$\delta > 0$$

2. Dados de entrada e saída da planta

$$d(n) = \mathbf{x}^T(n)\mathbf{p}(n)$$

e do filtro adaptativo

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

3. Sinal do erro

$$e(n) = d(n) - y(n) + v(n)$$

4. Calcule $d_i(n)$, $\tau_U(n)$, $g_i(n)$ e $f_i(n)$ para (4.10), (4.16), (4.17) e (4.27), respectivamente.

5. Calcule $g_i(n) = \frac{\phi_i(n)}{\sum_{i=1}^N \phi_i(n)}$, onde, $i = 1, 2, \dots, N$,

6. Monte a matriz de ganhos $\mathbf{G}(n) = \text{diag} [g_0(n), g_1(n) \dots g_{N-1}(n)]$

7. Atualize os pesos

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n)\mathbf{G}(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n) + \delta\rho} - \rho \text{sgn}[\mathbf{w}(n)],$$

4.4 MÉTRICAS DE AVALIAÇÃO

As simulações computacionais são realizadas considerando quatro plantas diferentes. Em todos os cenários de simulação o objetivo principal é identificar uma planta esparsa com $N = 100$ coeficientes. São realizadas simulações de Monte Carlo (média de 100 realizações independentes). Os valores das variáveis avaliadas são obtidos por

$$E\{V(n)\} = \frac{1}{R} \sum_{r=1}^R V_r(n), \quad (4.30)$$

sendo $E\{\cdot\}$ o operador esperança, $V_r(n)$ é o valor da variável avaliada $V(n)$ na n -ésima iteração da r -ésima realização e R é o número de realizações.

A figura de mérito utilizada para comparar o desempenho do algoritmo IAF-UT-ZA-PNLMS e comparar com o IAF-PNLMS e UT-ZA-PNLMS é o desalinhamento normalizado em dB dado por (Branco, 2016):

$$\eta(n) = 10 \log_{10} \left(\frac{\|\mathbf{p} - \mathbf{w}(n)\|_2^2}{\|\mathbf{p}\|_2^2} \right). \quad (4.31)$$

Além das métricas já mencionadas, os algoritmos serão avaliados também erro em regime permanente e tempo de transição da curva de aprendizado, ou seja, o número de iterações (aproximadas) para se chegar ao regime permanente. Na Figura 4.1 consta um exemplo dessas duas fases do aprendizado de um filtro adaptativo.

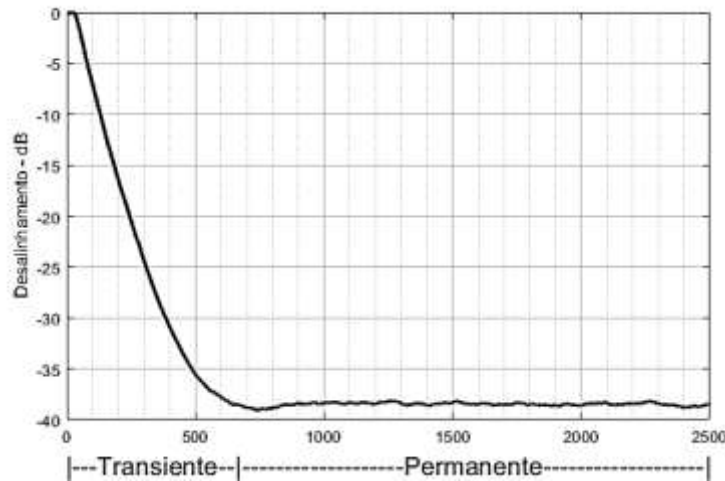


Figura 4.1 – Curva teórica do erro indicando os regimes transiente e permanente.

Nesse trabalho a determinação do erro em regime permanente e tempo de transição da curva de aprendizado será realizada por observação dos gráficos, ou seja, serão valores aproximados.

Capítulo 5

Simulações Computacionais

As simulações computacionais apresentadas nesse capítulo foram divididas em três grupos. No primeiro grupo, apenas os filtros UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) são comparados, pois o objetivo é avaliar o impacto no desempenho dos mesmos causado por alguns parâmetros inerentes a esses algoritmos. No segundo grupo de simulações os algoritmos IAF-PNLMS, UT-ZA-PNLMS e IAF-UT-ZA-PNLMS foram comparados entre si, considerando o objetivo de identificar quatro plantas esparsas diferentes. No terceiro grupo de simulações modificou-se a variância do ruído branco gaussiano e os filtros foram avaliados.

Em todas as simulações, utiliza-se como entrada um sinal correlacionado autorregressivo de ordem 2 de variância unitária dado por

$$x(n) = 0,4x(n-1) - 0,4x(n-2) + v(n), \quad (5.1)$$

sendo $v(n)$ um ruído branco gaussiano de variância σ_v^2 . O ruído de medição, $z(n)$, é gaussiano branco com variância $\sigma_z^2 = 10^{-3}$ (SNR = 30 dB) e o tamanho do passo μ é 0.5 para todos os algoritmos.

5.1 AVALIAÇÃO COMPARATIVA ENTRE OS ALGORITMOS UT-ZA-PNLMS E IAF-UT-ZA-PNLMS

Percebe-se que o algoritmo UT-ZA-PNLMS e conseqüentemente o IAF-UT-ZA-PNLMS possui diversos parâmetros. Por esse motivo resolveu-se nesse trabalho averiguar qual impacto que a variação desses parâmetros causa no desempenho dos referidos filtros. Essa verificação foi realizada escolhendo-se um parâmetro para variar e fixando os demais nos seguintes valores: $\beta_1 = 1$, $\beta_2 = 0.2$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$ e $\delta_g = \rho_g = 0.01$. Especificamente, para o algoritmo proposto (IAF-UT-ZA-PNLMS), o parâmetro ϵ foi definido como 0.01 e $f_i(0) = 10^{-3}$ (para todo i).

O primeiro parâmetro cujo impacto foi analisado foi β_1 . Para tal variou-se seu valor fazendo com que o mesmo recebesse os seguintes valores: 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8;

0,9 e 1. Na Figura 5.1 consta um resumo dos resultados de desalinhamento para cada valor de β_1 .

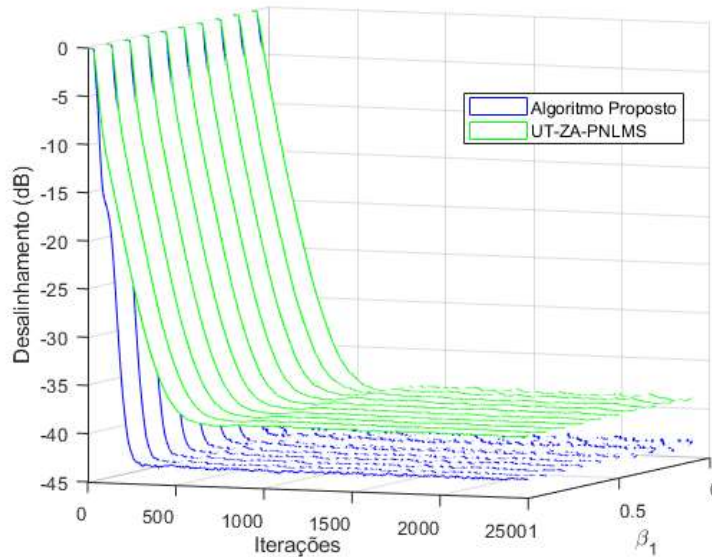


Figura 5.1 – Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando os seguintes valores de parâmetro $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0,95$, e $\beta_1 = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$.

Os resultados apresentados graficamente na Figura 5.1 indicam pouco impacto nos resultados causados pela variação do parâmetro. Tal observação pode ser confirmada quando se verifica na Tabela 5.1 o instante (em número de iterações) no qual os algoritmos analisados atingem um desalinhamento igual a -30 dB quando β_1 sofre variação.

Tabela 5.1 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro β_1 sofre variação.

β_1	IAF-UT-ZA-PNLMS	UT-ZA-PNLMS.
0.1	180	367
0.2	173	356
0.3	172	349
0.4	171	343
0.5	171	338
0.6	171	334
0.7	171	330
0.8	171	325
0.9	171	322
1	172	320

O segundo parâmetro que teve seu impacto nos resultados analisado foi β_2 . Para tal, variou-se seu valor, assim como feito com β_1 , fazendo com que o mesmo recebesse os seguintes valores: 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 e 1. Na Figura 5.2 consta um resumo dos resultados de desalinhamento para cada valor de β_2 .

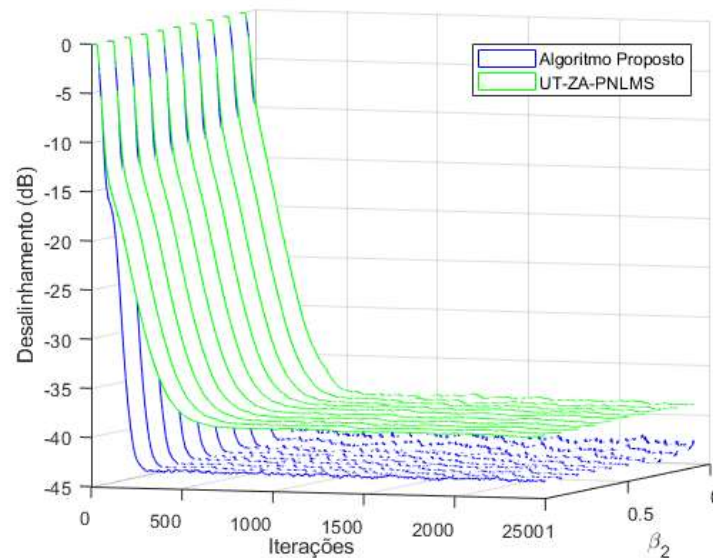


Figura 5.2 – Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0,5$, $\epsilon = \delta_N = 0,001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0,95$, $\beta_1 = 1$, $\beta_2 = \{0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1\}$, $\delta_g = \rho_g = 0,01$, $\epsilon = 0,01$ e $f_i(0) = 10^{-3}$.

Os resultados apresentados graficamente na Figura 5.2 também indicam pouco impacto nos resultados causados pela variação do parâmetro. Essa observação pode ser confirmada quando se verifica na Tabela 5.2 os valores referentes ao instante em que os filtros atingem o desalinhamento de -30 dB.

Tabela 5.2 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro β_2 sofre variação.

β_2	IAF-UT-ZA-PNLMS	UT-ZA-PNLMS.
0.1	181	322
0.2	177	314
0.3	173	306
0.4	173	302
0.5	172	300
0.6	171	298
0.7	171	296
0.8	172	296

0.9	172	296
1	172	294

O terceiro parâmetro que teve seu impacto nos resultados analisado foi λ_d . Assim como feito com β_1 e β_2 , atribuiu-se a este parâmetro os seguintes valores: 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 e 1. Na Figura 5.3, consta um resumo dos resultados de desalinhamento para cada valor de λ_d .

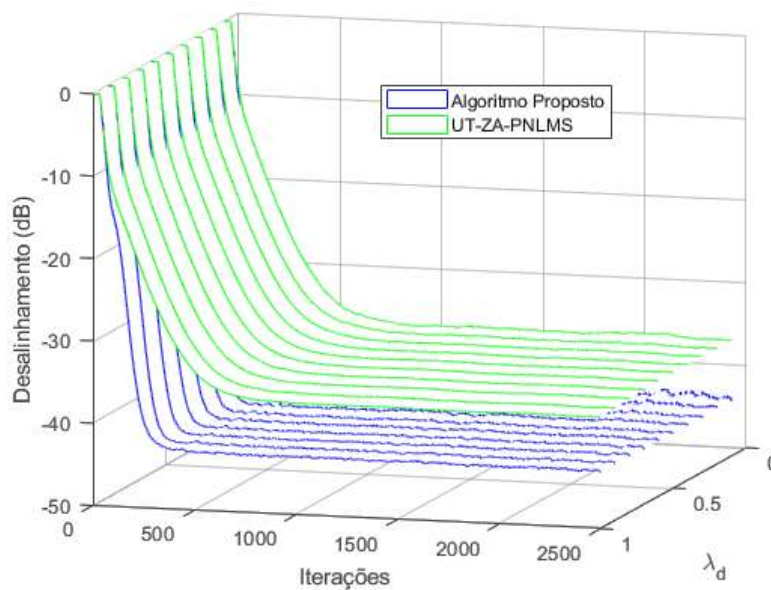


Figura 5.3 – Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$ $\lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$.

Novamente resultados apresentados graficamente na Figura 5.3 também indicam pouco impacto nos resultados causados pela variação do parâmetro. Verificação que pode ser comprovada ao analisar a Tabela 5.3.

Tabela 5.3 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro λ_d sofre variação.

λ_d	IAF-UT-ZA-PNLMS	UT-ZA-PNLMS.
0.1	182	402
0.2	182	397
0.3	182	397
0.4	181	396
0.5	181	394
0.6	180	392

0.7	179	389
0.8	178	383
0.9	177	374
1	183	383

Por fim, o impacto de dois parâmetro foi analisado, um parâmetro (ρ_g) específico do filtro UT-ZA-PNLMS e o outro ($f_i(0)$) específico do algoritmo proposto. Para o Parâmetro ρ_g foram testados os valores 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 e 1. Já para $f_i(0)$ os valores testado foram: 10^{-9} , 10^{-8} , 10^{-7} , 10^{-6} , 10^{-5} , 10^{-4} , 10^{-3} , 10^{-2} , 10^{-1} , 10^0 . Figura 5.4 consta um resumo dos resultados de desalinhamento para cada valor de $f_i(0)$ e de ρ_g . Nessa simulação a comparação é feita entre algoritmo proposto (Figura 5.4a) e UT-ZA-PNLMS (Figura 5.4b).

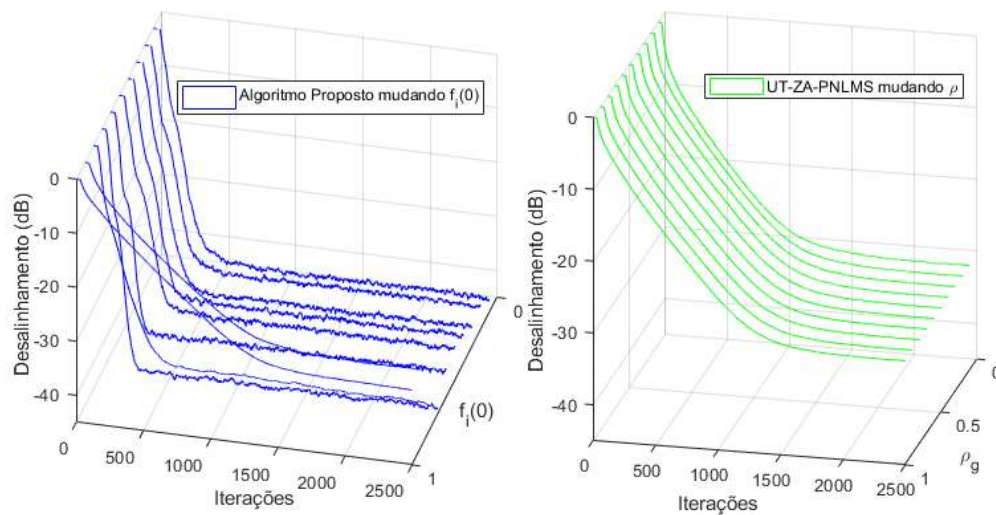


Figura 5.4 – Curvas de desalinhamento normalizado, $\eta(n)$, para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS (algoritmo proposto) considerando $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = \{10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ (para o algoritmo proposto) e $\rho_g = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\}$ (para UT-ZA-PNLMS).

O algoritmo proposto (Figura 5.4a), em sua melhor performance, alcançou $\eta(n) = -30$ dB na iteração 169 com $f_i(0) = 10^{-3}$. Na tabela 5.4 é possível verificar essas informações.

Tabela 5.4 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro $f_i(0)$ sofre variação.

$f_i(0)$	IAF-UT-ZA-PNLMS
10^{-9}	258
10^{-8}	272
10^{-7}	265
10^{-6}	265
10^{-5}	278
10^{-4}	192
10^{-3}	169
10^{-2}	312
10^{-1}	1361
1	1346

Vale ressaltar ainda que quando $f_i(0) = 10^{-1}$ e $f_i(0) = 10^0$ o algoritmo teve uma convergência lenta alcançando -30 dB de desalinhamento apenas após 1361 e 1346 iterações respectivamente, mostrando assim uma sensibilidade para valores altos de $f_i(0)$. Por outro lado, na Tabela 5.5 percebe-se pouca variação no algoritmo UT-ZA-PNLMS quando o parâmetro ρ_g sofre variação.

Tabela 5.5 - Número de iterações para que os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS atinjam $\eta(n) = -30$ dB quando o parâmetro ρ_g sofre variação.

ρ_g	UT-ZA-PNLMS
0.1	1179
0.2	1190
0.3	1210
0.4	1228
0.5	1237
0.6	1245
0.7	1251
0.8	1259
0.9	1263
1	1267

Observando todas as simulações aqui realizadas destacou-se aquela na qual os algoritmos UT-ZA-PNLMS e o proposto obtiveram os melhores resultados. A combinação que culminou nos melhores resultados foi à sequência $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\varepsilon = 0.01$ e $f_i(0) = 10^{-3}$. Na Figura 5.5 consta o resultado dessa simulação de comparação.

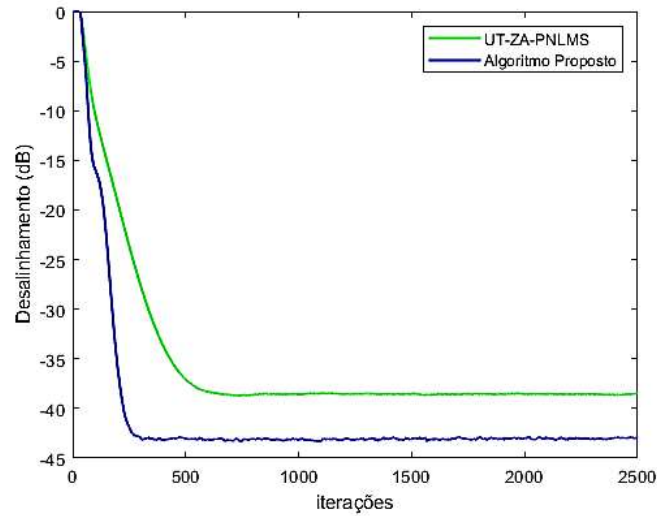


Figura 5.5 – Curvas de desalinhamto normalizado, $\eta(n)$, para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$.

Em resumo, percebe-se que, quando confrontado os melhores resultados, tanto do algoritmo proposto quanto do filtro UT-ZA-PNLMS, a superioridade, em termos de convergência e de alcance de menor desalinhamto, é característica do filtro proposto neste trabalho.

5.2 AVALIAÇÃO DOS ALGORITMOS NA IDENTIFICAÇÃO DE DIFERENTES PLANTAS ESPARSAS

Nessa seção são apresentadas simulações do desempenho dos algoritmos quando o objetivo é identificação de quatro plantas diferentes. Para os algoritmos UT-ZA-PNLMS e IAF-UT-ZA-PNLMS, são utilizados os mesmos parâmetros definidos na seção 5.1, fixando porém $\beta_1 = 1$, $\beta_2 = 0.2$. Vale lembrar ainda que os valores desses parâmetros definem, segundo Das (2020) o melhor desempenho em estado estacionário do algoritmo UT-ZA-PNLMS. Nas subseções a seguir são apresentados os resultados referentes a cada cenário de simulação.

5.2.1 Exemplo 1

O objetivo deste exemplo, conforme a Figura 5.6, em (a) é realizar a estimação dos coeficientes da planta \mathbf{p} com coeficientes ativos $\{1, 0.1\}$ nas posições $\{30, 85\}$. E em (b) após $n = 2500$, a planta \mathbf{p} sofre uma perturbação tornando-se $-\mathbf{p}$.

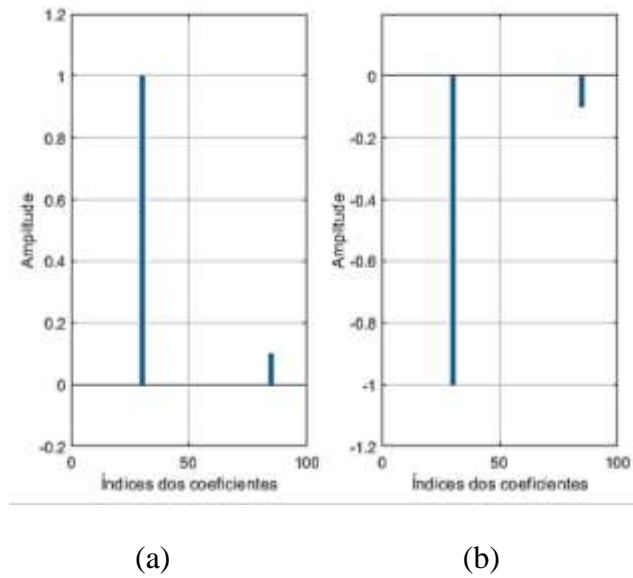


Figura 5.6 – Estimação dos coeficientes da planta \mathbf{p} que logo após $n = 2500$ sofre uma perturbação e passa a ser $-\mathbf{p}$. (a) coeficientes ativos $\{1, 0.1\}$ nas posições $\{30, 85\}$. (b) coeficientes ativos $\{-1, -0.1\}$ nas posições $\{30, 85\}$

O grau de esparsidade da resposta ao impulso para essa planta é $S(\mathbf{p}) = 0.9895$. Na Figura 5.7 são apresentados os resultados obtidos antes e depois da perturbação.

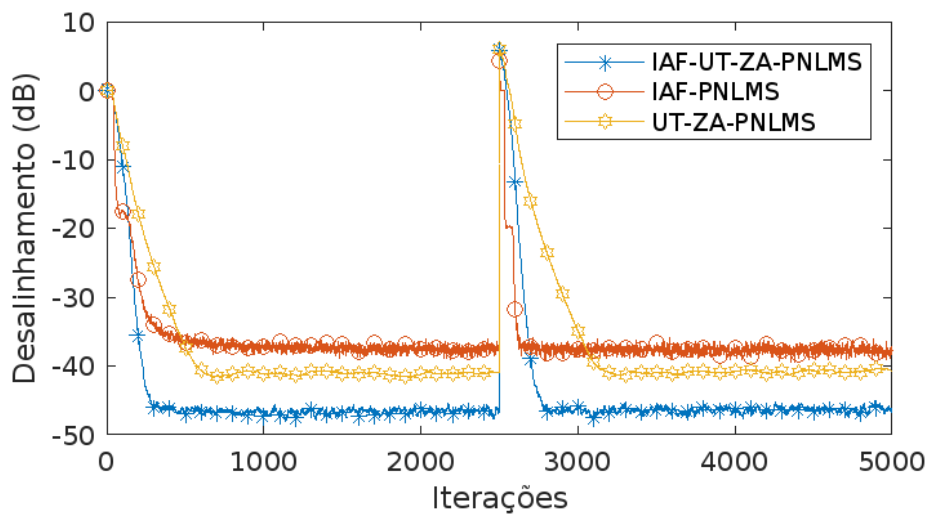


Figura 5.7 – Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.2$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Nos resultados apresentados na Figura 5.7, é possível observar que o algoritmo proposto nesse artigo, nos transitórios tanto antes quanto depois da perturbação, tem um

decaimento similar ao do IAF-PNLMS, porém em regime estacionário, a proposta apresenta menor nível de desalinhamento.

Para uma análise minuciosa, nas Figuras 5.8 a 5.10, observa-se o comportamento dos ganhos individuais dos coeficientes ativos $\{30, 85\}$ e de um inativo $\{2\}$. No caso do ganho inativo $g_2(n)$, note que os ganhos fornecidos pelos algoritmos do tipo UT apresentam uma limitação superior quando comparados aos ganhos fornecidos pelo IAF-PNLMS, conforme ilustra a Figura 5.8.

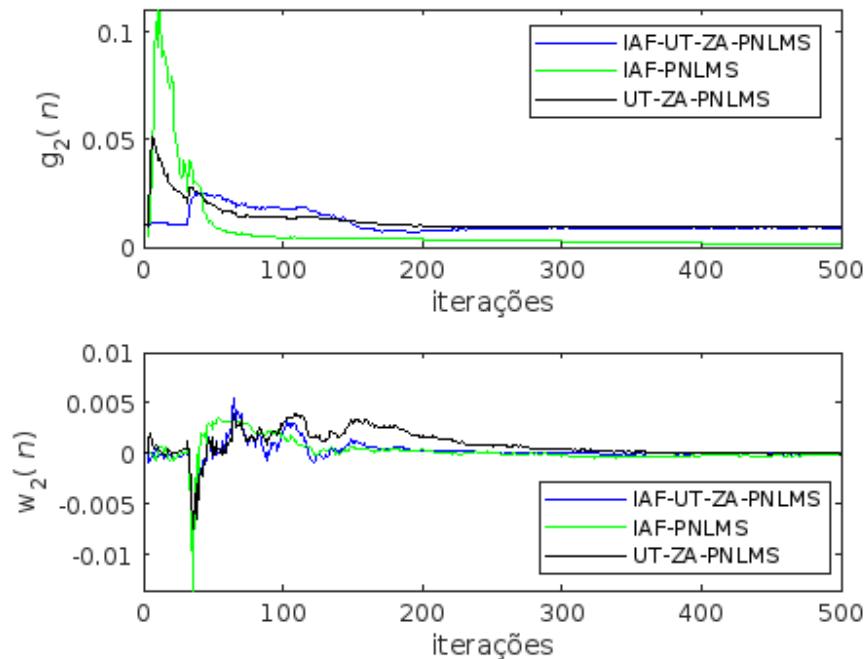


Figura 5.8 – Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0,01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Comparando o comportamento dos ganhos associados ao coeficiente ativo $w_{30}(n)$, observa-se que o algoritmo proposto impõe menores ganhos ao coeficiente ativo de maior magnitude comparado ao algoritmo IAF-PNLMS; entretanto, os ganhos fornecidos pela proposta são maiores do aqueles fornecidos pelo UT-ZA-PNLMS (veja a Figura 5.9). O comportamento do algoritmo proposto está de acordo com o esperado, isto é, uma distribuição de ganhos intermediária, ficando entre os algoritmos UT-ZA-PNLMS e IAF-PNLMS.

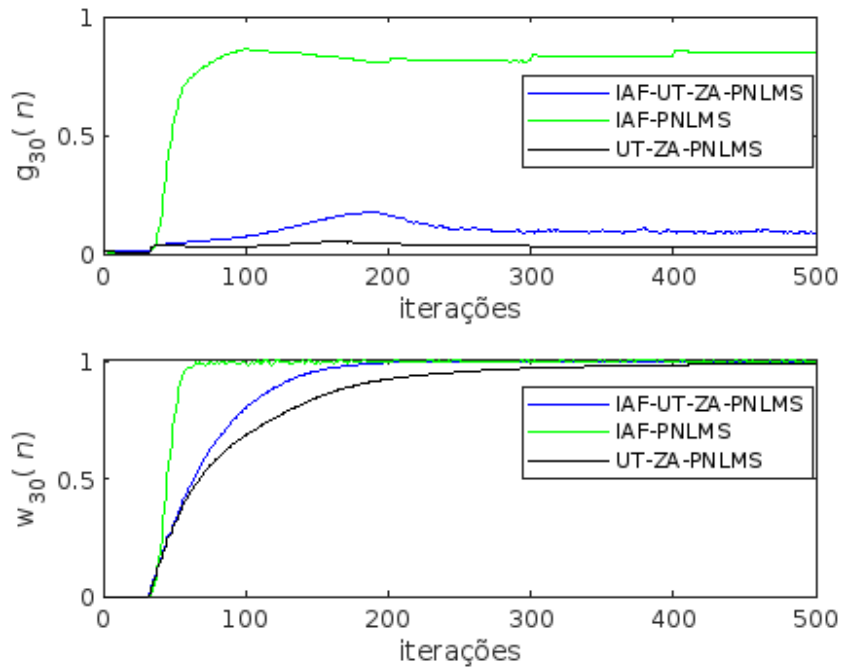


Figura 5.9 – Curvas de desalinhameno normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Da Figura 5.9, note que o algoritmo IAF-PNLMS impõe maior ganho relativo ao coeficiente $w_{30}(n)$ e com isso a convergência deste coeficiente é mais rápida do que os demais. Nesse caso, em especial, a limitação dos ganhos fornecidos pelo IAF-UT-ZA-PNLMS e UT-ZA-PNLMS retarda a convergência desse coeficiente.

Na Figura 5.10, o ganho $g_{85}(n)$ associado ao coeficiente ativo $w_{85}(n)$ é analisado. Nota-se que os resultados apresentados pelo o IAF-UT-ZA-PNLMS em termos de velocidade de convergência é superior aos demais algoritmos devido à redistribuição de ganhos imposta pela proposta.

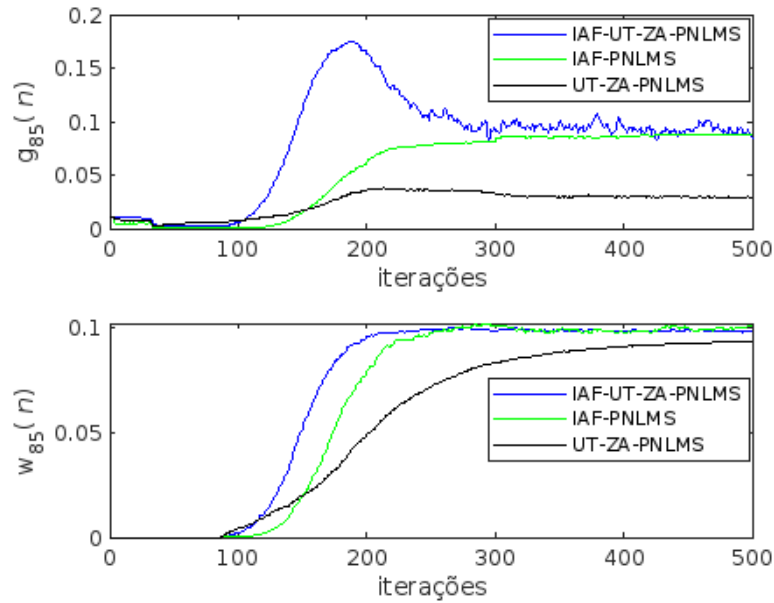


Figura 5.10 – Curvas de desalinhameno normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

5.2.2 Exemplo 2

Neste exemplo, conforme a Figura 5.11, considera-se a identificação de uma planta \mathbf{p} com coeficientes ativos $\{0.1, 1.0, -0.5, 0.1\}$ nas posições $\{1, 30, 35, 85\}$ que após $n = 2500$, a planta \mathbf{p} sofre uma perturbação tornando-se $-\mathbf{p}$.

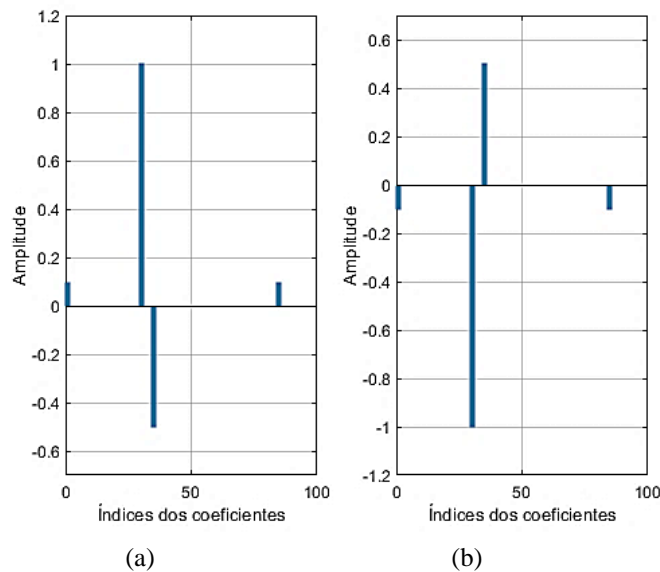


Figura 5.11 – Estimação dos coeficientes da planta \mathbf{p} que logo após $n = 2500$ sofre uma perturbação e passa a ser $-\mathbf{p}$. (a) coeficientes ativos $\{0.1, 1.0, -0.5, 0.1\}$ nas posições $\{1, 30, 35, 85\}$. (b) coeficientes ativos $\{-0.1, -1.0, 0.5, -0.1\}$ nas posições $\{1, 30, 35, 85\}$

O grau de esparsidade desta planta é $S(\mathbf{p}) = 0.9435$. Os resultados dessa simulação estão apresentados na Figura 5.12.

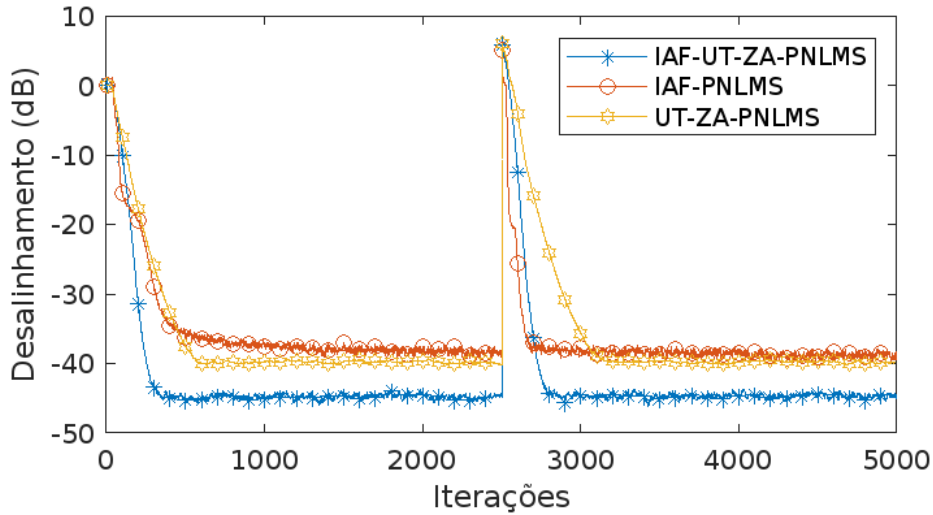


Figura 5.12 – Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Os resultados do Exemplo 2 ratificam o que foi observado na Figura 5.7, ou seja, o algoritmo proposto preserva a convergência rápida do IAF-PNLMS e o menor desalinhamento, tanto antes quanto depois da perturbação.

5.2.3 Exemplo 3

Para o terceiro exemplo, como mostra a Figura 5.13, realiza-se a estimação da planta \mathbf{p} com coeficientes ativos $\{0.1, 1.0, -0.5, 0.1, -0.1\}$ nas posições $\{1, 30, 35, 85, 95\}$. Após $n = 2500$, a planta \mathbf{p} sofre uma perturbação tornando-se $-\mathbf{p}$

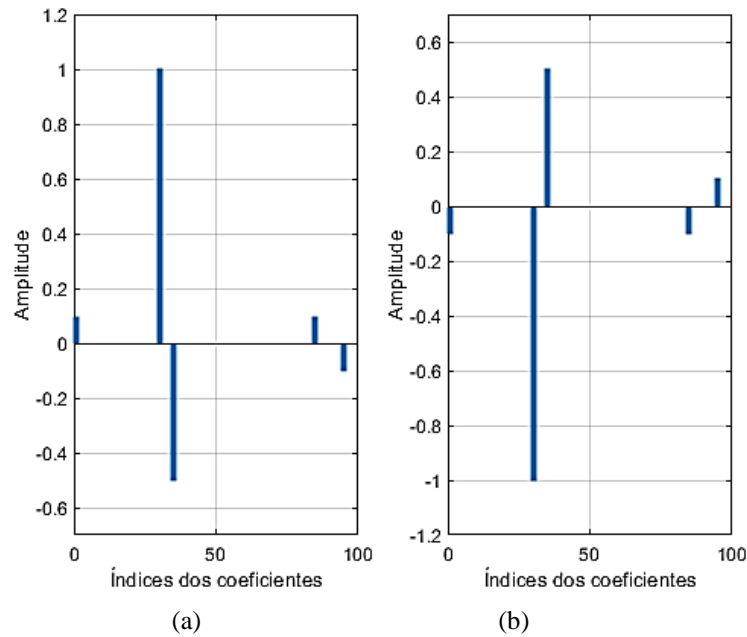


Figura 5.13 – Estimação dos coeficientes da planta \mathbf{p} que logo após $n = 2500$ sofre uma perturbação e passa a ser $-\mathbf{p}$. (a) coeficientes ativos $\{0.1, 1.0, -0.5, 0.1, -0.1\}$ nas posições $\{1, 30, 35, 85, 95\}$. (b) coeficientes ativos $\{-0.1, -1.0, 0.5, -0.1, 0.1\}$ nas posições $\{1, 30, 35, 85, 95\}$

O grau de esparsidade da resposta ao impulso para essa planta é $S(\mathbf{p}) = 0,9343$. Na Figura 5.14 estão representados graficamente os resultados do desalinhamento (em dB) para esse cenário de simulação.

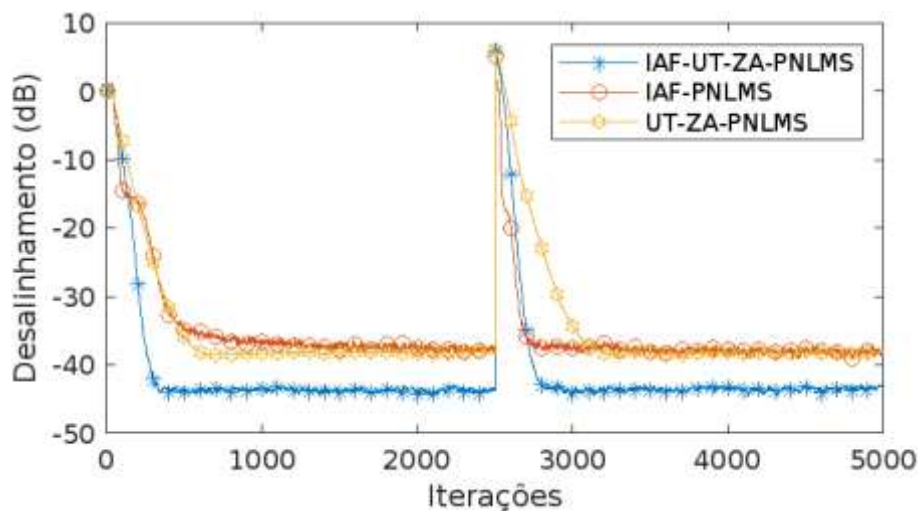


Figura 5.14 – Curvas de desalinhamento normalizado para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

De acordo com os testes realizados para este cenário de simulação (Exemplo 3),

observa-se da Figura 5.14 que a característica em manter a convergência elevada e o menor desalinhamento possível em estado estacionário permanece. Tal característica pode ser observada também antes e depois da perturbação.

5.4.1 Exemplo 4

Neste exemplo, a capacidade de rastreamento dos algoritmos IAF-PNLMS, UT-ZA-PNLMS e IAF-UT-ZA-PNLMS é avaliada. Aqui, a simulação começa com a planta \mathbf{p} com coeficientes ativos $\{0.1, 1.0, -0.5, 0.1\}$ nas posições $\{1, 30, 35, 85\}$, porém dois de seus coeficientes são modulados de acordo com a Equação (5.2), proposta em Martin (2002) e adaptada por Souza *et al.* (2010).

$$q_i = \begin{cases} p_i \left[1 + 0.5 \operatorname{sen} \left(\frac{2\pi n}{256} \right) \right] & i = 1, 30 \\ p_i & i = 35, 85 \\ 0 & \text{caso contrário} \end{cases} \quad (5.2)$$

Observando o desempenho dos algoritmos quanto ao rastreamento do primeiro coeficiente da planta como mostra a Figura 5.15, nota-se uma superioridade dos algoritmos IAF-PNLMS e do algoritmo proposto nesse trabalho.

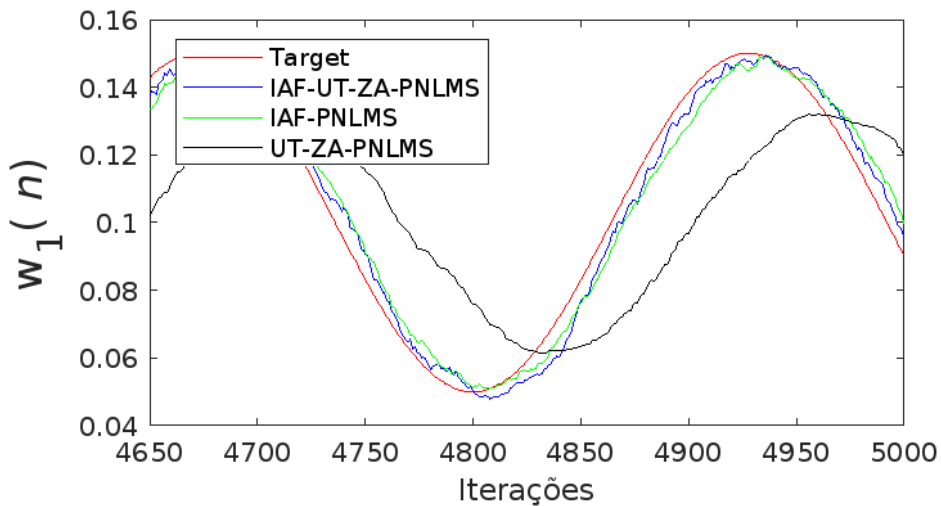


Figura 5.15 – Comportamento de $w_1(n)$ para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Uma análise considerando um coeficiente inativo também foi realizada, conforme a Figura 5.16.

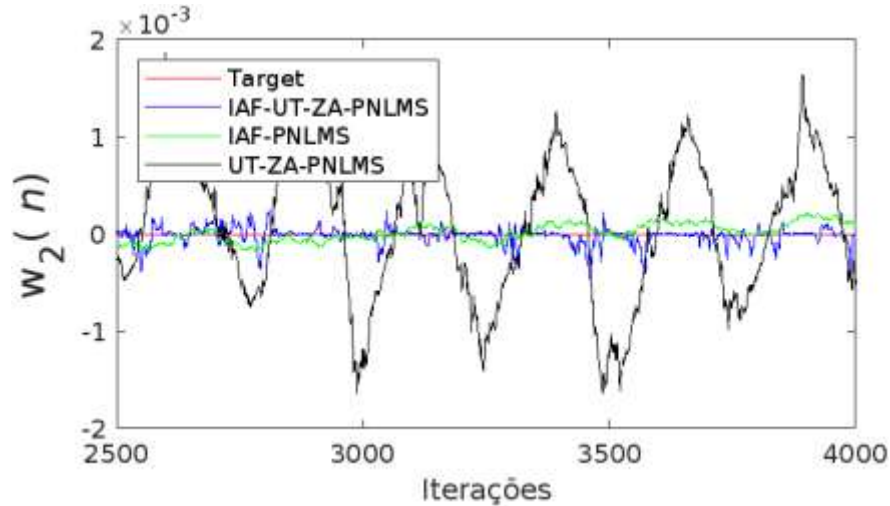


Figura 5.16 – Comportamento do coeficiente inativo $w_2(n)$ para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Verifica-se da Figura 5.16 que, o algoritmo IAF-UT-ZA-PNLMS juntamente com o IAF-UT-ZA-PNLMS convergem rapidamente no início das interações para o coeficiente inativo ilustrado.

O coeficiente ativo $w_{30}(n)$ é observado na Figura 5.17, onde é evidenciado mais uma vez a superioridade do IAF-PNLMS e do proposto em comparação ao UT-ZA-PNLMS.

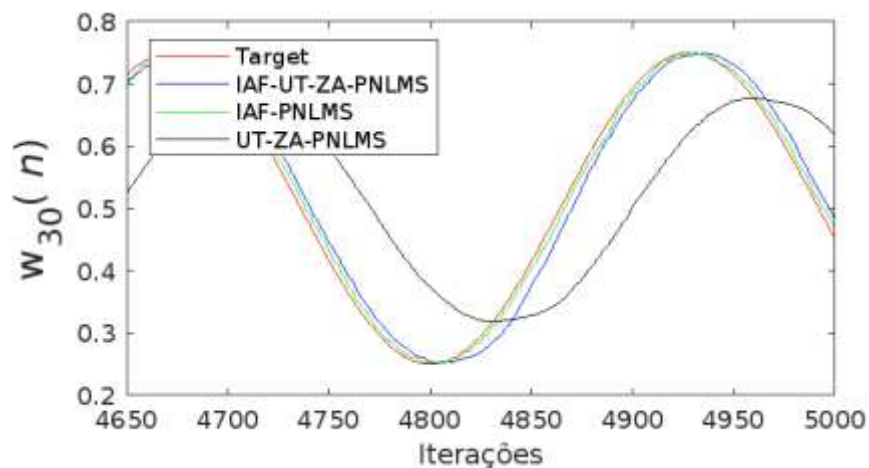


Figura 5.17 – Comportamento de $w_{30}(n)$ para o UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

Com o objetivo de se fazer uma análise mais geral do rastreamento, apresenta-se na Figura 5.18, o desalinhamento normalizado dos algoritmos durante o processo de adaptação.

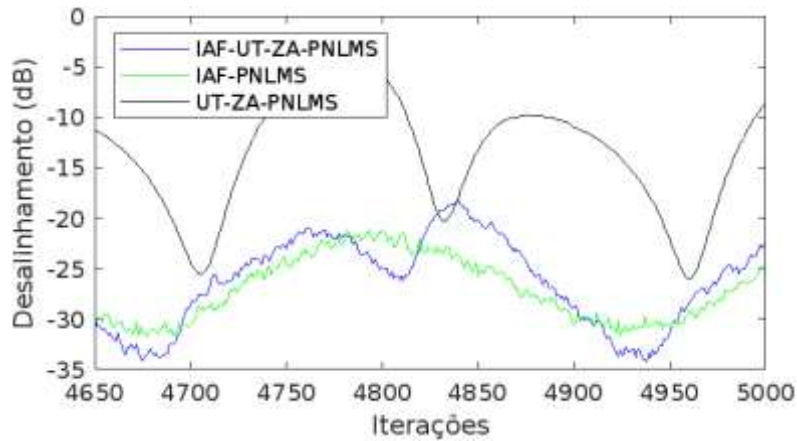


Figura 5.18 – Curvas de desalinhamento normalizado durante a tarefa de rastreamento para UT-ZA-PNLMS e IAF-UT-ZA-PNLMS com $\mu = 0.5$, $\epsilon = \delta_N = 0.001$, $\rho = 65 \cdot 10^{-6}$, $\lambda_d = \lambda_e = 0.95$, $\beta_1 = 1$, $\beta_2 = 0.02$, $\delta_g = \rho_g = 0.01$, $\epsilon = 0.01$ e $f_i(0) = 10^{-3}$. Para o IAF-PNLMS, $\xi = 0.001$ e $f_i(0) = 10^{-4}$.

5.3 AVALIAÇÃO DOS ALGORITMOS MEDIANTE A MUDANÇAS NA VARIÂNCIA DO RUÍDO

Nessa seção, apresenta-se a comparação dos filtros IAF-UT-ZA-PNLMS, IAF-PNLMS e UT-ZA-PNLMS em função de distintos valores de variância σ_v^2 de ruído branco gaussiano $v(n)$. Para a análise, observou-se os valores do instante (iteração aproximada) em que os mesmos atingem o regime permanente e o erro médio nessa fase. Na Tabela 5.6 consta um resumo dos resultados obtidos nas simulações.

Tabela 5.6 - Erro médio em regime permanente em função da variância do ruído $v(n)$.

Variância do Ruído	Iteração IAF-UT-ZA-PNLMS	Iteração IAF-PNLMS	Iteração UT-ZA-PNLMS	Erro IAF-UT-ZA-PNLMS	Erro IAF-PNLMS	Erro UT-ZA-PNLMS
0.10	321	564	540	-35	-31	-35
0.15	324	616	592	-37	-31	-35
0.20	320	543	563	-38	-33	-37
0.25	337	710	550	-39	-33	-37
0.30	323	660	562	-39	-33	-38
0.35	321	751	520	-40	-35	-37
0.40	328	742	546	-41	-36	-38
0.45	334	856	692	-42	-35	-37
0.50	316	694	565	-41	-35	-38
0.55	312	737	590	-41	-36	-38
0.60	320	763	543	-42	-36	-39
0.65	331	777	573	-42	-37	-39
0.70	318	782	536	-43	-38	-39

0.75	331	684	537	-43	-37	-38
0.80	316	852	530	-44	-38	-39
0.85	316	861	537	-43	-38	-39
0.90	329	867	537	-43	-39	-39
0.95	319	784	544	-44	-39	-38
1	324	719	542	-44	-37	-39

Ao considerar a Tabela 5.6 pode-se perceber que em todas as simulações o algoritmo proposto alcança o regime permanente antes dos demais algoritmos. Além disso, o erro obtido em regime permanente é, em média, sempre menor do que aqueles obtidos pelos demais algoritmos.

Capítulo 6

Considerações Finais

Neste capítulo são apresentadas as considerações finais da dissertação, incluindo a conclusão e trabalhos futuros.

6.1 CONCLUSÃO

Entre as técnicas de processamento digital de sinais, os filtros adaptativos têm ganhado destaque, pois apresentam um bom desempenho na solução de diversos problemas dos quais se podem destacar a equalização de canais, a predição de sinais, o realçamento de sinais e o cancelamento de eco. Dentre os problemas nos quais se utiliza filtros adaptativos está a identificação de sistemas que se torna uma tarefa ainda mais desafiadora quando este possui alto grau de esparsidade.

Para sistemas com alto grau de esparsidade, os algoritmos adaptativos clássicos como o LMS e o NLMS apresentam um desenvolvimento pobre em termos de velocidade de convergência. Em contrapartida os algoritmos da classe proporcional, como o PNLMS e o IPNLMS, têm se mostrado superior quando a tarefa envolve a identificação de plantas esparsas. A fim de melhorar o desempenho destes algoritmos existentes, vários outros foram surgindo, tais como o IAF-PNLMS e o UT-ZA-PNLMS.

Nesse contexto, um algoritmo (IAF-UT-ZA-PNLMS) que combina as melhores características dos algoritmos UT-ZA-PNLMS e IAF-PNLMS foi proposto neste trabalho de pesquisa. Nesta proposta foi possível desenvolver uma função de custo, e essa função de custo tem relação com os fatores de ativação individual por meio do método do gradiente descendente e a mesma foi utilizada no algoritmo UT-ZA-PNLMS.

Resultados de simulações numéricas demonstram que o algoritmo proposto apresenta desempenho superior ao UT-ZA-PNLMS e ao IAF-PNLMS em termos de velocidade de convergência. Foram realizados testes com diferentes valores de parâmetros e, observou-se que os algoritmos UT-ZA-PNLMS e o IAF-UT-ZA-PNLMS não são sensíveis às modificações dos parâmetros. No entanto, ainda assim, o algoritmo proposto apresenta um melhor desempenho. Como proposto, foram realizados também, testes em quatro plantas de natureza esparsa, com o objetivo de analisar o comportamento dos filtros em cada cenário

antes e depois de cada planta sofrer uma perturbação. Ao final da análise pode-se perceber que em todas as simulações o algoritmo proposto alcançou o regime permanente antes dos demais algoritmos.

Em suma o algoritmo proposto, IAF-UT-ZA-PNLMS, apresentou convergência semelhante ao IAF-PNLMS e comportamento em estado estacionário condizente ao apresentado pelo UT-ZA-PNLMS. Tais resultados preliminares podem ser interpretados como um algoritmo melhorado para plantas esparsas.

6.2 TRABALHOS FUTUROS

Como trabalhos futuros, pretende-se:

- Realizar estudos teóricos de convergência do algoritmo proposto, visando generalizar os resultados obtidos;
- Testar outras possibilidades de se estabelecer os limites superiores para os ganhos dos coeficientes do filtro, como, por exemplo, modificar a função utilizada na etapa de máxima verossimilhança;
- Verificar a possibilidade de diminuir a quantidade de parâmetros que compõe o algoritmo proposto;
- Testar o algoritmo proposto em outras aplicações tais como cancelamento de eco, previsão de séries temporais e etc.

REFERÊNCIAS

- BAKRI, K.J., KUHN, E.V. and SEARA, R., Considerações sobre a modelagem estocástica da combinação de filtros adaptativos usando o algoritmo NLMS. *In: XXXVIII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS – SBrT'20*. DOI: 10.14209/SBRT.2020.1570641724. Florianópolis, SC, nov. 2020. Disponível em: <https://biblioteca.sbrt.org.br/articles/2146>. Acesso em: 24 abr. 2023. Citado 1 vez na página 1.
- BENESTY, J. and GAY, S.L. An improved PNLMS algorithm. In *2002 IEEE international conference on acoustics, speech, and signal processing*, v. 2, p. II-1881. IEEE, maio 2002. Citado 2 vezes nas páginas 1 e 36.
- BOLFARINE, H. and SANDOVAL, M.C. *Introdução à inferência estatística* (Vol. 2). SBM, 2001, p. 35 - 36. Citado 4 vezes nas páginas 44 e 45.
- BRANCO, C. A. S. C. Algoritmos adaptativos LMS normalizados proporcionais: proposta de novos algoritmos para identificação de plantas esparsas. Universidade Federal do Maranhão, 2016. Citado 7 vezes nas páginas 3, 34, 52, 53 e 68.
- CHEN, Y., GU, Y. and HERO, A.O. Sparse LMS for system identification. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 3125-3128, abr. 2009. Citado 2 vezes nas páginas 40 e 41.
- DAS, R. L. and CHAKRABORTY, M. Improving the performance of the PNLMS algorithm using 1 norm regularization. *IEEE/ACM Trans. Audio, Speech, Language Process*, v. 24, n. 7, p. 1280–1290, jul. 2016. Citado 3 vezes nas páginas 2, 42 e 43.
- DAS, R.L. and TRIVEDI, V. An adaptive upper threshold based gain function for the ZA-PNLMS algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(10), pp.2274-2278, out. 2020. Citado 9 vezes nas páginas 2, 5, 46, 47, 48 e 59.
- DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation*. Springer. New York, NY, USA, 2008. Citado 2 vezes nas páginas 8 e 31.
- DUTTWEILER, D.L. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Transactions on Speech and Audio Processing*, v. 8, n.5, p.508-518, 2000. Citado 2 vezes nas páginas 4, 5 e 34.
- FARHANG-BOROJENY, B. *Adaptive filters: theory and applications*. [S.l.]: John Wiley & Sons, 2013, p.1. Citado 1 vez na página 1.
- FONSECA, J. d. R. S. Análise e implementação em circuito eletrônico de algoritmos adaptativos de gradiente estocástico baseado em função contínua, par e não linear. 2022. 88 f. Tese (Programa de Pós-Graduação em Engenharia de Eletricidade/CCET) - Universidade Federal do Maranhão, São Luís, 2022. Citado 1 vez na página 1.

GAY, S.L. An efficient, fast converging adaptive filter for network echo cancellation. In *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284)*, v.1, pp. 394-398, nov. 1998. Citado 2 vezes nas página 1.

GAY, S.L. and DOUGLAS, S.C. Normalized natural gradient adaptive filtering for sparse and non-sparse systems. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 2, pp. II-1405, maio 2002. Citado 2 vezes nas páginas 1 e 36.

HAYKIN, S. *Adaptive Filter Theory*, 3. ed. Englewood Cliffs, NJ. Ed. Prentice-Hall, p 339 a 438, 1996. Citado 3 vezes nas páginas 4, 8 e 33.

HAYKIN, S.S. *Adaptive filter theory*. Pearson Education India, p 246-326, 2014. Citado 2 vezes nas páginas 1 e 8.

HAYKIN, S.S. *Neural Networks and Learning Machines*. Prentice-Hall, p 91-104, 2008. Citado 2 vezes nas páginas 16.

HUANG, Y., BENESTY, J., and CHEN, J. *Acoustic MIMO signal processing*. Springer Science & Business Media, 2006. Citado 1 vez na página 1.

MANOLAKIS, D.G., INGLE, V. K. and KOGON, S. M. *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. Artech House. 2005, p. 47-51. Citado 1 vez na página 9.

MARTIN, R.K., SETHARES, W.A., Williamson, R.C. and Johnson, C.R. Exploiting sparsity in adaptive filters. *IEEE Transactions on Signal Processing*, 50(8), pp.1883-1894, 2002. Citado 1 vez na página 66.

MARTINS, R. N. M. Estimação de Componentes Harmônicos em Sistemas Elétricos de Potência via Combinação Convexa com Otimização de Parâmetros Baseada em Busca Tabu. Universidade Federal do Maranhão, 2020. Citado 1 vez na página 31.

MÜLLER, C. and PIBER, P. R.V. Uma Comparação do Desempenho da Filtragem Adaptativa em Banda Cheia e a Filtragem Adaptativa em Subbandas, Aplicada ao Cancelamento do Eco Elétrico. *Revista Tecnologia e Tendências*, v. 5, p. 52-71, 2006. Citado 3 vezes nas páginas 1 e 2.

PALEOLOGU, C., BENESTY, J. and CIOCHINA, S. Sparse adaptive filters for echo cancellation. *Synthesis Lectures on Speech and Audio Processing*, 6(1), pp.1-124, 2010. Citado 2 vezes nas páginas 1 e 4.

SOUZA, F.d.C. de, SEARA, R. and MORGAN, D.R. An enhanced IAF-PNLMS adaptive algorithm for sparse impulse response identification. *IEEE transactions on signal processing*, 60(6), pp.3301-3307, 2012. Citado 1 vez na página 36.

SOUZA, F.d.C. de, TOBIAS, O.J., SEARA, R. and MORGAN, D.R. A PNLMS algorithm with individual activation factors. *IEEE Transactions on Signal Processing*, 58(4), pp.2036-2047, 2010. Citado 4 vezes nas páginas 1, 2, 39 e 66.

SOUZA, F.d.C. de, TOBIAS, O.J., SEARA, R. and MORGAN, D.R. Alternative approach for computing the activation factor of the PNLMS algorithm. In *2009 17th European Signal Processing Conference*, pp. 2633-2637, ago. 2009. Citado 6 vezes nas páginas 35, 38, 39 e 49.

WIDROW, B. and STEARNS, S. D. *Adaptive Signal Processing*. Englewood Cliffs, New Jersey: Prentice Hall. 1985. Citado 1 vez na página 2.