**UNIVERSIDADE FEDERAL DO MARANHÃO**
**PRÓ-REITORIA DE PÓS-GRADUAÇÃO**
**DEPARTAMENTO DE ENGENHARIA DE ELETRICIDADE**
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE**

CÉSAR AUGUSTO SANTANA CASTELO BRANCO

# ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS PROPORCIONAIS: PROPOSTA DE NOVOS ALGORITMOS PARA IDENTIFICAÇÃO DE PLANTAS ESPARSAS

São Luís, MA

2016

**UNIVERSIDADE FEDERAL DO MARANHÃO**
**PRÓ-REITORIA DE PÓS-GRADUAÇÃO**
**DEPARTAMENTO DE ENGENHARIA DA ELETRICIDADE**
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE**

CÉSAR AUGUSTO SANTANA CASTELO BRANCO

# ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS PROPORCIONAIS: PROPOSTA DE NOVOS ALGORITMOS PARA IDENTIFICAÇÃO DE PLANTAS ESPARSAS

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Eletricidade 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

2016

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

Castelo Branco, César Augusto Santana.
Algoritmos Adaptativos LMS Normalizados Proporcionais: Proposta de Novos Algoritmos Para Identificação de Plantas Esparsas / César Augusto Santana Castelo Branco. - 2016.
196 p.

Orientador(a): Francisco das Chagas de Souza.
Dissertação (Mestrado) - Programa de Pós-graduação em Engenharia de Eletricidade/ccet, Universidade Federal do Maranhão, São Luís - MA, 2016.

1. Algoritmos adaptativos LMS normalizados proporcionais. 2. Filtragem adaptativa. 3. Identificação de plantas esparsas. 4. Otimização. I. Souza, Francisco das Chagas de. II. Título.

# ALGORITMOS ADAPTATIVOS LMS NORMALIZADOS PROPORCIONAIS: PROPOSTA DE NOVOS ALGORITMOS PARA IDENTIFICAÇÃO DE PLANTAS ESPARSAS

**César Augusto Santana Castelo Branco**

Dissertação aprovada em 12 de dezembro de 2016.

Prof. Francisco das Chagas de Souza, Dr.

(Orientador)

Prof. Vicente Leonardo Paucar Casas, Dr.

(Membro da Banca Examinadora)

Profa. Áurea Celeste da Costa Ribeiro, Dra.

(Membro da Banca Examinadora)

# AGRADECIMENTOS

Agradeço primeiramente a Deus, digno de toda a honra, toda a glória, todo o louvor e toda a adoração, por me permitir cursar a pós-graduação. Apesar de todas as dificuldades que passei durante este período, senti o mover da sua mão, Pai, me ajudando, protegendo, dando força e cuidando de mim e de minha família. Agradeço também a meu pai e minha mãe por todo o amor, cuidado, e sacrifícios que fizeram de bom grado para que eu pudesse chegar até aqui e, quem sabe, poder honrar um pouco todo o seu esforço para comigo. Deixo também um agradecimento especial à minha noiva e companheira insubstituível, Julianna Santos Farias, pelo incentivo e apoio tão necessários nos momentos mais difíceis.

Presto também aqui um enorme agradecimento ao meu orientador, o Prof. Dr. Francisco das Chagas de Souza, por sua paciência, dedicação, compreensão e ensinamentos valorosos que me fizeram evoluir como pesquisador e como pessoa. Com certeza levarei comigo seus conselhos, dicas, opiniões e explicações pelo resto da minha vida profissional.

Agradeço também à Universidade Federal do Maranhão, em especial ao Programa de Pós-Graduação em Engenharia de Eletricidade, pela oportunidade de fazer parte da sua história e dar a minha pequena parcela de contribuição. Deixo também meu agradecimento à CAPES e ao CNPq pelo apoio financeiro no desenvolvimento do projeto de pesquisa e na publicação e apresentação dos artigos científicos aceitos.

Dedico este trabalho a Raimundo Santana Castelo Branco e Maria da Paixão Castelo Branco, meus amados pais e verdadeiros heróis.

# RESUMO

Neste trabalho, novas metodologias para otimizar a escolha dos parâmetros dos algoritmos adaptativos LMS normalizados proporcionais (PNLMS) são propostas. As abordagens propostas usam procedimentos baseados em dois métodos de otimização, a saber, os métodos da razão áurea e da busca tabu. Tais procedimentos são empregados para determinar os parâmetros ótimos em cada iteração do processo de adaptação dos algoritmos PNLMS e PNLMS melhorado (IPNLMS). A função objetivo adotada pelos procedimentos propostos é baseada no erro de estimação *a posteriori*. O estudo de desempenho realizado para avaliar o impacto dos parâmetros dos algoritmos PNLMS e IPNLMS no comportamento dos mesmos mostram que, com o auxílio de técnicas de otimização para escolher adequadamente tais parâmetros, o desempenho destes algoritmos pode ser melhorado, em termos de velocidade de convergência, para a identificação de plantas com elevado grau de esparsidade. O principal objetivo das metodologias propostas é melhorar a distribuição da energia de ativação entre os coeficientes dos algoritmos PNLMS e IPNLMS, usando valores de parâmetros que levam ao erro de estimação mínimo em cada iteração do processo de adaptação. Testes numéricos realizados (considerando diversos cenários nos quais a resposta impulsiva da planta é esparsa) mostram que as metodologias propostas alcançam velocidades de convergência superiores às dos algoritmos PNLMS e IPNLMS, além de outros algoritmos da classe PNLMS, tais como o algoritmo IPNLMS com controle de esparsidade (SC-IPNLMS).

**Palavras-chave**: Filtragem adaptativa, algoritmos adaptativos LMS normalizados proporcionais, métodos de otimização, razão áurea, busca tabu, identificação de plantas esparsas.

# ABSTRACT

This work proposes new methodologies to optimize the choice of the parameters of the proportionate normalized least-mean-square (PNLMS) adaptive algorithms. The proposed approaches use procedures based on two optimization methods, namely, the golden section and tabu search methods. Such procedures are applied to determine the optimal parameters in each iteration of the adaptation process of the PNLMS and improved PNLMS (IPNLMS) algorithms. The objective function for the proposed procedures is based on the *a posteriori* estimation error. Performance studies carried out to evaluate the impact of the PNLMS and IPNLMS parameters in the behavior of these algorithms shows that, with the aid of optimization techniques to choose properly such parameters, the performance of these algorithms may be improved in terms of convergence speed for the identification of plants with high sparseness degree. The main goal of the proposed methodologies is to improve the distribution of the adaptation energy between the coefficients of the PNLMS and IPNLMS algorithms, using parameter values that lead to the minimal estimation error of each iteration of the adaptation process. Numerical tests performed (considering various scenarios in which the plant impulse response is sparse) show that the proposed methodologies achieve convergence speeds faster than the PNLMS and IPNLMS algorithms, and other algorithms of the PNLMS class, such as the sparseness controlled IPNLMS (SC-IPNLMS) algorithm.

**Keywords**: Adaptive filtering, proportionate normalized leas-mean-square adaptive algorithms, optimization methods, golden section, tabu search, sparse plant identification.

# LISTA DE FIGURAS

# LISTA DE TABELAS

# LISTA DE SIGLAS

# SUMÁRIO

# Capítulo 1

# Introdução

Sistemas adaptativos podem ser definidos como aqueles cuja estrutura é alterável ou ajustável. O objetivo destes ajustes é melhorar o desempenho do sistema em relação a algum critério desejado e às características do ambiente no qual o sistema está inserido (Astrom & Wittenmark, 1994). Dentre as características que tais sistemas podem assumir, tem-se a adaptação automática em ambientes que variam com o tempo, a possibilidade de serem treinados para realizar tarefas específicas (filtragem, tomada de decisões), a independência de procedimentos elaborados para sintetizá-los (geralmente necessários aos sistemas não-adaptativos) e a capacidade se "autoprojetarem" continuamente e de extrapolarem o seu padrão de comportamento para lidarem com novas situações (Widrow & Stearns, 1985). Em contrapartida, sistemas adaptativos mostram-se mais complexos e difíceis de analisar do que sistemas não-adaptativos. Isto deve-se a característica não-linear dos parâmetros dos sistemas adaptativos, os quais são variantes no tempo. No entanto, este tipo de sistema pode apresentar-se como uma alternativa eficaz para o controle de plantas cujas características são desconhecidas ou variantes no tempo, ou mesmo plantas instáveis ou sujeitas a perturbações internas (Ali, 2013) e (Widrow & Walach, 2007). Um exemplo de sistema que possui tais características é um filtro adaptativo (Farhang-Boroujeny, 2013).

O termo "filtro" pode ser usado para descrever um dispositivo físico (*hardware*) ou computacional (*software*) que coleta uma série de dados aleatórios e processa-os segundo um conjunto de regras pré-definidas para extrair informações a respeito de uma variável de interesse (Haykin, 1996). Quando os parâmetros de um filtro são capazes de alterarem seus respectivos valores com o tempo, diz-se que o filtro é "adaptativo" (Schilling & Harris, 2011). Um filtro adaptativo deve operar em um esquema de filtragem adaptativa, semelhante ao mostrado no exemplo da Figura 1.1. Neste esquema, filtro adaptativo produz, a partir de uma dada entrada, $x(n)$, e do vetor de coeficientes, $\mathbf{w}(n)$, uma saída, $y(n)$, a qual é comparada com um dado sinal desejado, $d(n)$, gerando um sinal de erro, $e(n)$. Então, o algoritmo adaptativo, A, é usado para modificar em cada instante de tempo, $n$, o vetor de coeficientes, $\mathbf{w}(n)$, com o intuito de tornar a saída do filtro adaptativo, $y(n)$, uma boa estimativa do sinal desejado, $d(n)$. Para atingir tal objetivo, o algoritmo adaptativo, A, usa os dados de $x(n)$ e $e(n)$ em uma regra de atualização do vetor de coeficientes, $\mathbf{w}(n)$, que busca otimizar uma função de custo escolhida apropriadamente (Farhang-Boroujeny, 2013). Durante a captação das amostras do sinal desejado, $d(n)$, um ruído de medição, $v(n)$, pode ser gerado. Este ruído interfere no processo de filtragem dificultando a estimação de $d(n)$.

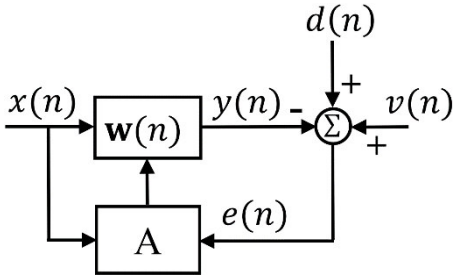



Figura 1.1. Esquema de Filtragem Adaptativa.

Alguns problemas nos quais os filtros adaptativos são empregados envolvem plantas esparsas, isto é, plantas nas quais apenas uma pequena parcela de sua resposta impulsiva é formada por coeficientes ativos, ou seja, não-nulos (Su, Jin & Gu, 2012). Para garantir um bom desempenho ao lidar com plantas esparsas, o filtro adaptativo pode fazer uso de algoritmos adaptativos que levam em consideração as características deste tipo de planta. Os Algoritmos adaptativos LMS (*least-mean-square*) normalizados proporcionais apresentam-se como alternativas eficientes para este tipo de problema (Souza, Tobias, Seara & Morgan, 2010) e (Wagner & Doroslovacky, 2011). Tais algoritmos são projetados para tirarem vantagem da estrutura e do comportamento das plantas esparsas (Huang, Benesty & Chen, 2006). Com o objetivo de situar o leitor a respeito da problemática alvo deste trabalho, a seguir são apresentados alguns conceitos básicos envolvendo plantas esparsas e algoritmos adaptativos LMS normalizados proporcionais.

## 1.1 Plantas Esparsas

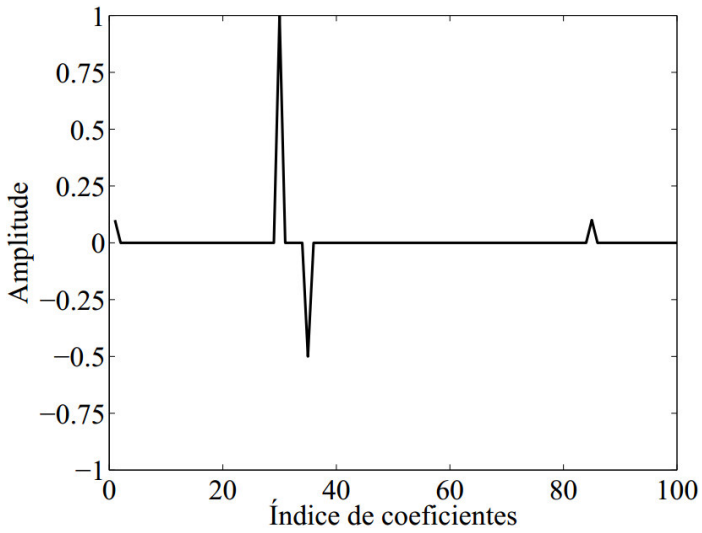
O termo "esparsidade" está relacionado com uma característica inerente a alguns vetores e matrizes. Em particular, considere um dado vetor

$$\mathbf{p} = [p_1\ p_2\ p_3\ \dots p_N]^{\mathrm{T}} \tag{1.1}$$

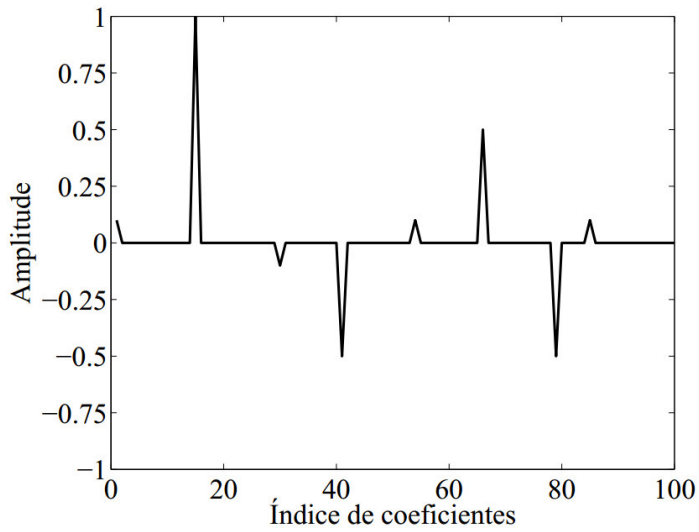
definido em um dado subespaço de dimensão finita de $\mathbb{R}^N$. Este vetor é dito esparso se a grande maioria de seus elementos, $p_i$, com $i = 1,2,\dots,N$, é nula. O sobrescrito T denota transposta de um vetor ou matriz. A característica esparsa de alguns sinais, plantas e ambientes tem sido objeto de estudo de várias áreas do conhecimento (Luo & Semlyem, 1990), (Hoyer, 2004) e (Starck, Murtagh & Fadili, 2010). Em especial, as plantas esparsas são encontradas em diversas aplicações, tais como estimação de harmônicas em sistemas elétricos de potência (Abdollahi, Zhang, Xue & Li, 2013), cancelamento de eco em telecomunicações (Deng & Doroslovacky, 2006) e (Loganathan, Khong & Naylor, 2009), estimação de canais de comunicação subaquáticos (Pelekanakis & Chitre, 2013) e de canais de comunicação com múltiplos percursos (Bajwa, Haupt, Sayeed & Nowak, 2010) e identificação de eventos sísmicos (Gabarda & Cristóbal, 2009). A título de exemplo, a Figura 1.2 mostra a disposição dos coeficientes de duas plantas consideradas esparsas. A Figura 1.2(a) refere-se a uma planta com esparsidade $S(\mathbf{p}) = 0{,}9435$ e a Figura 1.2(b) a outra planta desta vez com esparsidade $S(\mathbf{p}) = 0{,}8703$. A medida de esparsidade dessas plantas foi calculada segundo a definição (Hoyer, 2004) e (Huang, Benesty & Chen, 2006)

$$S(\mathbf{p}) = \frac{N}{\sqrt{N} - N}\left(1 - \frac{\|\mathbf{p}\|_1}{\sqrt{N}\|\mathbf{p}\|_2}\right) \tag{1.2}$$

onde $\|\mathbf{p}\|_1$ e $\|\mathbf{p}\|_2$ são, respectivamente, a norma-1 e a norma-2 do vetor de coeficientes da planta, $\mathbf{p}$. Tem-se também que $0 \leq S(\mathbf{p}) \leq 1$. Quanto mais próximo $S(\mathbf{p})$ estiver de 1, mais esparsa a planta é considerada. A variável $N$ representa a quantidade de coeficientes de $\mathbf{p}$. Cada uma das plantas esparsas mostradas na Figura 1.2 possui $N = 100$ coeficientes. A planta da Figura 2(a) possui apenas quatro coeficientes ativos localizados nas posições $\{1, 30, 35, 85\}$, com suas respectivas magnitudes iguais a $\{0{,}1\ , 1{,}0\ , -0{,}5\ , 0{,}1\}$. Já a planta da Figura 2(b) possui oito coeficientes ativos localizados nas posições $\{1, 15, 30, 41, 54, 66, 79, 85\}$, com suas respectivas magnitudes iguais a $\{0{,}1\ , 1{,}0\ , -0{,}1\ , -0{,}5\ , 0{,}1\ , 0{,}5\ , -0{,}5\ , 0{,}1\}$.




(a)




(b)

Figura 1.2. Exemplos de plantas esparsas com $N = 100$ coeficientes. (a) Planta com esparsidade $S(\mathbf{p}) = 0{,}9435$. (b) Planta com esparsidade $S(\mathbf{p}) = 0{,}8703$.

Para ilustrar a importância dos conceitos apresentados até este ponto, a seguir são mostrados alguns problemas que envolvem plantas e ambientes de característica esparsa.

## 1.2 Problemas Envolvendo Plantas Esparsas

Nesta seção são mostradas três aplicações práticas onde as plantas esparsas são encontradas, são eles: o cancelamento de eco em telecomunicações, a estimação de harmônicas em sistemas de energia elétrica e a estimação de canais de comunicação com múltiplos percursos.

### *1.2.1 Cancelamento de Eco em Telecomunicações*

Os sistemas de comunicação sem fio têm se mostrado como uma ferramenta essencial para os dias atuais devido à flexibilidade e velocidade na transmissão de informações proporcionada. Porém, para o caso da telefonia móvel, por exemplo, a experiência dos usuários pode ser prejudicada pela presença dos ecos acústicos. Já para os sistemas de comunicação via satélite, os caminhos percorridos pelos ecos da rede podem resultar em atrasos superiores a meio segundo em relação ao sinal de voz original, dificultando a comunicação. A Figura 1.3 mostra a disposição dos coeficientes de um caminho de eco de rede, obtida a partir da Recomendação ITU-T G.168, modelo #1 (ITU-T, 2015). Esta planta é composta por 512 coeficientes e é composta por duas regiões inativas, com 224 coeficientes nulos cada, e por uma região ativa representada por 64 coeficientes centrais.

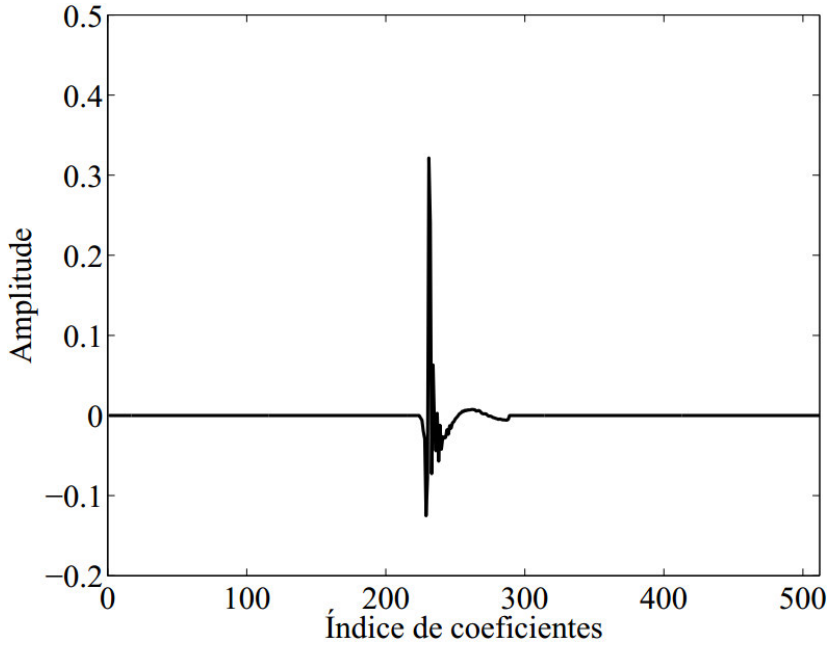



Figura 1.3. Resposta típica de um caminho de eco de rede.

Segundo a definição de (1.2), o grau de esparsidade do caminho de eco de rede da Figura 1.3 é $S(\mathbf{p}) = 0{,}8969$, que é considerado elevado.

*1.2.2 Estimação de Harmônicas em Sistemas Elétricos de Potência*

O espectro de frequências de um sinal de um sistema elétrico de potência também pode ser considerado um ambiente com um alto grau de esparsidade. Isto deve-se ao fato de que qualquer sinal transmitido pelo sistema (tensão ou corrente) é formado por uma componente fundamental e por componentes harmônicas e interharmônicas. A componente fundamental detém a maior parte da energia do sinal e, consequentemente, sua amplitude no espectro é consideravelmente superior à das demais. As componentes harmônicas estão localizadas em frequências múltiplas da fundamental. Uma vez que os sinais produzidos nas unidades geradoras são senóides, apenas harmônicas ímpares estão presentes. Além disso, para medição em alta tensão, as interharmônicas podem ser desprezadas (Abdollahi, Zhang, Xue & Li, 2013). A Figura 1.4 mostra o espectro de frequências de um sinal de tensão típico de um sistema elétrico de potência. Este sinal é formado pela componente fundamental em 60Hz e por duas componentes harmônicas (5ª e 7ª harmônica), além de uma componente interharmônica localizada na frequência 415Hz. A resolução deste espectro de frequências é de 1Hz e considera-se uma faixa de frequências que vai de 0 a 480Hz. As amplitudes das componentes fundamental, harmônicas e interharmônicas estão em p.u. (*per unit*). Este sinal pode ser descrito analiticamente por

$$v(t) = \cos(2\pi 60t) + 0{,}03\cos(2\pi 252t) + 0{,}1\cos(2\pi 300t) + 0{,}08\cos(2\pi 420t) \quad (1.3)$$

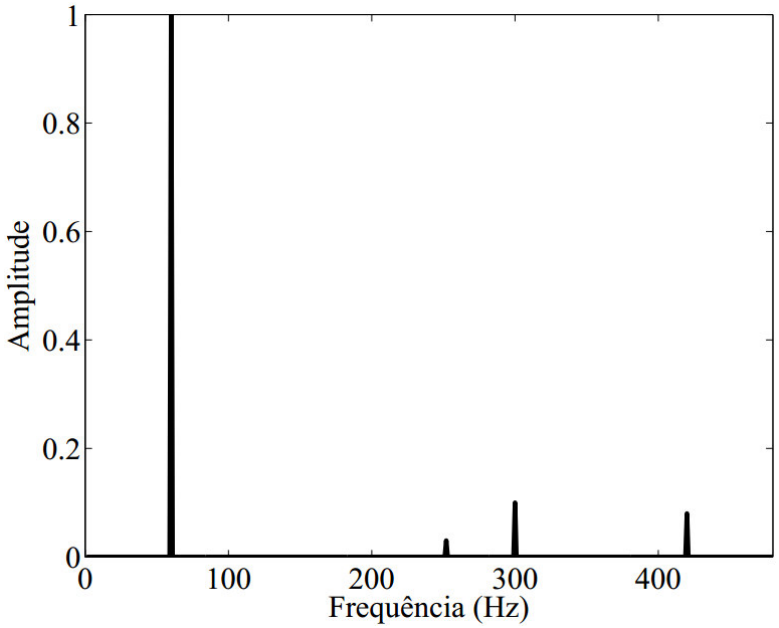



Figura 1.4. Espectro de frequências de um sinal de tensão típico de um sistema elétrico de potência.

O grau de esparsidade desse espectro de frequências é igual a $0{,}995$, segundo a definição de (1.2).

*1.2.3 Estimação de Canais de Comunicação com Múltiplos Percursos*

Um sinal de rádio transmitido em um ambiente de dispersão típico sofre reflexões, difrações dos objetos que compõem tal ambiente. Dessa forma, o sinal chega ao receptor como uma superposição de múltiplas cópias atrasadas, atenuadas e/ou com distorções no ângulo de fase em relação ao sinal original transmitido. Estas cópias são chamadas de componentes de múltiplos percursos do sinal. Em razão deste fenômeno, sistemas de comunicação sem fio com transmissão de dados em alta velocidade geralmente requerem um conhecimento da resposta do canal pelo receptor (Bajwa, Haupt, Sayeed & Nowak, 2010). A Figura 1.5 mostra um exemplo típico de um canal de comunicação com múltiplos percursos, o qual é usado em sistemas de comunicação digital de alta velocidade (Al-Shabilli *et. al*, 2015). Este canal possui tamanho igual a 32 coeficientes, porém apenas cinco deles são significativos. Os coeficientes ativos deste canal possuem valores iguais a $\{0{,}58\,,0{,}84\,,0{,}31\,,0{,}22\,,0{,}75\}$, localizados nas posições $\{2,\ 6, 14, 27,\ 0{,}75\}$, respectivamente.

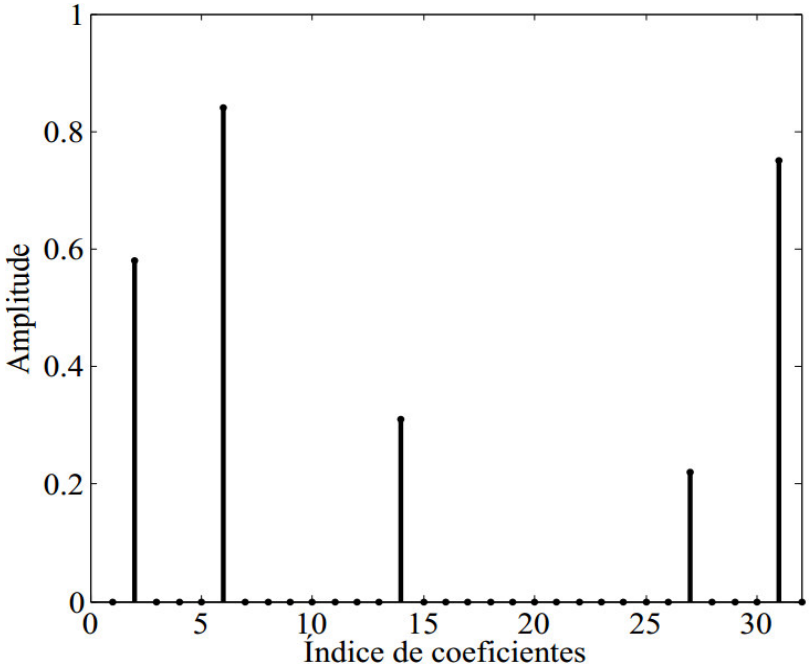



Figura 1.5. Exemplo de um canal de comunicação com múltiplos percursos.

O grau de esparsidade deste canal é igual a $0{,}7763$, segundo a definição de (1.2).

## 1.3 Algoritmos Adaptativos LMS Normalizados Proporcionais

Algoritmos adaptativos clássicos, tais como o LMS e o LMS normalizado (NLMS – *normalized* LMS), apresentam desempenho pobre, em termos de velocidade de convergência, ao lidarem com plantas esparsas (Widrow & Stearns, 1985) e (Huang, Benesty & Chen, 2006). Para solucionar este problema, foram desenvolvidos algoritmos que levam em consideração as características das plantas esparsas, tais como os algoritmos NLMS proporcional (PNLMS – *proportionate* NLMS), PNLMS melhorado (IPNLMS – improved PNLMS), e PNLMS com lei

$\mu$ (MPNLMS – $\mu$-*law* PNLMS). Estes algoritmos empregam uma filosofia proporcional de adaptação, ou seja, distribuem os ganhos de adaptação proporcionalmente à magnitude dos coeficientes do filtro adaptativo, resultando em uma maior velocidade de convergência para plantas com alto grau de esparsidade (Duttweiler, 2000), (Benesty & Gay, 2002) e (Deng & Doroslovacky, 2006).

O algoritmo PNLMS foi um dos primeiros algoritmos a empregar a filosofia proporcional no processo de adaptação dos coeficientes do filtro adaptativo (Duttweiler, 2000). No entanto, o desempenho deste algoritmo decai para plantas de esparsidade média ou baixa (Benesty & Gay, 2002). Além disso, a rápida velocidade inicial de convergência deste algoritmo não é mantida em todo o processo de adaptação (Deng & Doroslovacky, 2006). Para tentar contornar estes problemas, versões aprimoradas do algoritmo PNLMS foram desenvolvidas. O algoritmo PNLMS++ apresenta-se como opção para plantas de esparsidade média, pois alterna, em cada iteração do processo de adaptação, entre as regras de atualização dos algoritmos NLMS e PNLMS. Outra alternativa para este tipo de planta, e também para plantas de esparsidade elevada, é o algoritmo IPNLMS. Tal algoritmo foi desenvolvido a partir de modificações na estrutura do algoritmo PNLMS. O principal objetivo de tais modificações é explorar melhor a filosofia proporcional advinda do algoritmo PNLMS e, consequentemente, alcançar um desempenho superior não somente para plantas de alto grau de esparsidade, mas também para plantas de esparsidade média. Isto se dá pelo emprego de uma regra de distribuição dos ganhos de adaptação baseada em uma combinação convexa, a qual pondera entre contribuições relativas às regras de adaptação dos algoritmos NLMS e PNLMS. Porém, apesar do desempenho satisfatório para plantas de esparsidade média, o algoritmo IPNLMS não atinge a mesma velocidade inicial de convergência do algoritmo PNLMS para plantas altamente esparsas (Deng & Doroslovacky, 2006). Além disso, seu desempenho depende de uma constante própria deste algoritmo denominada parâmetro de proporcionalidade. Resultados experimentais demonstram que valores iguais a $0{,}0$, $-0{,}5$ e $-0{,}75$ são boas escolhas para tal parâmetro (Loganathan, Khong & Naylor, 2009).

Dentre as principais versões aprimoradas dos algoritmos PNLMS e IPNLMS desenvolvidas nos últimos anos tem-se os algoritmos MPNLMS, IIPNLMS, SC-IPNLMS (*sparseness controlled* PNLMS) e IAF-PNLMS (*individual activation factors* PNLMS). O algoritmo IIPNLMS usa a mesma regra de adaptação do algoritmo IPNLMS, mas com o emprego de dois parâmetros de proporcionalidade. O algoritmo MPNLMS distribui os ganhos de adaptação de forma proporcional ao logaritmo da magnitude dos coeficientes do filtro adaptativo. Já os algoritmos PNLMS com controle de esparsidade, a saber, os algoritmos SC-PNLMS, SC-MPNLMS e SC-IPNLMS, empregam estimativas da esparsidade da planta em suas respectivas regras de adaptação. Tais algoritmos são opções para plantas de esparsidade variável (Loganathan, 2011). O algoritmo IAF-PNLMS emprega fatores de ativação individuais para cada coeficiente do filtro adaptativo. Tais fatores são concebidos de forma que seus valores convirjam para as respectivas magnitudes dos coeficientes do filtro adaptativo. Outras abordagens correlatas aos algoritmos adaptativos proporcionais incluem os algoritmos LMS com atração para zero (ZA-LMS – *zero attraction* LMS), algoritmos LMS com restrição de

norma ($l_x$-LMS – $l_x$-*norm constraint* LMS), algoritmos LMS com esparsidade em bloco induzida (BS-LMS – *block sparse* LMS) e combinação convexa de filtros proporcionais (Chen, Gu & Hero III, 2009), (Gu, Jin, & Mei, 2009), (Shi & Shi, 2010), (Jiang & Gu, 2015) e (Arenas-García & Figueiras-Vidal, 2009).

## 1.4 Motivações para o Desenvolvimento de Algoritmos Adaptativos Proporcionais com Otimização de Parâmetros

O comportamento dos algoritmos PNLMS e IPNLMS depende de parâmetros que controlam a proporcionalidade e a inicialização, os quais mostram-se como sendo de difícil ajuste (Jelfs, Mandic & Benesty, 2007). Os parâmetros do algoritmo PNLMS, por exemplo, afetam diretamente na adaptação dos coeficientes considerados inativos. Além disso, uma das principais funções destes parâmetros é evitar a estagnação dos coeficientes do filtro (Souza, Tobias, Morgan & Seara, 2010). Já para o algoritmo IPNLMS, a principal função de seu respectivo parâmetro de proporcionalidade é controlar as contribuições das parcelas proporcional e não-proporcional, de forma a garantir uma regra de atualização adequada ao grau de esparsidade da planta considerada (Loganathan, Khong & Naylor, 2009). Porém, sabe-se que a esparsidade de uma planta pode variar com fatores como temperatura e pressão (Khong & Naylor, 2006). Para o algoritmo PNLMS, valores inadequados para o parâmetro de porpocionalidade ou para o fator de ativação, podem tornar mais lenta a adaptação dos coeficientes de menor magnitude. Além disso, valores constantes para tais parâmetros podem comprometer o desempenho do algoritmo PNLMS quando da identificação de plantas de esparsidade variável no tempo. Analogamente, para o algoritmo IPNLMS, o uso de um valor de parâmetro de proporcionalidade constante pode comprometer o desempenho do IPNLMS ao lidar com plantas com resposta impulsiva variável. Além disso, o valor desse parâmetro deve ser escolhido de acordo com o grau de esparsidade da planta. Dessa forma, um valor inapropriado para tal parâmetro pode degradar o desempenho do IPNLMS.

Assim, o ponto central para aprimorar o desempenho dos algoritmos adaptativos proporcionais consiste em estabelecer valores adequados para os parâmetros destes algoritmos, uma vez que, conforme as análises realizadas neste trabalho de pesquisa, tem-se que tais parâmetros impactam, durante todo o processo de adaptação, no comportamento dos referidos algoritmos para a identificação de plantas esparsas. Dessa forma, neste trabalho são propostas novas metodologias para otimizar a escolha dos parâmetros dos algoritmos PNLMS e IPNLMS. Para tanto, são empregados procedimentos baseados nos métodos de otimização da razão áurea e da busca tabu. Resultados de testes numéricos mostram que os novos algoritmos propostos alcançam velocidades de convergência superiores, bem como resposta mais rápida a perturbações na planta, quando comparados com os algoritmos PNLMS e IPNLMS para a identificação de plantas com elevado grau de esparsidade.

## 1.5 Objetivos da Pesquisa

Os objetivos pretendidos por este projeto de pesquisa estão listados a seguir:

- Realizar análises dos impactos dos parâmetros dos algoritmos adaptativos LMS normalizados proporcionais na velocidade de convergência e na distribuição dos ganhos de adaptação destes algoritmos.
- Verificar a viabilidade do uso de metodologias para aprimorar o desempenho dos algoritmos adaptativos proporcionais em termos de velocidade de convergência.
- Empregar procedimentos com requisitos computacionais reduzidos baseados em métodos de otimização para otimizar a escolha dos parâmetros dos algoritmos adaptativos proporcionais para identificação de plantas esparsas.
- Desenvolvimento de novos algoritmos adaptativos proporcionais com otimização de parâmetros dedicados à identificação de plantas de elevado grau de esparsidade.
- Comparar o desempenho dos novos algoritmos desenvolvidos, em termos de velocidade de convergência, perturbações ocorridas no vetor de coeficientes e variações na esparsidade da planta, com outros algoritmos clássicos dedicados à identificação de plantas esparsas.

A seguir, são mostrados os trabalhos publicados a partir deste projeto de pesquisa.

## 1.6 Trabalhos Publicados

Durante a execução deste projeto de pesquisa, foram publicados e apresentados dois artigos científicos em fóruns especializados, os quais estão listados a seguir:

1) Branco, C. A. S. C., Souza, F. C. (2016) Algoritmo IPNLMS com parâmetro de proporcionalidade ótimo. XXXIV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2016), Santarém-PA, Brasil, 30 de agosto a 02 de setembro de 2011.
2) Branco, C. A. S. C., Souza, F. C. (2016) Algoritmo IPNLMS com otimização do parâmetro de proporcionalidade baseada em busca tabu para identificação de plantas esparsas. XXI Congresso Brasileiro de Automática (CBA 2016), Vitória-ES, Brasil, 03 a 07 de outubro de 2016.

## 1.7 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 especial para os algoritmos adaptativos da classe LMS. Primeiramente, são mostrados problemas práticos nos quais os filtros adaptativos são aplicados e a estrutura de filtragem adotada pelos algoritmos adaptativos considerados neste trabalho. Em seguida, são feitas breves discussões a respeito do funcionamento dos filtros lineares ótimos, os quais deram origem aos filtros adaptativos. Por fim, são apresentados os algoritmos de descida mais íngreme (SD – *steepest descent*), LMS e NLMS.

- Capítulo 3: apresenta algoritmos adaptativos LMS normalizados proporcionais, os quais foram concebidos para melhorar o desempenho dos filtros adaptativos para identificação e controle de plantas esparsas. Inicialmente, é mostrada estrutura dos algoritmos PNLMS padrão e IPNLMS, juntamente com uma análise dos parâmetros destes algoritmos no comportamento dos mesmos para a identificação de plantas esparsas. Por fim, são mostrados alguns dos principais algoritmos adaptativos proporcionais desenvolvidos recentemente, os quais são derivações dos algoritmos PNLMS e IPNLMS.
- Capítulo 4: apresenta conceitos básicos envolvendo problemas e métodos de otimização. Também são apresentados os métodos adotados por este trabalho de pesquisa para otimizar a escolha dos parâmetros dos algoritmos adaptativos proporcionais. São eles os métodos da razão áurea e da busca tabu.
- Capítulo 5: neste capítulo, são obtidos os novos algoritmos adaptativos proporcionais propostos, os quais empregam métodos de otimização para determinar os valores adequados dos parâmetros dos algoritmos PNLMS e IPNLMS para a identificação de plantas esparsas. Primeiramente, é apresentada a função objetivo a ser minimizada pelas metodologias adotadas, juntamente com uma análise do efeito dos parâmetros dos algoritmos PNLMS e IPNLMS no comportamento desta função. Por fim, os novos algoritmos PNLMS e IPNLMS propostos são obtidos.
- Capítulo 6: neste capítulo, são apresentados os resultados obtidos com o emprego dos algoritmos propostos em problemas de identificação de plantas esparsas. O desempenho dos algoritmos propostos é comparado com o dos algoritmos PNLMS, IPNLMS e SC-IPNLMS em termos de velocidade de convergência, resposta a perturbações na planta, além de identificação de plantas de esparsidade variável.
- Capítulo 7: apresenta as conclusões deste trabalho de pesquisa, além de indicar possíveis sugestões de trabalhos futuros.

# Capítulo 2

# Filtragem Adaptativa

Filtros adaptativos são ferramentas fundamentais para a resolução de vários problemas da área de processamento de sinais. Dentre as aplicações destes dispositivos pode-se citar a identificação de sistemas, o cancelamento de eco em telecomunicações, o controle ativo de ruído, a equalização de canais e a predição de entradas futuras. Em linhas gerais, o estudo destes filtros consiste na análise e no desenvolvimento de algoritmos adaptativos, os quais são implementados em estruturas de filtragem dedicadas à resolução de determinados problemas envolvendo processamento de sinais. Dependendo do tempo necessário para atingir o objetivo do processo de adaptação, além do compromisso entre o esforço requerido e os recursos computacionais disponíveis, pode-se escolher entre uma variedade de algoritmos adaptativos e de estruturas de filtro (Nascimento & Silva, 2014), (Farhang-Buroujeny, 2013) e (Sayed, 2003).

Os filtros podem ser classificados em lineares e não-lineares. A saída de um filtro linear é uma função linear dos valores aplicados na entrada. Caso isto não ocorra, o filtro é dito não-linear. Como, na prática, a maioria dos sinais envolvidos em problemas de filtragem linear possuem certo grau de aleatoriedade, geralmente recorre-se a uma abordagem estatística. Esta, por sua vez, requer o conhecimento de alguns parâmetros, tais como as funções de valor médio e de autocorrelação. Uma maneira eficiente de resolver este tipo de problema dá-se a partir da minimização do valor quadrático médio do erro entre o sinal desejado e a saída do filtro. Quando os sinais envolvidos no processo de filtragem são processos estocásticos estacionários, a solução ótima pode ser obtida a partir do filtro de Wiener. Tal solução é dita ótima no sentido médio quadrático. No entanto, a abordagem de Wiener requer conhecimento *a priori* das propriedades estatísticas dos dados a serem processados, além de ser inadequada para situações onde os sinais envolvidos não são processos estacionários, onde o filtro precisa assumir um caráter variante no tempo.

Uma alternativa viável para lidar com a não-estacionariedade, ou mesmo a falta de conhecimento das propriedades estatísticas dos sinais envolvidos, são os filtros adaptativos. Estes filtros foram desenvolvidos tendo como base a teoria de filtragem linear ótima e demonstram-se eficientes para aplicações em tempo real (Haykin, 1996). Isto deve-se à capacidade destes dispositivos de se auto ajustarem às mudanças do ambiente no qual estão inseridos. Tal capacidade reside no uso de um algoritmo adaptativo, o qual converge para a solução ótima de Wiener para o caso estacionário. Em contrapartida, para o caso não estacionário, este tipo de algoritmo possui uma característica de rastreamento, a despeito das variações na estatística dos sinais envolvidos. Vários algoritmos adaptativos têm sido desenvolvidos na literatura para a operação em filtros lineares adaptativos (Diniz, 2012). A

escolha do algoritmo mais adequado para uma determinada aplicação deve ser feita com base em fatores tais como velocidade de convergência, capacidade de rastreamento, robustez a pequenas perturbações, estrutura de implementação, dentre outros (Haykin, 1996).

Este capítulo aborda alguns dos aspectos fundamentais da teoria dos filtros adaptativos, com enfoque voltado para os algoritmos adaptativos da classe LMS. Primeiramente, são mostrados alguns problemas práticos nos quais os filtros adaptativos são empregados. Em seguida, é apresentada uma estrutura chamada de filtro transversal, o qual é usado na implementação dos algoritmos adaptativos considerados neste trabalho. Logo após, é feita uma breve discussão a respeito de processos estocásticos estacionários, com o objetivo de apresentar algumas funções e elementos estatísticos de interesse para a operação dos filtros lineares ótimos, os quais deram origem aos filtros adaptativos. Posteriormente, são apresentados alguns conceitos do filtro de Wiener, com o intuito de construir a base teórica necessária para entender o funcionamento dos filtros adaptativos. Por fim, são apresentados três algoritmos. Os dois primeiros são o algoritmo de descida mais íngreme (SD – *steepest descent*) e a implementação estocástica deste, a saber, o algoritmo LMS (Widrow & Stearns, 1985). Já o terceiro é uma evolução do algoritmo LMS, o qual é chamado de LMS normalizado ou NLMS. Os algoritmos adaptativos proporcionais, objeto de estudo deste trabalho, são derivações do algoritmo NLMS.

## 2.1 Aplicações dos Filtros Adaptativos

A natureza auto projetável dos filtros adaptativos confere a estes dispositivos a possibilidade de serem aplicados em vários campos da ciência e engenharias (Nascimento & Silva, 2012). Em outras palavras, a capacidade destes filtros de se ajustarem automaticamente às mudanças do ambiente faz com que estes dispositivos desempenhem um papel relevante em vários problemas práticos. Dentre as áreas nas quais os filtros adaptativos têm sido empregados com sucesso pode-se citar o controle de processos, as telecomunicações, a engenharia biomédica, a sismologia, aplicações envolvendo radar, sonar, dentre outras (Widrow & Stearns, 1985), (Sayed, 2003), (Adali & Haykin, 2010) e (Diniz, 2012). Embora distingam-se devido à natureza de cada problema, tais aplicações possuem como característica comum o uso de um vetor de entrada e de uma saída desejada para computar um erro de estimação, o qual é por sua vez usado para ajustar os valores de um conjunto de parâmetros variáveis do filtro adaptativo. Entretanto, a principal diferença entre as várias aplicações destes filtros reside na forma como a saída desejada é obtida (Haykin, 1996). Com o objetivo de mostrar a importância dos filtros adaptativos, esta seção apresenta quatro exemplos práticos de aplicação destes dispositivos. São eles a identificação de sistemas, o cancelamento de eco em telecomunicações, a equalização de canais de comunicação e o controle ativo de ruído.

### *2.1.1 Identificação de Sistemas*

A Figura 2.1 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, consequentemente, tornar $y(n)$ uma boa estimativa de $d(n)$.

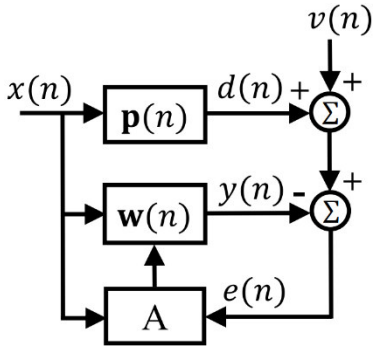



Figura 2.1. Aplicação de um filtro adaptativo em um problema de identificação de sistemas.

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.

A maioria dos sistemas de controle modernos realiza uma identificação on-line da planta a ser controlada. O resultado desta identificação pode ser usado, por exemplo, em um Controlador Auto-ajustável (STR – *Self-Tuning Regulator*), como o mostrado no exemplo da Figura 2.2. Para este tipo de sistema de controle, é necessário que o modelo da planta seja especificado. A variável $r(n)$ representa o sinal de referência. Já as variáveis $x(n)$ e $d(n)$ são, respectivamente, a entrada e a saída da "Planta". A identificação ou estimação on-line dos parâmetros da "Planta" é realizada pelo "Filtro Adaptativo", o qual usa o sinal de erro, $e(n)$, para ajustar os seus coeficientes com o objetivo de assimilar o comportamento da "Planta". O resultado da estimação é então enviado ao bloco denominado "Projeto do Controlador", o qual usa os dados dos parâmetros estimados da "Planta", além das "Especificações" da política de controle adotada, para ajustar os parâmetros do "Controlador". Este último, por sua vez, aplica um sinal de entrada, $x(n)$, na "Planta" com o objetivo fazer com que a sua saída, $d(n)$, siga a referência, $r(n)$.

Note, da Figura 2.2, que este tipo de sistema de controle possui dois laços de realimentação. O primeiro é o laço interno de realimentação unitária entre a planta e o controlador. O outro é o laço externo responsável por ajustar os parâmetros do controlador. O termo auto-ajustável foi usado para expressar a capacidade dos parâmetros do controlador de convergirem para o controlador que deveria ser projetado caso a dinâmica da planta fosse conhecida. Um resultado interessante do uso deste sistema de controle adaptativo é que isto pode ocorrer mesmo se a estrutura escolhida para estimar os parâmetros da planta estiver incorreta (Astrom & Wittenmark, 1994). Para realizar a estimação dos parâmetros da planta utiliza-se um esquema semelhante ao da Figura 2.1.

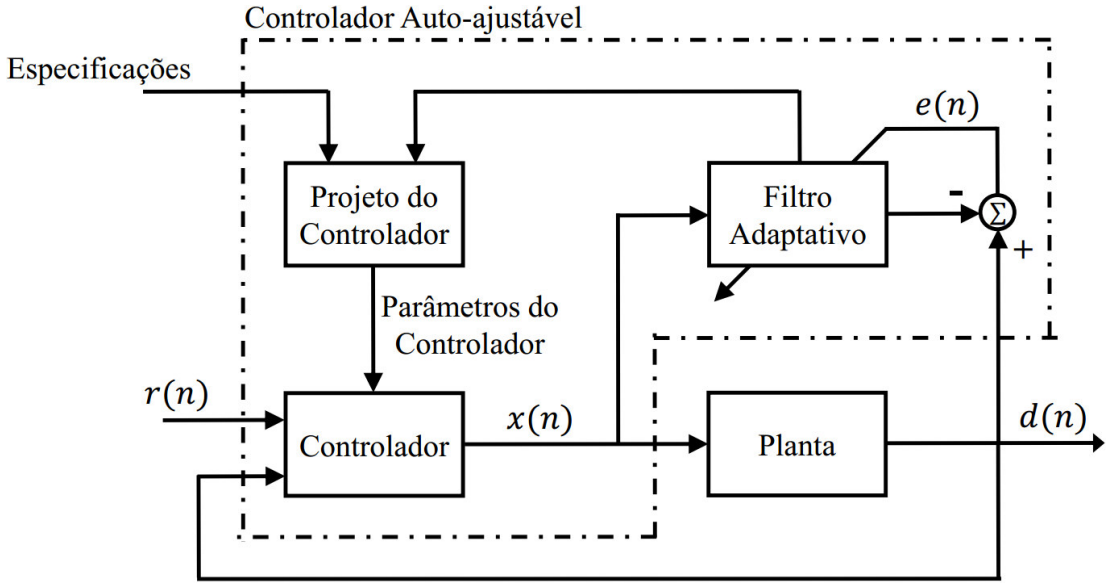



Figura 2.2. Diagrama de blocos de um controlador auto-ajustável

Como exemplos de algoritmos adaptativos que podem ser empregados neste processo de estimação, tem-se o de mínimos quadrados recursivo (RLS – *recursive least squares*) e o LMS. Pode-se escolher entre uma ampla variedade de modelos de planta e estruturas de controlador. Também existem várias técnicas de projeto de controlador que podem ser usadas.

### *2.1.2 Cancelamento de Eco em Telecomunicações*

Raramente uma conversa entre dois usuários de um sistema de telecomunicações acontece sem eco. Um dado usuário "A" escuta o eco da própria voz durante uma conversa com um outro usuário "B" quando as ondas do sinal de voz de "A", as quais são emitidas pelo autofalante de "B", sofrem reflexão nas paredes e demais objetos do ambiente onde "B" está e são captadas pelo microfone deste último (Nascimento & Silva, 2014). Há também a possibilidade de parte das ondas emitidas pelo autofalante de "B" serem captadas diretamente pelo microfone sem que haja reflexão das mesmas. O usuário "B" também escuta o eco da própria voz quando algo semelhante ocorre do outro lado da linha, ou seja, onde o usuário "A" está. Como exemplo, considere que "B" está se comunicando à distância com "A" a partir de um telefone celular em modo viva-voz. Neste caso, as ondas de sinal de voz de "A", transmitidas pelo sistema, serão emitidas pelo autofalante do celular de "B" e irão se propagar dentro do carro. Então, estas ondas sofrerão reflexões toda vez que encontrarem obstáculos dentro do carro, gerando um sinal de eco da voz de "A". O microfone de "B" capta então parte deste eco, o qual será transmitido pelo sistema de volta até "A". Consequentemente, "A" escutará o eco da própria voz durante a conversa com "B", caso o sistema não possua aparelhos capazes de eliminar tal eco.

O exemplo do parágrafo anterior refere-se à ocorrência de um eco acústico. Contudo, ecos também podem ser gerados dentro dos circuitos de telefonia. Neste caso, quando um sinal transmitido encontra, em algum ponto, um desbalanceamento de impedâncias, parte deste sinal acaba sendo refletido, tornando-se em eco. O dano que tal eco pode causar à operação do

sistema depende da potência do sinal refletido, da distorção causada no espectro de frequências e do atraso em relação ao sinal original (Sondhi & Berkley, 1980). Quando a defasagem entre o sinal original e o refletido seja pequena, o usuário do sistema não é capaz de identificar o eco. Porém, nesta situação, os efeitos causados pelo eco são percebidos na forma de distorção espectral ou reverberação. Em particular, um tipo de eco praticamente instantâneo, chamado de tom lateral (*sidetone*), ocorre sempre que há acoplamento acústico entre a orelha do usuário e o aparelho de telefone, ou seja, quando estes estão muito próximos ou mesmo em contato físico. O efeito deste eco é sentido como uma leve vibração do aparelho, resultante das múltiplas reflexões das ondas sonoras no ouvido do usuário e no telefone. Por outro lado, ecos acústicos ou de rede defasados de algumas centenas de milissegundos causam uma sensação de irritação no usuário, impedindo que este consiga se comunicar corretamente, isto é, sem balbuciar ou interromper o raciocínio repetidas vezes. Este incômodo ocorre mesmo em situações onde a potência do eco é muito menor do que a do sinal original.

Ecos com defasagem suficiente para prejudicarem sistemas de telecomunicações são observados somente em conexões de longa distância, tais como as de sistemas de comunicação via satélite. Como um exemplo da magnitude que estas distâncias podem alcançar, considere um sistema que utiliza satélites geoestacionários. Tais satélites ficam posicionados a cerca de 37 mil quilômetros acima da superfície da terra. O caminho de ida e volta a ser percorrido por um sinal de eco nesse sistema corresponde a quatro vezes esta distância. Isto resulta em um atraso de pelo menos 500 milissegundos em relação ao sinal de voz original (Sondhi, 2006).

Um esquema simplificado de um sistema telefônico de longa distância é mostrado na Figura 2.3. Neste esquema, cada linha sólida representa um circuito de dois fios. Cada telefone (usuários "A" e "B") conecta-se com a central de telefonia a partir de uma linha de comunicação local composta por dois fios, denominada de linha de assinante (*subscriber's loop*). Esta linha é usada para envio e recebimento de sinais de voz. O uso destas linhas com dois fios deve-se primeiramente à economia proporcionada com o uso de menos fios e dispositivos de chaveamento. A outra vantagem desta prática é que, no caso de uma ligação local, basta conectar diretamente na central as linhas dos dois assinantes envolvidos. Porém, para comunicações de longa distância, são necessárias duas linhas distintas, $L_1$ e $L_2$, uma para envio e outra para recepção de dados, respectivamente.

Uma das razões desta prática deve-se ao fato de que circuitos longos requerem o uso de amplificadores, os quais são dispositivos unidirecionais. A outra razão é econômica, uma vez que chamadas de longa distância devem ser multiplexadas. Diferentes chamadas usam porções distintas da largura de banda de um dado canal de comunicação. A multiplexação, por sua vez, requer que os sinais transmitidos em cada direção sejam enviados através de canais distintos. Logo, quando tal separação de caminhos é necessária, deve-se providenciar uma forma de conectar os circuitos de dois fios dos assinantes ao circuito de quatro fios do sistema, o qual é formado por $L_1$ e $L_2$.

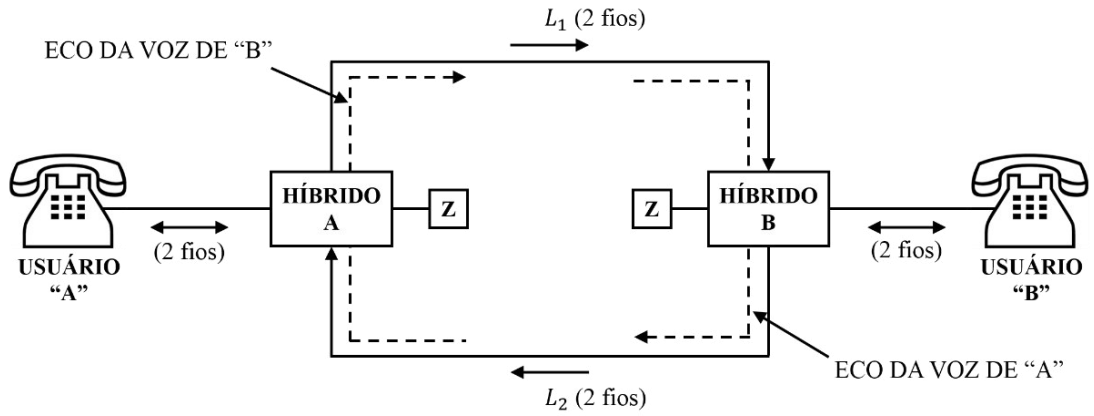



Figura 2.3. Esquema simplificado de um circuito de telefonia para uma conexão de longa distância

*Fonte: Autor*

O responsável por executar tal tarefa é o transformador diferencial ou transformador híbrido ou, simplesmente, híbrido (*hybrid*). Como mostra a Figura 2.3, é neste dispositivo onde surgem os ecos da rede telefônica. Isto se dá pelo desbalanceamento entre as impedâncias da linha do assinante e do circuito de quatro fios. Para tentar resolver este problema, cada híbrido faz uso de uma impedância de balanceamento, representada pelo bloco com a letra "Z" na Figura 2.3. No entanto, um dado híbrido (o qual faz parte de um circuito de quatro fios) é responsável por se conectar com um grupo de várias linhas de assinante. Cada uma destas, por sua vez, pode ser composta pelos mais comprimentos e tipos de materiais. Portanto, é praticamente impossível balancear a impedância "Z" de um mesmo híbrido a cada uma das impedâncias das suas respectivas linhas de assinante. Então, conforme exposto anteriormente, uma consequência inevitável deste desbalanceamento é o eco de rede (também chamado de eco de linha).

Os primeiros canceladores de eco foram idealizados na década de 1960 (Sondhi & Presti, 1966) e (Sondhi, 1967). O princípio básico de tais dispositivos consiste em produzir uma réplica sintética do sinal de eco para então subtraí-la do mesmo. Isto deve ser feito sem que a comunicação entre os usuários seja interrompida. Se a resposta impulsiva do canal por onde o eco se propaga possuir característica linear, então um filtro linear pode ser usado para produzir uma réplica do sinal desejado e, portanto, este eco pode ser cancelado (Sondhi, 2006). Os filtros empregados neste método necessitam da injeção de sinais de teste na sua entrada para o aprendizado dos mesmos. Isto deve ser feito de forma itermitente e durante as conversações. Como os caminhos de propagação dos ecos não são ambientes perfeitamente estacionários, isto é, suas propriedades estatísticas não são totalmente invariantes com o tempo, foram desenvolvidos os canceladores de eco adaptativos. Tais canceladores utilizam o próprio sinal de voz do usuário como entrada do filtro (Sondhi, 1967) e (Kelly & Logan, 1970).

Concebido inicialmente como um dispositivo analógico (Sondhi & Presti, 1966), o cancelador de eco só pôde ser comercializado após 1980, ano da primeira implementação deste tipo de dispositivo em um circuito integrado (Duttweiler & Chen, 1980). A partir daí a capacidade de processamento dos canceladores de eco só aumentou. Este fato também possibilitou o desenvolvimento e a implementação de algoritmos adaptativos de estrutura

simples e robusta tais como o LMS, o NLMS e o PNLMS. Em particular, o NLMS foi e continua sendo o algoritmo mais usado em canceladores de eco de todo o mundo (Paleologu, Ciochina, Benesty & Gay, 2015). Embora, na prática, o algoritmo PNLMS demonstre possuir maior velocidade de convergência do que o NLMS, ainda não existe uma prova matemática que explique as razões de sua convergência (Sondhi, 2006).

A Figura 2.4 mostra o diagrama de blocos de um filtro adaptativo empregado na resolução de um problema de cancelamento de eco de rede. Note que o filtro, $\mathbf{w}(n)$, usa como entrada o sinal de voz de "A", transmitido pela linha $L_1$, para gerar uma saída que por sua vez é subtraída do eco de "A", o qual é gerado no híbrido conectado a "B" e transportado pela linha $L_2$. O resultado dessa subtração é então usado pelo algoritmo adaptativo para modificar os coeficientes do filtro, $\mathbf{w}(n)$, de forma que a sua saída seja uma boa estimativa do eco de "A" e, consequentemente, este eco possa ser devidamente cancelado.

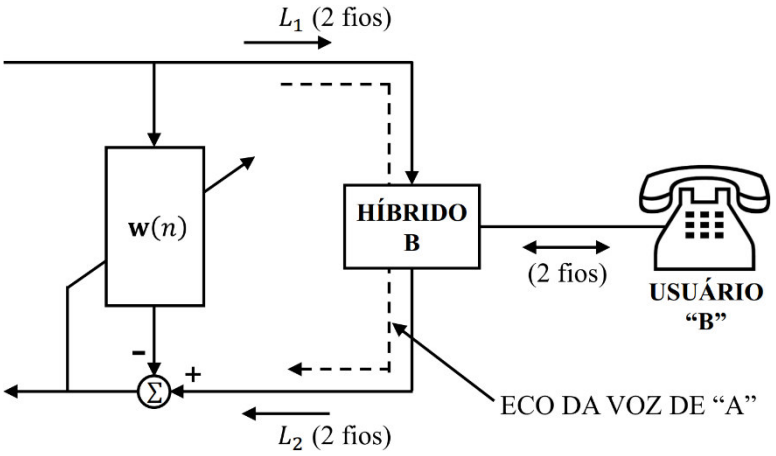



Figura 2.4. Filtro adaptativo empregado em um problema de cancelamento de eco de rede.

Desde a década de 1990, um problema similar ao cancelamento de eco rede vem ganhando bastante destaque. Trata-se do cancelamento de eco acústico. As principais aplicações dos canceladores de eco acústico são as teleconferências e as comunicações sem fio em geral (Sondhi, 2006). Apesar das semelhanças entre estes dois problemas, existem algumas diferenças de cunho prático que diferenciam o cancelamento de eco acústico do de rede. A primeira está no comprimento da resposta impulsiva dos caminhos de eco. Os caminhos de eco acústico possuem respostas impulsivas com número de coeficientes superior aos de rede. Como consequência disto, tem-se que o cancelamento de eco acústico exige filtros de ordem superior aos usados nos canceladores de eco de rede. Outra diferença está na variabilidade da resposta do caminho de eco. Os caminhos de eco acústico demonstram-se mais sensíveis às variações no ambiente (Loganathan, Khong & Naylor, 2009) e (Farhang-Boroujeny, 2013).

Claramente, o cancelamento de eco acústico também pode ser visto como um problema de identificação de sistemas, como mostra o exemplo da Figura 2.5. Neste caso, $x(n)$ é um sinal de voz proveniente de um usuário de longa distância do sistema de telecomunicações. Este sinal de voz é então recebido por um usuário local do sistema através de um dispositivo tal como um

telefone celular, por exemplo. Então, o autofalante (AF) reproduz $x(n)$ que se propaga pelo ambiente onde o usuário local está, gerando eco. Este ambiente também é chamado de "caminho de eco" e pode ser modelado por uma planta cuja resposta impulsiva é $\mathbf{p}(n)$. Então, o eco $d(n)$ é captado e transmitido pelo microfone (MIC). Este eco deve ser cancelado para evitar que o usuário de longa distância escute sua própria voz enquanto estiver conversando com o usuário local. Note que o microfone também capta o ruído do ambiente onde o usuário local está, além da voz do mesmo. A composição destes sinais é representada por $v(n)$. O filtro adaptativo $\mathbf{w}(n)$ é usado para modelar a resposta impulsiva $\mathbf{p}(n)$ do ambiente que gera $d(n)$. Uma réplica $y(n)$ de $d(n)$ é então obtida e subtraída deste último. Resultados de aplicações de filtros adaptativos como canceladores de eco acústico mostram que, para este tipo de problema, apenas uma pequena parcela de $\mathbf{p}(n)$ é formada por coeficientes não-nulos (Loganathan, Khong & Naylor, 2009).

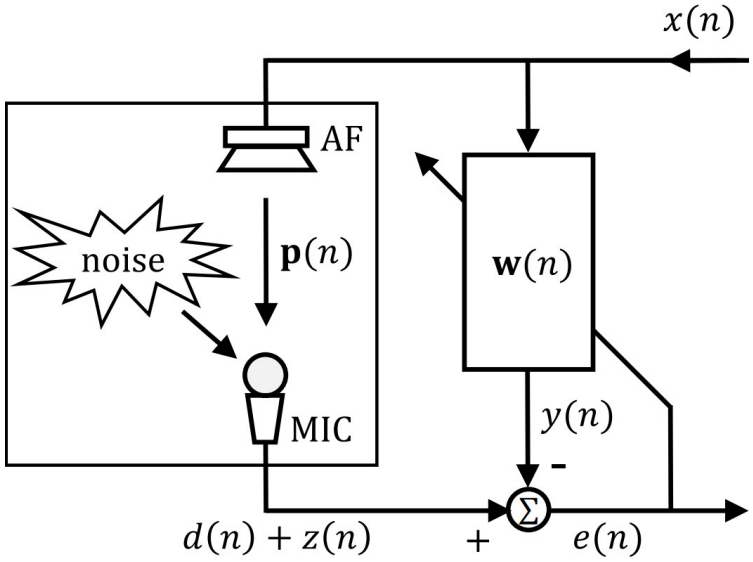



Figura 2.5. Filtro adaptativo aplicado em um problema de cancelamento de eco acústico.

Os problemas de cancelamento de eco acústico e de rede têm sido, desde o final da década de 1990, os principais motivadores para o desenvolvimento dos algoritmos adaptativos proporcionais (Benesty, Paleologu & Ciochina, 2010). Isto se deve à característica esparsa dos caminhos de eco. Algoritmos tais como o PNLMS, o IPNLMS, o MPNLMS, o SC-IPNLMS e o SC-PNLMS foram concebidos com o intuito de aprimorar o desempenho dos filtros adaptativos empregados na resolução deste tipo de problema (Duttweiler, 2000), (Benesty & Gay, 2002), Deng & Doroslovacky, 2006) e (Loganathan, Khong & Naylor, 2009). Outras abordagens correlatas dedicadas, tais como os algoritmos da classe ZA-LMS, também foram desenvolvidas na tentativa de resolver problemas envolvendo eco (Das & Chakraborty, 2014).

### *2.1.3 Equalização de Canais de Comunicação*

Outra aplicação de filtros adaptativos empregada em várias áreas da engenharia é a chamada modelagem inversa ou deconvolução. Em particular, o uso de um filtro adaptativo como uma espécie de "modelo inverso" de uma determinada planta é de especial interesse do

ramo das telecomunicações. Especificamente, o filtro usado neste tipo de aplicação tem como função eliminar as distorções inseridas por um dado canal de comunicação nos dados transmitidos pelo mesmo. O filtro empregado para realizar tal tarefa é chamado de equalizador.

A transmissão digital de dados em alta velocidade pode causar distorções nos símbolos transmitidos por um canal de comunicação. Isto acontece em razão da característica dispersiva de alguns canais e das não-linearidades introduzidas pelos processos de modulação e demodulação. O equalizador precisa ser rápido o bastante para garantir a comunicação eficiente em tempo real. A Figura 2.6 mostra o diagrama de blocos de um sistema de transmissão de dados equipado com um equalizador de canais. A função de transferência do canal, $H(z)$, representa a resposta combinada do próprio canal de comunicação, juntamente com os dispositivos de emissão e recepção dos símbolos transmitidos, $s(n)$. O ruído, $v(n)$, pode ocorrer devido ao aquecimento dos circuitos do canal e também pela interferência decorrente de possíveis conversas transmitidas por canais vizinhos. Estas distorções produzidas pelo canal podem causar um fenômeno conhecido como interferênicia entre símbolos (ISI – *Intersymbol Interference*), resultando em uma queda de desempenho dos dispositivos de interpretação e recepção dos símbolos. Idealmente, o equalizador deve possuir uma função de transferência, $W(z)$, igual ao inverso da do canal de comunicação, ou seja

$$W(z) = \frac{1}{H(z)} \tag{2.1}$$

de forma que

$$H(z)W(z) = 1. \tag{2.2}$$

Contudo, note que um equalizador com uma função de transferência dada por (2.1) pode ser não-causal caso $H(z)$ possua algum zero fora do círculo unitário, $|z| \leq 1$. Para solucionar este problema, pode-se escolher um equalizador tal que

$$H(z)W(z) = z^{-\Delta} \tag{2.3}$$

onde $\Delta$ é um valor inteiro apropriadamente escolhido para o atraso. Logo, a saída do equalizador, $y(n)$, é uma réplica atrasada dos símbolos transmitidos pelo canal, ou seja

$$y(n) = \hat{s}(n - \Delta)\,. \tag{2.4}$$

Note também que a escolha de um equalizador tal que $W(z) = z^{-\Delta}/H(z)$ pode levar a um aumento na amplitude do ruído aditivo, $v(n)$, nas frequências onde a magnitude de $H(z)$ é pequena. Portanto, a escolha de um equalizador, $W(z)$, deve ser feita levando em consideração o balanço entre a redução da ISI e a da magnitude do ruído na saída do equalizador (Farhang-Boroujeny, 2013). O filtro de Wiener pode prover uma solução que garanta tal balanço. No entanto, esta abordagem está limitada a um regime de estacionariedade dos dados envolvidos (Haykin, 1996). Uma alternativa eficiente capaz de contornar este problema é o uso de um equalizador adaptativo.

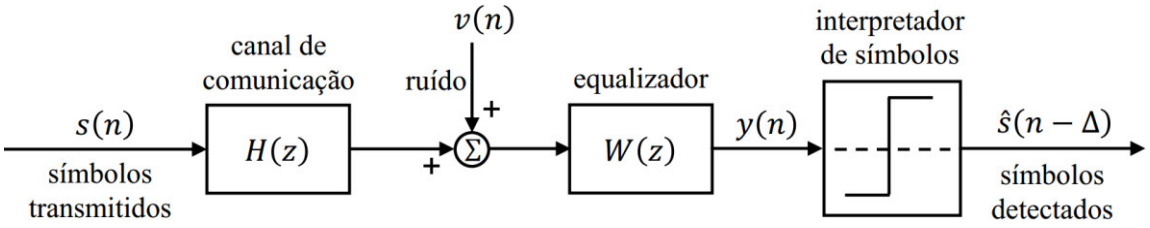



Figura 2.6. Sistema de transmissão de dados com um equalizador de canais

*Fonte: Autor*

A Figura 2.7 mostra o diagrama de blocos de um sistema de transmissão de dados equipado com um equalizador adaptativo. Note que este equalizador necessita de uma sequência inicial de treinamento (idealmente, uma réplica atrasada dos símbolos transmitidos pelo canal) para a adaptação dos coeficientes do equalizador. Então, antes da operação do equalizador, é necessário que haja um período de inicialização, durante o qual o emissor envia uma sequência de símbolos de treinamento que já são conhecidos pelo receptor. Este período é chamado de modo de treino. Os símbolos de treinamento normalmente são especificados pelos fabricantes dos dispositivos de envio e recepção de dados como parte de padrões pré-estabelecidos para os sistemas de comunicação. O objetivo desta padronização é garantir que dispositivos de diferentes fabricantes possam se comunicar sem perda de qualidade (Farhang-Boroujeny, 2013).

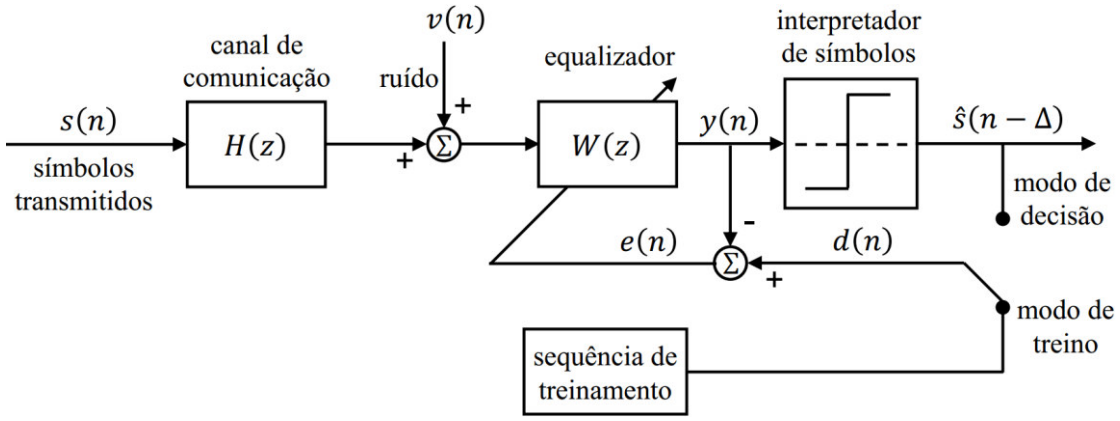



Figura 2.7. Sistema de transmissão de dados com equalizador adaptativo

Ao final do modo de treino, espera-se que os coeficientes do equalizador tenham convergido para valores próximos dos ótimos. Assim, os símbolos detectados pelo interpretador podem ser usados como o sinal desejado para o restante do processo de adaptação. Este segundo modo de operação é chamado de modo orientado por decisão ou simplesmente modo de decisão. Tal modo funciona satisfatoriamente enquanto as variações no canal forem suficientemente pequenas de forma que o algoritmo adaptativo seja capaz de rastreá-las. Teoricamente, os símbolos gerados pelo emissor seriam a melhor escolha para o sinal desejado, $d(n)$, do equalizador adaptativo. Porém, na prática, o este último fica localizado juntamente com o receptor, ou seja, há uma separação física entre o equalizador e a sua resposta ideal desejada. Daí a importância da sequência padronizada de treinamento para ajustar os

coeficientes do equalizador adaptativo e, consequentemente, garantir que o modo de decisão opere satisfatoriamente (Haykin, 1996).

### *2.1.4 Controle Ativo de Ruído*

Quando uma determinada situação requer a eliminação de um ruído presente em um dado sinal, o qual é constituído por um sinal desejado e por este ruído, diz-se que o problema em questão é de cancelamento de interferência. Em um problema geral de cancelamento de interferência, o qual é mostrado no diagrama da Figura 2.8, tem-se acesso a um dado sinal, $d(n)$, chamado de entrada primária e que é composto pela mistura de dois outros sinais. O primeiro é o sinal desejado, $y(n)$, enquanto que o segundo é a interferência, $v(n)$, que foi adicionada a $y(n)$. Deseja-se separar $v(n)$ de $y(n)$ ou, em outras palavras, eliminar $v(n)$. Embora não se tenha conhecimento completo a respeito dos sinais $v(n)$ e $y(n)$, normalmente dispõe-se de alguma informação a respeito de $v(n)$ na forma de um dado sinal ou entrada de referência, $x(n)$. Então, um filtro adaptativo, $\mathbf{w}(n)$, pode ser usado para produzir, a partir de $x(n)$, uma estimativa, $\hat{v}(n)$, da interferência, $v(n)$. Esta estimativa é então subtraída da entrada primária, $d(n)$, para obter um sinal de erro, $e(n)$. Este, por sua vez, é utilizado pelo algoritmo adaptativo para ajustar iterativamente os coeficientes do filtro adaptativo, $\mathbf{w}(n)$. O objetivo destes ajustes é aproximar $\hat{v}(n)$ de $v(n)$. Quando a estimativa, $\hat{v}(n)$, for uma réplica da interferência, $v(n)$, o sinal de erro, $e(n)$, será igual ao sinal desejado, $y(n)$. Por isso o nome cancelamento de interferência (Nascimento & Silva, 2014).

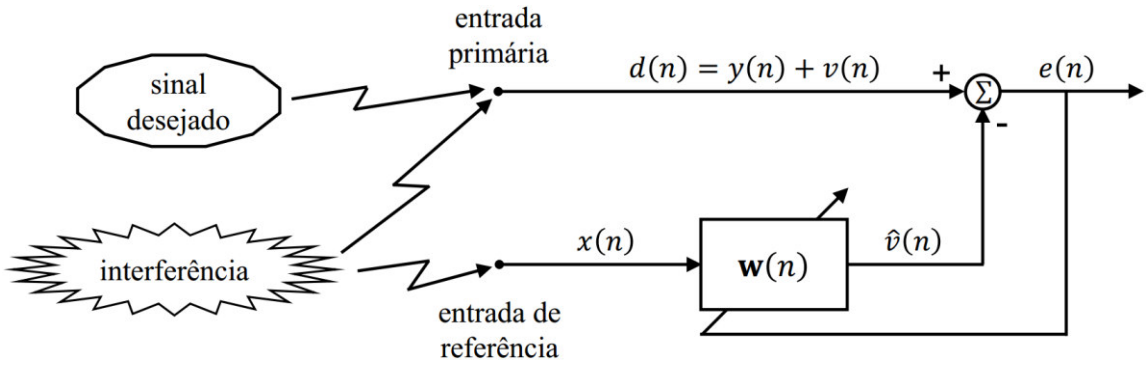



Figura 2.8. Diagrama de um problema de cancelamento de interferência

Um exemplo clássico de cancelamento de interferência pode ser encontrado em um exame clínico neurofisiológico chamado de eletroencefalograma (EEG). Este exame avalia o funcionamento do cérebro do paciente a partir da análise da atividade elétrica cerebral espontânea do mesmo, e é indicado para diagnosticar os mais variados transtornos neurológicos. A Figura 2.9 mostra um diagrama simplificado de um EEG onde a própria fonte de alimentação do aparelho que realiza o exame contamina o sinal coletado com um ruído de frequência 60Hz. Como este ruído exerce influência sob o paciente, o sinal aleatório do EEG é coletado pela entrada primária juntamente com a interferência, $v(n)$, causada pela fonte de alimentação. Para cancelar desta interferência, toma-se amostras do sinal da fonte de ruído como a entrada de

referência, $x(n)$, de um filtro adaptativo, $\mathbf{w}(n)$. Este filtro, por sua vez, usa esta entrada e o sinal de erro, $e(n)$, para ajustar seus coeficientes de forma que a sua saída, $\hat{v}(n)$, seja uma réplica do ruído, $v(n)$, que contamina o sinal, $d(n)$, colhido pelo eletrocardiograma. Então, $\hat{v}(n)$ é subtraído de $d(n)$ para que se possa isolar o sinal desejado, $y(n)$, a partir do sinal restaurado, $e(n)$. Em outras palavras, se $\hat{v}(n)$ for uma réplica de $v(n)$, tem-se que $e(n)$ é igual a $y(n)$ (Hagan, Demuth, Beale & De Jesús, 2015). Um sistema similar é usado para eliminar a interferência presente em sinais de eletrocardiogramas (Widrow & Walach, 2007).

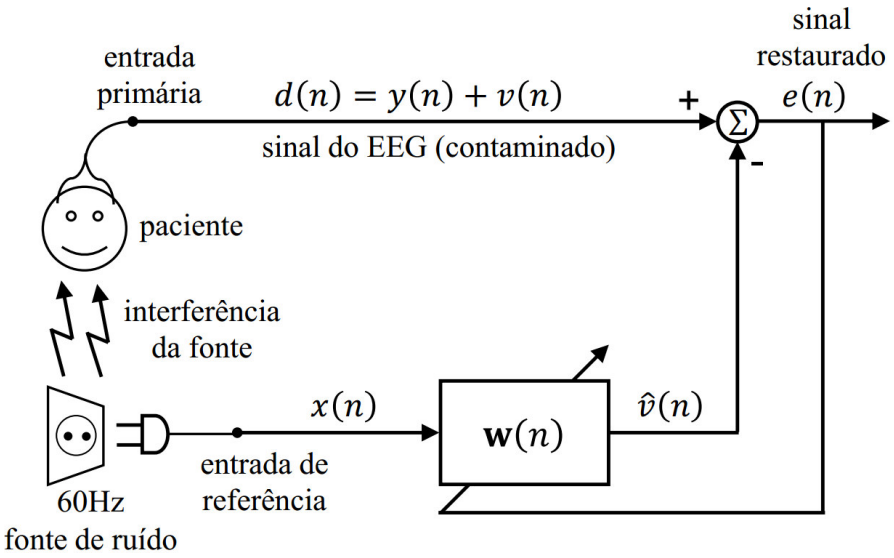



Figura 2.9. Filtro adaptativo aplicado no cancelamento da interferência causada por uma fonte de alimentação de 60Hz em um sinal de EEG.

Outra aplicação de filtros adaptativos envolvendo cancelamento de interferência é o chamado de controle ativo de ruído. Em particular, ruídos de natureza acústica podem ocorrer em vários ambientes de natureza industrial. Nestes ambientes, geralmente estão presentes máquinas rotativas de grande porte tais como exaustores, moinhos, correias transportadoras e compressores de ar, além de dutos exaustão de gases e sistemas de ventilação. Tais equipamentos estão entre as principais fontes de ruído do setor industrial primário. Como exemplo de aplicação de controle ativo de ruído neste tipo de ambiente, a Figura 2.10 mostra o diagrama simplificado de um sistema de controle ativo de ruído em um duto de exaustão de gases. A fonte de ruído (possivelmente uma máquina rotativa) gera um ruído primário que é captado pelo microfone de referência. Este microfone funciona como entrada de referência, $x(n)$, para o controlador ativo de ruído, o qual também usa dados do ruído secundário, $e(n)$, captado pelo microfone de erro para produzir uma saída, $y(n)$. Esta saída é então convertida em uma onda acústica pelo autofalante de cancelamento. Tal onda é usada para cancelar o ruído primário que se propaga pelo duto.

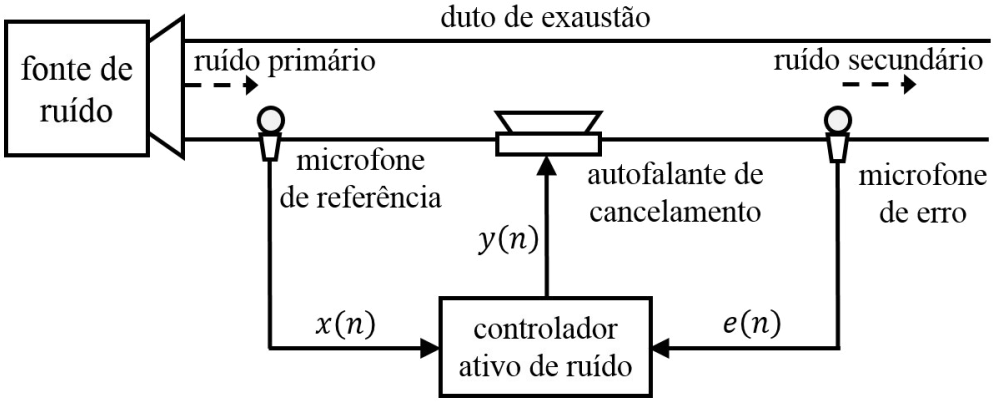



Figura 2.10. Sistema de controle ativo de ruído em um duto de exaustão.

*Fonte: Autor*

Um algoritmo adaptativo amplamente empregado no controle ativo de ruído é o LMS de entrada filtrada (FXLMS – *filtered-x least-mean-square*). Proposto por Dennis R. Morgan em 1980, o algoritmo FXLMS recebeu este nome pelo fato de o sinal de referência normalmente ser descrito pela letra $x$ (Morgan, 1980) e (Widrow & Stearns, 1985), embora tal terminologia não fosse usada pelo autor no momento da concepção do algoritmo FXLMS (Morgan, 2013). Outras áreas de aplicação de controle ativo de ruído a partir do algoritmo FXLMS incluem engenharias automotiva e aeronáutica, ruído de transformadores de potência e de geradores de grande porte, dentre outras (Nascimento & Silva, 2014).

Muitas aplicações práticas de filtros adaptativos, juntamente com os algoritmos adaptativos desenvolvidos para estas aplicações, usam uma estrutura de filtro de resposta ao impulso finita (FIR – *finite impulse response*) chamada de filtro adaptativo transversal (Widrow & Walach, 2007). Esta estrutura de filtro é discutida a seguir em detalhes.

## 2.2 Filtro Adaptativo Transversal

Considere um sistema dinâmico com $N$ entradas, $x_i(n)$, com $i = 1,2, \dots, N$, e uma saída, $d(n)$, mostrado na Figura 2.11(a), para o qual deseja-se determinar sua caracterização matemática. Este sistema pode ser modelado na forma de um Combinador Linear Adaptativo (CLA), o qual é mostrado na Figura 2.11(b). Neste caso, o CLA possui um algoritmo adaptativo que usa o erro, $e(n)$, entre sua a saída, $y(n)$, e a saída do sistema dinâmico, $d(n)$, para ajustar, em cada instante de tempo, $n$, os coeficientes, $w_i(n)$, com $i = 1,2, \dots, N$, com o objetivo de minimizar a diferença entre $y(n)$ e $d(n)$ e, consequentemente, assimilar o comportamento do sistema dinâmico desconhecido. Note da Figura 2.11(b) que a saída do CLA, $y(n)$, é igual à soma das $N$ entradas, $x_i(n)$, com $i = 1,2, \dots, N$, ponderadas pelos seus respectivos coeficientes, $w_i(n)$, com $i = 1,2, \dots, N$, ou seja

$$y(n) = \sum_{i=1}^{N} x_i(n) w_i(n)$$

$$= \mathbf{u}^{\mathrm{T}}(n)\mathbf{w}(n) \tag{2.5}$$

onde

$$\mathbf{u}(n) = [x_1(n)\ x_2(n)\ \ldots x_N(n)]^{\mathrm{T}} \tag{2.6}$$

e

$$\mathbf{w}(n) = [w_1(n)\ w_2(n)\ \ldots w_N(n)]^{\mathrm{T}} \tag{2.7}$$

são os vetores de entrada e de coeficientes do CLA, respectivamente. O sobrescrito T denota transposta de um vetor ou matriz.

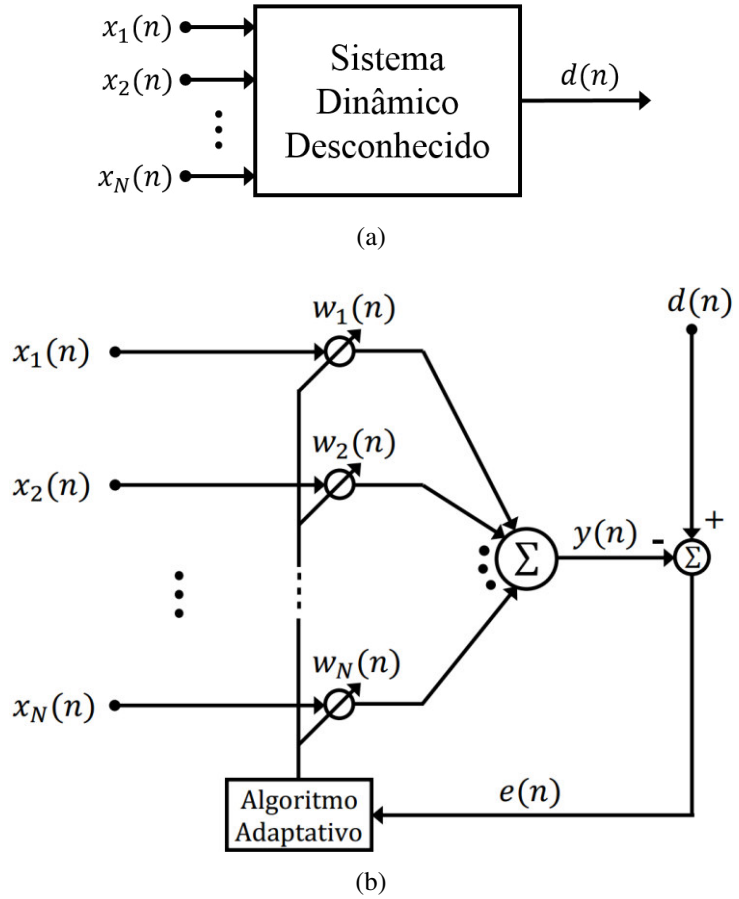



Figura 2.11. Modelagem de um sistema dinâmico de caracterização matemática desconhecida. (a) Diagrama de blocos do sistema dinâmico. (b) Modelagem do sistema dinâmico a partir de um CLA.

*Fonte: Autor*

A operação do CLA consiste na execução contínua e sequencial de duas etapas. São elas:

- Filtragem: a partir de (2.5), computa-se a saída do CLA, $y(n)$, e calcula-se o sinal de erro, $e(n)$, dado por

$$e(n) = d(n) - y(n). \tag{2.8}$$

- Adaptação: o algoritmo adaptativo é usado para atualizar do vetor de coeficientes do CLA, $\mathbf{w}(n)$. Uma nova estimativa deste vetor, denotada por $\mathbf{w}(n+1)$, é então usada

para computar a nova saída, $y(n+1)$, e calcular o novo valor do erro, $e(n+1)$, correspondentes à etapa de filtragem da iteração seguinte, $n+1$.

Note que, exceto durante a etapa de adaptação, a saída do CLA é uma combinação linear de suas entradas. Porém, enquanto os coeficientes do CLA estão sendo atualizados, no intervalo entre uma iteração e outra, tais coeficientes tornam-se dependentes dos dados de entrada e saída do sistema dinâmico e dos próprios parâmetros do CLA. Isto é uma consequência direta da aplicação do algoritmo adaptativo e confere uma característica não-linear ao CLA, uma vez que, durante a etapa de adaptação, o princípio da superposição não é atendido (Haykin, 1996).

Um sistema adaptativo de estrutura semelhante à do CLA é o filtro adaptativo transversal. O diagrama de blocos deste sistema é mostrado na Figura 2.12. Tal filtro possui apenas uma entrada, $x(n)$, e uma saída, $y(n)$. A variável $d(n)$ é a saída ou o sinal desejado. A saída do filtro adaptativo transversal, $y(n)$, é dada pela combinação linear das amostras atrasadas da entrada, $x(n)$, ou seja

$$y(n) = \sum_{i=1}^{N} x(n-i+1)w_i(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n) \tag{2.9}$$

onde

$$\mathbf{x}(n) = [x(n)\ x(n-1)\ x(n-2)\ \ldots x(n-N+1)]^{\mathrm{T}} \tag{2.10}$$

é o vetor das amostras atrasadas da entrada ou, simplesmente, vetor de entrada. A variável $N$ representa a ordem do filtro.

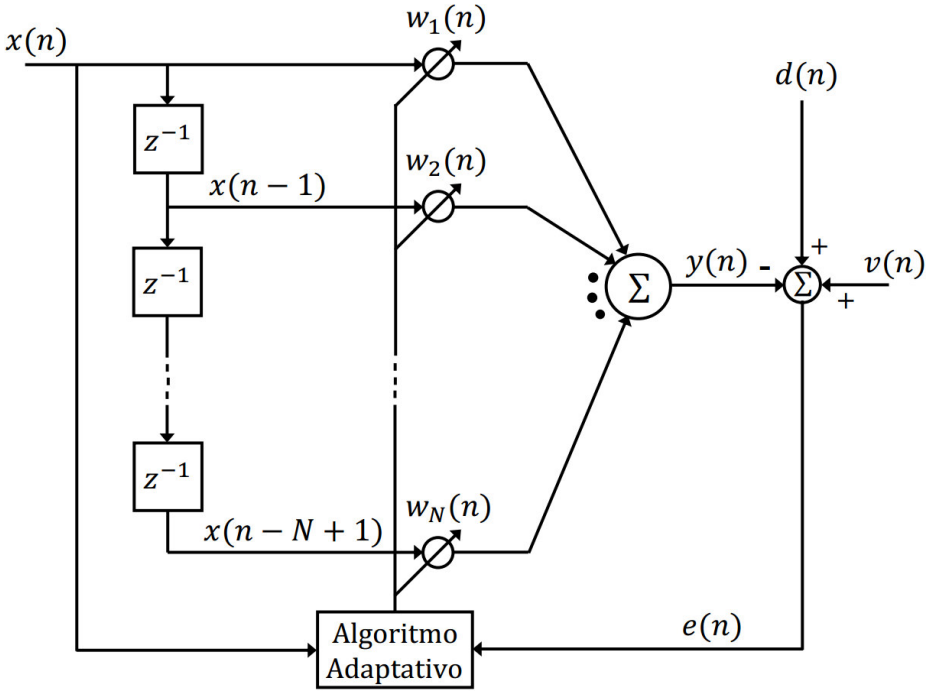



Figura 2.12. Diagrama de blocos de um filtro adaptativo transversal.

O filtro adaptativo transversal é um filtro FIR, ou seja, é um sistema de tempo discreto que possui, em um dado instante de tempo, $n$, a seguinte função de transferência

$$W(z) = w_1(n) + w_2(n)z^{-1} + \cdots + w_N(n)z^{-N}. \quad (2.11)$$

Os filtros FIR possuem a característica de serem sempre estáveis, uma vez que os pólos destes filtros estão todos localizados na origem do plano $z$. Além disso, tais filtros podem ser projetados para estimar com precisão respostas impulsivas arbitrárias se a ordem do filtro for suficientemente grande. Em contrapartida, quanto maior a ordem requerida para o filtro FIR, maiores serão os requisitos de memória e o esforço computacional necessário (Schilling & Harris, 2011). Isto pode prejudicar o desempenho do filtro em termos de velocidade de convergência e capacidade de rastreamento (Homer, Mareels & Hoang, 2006).

O filtro adaptativo transversal é a estrutura mais comumente usada em implementação de filtros adaptativos. Isto deve-se à simplicidade das operações que são efetuadas neste dispositivo, que são basicamente adições e multiplicações (Farhang-Boroujeny, 2013). Todos os algoritmos adaptativos abordados neste trabalho consideram a estrutura de um filtro adaptativo transversal.

A seguir, são apresentados alguns conceitos da base teórica necessária para o desenvolvimento dos algoritmos adaptativos LMS normalizados proporcionais, objeto de estudo deste trabalho. Primeiramente é feito um breve estudo a respeito de processos estocásticos estacionários. Logo após, o filtro de Wiener é apresentado. Posteriormente, os algoritmos SD, LMS e NLMS são derivados e suas principais características são discutidas. Por fim, são apresentadas duas medidas usualmente empregadas para avaliar o desempenho dos algoritmos adaptativos, a saber, o desalinhamento normalizado em dB e o erro quadrático médio.

## 2.3 Processos Estocásticos Estacionários

A maioria dos sinais envolvidos em problemas reais de engenharia não podem ser descritos a partir de uma análise matemática precisa. Em contrapartida, tais sinais podem ser caracterizados de forma probabilística, usando elementos de análise estatística, ou seja, podem ser descritos como processos estocásticos. O termo “processo estocástico” pode ser usado para definir a evolução temporal de um dado fenômeno para o qual não é possível descrever seu comportamento de maneira exata ou determinística. Um processo estocástico pode ser representado por uma função do tempo definida em um dado intervalo de observação. Tal função, teoricamente, deve ser capaz de representar infinitas e distintas realizações do mesmo processo (Haykin, 1996). Exemplos de processos estocásticos incluem sinais de voz, de entrada e saída de canais de comunicação, dados transmitidos por protocolos de redes industriais, dados de fenômenos sísmicos, dentre outros.

Um tipo de processo de particular interesse da área de filtragem adaptativa são os processos estocásticos de tempo discreto. A evolução temporal de tais processos é dada em instantes de tempo discretos e uniformemente espaçados. Um processo estocástico de tempo discreto, $\{u(n)\}$, pode ser representado por uma sequência

$$\{u(0), u(1), u(2), \dots\} \tag{2.12}$$

de variáveis aleatórias, $u(n)$, indexadas pelo índice temporal $n$, com $n = 0, 1, 2, \dots$ . Assume-se que cada variável aleatória é uma função contínua em qualquer instante de tempo, $n$. A sequência dada por (2.12) é denominada série de tempo discreto ou simplesmente série temporal. Esta série também pode ser escrita da forma

$$\{u(n), u(n-1), u(n-2), \dots, u(n-N+1)\} \tag{2.12}$$

contendo uma amostra de um dado sinal, $u(n)$, coletada no instante de tempo $n$, e outras $N-1$ amostras deste mesmo sinal coletadas nos instantes $n-1, n-2, \dots, n-N+1$. Tais amostras podem assumir valores reais ou complexos. Diz-se que uma série temporal é um tipo particular de realização de um dado processo estocástico (Haykin, 1996).

Um processo estocástico é dito estritamente estacionário se suas propriedades estatísticas são invariantes com o tempo. Em particular, para que um processo descrito por uma série temporal semelhante à dada por (2.12) seja estritamente estacionário, a função de densidade de probabilidade de cada uma das amostras colhidas nos instantes de $n$ a $n-N+1$ deve permanecer a mesma, independentemente dos valores de $n$ e $N$ (Farhang-Boroujeny, 2013). Porém, na prática, geralmente não é possível determinar a função densidade de probabilidade de uma série temporal arbitrária. Todavia, a partir das amostras de tal série, pode-se obter uma caracterização parcial do processo. Para tal, é necessário especificar os momentos de primeira e segunda ordem do processo. São eles, as funções de valor médio, de autocorrelação e de autocovariância (Manolakis, Ingle & Kogon, 2005).

Seja um processo estocástico de tempo discreto, $\{u(n)\}$, representado pela série temporal de (2.12). A função de valor médio ou, simplesmente, a média deste processo é dada por

$$\mu(n) = \mathrm{E}[u(n)] \tag{2.13}$$

onde $\mathrm{E}[\cdot]$ é o operador valor esperado. Já a função de autocorrelação de $\{u(n)\}$ é dada por

$$r(n, n-k) = \mathrm{E}[u(n)u^*(n-k)] \tag{2.14}$$

com $k = 0, \pm 1, \pm 2, \dots$ , e o sobrescrito $*$ denotando conjugação complexa. A função de autocovariância de $\{u(n)\}$ é dada por

$$c(n, n-k) = \mathrm{E}\big[\big(u(n) - \mu(n)\big)\big(u(n-k) - \mu(n-k)\big)^*\big] \tag{2.15}$$

com $k = 0, \pm 1, \pm 2, \dots$ . Note que as funções de valor médio, autocorrelação e autocovariância, dadas respectivamente por (2.13), (2.14) e (2.15), estão relacionadas da seguinte forma

$$c(n, n-k) = r(n, n-k) - \mu(n)\mu^*(n-k). \tag{2.16}$$

Assim, para uma caracterização parcial do processo, $\{u(n)\}$, faz-se necessário especificar a função de valor médio, $\mu(n)$, e a função de autocorrelação, $r(n, n-k)$ ou a função de autocovariância, $c(n, n-k)$, para vários valores de $n$ e $k$ que forem de interesse. A vantagem de tal caracterização é que a mesma pode ser feita a partir de amostras reais do

processo, além de ser adequada para situações nas quais são realizadas operações lineares com processos estocásticos (Haykin, 1996).

Caso o processo em questão for estritamente estacionário, as funções definidas nas equações de (2.13) a (2.15) assumem formas mais simples. Em particular, a função de valor médio torna-se uma constante, ou seja

$$\mu(n) = \mu, \qquad \forall\, n\,. \tag{2.17}$$

Já as funções de autocorrelação e autocovariância tornam-se dependentes apenas da diferença entre os instantes de observação $n$ e $n-k$, ou seja, do valor de $k$. Logo

$$r(n, n-k) = r(k) \tag{2.18}$$

e

$$c(n, n-k) = c(k) \tag{2.19}$$

onde, obviamente, a diferença $n-(n-k) = k$ indica o valor de atraso (*lag*) entre as amostras. Quando $k = 0$, tem-se que $r(0)$ é igual ao valor quadrático médio de $u(n)$

$$r(0) = E[|u(n)|^2] \tag{2.20}$$

e $c(k)$ é igual à variância de $u(n)$

$$c(0) = \sigma_u^2\,. \tag{2.21}$$

Note que as equações de (2.17) a (2.21) não são condições suficientes para definir se um processo estocástico de tempo discreto, $\{u(n)\}$, é estritamente estacionário ou não. No entanto, se $\{u(n)\}$ atende estas condições, o mesmo é dito estacionário de 2ª ordem ou estacionário no sentido geral (Manolakis, Ingle & Kogon, 2005). Um processo estritamente estacionário, $\{u(n)\}$, também será estacionário no sentido geral se, e somente se

$$E[|u(n)|^2] < -\infty, \qquad \forall\, n\,. \tag{2.22}$$

A condição dada por (2.22) é satisfeita pela maioria dos processos estocásticos encontrados em problemas de engenharia (Haykin, 1996). Uma série temporal dada por (2.12) também pode ser representada pelo vetor de observação

$$\mathbf{u}(n) = [u(n)\ \ u(n-1)\ \ \ldots\ \ u(n-N+1)]^{\mathrm{T}} \tag{2.23}$$

correspondente às amostras de uma realização de $\{u(n)\}$. A partir deste vetor, define-se a matriz de autocorrelação, **R**, de um processo estocástico estacionário como sendo a expectativa matemática do produto externo do vetor de observação por ele mesmo, ou seja

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}(n)^{\mathrm{H}}] \tag{2.24}$$

onde o sobrescrito H denota hermitiano, ou transposto e conjugado complexo, de um vetor ou matriz. Caso os elementos do vetor $\mathbf{u}(n)$ forem reais, a matriz **R** pode ser reescrita da seguinte forma

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}(n)^{\mathrm{T}}] \tag{2.25}$$

que, na forma expandida, é dada por

$$\mathbf{R} = \begin{bmatrix} E[u(n)u(n)] & \cdots & E[u(n)u(n-N+1)] \\ E[u(n-1)u(n)] & \cdots & E[u(n-1)u(n-N+1)] \\ \vdots & \ddots & \vdots \\ E[u(n-N+1)u(n)] & \cdots & E[u(n-N+1)u(n-N+1)] \end{bmatrix}. \quad (2.26)$$

Assumindo que $\{u(n)\}$ seja estritamente estacionário, a matriz de (2.26) pode ser reescrita da forma

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \cdots & r(N-1) \\ r(-1) & r(0) & \cdots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-N+1) & r(-N+2) & \cdots & r(0) \end{bmatrix} \quad (2.27)$$

onde $r(k)$, com $k = -N+1, \ldots, -1, 0, 1, \ldots, N-1$, é a autocorrelação de (2.18) para um *lag* igual a $k$. A matriz **R** possui um papel fundamental no projeto de filtros de tempo discreto. É importante notar que, para um processo estocástico estacionário, a matriz **R** possui algumas propriedades que facilitam a análise matemática, tais como ser simétrica, toeplitz, positiva semidefinida, dentre outras (Haykin, 1996).

Na prática, a estimação das médias (ou funções) estocásticas dadas pelas equações de (2.13) a (2.15) não é viável, uma vez que seriam necessários dados de várias realizações (ou várias séries temporais) de um mesmo processo, mesmo se este último for estacionário. Como na prática dispõe-se de apenas uma única realização do processo, uma alternativa viável é o uso de médias temporais para aproximar médias estocásticas de interesse. A característica de médias temporais serem capazes de igualar médias estocásticas, para um dado processo estocástico, é chamada de ergodicidade. Em filtragem adaptativa linear é sempre assumido que os processos envolvidos são ergódicos no sentido restrito, ou seja, todas as funções estocásticas são obtidas a partir de médias temporais (Farhang-Boroujeny, 2013). A seguir é apresentado um tipo de filtro linear de desempenho satisfatório para ambientes estacionários chamado de filtro de Wiener.

## 2.4 Filtro de Wiener

O filtro de Wiener é parte integrante da classe dos filtros lineares ótimos (Haykin, 1996). Os filtros adaptativos são derivados da teoria de filtragem de Wiener. Assim, alguns conceitos envolvendo filtro de Wiener são essenciais para entender do funcionamento dos filtros adaptativos (Farhang-Boroujeny, 2013). Conforme a teoria de Wiener, os coeficientes ótimos de um filtro linear podem ser obtidos a partir da minimização de uma dada função de desempenho chamada de erro quadrático médio (MSE – *mean square error*). O cálculo de tal função requer o conhecimento de certas funções estatísticas que não são possíveis de obter em aplicações práticas. Este problema pode ser resolvido usando ergodicidade. O comportamento do MSE para o filtro de Wiener é apresentado a seguir em detalhes.

Considere o esquema de filtragem da Figura 2.13 onde um filtro linear de tempo discreto, $H(z)$, é usado para estimar um sinal desejado, $d(n)$, a partir do processamento de uma dada entrada, $x(n)$. Este filtro produz uma saída, $y(n)$, a qual é subtraída do sinal desejado,

$d(n)$, gerando um erro de estimação, $e(n)$. Obviamente, quanto mais próximo de zero o erro de estimação, melhor o desempenho do filtro. Então, uma abordagem adequada para a determinação dos parâmetros do filtro deve se basear na otimização (ou minimização) de uma função de custo que possua relação com o erro de estimação. Esta função de custo pode também ser chamada de superfície de desempenho ou função de desempenho (Farhang-Boroujeny, 2013). Tal função deve ser matematicamente rastreável, de forma que a análise do filtro e o desenvolvimento de algoritmos iterativos para ajustar os parâmetros do filtro sejam possíveis. Além disso, a função de desempenho também deve possuir, preferencialmente, um único ponto ótimo (ou de mínimo), para que os parâmetros ótimos do filtro possam ser determinados de forma única, ou seja, sem ambiguidades.

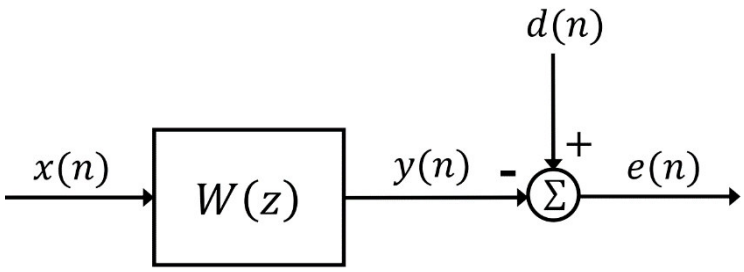



Figura 2.13. Filtro linear de tempo discreto empregado em um problema de estimação.

Devido à sua estrutura, os filtros FIR apresentam um único ponto ótimo (ou mínimo) global caso a função de desempenho seja escolhida apropriadamente. Para o filtro de Wiener, toma-se o MSE dado por

$$\xi = \mathrm{E}[|e(n)|^2] \tag{2.28}$$

onde $\mathrm{E}[\cdot]$ é o operador valor esperado. A função de desempenho de (2.28) possui uma estrutura simples, além de satisfazer os requisitos de rastreamento e de único ótimo global. Em particular, para um filtro FIR, $\xi$ é um hiperparabolóide com apenas um ponto de mínimo. Para verificar esta afirmação, considere o filtro transversal mostrado na Figura 2.14. Tal filtro é aplicado em um problema de estimação de uma saída desejada, $d(n)$, a partir do processamento de uma entrada, $x(n)$. O vetor de coeficientes deste filtro é definido como sendo

$$\mathbf{w} = [w_1\ w_2\ \ldots w_N]^{\mathrm{T}} \tag{2.29}$$

e o sinal de entrada, em um dado instante de tempo, $n$, é dado por

$$\mathbf{x}(n) = [x(n)\ x(n-1)\ \ldots x(n-N+1)]^{\mathrm{T}} \tag{2.30}$$

onde o sobrescrito T denota transposta de um vetor ou matriz e $N$ é a ordem do filtro. O sinal desejado, $d(n)$, e a entrada, $x(n)$, são assumidos como sendo processos estocásticos estacionários. A saída, $y(n)$, deste filtro é dada por

$$y(n) = \sum_{i=1}^{N} w_i x(n-i+1) \tag{2.31}$$

a qual também pode ser reescrita da forma vetorial

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

$$= \mathbf{w}^{\mathrm{T}}\mathbf{x}. \tag{2.32}$$

O sinal de erro é dado por

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= d(n) - \mathbf{w}^{\mathrm{T}}\mathbf{x} \\ &= d(n) - \mathbf{x}^{\mathrm{T}}\mathbf{w}\,. \end{aligned} \tag{2.33}$$

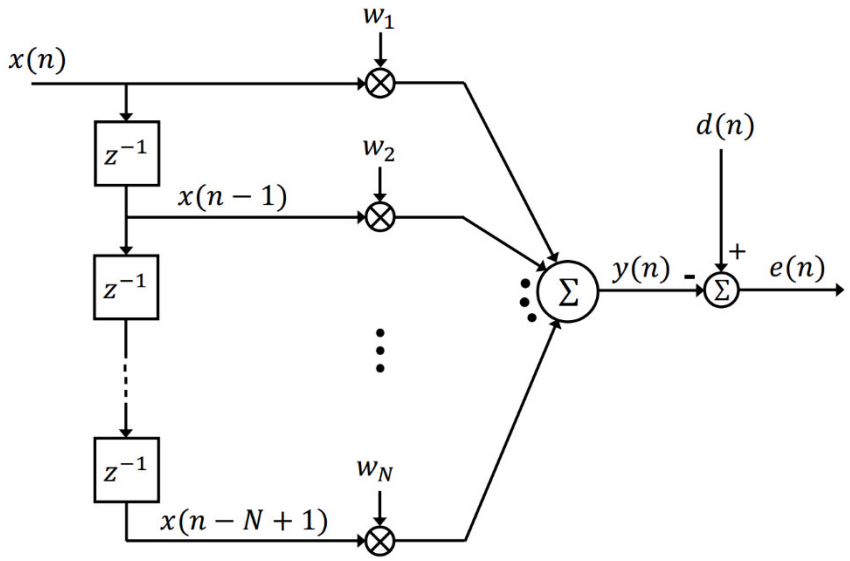



Figura 2.14. Filtro transversal.

Substituindo (2.33) em 2.28, obtém-se

$$\xi = \mathrm{E}[|e^2(n)|] = \mathrm{E}[(d(n) - \mathbf{w}^{\mathrm{T}}\mathbf{x}(n))(d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w})]. \tag{2.34}$$

Expandindo o lado direito de (2.34), tem-se que

$$\xi = \mathrm{E}[d^2(n)] - \mathbf{w}^{\mathrm{T}}\mathrm{E}[\mathbf{x}(n)d(n)] - \mathrm{E}[d(n)\mathbf{x}^{\mathrm{T}}(n)]\mathbf{w} + \mathbf{w}^{\mathrm{T}}\mathrm{E}[\mathbf{x}(n)\mathbf{x}^{\mathrm{T}}(n)]\mathbf{w} \tag{2.35}$$

uma vez que $\mathbf{w}$ não é uma variável estatística. Agora, considerando (2.25) e definindo o vetor de correlação cruzada dado por

$$\begin{aligned} \mathbf{q} &= \mathrm{E}[\mathbf{x}(n)d(n)] \\ &= [q_1\ q_2\ \ldots q_N] \end{aligned} \tag{2.36}$$

tem-se que (2.35) pode ser reescrita como sendo

$$\xi = \mathrm{E}[d^2(n)] - 2\mathbf{w}^{\mathrm{T}}\mathbf{q} + \mathbf{w}^{\mathrm{T}}\mathbf{R}\mathbf{w} \tag{2.37}$$

já que $\mathbf{w}^{\mathrm{T}}\mathbf{q} = \mathbf{q}^{\mathrm{T}}\mathbf{w}$ e $E[d(n)\mathbf{x}^{\mathrm{T}}(n)] = \mathbf{q}^{\mathrm{T}}$. Note que, se $\mathbf{R}$ é positiva definida, o lado direito de (2.37) corresponde a uma curva convexa com apenas um ponto de mínimo. Em outras palavras, se $x(n)$ e $d(n)$ são processos estocásticos estacionários, (2.37) é uma função quadrática do vetor de coeficientes do filtro, $\mathbf{w}$, com apenas um mínimo global. Para determinar tal ponto, é

necessário resolver o sistema de equações das derivadas parciais de $\xi$ em relação a cada elemento de $\mathbf{w}$, ou seja

$$\frac{\partial \xi}{\partial w_i} = 0, \quad i = 1,2,\dots,N \tag{2.38}$$

que pode ser reescrita na forma vetorial

$$\nabla \xi = \mathbf{0} \tag{2.39}$$

sendo $\nabla$ o operador gradiente dado por

$$\nabla = \left[\frac{\partial}{\partial w_1} \;\; \frac{\partial}{\partial w_2} \;\; \dots \;\; \frac{\partial}{\partial w_N}\right]^{\mathrm{T}} \tag{2.40}$$

e $\mathbf{0}$ é o vetor nulo de ordem $N$. Considerando (2.18), (2.36) e (2.37), e desenvolvendo o lado esquerdo de (2.38), obtém-se

$$\frac{\partial \xi}{\partial w_i} = 2\sum_{l=1}^{N} r(i-l)w_l - 2q_i\,, \qquad i = 1,2,\dots,N \tag{2.41}$$

que pode ser reescrita na forma matricial

$$\nabla \xi = 2\mathbf{R}\mathbf{w} - 2\mathbf{q}\,. \tag{2.42}$$

Fazendo $\nabla \xi = \mathbf{0}$, tem-se a equação para determinar os coeficientes ótimos do filtro de Wiener dada por

$$\mathbf{R}\mathbf{w}_{\mathrm{opt}} = \mathbf{q} \tag{2.43}$$

onde o subscrito "opt" é usado para enfatizar que $\mathbf{w}_{\mathrm{opt}}$ é o vetor dos coeficientes ótimos do filtro de Wiener. A equação (2.43) é conhecida como equação de Wiener-Hopf e possui a seguinte solução

$$\mathbf{w}_{\mathrm{opt}} = \mathbf{R}^{-1}\mathbf{q} \tag{2.44}$$

considerando que $\mathbf{R}$ possui inversa.

Substituindo (2.43) e $\mathbf{w}$ por $\mathbf{w}_{\mathrm{opt}}$ em (2.37), obtém-se o mínimo MSE que pode ser alcançado pelo filtro de Wiener dado por

$$\begin{aligned}\xi_{\min} &= \mathrm{E}[d^2(n)] - \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathbf{q} \\ &= \mathrm{E}[d^2(n)] - \mathbf{w}_{\mathrm{opt}}^{\mathrm{T}}\mathbf{R}\mathbf{w}_{\mathrm{opt}}\,.\end{aligned} \tag{2.45}$$

O resultado de (2.45) é obtido quando os coeficientes do filtro de Wiener são escolhidos de acordo com a solução de (2.43). Substituindo (2.43) em (2.45) obtém-se

$$\xi_{\min} = \mathrm{E}[d^2(n)] - \mathbf{q}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{q}\,. \tag{2.46}$$

Note de (2.46) que o valor mínimo para o MSE de um filtro de Wiener pode ser obtido apenas de posse das propriedades estatísticas dos sinais envolvidos.

Diz-se que o filtro da Figura 2.14 é um filtro de Wiener transversal se os coeficientes do mesmo são determinados a partir da minimização de (2.28). Note, da estrutura deste filtro,

que o mesmo é um filtro FIR. Com o intuito de avaliar o comportamento do MSE para este tipo de filtro, considere o problema de estimação da Figura 2.15 onde um filtro de Wiener transversal, $W(z)$, de ordem 2 é usado para identificar uma planta que também é um filtro de ordem 2 e cuja função de transferência é $P(z) = 1 + 2z^{-1}$. Considera-se que a entrada $x(n)$ é um processo estacionário branco de variância unitária. O ruído aditivo de medição, $v(n)$, é branco de variância $\sigma_v^2 = 0{,}1$ e não-correlacionado com $x(n)$. O objetivo deste problema é determinar os valores ótimos dos coeficientes do filtro de Wiener, $w_1$ e $w_2$, tal que (2.28) seja minimizada.

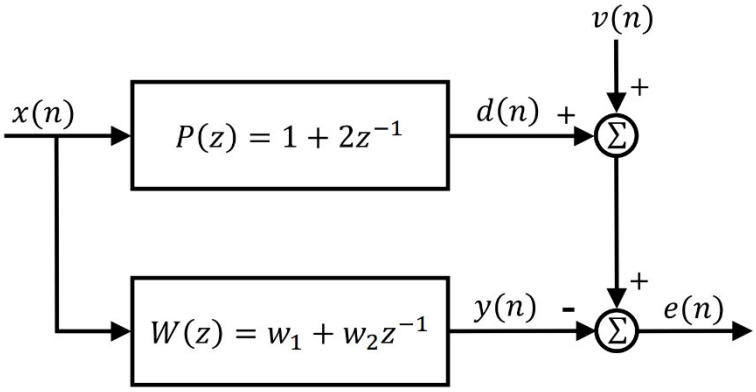



Figura 2.15. Filtro de Wiener transversal aplicado em um problema de identificação de sistemas.

Para determinar os valores ótimos de $w_1$ e $w_2$, é necessário calcular $\mathbf{R}$ e $\mathbf{q}$. Para tanto, considera-se (2.18), (2.20), (2.26), (2.27) e (2.36) para obter

$$\begin{aligned}\mathbf{R} &= \begin{bmatrix} E[x^2(n)] & E[x(n)x(n-1)] \\ E[x(n-1)x(n)] & E[x^2(n)] \end{bmatrix} \\ &= \begin{bmatrix} r(0) & r(1) \\ r(-1) & r(0) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{2.47}$$

uma vez que $x(n)$ é branco e possui variância unitária. Agora, para o cálculo de $\mathbf{q}$, tem-se

$$\begin{aligned}\mathbf{q} &= \begin{bmatrix} \mathrm{E}\big[x(n)\big(x(n) + 2x(n-1) + v(n)\big)\big] \\ \mathrm{E}\big[x(n-1)\big(x(n) + 2x(n-1) + v(n)\big)\big] \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{E}[x^2(n)] + 2\mathrm{E}[x(n)x(n-1)] + \mathrm{E}[x(n)v(n)] \\ \mathrm{E}[x(n)x(n-1)] + 2\mathrm{E}[x^2(n-1)] + \mathrm{E}[x(n-1)v(n)] \end{bmatrix}. \end{aligned} \tag{2.48}$$

Como $\mathrm{E}[x(n)x(n-1)] = \mathrm{E}[x(n)v(n)] = \mathrm{E}[x(n-1)v(n)] = 0$ e $\mathrm{E}[x^2(n)] = \mathrm{E}[x^2(n-1)] = 1$, tem-se que

$$\mathbf{q} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}. \tag{2.48}$$

Similarmente, para a primeira parcela de (2.46), obtém-se

$$\begin{aligned}\mathrm{E}\left[\left(d(n)+v(n)\right)^2\right] &= \mathrm{E}\left[\left(x(n)+2x(n-1)+v(n)\right)^2\right] \\ &= \mathrm{E}[x^2(n)] + 4E[x^2(n-1)] + \sigma_v^2 \\ &= 5{,}1 \end{aligned} \tag{2.49}$$

Então, sabendo que $\mathbf{w} = [w_1\ w_2]^{\mathrm{T}}$, substitui-se (2.47), (2.48) e (2.49) em (2.37) para obter

$$\begin{aligned}\xi &= 5{,}1 - 2[w_1\ w_2]\begin{bmatrix}1\\2\end{bmatrix} + [w_1\ w_2]\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}w_1\\w_2\end{bmatrix} \\ &= 5{,}1 - 2w_1 - 4w_2 + w_1^2 + w_2^2 \\ &= 0{,}1 + w_1^2 - 2w_1 + 1 + w_2^2 - 4w_2 + 4 \\ &= 0{,}1 + (w_1-1)^2 + (w_2-2)^2 \end{aligned} \tag{2.50}$$

que é um paraboloide com um ponto mínimo em $\mathbf{w}_{\min} = [1\ \ 2]^{\mathrm{T}}$, como mostra a Figura 2.16.

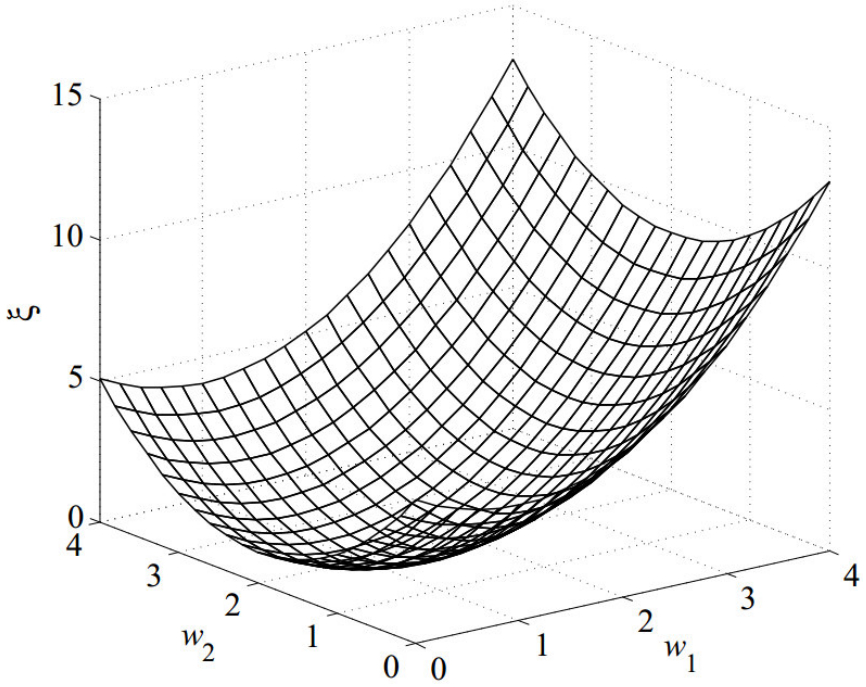



Figura 2.16. Superfície de desempenho para o problema de identificação de sistemas da Figura 2.15.

Substituindo (2.47) e (2.48) em (2.44), obtém-se os valores ótimos de $w_1$ e $w_2$

$$\begin{aligned}\mathbf{w}_{\mathrm{opt}} &= \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix} \\ &= \begin{bmatrix}1\\2\end{bmatrix} \end{aligned} \tag{2.51}$$

que correspondem ao ponto mínimo de (2.50). Agora, substitui-se (2.47), (2.48) e (2.49) em (2.46) para obter

$$\xi_{\min} = 5{,}1 - [1\ 2]\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix}$$
$$= 0{,}1\,. \tag{2.52}$$

que corresponde ao valor mínimo de (2.50), como mostra a Figura 2.16. Note do exemplo da Figura 2.15 que o projeto de um filtro de Wiener pode ser feito a partir da solução da equação de Wiener-Hopf dada por (2.44), uma vez que a estatística dos sinais envolvidos é conhecida. Esta solução pode ser obtida a partir da minimização do MSE dado por (2.28). Contudo, a solução de Wiener é adequada apenas para ambientes estacionários, ou seja, se as propriedades estatísticas forem variantes com o tempo a solução de (2.44) deixará de ser ótima. Um método alternativo para encontrar $\mathbf{w}_{\text{opt}}$ consiste no uso de um algoritmo iterativo de busca. Este algoritmo deve partir de um ponto inicial e mover-se progressivamente sobre a superfície da função de desempenho através de passos orientados. Caso a função de desempenho seja convexa, tal como a de um filtro transversal, este processo de busca iterativa certamente convergirá para a solução ótima. A seguir, um método iterativo baseado no gradiente de $\xi$ é apresentado. Este método realiza uma busca sobre a função de desempenho do filtro de Wiener transversal usando o algoritmo SD.

## 2.5 Algoritmo de Descida mais Íngreme

A determinação do vetor ótimo de coeficientes de um filtro a partir da minimização de uma dada função de custo usando um algoritmo iterativo constitui um conceito fundamental para o desenvolvimento de algoritmos adaptativos. Em particular, a ideia por trás do algoritmo SD é simples. Assumindo que a função de desempenho seja convexa, parte-se de um ponto inicial arbitrário e adota-se pequenos passos sucessivos em direção ao ponto mínimo da função. Apesar de a convergência ser garantida, este método pode apresentar convergência lenta para filtros de Wiener transversais.

Para analisar o desempenho do algoritmo SD, considere o filtro de Wiener transversal da Figura 2.14. O vetor de coeficientes ótimos, $\mathbf{w}_{\text{opt}}$, pode ser obtido a partir da minimização de

$$\xi = \mathrm{E}[e^2(n)] \tag{2.53}$$

uma vez que qualquer valor elevado ao quadrado é sempre um número positivo e $e(n)$ é o erro de estimação do filtro dado por

$$e(n) = d(n) - y(n). \tag{2.54}$$

Sabe-se da seção anterior que $\xi$ pode ser expandida da forma

$$\xi = \mathrm{E}[d^2(n)] - 2\mathbf{w}^{\mathrm{T}}\mathbf{q} + \mathbf{w}^{\mathrm{T}}\mathbf{R}\mathbf{w} \tag{2.55}$$

sendo $\mathbf{R}$ a matriz de autocorrelação da entrada do filtro, $x(n)$, e $\mathbf{q}$ é o vetor de correlação cruzada entre a entrada e a saída desejada, $d(n)$. A função de desempenho de (2.55) é uma função quadrática de $\mathbf{w}$ com um único mínimo global que pode ser obtido pela solução da equação de Wiener-Hopf dada por (2.44), se $\mathbf{R}$ e $\mathbf{q}$ são conhecidos. O algoritmo SD pode ser

empregado partindo-se de uma estimativa inicial de $\mathbf{w}_{\text{opt}}$ e adotando-se o procedimento a seguir.

1. Determine o gradiente da função de desempenho em relação aos coeficientes do filtro no ponto em questão;
2. Atualize a estimativa dos coeficientes do filtro tomando um passo na direção oposta do vetor gradiente obtido no passo anterior;
3. Repita os passos 1 e 2 até que são seja mais observada uma mudança significativa nos valores dos coeficientes do filtro.

A direção tomada no passo 2 corresponde à de descida mais íngreme da função de desempenho no ponto considerado. Para implementar tal procedimento, assumindo que $\mathbf{R}$ e $\mathbf{q}$ são conhecidos, usa-se o resultado de (2.42) dado por

$$\nabla\xi = 2\mathbf{R}\mathbf{w} - 2\mathbf{q} \tag{2.56}$$

onde $\nabla$ é o operador gradiente dado pelo vetor

$$\nabla = \left[\frac{\partial}{\partial w_1} \;\; \frac{\partial}{\partial w_2} \;\; \ldots \;\; \frac{\partial}{\partial w_N}\right]^{\mathrm{T}}. \tag{2.57}$$

Conforme o procedimento descrito acima, considere o vetor $\mathbf{w}(n)$, correspondente à $n$-ésima iteração. Este vetor pode ser atualizado usando a seguinte equação recursiva

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla_n\xi \tag{2.58}$$

onde $\mu$ é um escalar positivo chamado de parâmetro de passo e $\nabla_n\xi$ denota o vetor gradiente $\nabla\xi$ calculado na $n$-ésima iteração do algoritmo SD e no ponto $\mathbf{w}(n)$. Substituindo (2.56) em (2.58) obtém-se

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\mu[\mathbf{R}\mathbf{w}(n) - \mathbf{q}] \tag{2.59}$$

que pode ser rearranjada da seguinte forma

$$\mathbf{w}(n+1) = (\mathbf{I} - 2\mu\mathbf{R})\mathbf{w}(n) + 2\mu\mathbf{q} \tag{2.60}$$

onde $\mathbf{I}$ é a matriz identidade de ordem $N$. Assim, substituindo (2.43) e subtraindo $\mathbf{w}_{\text{opt}}$ nos dois lados de (2.60), tem-se

$$\begin{aligned}\mathbf{w}(n+1) - \mathbf{w}_{\text{opt}} &= (\mathbf{I} - 2\mu\mathbf{R})\mathbf{w}(n) + 2\mu\mathbf{R}\mathbf{w}_{\text{opt}} - \mathbf{w}_{\text{opt}} \\ &= (\mathbf{I} - 2\mu\mathbf{R})\big(\mathbf{w}(n) - \mathbf{w}_{\text{opt}}\big).\end{aligned} \tag{2.61}$$

Agora, definindo o vetor

$$\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}_{\text{opt}} \tag{2.62}$$

e substituindo (2.62) em (2.61), obtém-se

$$\mathbf{v}(n+1) = (\mathbf{I} - 2\mu\mathbf{R})\mathbf{v}(n). \tag{2.63}$$

O resultado de (2.63) pode ser simplificado aplicando a seguinte transformação de similaridade

$$\mathbf{R} = \mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{Q} \tag{2.64}$$

onde $\mathbf{\Lambda}$ é uma matriz diagonal composta pelos autovalores de $\mathbf{R}$, ou seja

$$\mathbf{\Lambda} = \mathrm{diag}[\lambda_1 \ \ \lambda_2 \ \ \ldots \ \ \lambda_N] \tag{2.65}$$

e $\mathbf{Q}$ é uma matriz cujas colunas são formadas pelos respectivos autovetores ortornormais de $\mathbf{R}$. Logo, substituindo (2.64) em (2.63) tem-se

$$\begin{aligned}\mathbf{v}(n+1) &= (\mathbf{Q}\mathbf{Q}^{\mathrm{T}} - 2\mu\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}})\mathbf{v}(n) \\ &= \mathbf{Q}(\mathbf{I} - 2\mu\mathbf{\Lambda})\mathbf{Q}^{\mathrm{T}}\mathbf{v}(n).\end{aligned} \tag{2.66}$$

Multiplicando à esquerda ambos os lados de (2.66) por $\mathbf{Q}^{\mathrm{T}}$ e aplicando a seguinte transformação

$$\mathbf{v}'(n) = \mathbf{Q}^{\mathrm{T}}\mathbf{v}(n) \tag{2.67}$$

obtém-se

$$\mathbf{v}'(n+1) = (\mathbf{I} - 2\mu\mathbf{\Lambda})\mathbf{v}'(n) \tag{2.68}$$

que é a equação recursiva de atualização dos coeficientes do filtro representada nos $N$ eixos $v'$, que equivalem a uma rotação dos $N$ eixos $v$ que, por sua vez, equivalem a um deslocamento da origem do espaço euclidiano $N$-dimensional definido pelos eixos $w_1, w_2, \ldots, w_N$. A equação vetorial (2.68) pode ser separada em $N$ equações escalares recursivas

$$v_i'(n+1) = (1 - 2\mu\lambda_i)v_i'(n)\,, \qquad i = 1,2,\ldots,N\,. \tag{2.69}$$

Então, partindo de um conjunto de valores iniciais, $v_1'(0), v_2'(0), \ldots, v_N'(0)$, tem-se, na primeira iteração ($n = 1$)

$$v_i'(1) = (1 - 2\mu\lambda_i)v_i'(0)\,, \qquad i = 1,2,\ldots,N \tag{2.70}$$

enquanto que , na segunda iteração ($n = 2$), tem-se

$$\begin{aligned}v_i'(2) &= (1 - 2\mu\lambda_i)v_i'(1) \\ &= (1 - 2\mu\lambda_i)(1 - 2\mu\lambda_i)v_i'(0) \\ &= (1 - 2\mu\lambda_i)^2 v_i'(0), \qquad i = 1,2,\ldots,N.\end{aligned} \tag{2.71}$$

Logo, na $n$-ésima iteração tem-se

$$v_i'(n) = (1 - 2\lambda_i)^n v_i'(n), \qquad i = 1,2,\ldots,N. \tag{2.72}$$

Sabe-se de (2.62) e (2.67) que $\mathbf{w}(n)$ converge para $\mathbf{w}_{\mathrm{opt}}(n)$ se, e somente se, $\mathbf{v}'(n)$ converge para um vetor nulo. Porém, note de (2.72) que $\mathbf{v}'(n)$ converge para zero somente se o parâmetro de passo, $\mu$, for escolhido tal que

$$|1 - 2\mu\lambda_i| < 1, \qquad i = 1,2,\ldots,N. \tag{2.73}$$

Caso (2.73) seja satisfeita, os escalares $v_i'(n)$, com $i = 1,2,\ldots,N$, decaem exponencialmente para zero de forma proporcional ao número de iterações, $n$. Portanto, quanto

maior o número de iterações, $n$, mais próximos de zero os escalares $v_i'(n)$, com $i = 1,2,\dots,N$, estão. A inequação (2.73) descreve a condição necessária para a convergência e estabilidade do algoritmo SD. Tal inequação pode ser expandida da seguinte forma

$$-1 < 1 - 2\mu\lambda_i < 1$$

ou

$$0 < \mu < \frac{1}{\lambda_i}, \qquad i = 1,2,\dots,N. \tag{2.74}$$

Como o parâmetro $\mu$ é o mesmo para todos os $N$ coeficientes do filtro, a convergência (ou estabilidade) do algoritmo SD é garantida somente se

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{2.75}$$

onde $\lambda_{\max}$ é o maior autovalor da matriz $\mathbf{R}$. Note de (2.58) e (2.75) que $\mu$ deve sempre ser um escalar positivo para garantir que a correção no vetor $\mathbf{w}(n)$ seja feita na direção oposta à indicada por $\nabla_n\xi$. Note também que o limite à direita de (2.75) garante o decaimento exponencial de (2.53) proporcionalmente à quantidade de iterações do algoritmo SD.

O comportamento transitório do algoritmo SD pode ser descrito em termos do vetor de coeficientes do filtro transversal, $\mathbf{w}(n)$. Note de (2.62) que

$$\begin{aligned}
\mathbf{w}(n) &= \mathbf{w}_{\text{opt}} + \mathbf{v}(n) \\
&= \mathbf{w}_{\text{opt}} + \mathbf{Q}\mathbf{v}'(n) \\
&= \mathbf{w}_{\text{opt}} + [\mathbf{r}_1\ \mathbf{r}_2\ \dots\ \mathbf{r}_N]\begin{bmatrix} v_1'(n) \\ v_2'(n) \\ \vdots \\ v_N'(n) \end{bmatrix} \\
&= \mathbf{w}_{\text{opt}} + \sum_{i=1}^{N} \mathbf{r}_i v_i'(n)
\end{aligned} \tag{2.76}$$

sendo $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N$ os autovetores associados aos autovalores $\lambda_1, \lambda_2, \dots, \lambda_N$ da matriz de autocorrelação, $\mathbf{R}$. Agora, substituindo (2.72) em (2.76), tem-se que

$$\mathbf{w}(n) = \mathbf{w}_{\text{opt}}(n) + \sum_{i=1}^{N} v_i'(0)(1 - 2\mu\lambda_i)^n \mathbf{r}_i\ . \tag{2.77}$$

O resultado de (2.77) mostra que o comportamento transitório do algoritmo SD para um filtro transversal de ordem $N$ é determinado por uma soma de $N$ termos exponenciais. Cada termo é controlado por um dos autovalores de $\mathbf{R}$. Cada autovalor, $\lambda_i$, determina um modo particular de convergência na direção definida pelo seu autovetor associado, $\mathbf{r}_i$. Cada modo, por sua vez, opera independentemente dos demais.

Note que o algoritmo SD não é um algoritmo adaptativo, pois depende unicamente dos momentos de segunda ordem $\mathbf{R}$ e $\mathbf{q}$. Além disso, o índice de iteração, $n$, não possui relação

alguma com o tempo. Resumidamente, o algoritmo SD proporciona apenas uma solução iterativa para o problema de filtragem ótima apresentado pelo filtro de Wiener e que possui como solução a equação de Wiener-Hopf de (2.43). Pode-se dizer que o algoritmo SD é uma versão idealizada dos algoritmos práticos (Manolakis, Ingle & Kogon, 2005). A exemplo de ilustração, a Figura 2.17 mostra as diferenças de operação entre (a) um filtro ótimo e (b) um filtro adaptativo. O vetor de coeficientes, $\mathbf{w}_{\text{opt}}$, de um filtro ótimo (tal como um filtro de Wiener) é constituído por valores determinísticos obtidos a partir da solução de (2.43). Já o filtro adaptativo é um sistema variante no tempo que realiza a busca pelo vetor ótimo de coeficientes, $\mathbf{w}_{\text{opt}}(n)$, a cada instante de tempo, $n$. Este tipo de filtro é adequado para ambientes não-estacionários, ao contrário dos filtros ótimos.

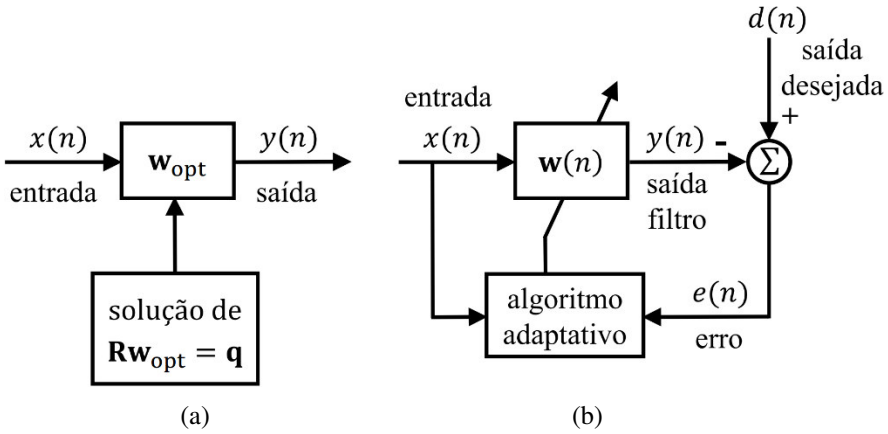



Figura 2.17. Diferença de operação entre (a) um filtro ótimo e (b) um filtro adaptativo.

A análise das equações de (2.58) a (2.77) mostra que o desempenho do algoritmo SD é altamente dependente dos autovalores da matriz **R**. Especificamente, a relação

$$\chi(\mathbf{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{2.78}$$

onde $\lambda_{\min}$ é o menor autovalor de **R**, é fundamental na determinação do desempenho de convergência do algoritmo SD. Tal relação é denominada dispersão ou número de condição (*eigenvalue spread*). Uma vez que $\lambda_{\max}$ e $\lambda_{\min}$ estão diretamente relacionados com a densidade espectral do respectivo sinal, pode-se afirmar que o desempenho do algoritmo SD possui relação direta com a densidade espectral do sinal de entrada do filtro. Quanto mais próximos estiverem os valores de $\lambda_{\max}$ e $\lambda_{\min}$, melhor será o desempenho do algoritmo SD (Manolakis, Ingle & Kogon, 2005) e (Farhang-Boroujeny, 2013).

A próxima seção apresenta um algoritmo adaptativo que foi derivado a partir do algoritmo SD, a saber, o algoritmo LMS. Este algoritmo preserva algumas semelhanças importantes com o algoritmo SD. Porém, a principal diferença entre estes dois algoritmos está no fato de que o algoritmo SD usa valores determinísticos enquanto que o algoritmo LMS usa valores aleatórios ou estocásticos. A derivação do algoritmo LMS a partir do algoritmo SD, bem como suas principais características são discutidas a seguir em detalhes.

## 2.6 Algoritmo LMS

O algoritmo LMS é o algoritmo adaptativo mais básico e o mais amplamente empregado em várias aplicações envolvendo filtragem adaptativa linear (Haykin, 1996) e (Manolakis, Ingle & Kogon, 2005). Tal algoritmo consiste de uma implementação estocástica do algoritmo SD, e foi introduzido por Bernard Widrow e Marcian Hoff em 1960 juntamente com a rede neural ADALINE (*ADAptive LInear NEuron*). O algoritmo LMS é uma técnica simples e computacionalmente eficaz de atualização do vetor de coeficientes de um filtro adaptativo, uma vez que usa, em sua regra de adaptação, uma estimativa do MSE baseada no erro quadrático instantâneo dado por (Widrow & Hoff, 1960)

$$\begin{aligned}\epsilon(n) &= e^2(n) \\ &= [d(n) - y(n) + v(n)]^2\end{aligned} \tag{2.79}$$

considerando um ruído aditivo de medição, $v(n)$. Devido à sua simplicidade e robustez a variações na estatística dos sinais envolvidos, o algoritmo LMS tem sido objeto de estudo de vários pesquisadores ao redor do mundo. Ao longo das últimas décadas, várias modificações e aprimoramentos deste algoritmo têm sido propostos. A seguir é apresentada a derivação do algoritmo LMS a partir do algoritmo SD.

### *2.6.1 Derivação do Algoritmo LMS*

Considere que medidas precisas do gradiente $\nabla\xi$ de (2.56) pudessem ser obtidas em cada iteração, $n$, e que o parâmetro de passo, $\mu$, pudesse ser escolhido corretamente de acordo com os autovalores de $\mathbf{R}$. Neste caso, o algoritmo SD poderia ser perfeitamente usado para encontrar o vetor $\mathbf{w}_{\text{opt}}$. Porém, na prática, não se dispõe de um conhecimento a priori de $\mathbf{R}$ e $\mathbf{q}$, o que torna impossível obter medidas exatas de $\nabla\xi$. Uma alternativa viável é o uso de um algoritmo iterativo que dispensa o conhecimento estatístico dos sinais envolvidos, tal como o algoritmo LMS. Este algoritmo é um membro importante da família dos algoritmos de gradiente estocástico. O termo "gradiente estocástico" é usado para distinguir o algoritmo LMS do algoritmo SD, uma vez que o primeiro usa um gradiente estocástico dado por (2.79) ao passo que o último usa um gradiente determinístico dado por (2.53) na busca iterativa pelos coeficientes ótimos do filtro linear. Com base nesta informação, considere o filtro adaptativo transversal de ordem $N$ mostrado na Figura 2.18. A saída deste filtro é dada por

$$\begin{aligned}y(n) &= \sum_{i=1}^{N} x(n-i+1)w_i(n) \\ &= \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)\end{aligned} \tag{2.80}$$

onde os coeficientes do filtro são ajustados iterativamente de forma a minimizar erro de estimação dado por

$$e(n) = d(n) - y(n) + v(n) \tag{2.81}$$

sendo $v(n)$ um ruído de medição.

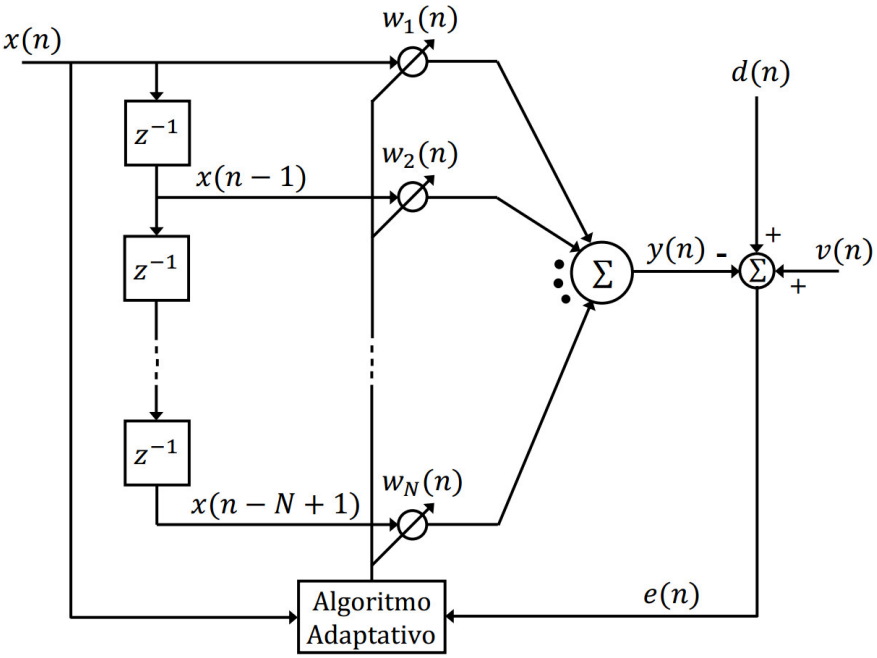



Figura 2.18. Filtro adaptativo transversal.

Sabe-se de (2.58) que o algoritmo SD possui a seguinte regra de atualização

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\nabla_n\xi \tag{2.82}$$

onde o operador $\nabla_n$ é dado por

$$\nabla_n = \left[\frac{\partial}{\partial w_1(n)} \quad \frac{\partial}{\partial w_2(n)} \quad \cdots \quad \frac{\partial}{\partial w_N(n)}\right]^{\mathrm{T}} \tag{2.83}$$

e

$$\xi = \mathrm{E}[e^2(n)]. \tag{2.84}$$

Substituindo o MSE dado por (2.84) por uma estimativa baseada no erro quadrático instantâneo dado por (2.79), obtém-se uma nova regra de atualização

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu\nabla_n\epsilon \\ &= \mathbf{w}(n) - \mu\nabla_n e^2(n)\end{aligned} \tag{2.85}$$

onde

$$\nabla_n e^2(n) = \left[\frac{\partial e^2(n)}{\partial w_1(n)} \quad \frac{\partial e^2(n)}{\partial w_2(n)} \quad \cdots \quad \frac{\partial e^2(n)}{\partial w_N(n)}\right]^{\mathrm{T}}. \tag{2.86}$$

Para $w_i(n)$, com $i = 1,2,\ldots,N$, obtém-se

$$\frac{\partial e^2(n)}{\partial w_i(n)} = 2e(n)\frac{\partial e(n)}{\partial w_i(n)}$$

$$= 2e(n)\frac{\partial[d(n) - \mathbf{x}^T(n)\mathbf{w}(n) + v(n)]}{\partial w_i(n)}$$

$$= -2e(n)x(n - i + 1). \quad (2.87)$$

Logo, tem-se que

$$\nabla_n e^2(n) = -2e(n)\mathbf{x}(n) \quad (2.88)$$

Agora, substituindo (2.88) em (2.85) obtém-se a regra de atualização do algoritmo LMS como sendo

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n). \quad (2.89)$$

Note que (2.89) apresenta-se como um procedimento simples de adaptação recursiva dos coeficientes do filtro adaptativo. A equação (2.89), juntamente com (2.80) e (2.81) compõem os passos a serem executados em cada iteração do processo de adaptação do algoritmo LMS. A Tabela 2.1 contém um sumário de execução deste algoritmo aplicado em um problema de identificação de uma planta $\mathbf{p}(n)$.

Tabela 2.1. Sumário do algoritmo adaptativo LMS

1. Inicialização

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

   e parâmetros

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

2. Obtenção dos dados de entrada e saída

   da planta

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

   e do filtro adaptativo

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

3. Cálculo do sinal de erro

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

4. Atualização dos coeficientes do filtro

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

A seguir é feita uma breve análise comparativa entre os algoritmos SD e LMS.

### *2.6.2 Comparação entre os Algoritmos SD e LMS*

Assim como o algoritmo SD, o desempenho do algoritmo LMS é altamente dependente da densidade espectral do sinal de entrada do filtro. Quando $x(n)$ possui um espectro plano (um

ruído branco, por exemplo), o algoritmo LMS converge rapidamente. Em outras palavras, quanto mais próxima da unidade for a relação dada por (2.78), melhor será o desempenho do algoritmo LMS. Outra semelhança entre os algoritmos SD e LMS reside no parâmetro de passo. Note de (2.82) e (2.89) que o parâmetro de passo, $\mu$, do algoritmo LMS é o mesmo usado pelo algoritmo SD

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{2.90}$$

sendo $\lambda_{\max}$ o maior autovalor de $\mathbf{R}$ (Widrow & Stearns, 1985). Isto implica que, assim como para o algoritmo SD, para garantir a convergência e estabilidade do algoritmo LMS o parâmetro de passo, $\mu$, deve ser escolhido de acordo com os limites de (2.90).

O algoritmo SD é capaz de alcançar o vetor de coeficientes ótimos, $\mathbf{w}_{\mathrm{opt}}$, definido pela equação de Wiener-Hopf, quanto maior for o número de iterações, $n$. Isto se dá pelo uso de medidas exatas do gradiente do MSE. Já o algoritmo LMS, por sua vez, usa uma estimativa ruidosa de $\nabla\xi$. Consequentemente, o algoritmo LMS, após várias iterações, resulta em um MSE em regime, $\xi(\infty)$, maior do que o MSE mínimo, $\xi_{\min}$, da função de desempenho do filtro de Wiener. Outra diferença básica entre os algoritmos SD e LMS reside na curva de aprendizagem de cada algoritmo. Esta curva é obtida traçando o gráfico da evolução do MSE para o algoritmo considerado em relação ao número de iterações. Nota-se de (2.72) que a curva de aprendizagem do algoritmo SD consiste de uma soma componentes exponenciais decrescentes bem definidas. Já para o algoritmo LMS, a curva de aprendizagem consiste de uma soma de exponenciais decrescentes ruidosas. A amplitude do ruído pode ser reduzida a partir da escolha de um menor valor para o parâmetro de passo, $\mu$ (Haykin, 1996).

A partir do algoritmo LMS surgiram outros dedicados aos mais variados processos (Paleologu, Ciochina, Benesty & Grant, 2015). O algoritmo LMS e suas derivações representam uma família padrão de aplicações envolvendo filtragem adaptativa (Jelfs, Mandic & Benesty, 2007). A seguir é apresentado um algoritmo LMS que usa um parâmetro de passo variável obtido a partir de dados do vetor de entrada. Este algoritmo proposto no final da década de 1960 e é denominado algoritmo LMS normalizado ou NLMS (Nagumo & Noda, 1967) e (Albert & Gardner Jr., 1967).

## 2.7 Algoritmo LMS Normalizado

O algoritmo NLMS é particularmente útil para operação em ambientes não-estacionários. Isto se dá pelo uso de um parâmetro de passo adaptativo que permite a adequação do filtro a entradas de natureza estatística variante. Este algoritmo pode ser visto como a solução para um problema de otimização com restrições (Goodwin & Sin, 1984) e (Haykin, 1996). Especificamente, considere o seguinte problema:

Minimizar a diferença

$$\boldsymbol{\eta}(n) = \mathbf{w}(n+1) - \mathbf{w}(n) \tag{2.91}$$

sujeito à restrição

$$\mathbf{w}(n+1)\mathbf{x}(n) = d(n). \tag{2.92}$$

Para resolver este problema de otimização com restrições usa-se o método dos multiplicadores de Lagrange (Bazaraa, Sherali & Shetty, 2006). Para tanto, primeiramente obtém-se a norma-2 elevada ao quadrado de $\boldsymbol{\eta}(n)$

$$\begin{aligned}\|\boldsymbol{\eta}(n)\|_2^2 &= \boldsymbol{\eta}^{\mathrm{T}}(n)\boldsymbol{\eta}(n)\\ &= [\mathbf{w}(n+1)-\mathbf{w}(n)]^{\mathrm{T}}[\mathbf{w}(n+1)-\mathbf{w}(n)]\\ &= \sum_{i=1}^{N}[w_i(n+1)-w_i(n)]^2. \end{aligned} \tag{2.93}$$

Agora, reescrevendo (2.92), obtém-se

$$\sum_{i=1}^{N} w_i(n+1)x(n-i+1) = d(n) \tag{2.94}$$

Então, a partir de (2.93) e (2.94), formula-se a função de custo para o problema de otimização com restrições dada por

$$J(n) = \sum_{i=1}^{N}[w_i(n+1)-w_i(n)]^2 + \gamma\left[d(n) - \sum_{i=1}^{N} w_i(n+1)x(n-i+1)\right] \tag{2.95}$$

onde $\gamma$ é um multiplicador de Lagrange. Para encontrar os valores ótimos de $w_i(n+1)$, com $i = 1,2,\dots,N$, é necessário derivar a função de custo $J(n)$ em relação a estes parâmetros e igualar os respectivos resultados a zero, ou seja

$$\frac{\partial J(n)}{\partial w_i(n+1)} = 0 \tag{2.96}$$

que resulta em

$$2[w_i(n+1)-w_i(n)] - \gamma x(n-i+1) = 0. \tag{2.97}$$

Logo, de (2.97) tem-se que

$$2[\mathbf{w}(n+1)-\mathbf{w}(n)] = \gamma\mathbf{x}(n). \tag{2.98}$$

Agora, isola-se $\gamma$ em (2.98) para obter

$$\gamma\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) = 2[\mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)]$$

$$\gamma\sum_{i=1}^{N}[x(n-i+1)]^2 = 2\left[\sum_{i=1}^{N} w_i(n+1)x(n-i+1) - \sum_{i=1}^{N} w_i(n)x(n-i+1)\right]$$

$$\gamma = \frac{2}{\|\mathbf{x}(n)\|_2^2}[\mathbf{w}^{\mathrm{T}}(n+1)\mathbf{x}(n) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)] \tag{2.99}$$

onde $\|\mathbf{x}(n)\|_2$ é a norma-2 do vetor de entrada $\mathbf{x}(n)$. Aplicando a restrição (2.92) em (2.99), tem-se

$$\gamma = \frac{2}{\|\mathbf{x}(n)\|_2^2}[d(n) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)]\,. \tag{2.100}$$

Aplicando a definição de erro de estimação, dada por (2.33), em (2.100), obtém-se

$$\gamma = \frac{2}{\|\mathbf{x}(n)\|_2^2} e(n). \tag{2.101}$$

Então, substituindo (2.101) em (2.98), tem-se que

$$\begin{aligned} \boldsymbol{\eta}(n) &= \mathbf{w}(n+1) - \mathbf{w}(n) \\ &= \frac{1}{\|\mathbf{x}(n)\|_2^2} e(n)\mathbf{x}(n). \end{aligned} \tag{2.102}$$

Para efetuar o controle sobre o ajuste dos coeficientes do vetor $\mathbf{w}(n)$, introduz-se um parâmetro de passo, $\mu$, e reescreve-se (2.89) da seguinte forma

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) + \varepsilon} e(n)\mathbf{x}(n) \tag{2.103}$$

sendo que (2.103) corresponde à regra de atualização dos coeficientes do algoritmo NLMS, e $\varepsilon > 0$ é um parâmetro de regularização, responsável por evitar divisão por zero e, consequentemente, estabilizar a solução. Note que, para este algoritmo, o parâmetro de passo, $\mu$, sofre uma normalização em relação à norma-2 elevada ao quadrado do vetor de entrada, $\mathbf{x}(n)$. A estabilidade (e convergência) do algoritmo NLMS é garantida para (Haykin, 1996)

$$0 < \mu < 2\,. \tag{2.104}$$

A Tabela 2.2 contém um sumário do algoritmo adaptativo NLMS.

Tabela 2.2. Sumário do algoritmo adaptativo NLMS

1. Inicialização
   $$\mathbf{w}(0) = \mathbf{0}$$
   e parâmetros
   $$0 < \mu < 2 \quad ; \quad \varepsilon > 0$$
2. Obtenção dos dados de entrada e saída da planta
   $$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n)$$
   e do filtro adaptativo
   $$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$$
3. Cálculo do sinal de erro
   $$e(n) = d(n) - y(n) + z(n)$$
4. Atualização dos coeficientes do filtro adaptativo
   $$\theta(n) = \mu \frac{1}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) + \varepsilon}$$
   $$\mathbf{w}(n+1) = \mathbf{w}(n) + \theta(n)e(n)\mathbf{x}(n)$$

A variável $\theta(n)$ é o parâmetro de passo adaptativo do algoritmo NLMS. Note que, ao contrário do algoritmo LMS, a escolha do parâmetro de passo do algoritmo NLMS independe

dos autovalores da matriz de autocorrelação da entrada, **R**. Analogamente ao caso do algoritmo LMS, o algoritmo NLMS pode ser entendido como uma implementação estocástica de outro algoritmo determinístico da mesma classe do algoritmo SD, a saber, o algoritmo de Newton (Sayed, 2003) e (Manolakis, Ingle & Kogon, 2005). Note também, de (2.102) e (2.103), que o algoritmo NLMS atualiza o vetor de coeficientes do filtro de tal forma que estimativa, $\mathbf{w}(n+1)$, da próxima iteração, $n+1$, exibe uma mudança mínima, no sentido da norma-2, em relação à estimativa, $\mathbf{w}(n)$, da iteração anterior, $n$. O algoritmo NLMS possui uma taxa de convergência potencialmente mais rápida do que o algoritmo LMS original para sinais de entrada correlacionados e não-correlacionados (Haykin, 1996).

## 2.8 Medidas de Desempenho

A seguir são apresentadas duas medidas de desempenho comumente empregadas na avaliação dos algoritmos adaptativos para os mais variados problemas. Estas medidas serão utilizadas para avaliar os algoritmos adaptativos desenvolvidos neste trabalho. São elas o erro quadrático e o desalinhamento normalizado em dB.

### *2.8.1 Erro Quadrático em dB*

Sabe-se da seção 2.4 que o MSE é a função objetivo usada pelos filtros lineares ótimos e que o mesmo é dado por (2.84). Por outro lado, os algoritmos da classe LMS buscam a minimização do erro quadrático instantâneo dado por (2.79). Neste trabalho, uma das medidas de desempenho que serão empregadas na avaliação dos algoritmos adaptativos desenvolvidos neste trabalho é o erro quadrático em dB dado por

$$\epsilon_{\mathrm{dB}} = 10\log_{10} e^2(n). \tag{2.105}$$

A outra medida de desempenho adotada neste trabalho é o desalinhamento normalizado em dB, o qual é mostrado a seguir.

### *2.8.2 Desalinhamento Normalizado em dB*

O desalinhamento normalizado em dB é uma das medidas mais comumente usadas na avaliação do desempenho de algoritmos adaptativos em problemas de identificação de sistemas e cancelamento de eco (Paleologu, Benesty & Ciochina, 2010). Esta medida é definida por

$$\Psi(n) = 10\log_{10}\left(\frac{\left\|\mathbf{w}_{\mathrm{opt}} - \mathbf{w}(n)\right\|_2^2}{\left\|\mathbf{w}_{\mathrm{opt}}\right\|_2^2}\right). \tag{2.106}$$

Note de (2.106) que $\Psi(n)$ depende unicamente do vetor de coeficientes ótimos, $\mathbf{w}_{\mathrm{opt}}$, que representa a resposta impulsiva da planta a ser identificada, e da estimativa atual, $\mathbf{w}(n)$ do vetor de coeficientes do filtro. Esta medida não é aplicável na prática, uma vez que requer conhecimento a priori da resposta impulsiva da planta verdadeira.

# Capítulo 3

# Algoritmos Adaptativos LMS Normalizados Proporcionais

Neste capítulo são estudados algoritmos adaptativos LMS normalizados proporcionais. Tais algoritmos foram desenvolvidos com o objetivo de melhorar o desempenho dos filtros adaptativos para identificação e controle de plantas esparsas. Historicamente, algoritmos da classe PNLMS encontram sua principal aplicação em problemas de cancelamento de eco acústico e de rede, onde geralmente as plantas encontradas neste tipo de problema possuem alto grau de esparsidade (Huang, Benesty & Chen, 2006), (Loganatham, Khong & Naylor, 2009) e (Wagner & Doroslovacky, 2013). Especificamente, a presença de atrasos cada vez maiores nos sistemas de comunicação modernos tem aumentado a necessidade por filtros adaptativos de convergência rápida para ambientes esparsos. Atualmente, além de cancelamento de eco em sistemas de telefonia, aplicações dos algoritmos adaptativos proporcionais incluem identificação de canais de comunicação subaquáticos (Pelekanakis & Chitre, 2013), transmissões televisivas terrestres em alta definição (Fan, He, Wang & Jiang, 2005), dentre outras.

Inicialmente, é apresentada a estrutura geral dos algoritmos adaptativos proporcionais. Logo após, os algoritmos PNLMS padrão e IPNLMS são mostrados e o comportamento destes algoritmos para a identificação de plantas esparsas é discutido em detalhes. Por fim, são mostrados os principais algoritmos adaptativos proporcionais desenvolvidos ao longo das últimas décadas, os quais foram todos derivados dos algoritmos PNLMS e IPNLMS.

## 3.1 Estrutura Geral dos Algoritmos Adaptativos Proporcionais

Para os algoritmos adaptativos proporcionais, a regra geral de atualização dos coeficientes do filtro adaptativo é dada por (Souza, Seara & Morgan, 2012)

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

onde

$$\mathbf{x}(n) = [x(n) \;\; x(n-1) \;\; \ldots \;\; x(n-N+1)]^{\mathrm{T}} \tag{3.2}$$

e

$$\mathbf{w}(n) = [w_1(n) \;\; w_2(n) \;\; \ldots \;\; w_N(n)]^{\mathrm{T}} \tag{3.3}$$

são, respectivamente, os vetores de entrada e de coeficientes do filtro adaptativo. As variáveis $0 < \mu < 2$, $\varepsilon > 0$ e são, respectivamente, os parâmetros de passo e de regularização. A variável $N$ é a quantidade de coeficientes do filtro. A matriz

$$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \ldots \quad g_N(n)] \tag{3.4}$$

de ordem $N \times N$ é responsável pela distribuição dos ganhos individuais entre os coeficientes do filtro adaptativo. A forma de distribuição desses ganhos varia de acordo com o algoritmo considerado. O sinal de erro é calculado por

$$e(n) = d(n) - y(n) + v(n) \tag{3.5}$$

onde $d(n)$ é a saída desejada, $y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$ é a saída do filtro adaptativo e $v(n)$ é um ruído de medição de variância $\sigma_v^2$. A seguir é apresentado o algoritmo PNLMS, o qual foi um dos primeiros algoritmos adaptativos proporcionais desenvolvidos e também deu origem, posteriormente, a outros algoritmos da mesma classe.

## 3.2 Algoritmo PNLMS

A necessidade de algoritmos adaptativos capazes de proporcionar velocidade de convergência rápida para plantas esparsas levou ao desenvolvimento do algoritmo PNLMS. Tal algoritmo foi talvez o primeiro a empregar a filosofia proporcional de adaptação, explorando a característica esparsa presente em algumas plantas tais como os caminhos de eco de rede (Wagner & Doroslovacky, 2011).

O algoritmo PNLMS atinge uma velocidade de convergência significativamente superior ao NLMS quando a planta a ser identificada é esparsa. Isto ocorre sem que haja perda de qualidade na estimação, porém há um acréscimo nos requisitos computacionais. O ganho individual que controla o ajuste do $i$-ésimo coeficiente do algoritmo PNLMS padrão é obtido a partir de (Duttweiler, 2000)

$$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N} \phi_j(n)} \tag{3.6}$$

onde

$$\phi_i(n) = \max[f(n), |w_i(n)|] \tag{3.7}$$

e

$$f(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_\infty]. \tag{3.8}$$

A função de proporcionalidade $\phi_i(n)$ de (3.7) determina o valor do $i$-ésimo ganho, $g_i(n)$, conforme a magnitude do respectivo $i$-ésimo coeficiente, $w_i(n)$. O fator de ativação $f(n)$ de (3.8) influencia diretamente na adaptação dos coeficientes considerados inativos (Souza, Tobias, Seara & Morgan, 2010). O parâmetro de inicialização, $\delta$, é usado somente no início do processo de adaptação ($n = 0$), quando todos os coeficientes do filtro são nulos, ou seja, $\mathbf{w}(0) = \mathbf{0}$. O parâmetro de proporcionalidade, $\rho$, evita a estagnação de um dado coeficiente cuja magnitude, em um dado instante $n$, seja muito menor do que a magnitude do maior coeficiente (Souza, Tobias, Seara & Morgan, 2010). O fator $\|\mathbf{w}(n)\|_\infty$ de (3.8) é a norma infinita do vetor de coeficientes do filtro adaptativo.

Diferentemente da regra geral dos algoritmos adaptativos proporcionais dada por (3.1), a regra de atualização dos coeficientes do algoritmo PNLMS original é dada por (Duttweiler, 2000)

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu \mathbf{G}(n) e(n) \mathbf{x}(n)}{\mathbf{x}^{\mathrm{T}} \mathbf{x}(n) + \varepsilon}. \tag{3.9}$$

Note que, em (3.9), a matriz $\mathbf{G}(n)$ não está presente no denominador do termo de correção dos coeficientes. Esta é a única diferença entre a regra proposta por Donald L. Duttweiler e a regra padrão dos algoritmos adaptativos proporcionais. A partir do algoritmo PNLMS original obtém-se o algoritmo PNLMS padrão, o qual é descrito em detalhes na Tabela 3.1, considerando um problema de identificação de uma planta $\mathbf{p}(n)$.

Tabela 3.1. Sumário do algoritmo adaptativo PNLMS padrão

1. Inicialização

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

   e parâmetros

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

2. Obtenção dos dados de entrada e saída

   da planta

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

   e do filtro adaptativo

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

3. Cálculo do sinal de erro

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

4. Fator de ativação

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

5. Função de proporcionalidade

   $$\phi_i(n) = \max[f(n), |w_i(n)|], \qquad i = 1,2,3, \ldots, N$$

6. Ganho individual dos coeficientes do filtro adaptativo

   $$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N} \phi_j(n)}, \qquad i = 1,2,3, \ldots, N$$

7. Matriz de ganhos individuais ($N$ x $N$)

   $$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \ldots \quad g_N(n)]$$

8. Atualização dos coeficientes do filtro adaptativo

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

Note que o algoritmo NLMS pode ser obtido a partir do PNLMS padrão simplesmente adotando

$$\mathbf{G}(n) = \mathbf{I} \tag{3.10}$$

onde **I** é a matriz identidade. Para efeitos de análise comparativa, o parâmetro de regularização do algoritmo PNLMS deve ser escolhido como sendo (Huang, Benesty & Chen, 2006)

$$\varepsilon_{\mathrm{PNLMS}} = \frac{\varepsilon_{\mathrm{NLMS}}}{N}\,. \tag{3.11}$$

A seguir, é feita uma análise do comportamento do algoritmo PNLMS. Especificamente, o efeito do parâmetro de proporcionalidade, $\rho$, e do fator de ativação, $f(n)$, na velocidade de convergência e na distribuição dos ganhos do algoritmo PNLMS é estudado.

### *3.2.1 Análise do Comportamento do Algoritmo PNLMS Padrão*

Nesta seção é estudado o comportamento do algoritmo PNLMS padrão em relação ao parâmetro de proporcionalidade, $\rho$, e ao fator de ativação, $f(n)$. O objetivo de tal estudo é avaliar o impacto destes fatores na velocidade de convergência e na distribuição dos ganhos do algoritmo PNLMS padrão, considerando um problema de identificação de plantas esparsas. Antes de realizar este estudo, deve-se definir duas figuras de mérito. A primeira delas é a distribuição de ganho total dada por

$$\varphi_i = \sum_{n=0}^{L-1} g_i(n) \tag{3.12}$$

onde $g_i(n)$ é o valor do ganho do *i*-ésimo coeficiente na *n*-ésima iteração e *L* é o número total de iterações consideradas para o processo de adaptação do algoritmo em questão. A distribuição de ganho total do *i*-ésimo coeficiente, $\varphi_i$, acumula os ganhos destinados a este coeficiente em todas as iterações do processo de adaptação. Por definição, um algoritmo adaptativo proporcional deve distribuir os ganhos de adaptação proporcionalmente à magnitude dos coeficientes do filtro. Logo, quanto maior o valor de $\varphi_i$ para os coeficientes ativos, mais "proporcional" é o algoritmo em questão. Outra figura de mérito considerada por esta análise é a média dos ganhos totais atribuídos aos coeficientes inativos dada por

$$\varphi_{\mathrm{médio}}^{\mathrm{inativos}} = \frac{1}{N - N_{\mathrm{ativos}}} \sum_{i \notin A} \varphi_i \tag{3.13}$$

onde $N_{\mathrm{ativos}}$ é a quantidade de coeficientes ativos e $A$ é o conjunto dos coeficientes ativos do filtro. Analogamente à distribuição de ganho total, quanto menor o valor de $\varphi_{\mathrm{médio}}^{\mathrm{inativos}}$, melhor explorada será a filosofia proporcional pelo algoritmo em questão.

Para avaliar o comportamento do algoritmo PNLMS padrão em relação ao parâmetro de proporcionalidade, $\rho$, e ao fator de ativação, $f(n)$, considera-se simulações de Monte Carlo (média de 100 realizações independentes) para um problema de identificação de sistemas. Os valores das variáveis avaliadas por uma simulação de Monte Carlo são obtidas a partir de (Fishman, 1996), (Manolakis, Ingle & Kogon, 2005) e (Rubstein, 2008)

$$\mathrm{E}[V(n)] \cong \frac{V_1(n) + V_2(n) + \cdots + V_R(n)}{R}$$

$$= \frac{1}{R}\sum_{r=1}^{R} V_r(n) \tag{3.14}$$

onde E[·] é o operador valor esperado, $V_r(n)$ é o valor da variável avaliada, $V$, na $n$-ésima iteração da $r$-ésima realização e $R$ é o número de realizações independentes consideradas pela simulação de Monte Carlo. O cenário de todas as simulações consiste de uma planta esparsa com $N = 100$ coeficientes (Martin, Seathers, Williamson & Johnson, 2002). Os coeficientes ativos (não-nulos) desta planta estão localizados nas posições $\{1, 30, 35, 85\}$, com seus respectivos valores iguais a $\{0{,}1\ ,\ 1{,}0\ ,\ -0{,}5\ ,\ 0{,}1\}$. segundo a definição de (1.2), a medida de esparsidade desta planta é igual a $S(\mathbf{p}) = 0{,}9435$, onde $\mathbf{p} = [p_1\ \ p_2\ \ \ldots\ \ p_N]^{\mathrm{T}}$ é a resposta ao impulso da planta a ser identificada. O sinal de entrada é correlacionado, com média zero e variância unitária, obtido através de um processo autorregressivo de ordem 2, AR(2), dado por

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + b(n) \tag{3.15}$$

sendo $a_1 = 0{,}4$, $a_2 = -0{,}4$ e $b(n)$ é um ruído branco com média zero e variância $\sigma_b^2 = 0{,}77$. A dispersão da matriz de autocorelação da entrada é $\chi(\mathbf{R}) = 10$. Cada elemento $r(m)$ da matriz $\mathbf{R}$ é calculado a partir de (Haykin, 1996)

$$r(m) = \frac{\sigma_x^2 p_1 (p_2^2 - 1) p_1^m}{(p_2 - p_1)(p_1 p_2 + 1)} - \frac{\sigma_x^2 p_2 (p_1^2 - 1) p_2^m}{(p_2 - p_1)(p_1 p_2 + 1)} \tag{3.16}$$

onde $m$ é o *lag* entre as amostras da entrada, $\sigma_x^2$ é a variância de $x(n)$ e

$$p_{1,2} = \frac{1}{2}\left(-a_1 \pm \sqrt{a_1^2 - 4a_2}\right). \tag{3.17}$$

O ruído de medição, $v(n)$, é branco gaussiano com variância $\sigma_v^2 = 10^{-3}$ (SNR = 30dB). A Figura de mérito considerada para avaliar a velocidade de convergência do algoritmo PNLMS padrão em relação a $\rho$ e $f(n)$ é o desalinhamento normalizado em dB dado por (2.106).

Para avaliar a dependência do algoritmo PNLMS padrão em relação ao parâmetro de proporcionalidade, $\rho$, curvas de desalinhamento são mostradas na Figura 3.1, considerando 2500 iterações de identificação da planta esparsa $\mathbf{p}$, com $\rho$ assumindo cada um dos seguintes valores: 0,001, 0,005, 0,01, 0,05, 0,1 e 0,5. Similarmente, a Figura 3.2 mostra o desempenho do algoritmo PNLMS padrão para 2000 iterações de identificação da planta esparsa $\mathbf{p}$, considerando diversos valores de $\rho$, que vão de $10^{-8}$ a $10^{-3}$.

Para as Figuras 3.1 e 3.2, considera-se um parâmetro de passo igual a $\mu = 0{,}5$ e um parâmetro de inicialização igual a $\delta = 0{,}01$. Note, das curvas das Figuras 3.1 e 3.2, que o valor do parâmetro de proporcionalidade, $\rho$, impacta diretamente na velocidade de convergência durante todo o processo de adaptação do algoritmo PNLMS padrão.

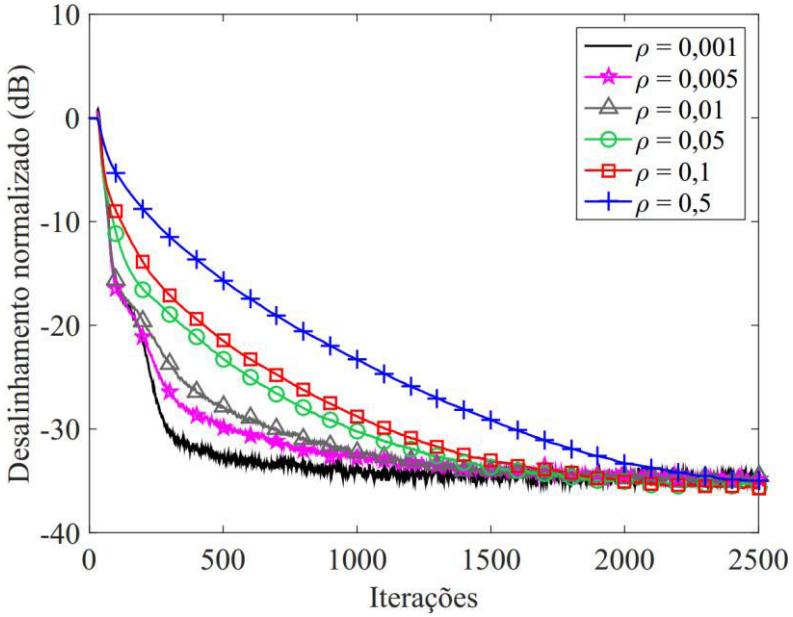



Figura 3.1. Desalinhamento normalizado do algoritmo PNLMS padrão para valores do parâmetro de proporcionalidade, $\rho$, iguais a 0,001, 0,005, 0,01, 0,05, 0,1 e 0,5.

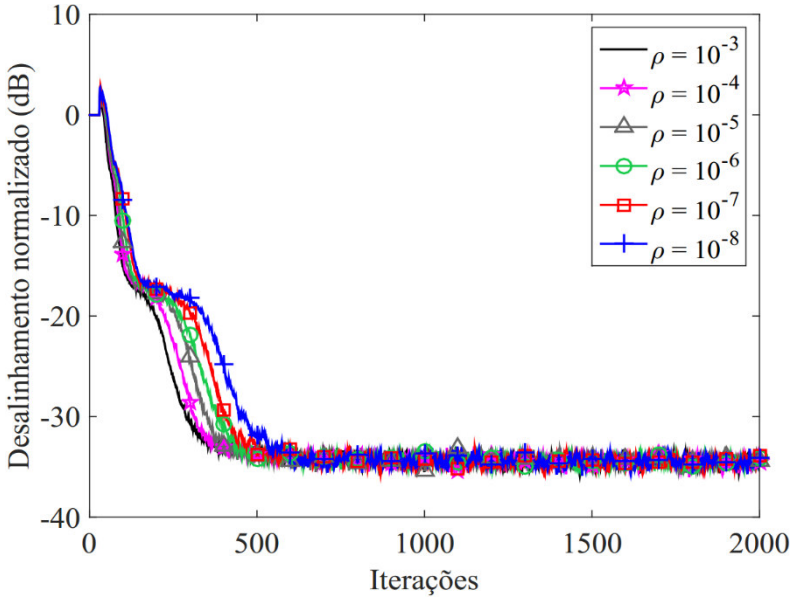



Figura 3.2. Desalinhamento normalizado do algoritmo PNLMS padrão para valores do parâmetro de proporcionalidade, $\rho$, iguais a $10^{-3}$, $10^{-4}$, $10^{-5}$, $10^{-6}$, $10^{-7}$ e $10^{-8}$.

Os resultados mostrados nas Figuras 3.1 e 3.2 podem ser explicados pela análise da Figura 3.3, a qual mostra o ganho de adaptação atribuído ao $i$-ésimo coeficiente em função da sua respectiva magnitude (normalizada) para vários valores de $\rho$, que vão de 0,001 a 1. As magnitudes normalizadas dos coeficientes do filtro adaptativo são obtidas a partir de

$$w_i^{\text{norm}}(n) = \frac{|w_i(n)|}{\max[|\mathbf{w}(n)|]} \tag{3.18}$$

onde $w_i^{\text{norm}}(n)$ é a magnitude normalizada do $i$-ésimo coeficiente e max[·] é o operador valor máximo. As curvas da Figura 3.3 mostram que, quanto maior o valor do parâmetro $\rho$, menos proporcional é a atualização dos coeficientes do filtro. Logo, tem-se que o valor do parâmetro de proporcionalidade afeta diretamente os ganhos de adaptação $g_i(n)$ dos coeficientes $w_i(n)$, com $i = 1,2,\ldots,N$, do filtro adaptativo.

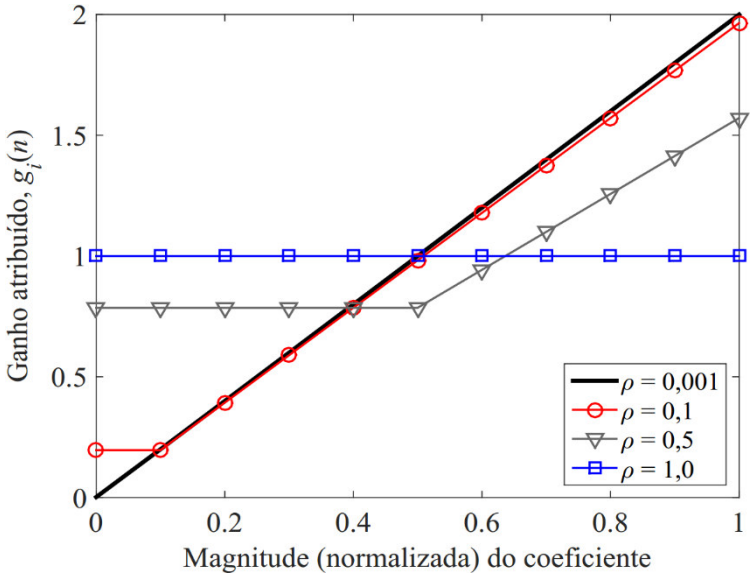



Figura 3.3. Ganho atribuído pelo algoritmo PNLMS padrão aos coeficientes de um filtro adaptativo conforme a magnitude (normalizada) dos mesmos, considerando valores do parâmetro de proporcionalidade, ρ, iguais a 0,001, 0,1, 0,5 e 1,0.

Os resultados das Figuras 3.1, 3.2 e 3.3 podem ser melhor observados através das Tabelas 3.2 e 3.3, as quais mostram as distribuições de ganho totais dos coeficientes ativos, além das médias dos ganhos totais dos coeficientes inativos, obtidas para os casos considerados nas Figuras 3.1 e 3.2, respectivamente. Os valores destas Tabelas são obtidos a partir de (3.12) e (3.13), sendo que $N_{\text{ativos}} = 4$ e $A = \{1, 30, 35, 85\}$, conforme os dados da planta esparsa **p**. Note dos valores das Tabelas 3.2 e 3.3 que, quanto menor o valor de $\rho$, maiores são os ganhos destinados aos coeficientes ativos e menores são os ganhos destinados aos coeficientes inativos.

Tabela 3.2. Distribuições de ganho totais dos coeficientes ativos, e médias dos ganhos totais dos coeficientes inativos, considerando 2500 iterações de identificação da planta esparsa **p** usando o algoritmo PNLMS padrão e com $\rho$ assumindo diversos valores, que vão de 0,001 a 0,5.

| **Parâmetros** | $\varphi_1$ | $\varphi_{30}$ | $\varphi_{35}$ | $\varphi_{85}$ | $\varphi_{médio}^{inativos}$ |
|---|---|---|---|---|---|
| $\rho = 0{,}001$ | 157,5 | 1359,7 | 669,7 | 124,5 | 1,9 |
| $\rho = 0{,}005$ | 129,3 | 1129,4 | 555,0 | 102,2 | 6,1 |
| $\rho = 0{,}01$ | 104,2 | 927,7 | 454,7 | 83,2 | 9,7 |
| $\rho = 0{,}05$ | 42,1 | 380,5 | 184,0 | 33,8 | 19,3 |
| $\rho = 0{,}1$ | 25,7 | 218,6 | 104,2 | 22,2 | 22,2 |
| $\rho = 0{,}5$ | 25,1 | 49,1 | 24,8 | 24,8 | 24,7 |

Tabela 3.3. Distribuições de ganho totais dos coeficientes ativos e médias dos ganhos totais dos coeficientes inativos considerando 2000 iterações de identificação da planta esparsa **p** usando o algoritmo PNLMS padrão e com $\rho$ assumindo diversos valores, que vão de $10^{-8}$ a $10^{-3}$.

| **Parâmetros** | $\varphi_1$ | $\varphi_{30}$ | $\varphi_{35}$ | $\phi_{85}$ | $\varphi_{\text{médio}}^{\text{inativos}}$ |
|---|---|---|---|---|---|
| $\rho = 10^{-8}$ | 147,9 | 1153,2 | 560,3 | 96,1 | 0,4 |
| $\rho = 10^{-7}$ | 145,9 | 1152,1 | 561,5 | 97,5 | 0,4 |
| $\rho = 10^{-6}$ | 144,8 | 1151,3 | 562,4 | 99,7 | 0,4 |
| $\rho = 10^{-5}$ | 143,2 | 1147,0 | 560,1 | 101,3 | 0,5 |
| $\rho = 10^{-4}$ | 139,7 | 1135,7 | 557,1 | 101,5 | 0,7 |
| $\rho = 10^{-3}$ | 130,6 | 1080,5 | 530,6 | 96,8 | 1,7 |

Note, da Figura 3.3 e das Tabelas 3.2 e 3.3, que um valor pequeno de $\rho$ distribui melhor os ganhos de adaptação entre os coeficientes do filtro quando a planta é esparsa. Os resultados mostrados nas Figuras 3.1, 3.2 e 3.3 e nas Tabelas 3.2 e 3.3 demonstram que um procedimento para otimizar a escolha do parâmetro $\rho$ pode melhorar o desempenho do algoritmo PNLMS para a identificação de plantas esparsas.

Para avaliar a dependência do algoritmo PNLMS padrão em relação ao fator de ativação, $f(n)$, faz-se uma análise semelhante à realizada para o parâmetro de proporcionalidade, $\rho$. Para tal, na Figura 3.4 são mostradas curvas de desalinhamento, considerando 2000 iterações de identificação da planta esparsa **p**, com $f(n)$ assumindo os seguintes valores: $10^{-7}$, $10^{-6}$, $10^{-5}$, $10^{-4}$, $10^{-3}$ e $10^{-2}$. Os parâmetros de passo e de inicialização considerados são $\mu = 0{,}5$ e $\delta = 0{,}01$, respectivamente. Note, das curvas da Figura 3.4, que o valor do fator de ativação, $f(n)$, impacta diretamente na velocidade de convergência do algoritmo PNLMS padrão durante todo o processo de adaptação.

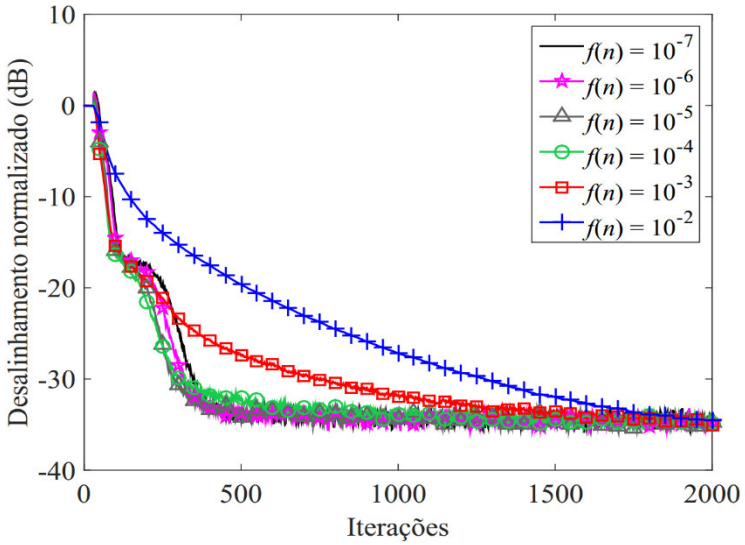



Figura 3.4. Desalinhamento normalizado do algoritmo PNLMS padrão para valores do fator de ativação, f(n), iguais a $10^{-7}$, $10^{-6}$, $10^{-5}$, $10^{-4}$, $10^{-3}$ e $10^{-2}$.

A Figura 3.5 mostra o ganho de adaptação atribuído ao $i$-ésimo coeficiente em função da sua respectiva magnitude (normalizada) para vários valores de $f(n)$, que vão de 0,001 a 1. As magnitudes normalizadas dos coeficientes do filtro adaptativo são obtidas a partir de (3.18). As curvas da Figura 3.5 mostram que, quanto maior o valor do fator de ativação, $f(n)$, menos proporcional é a atualização dos coeficientes do filtro. Logo, tem-se que o valor de $f(n)$ afeta diretamente os ganhos de adaptação $g_i(n)$ dos coeficientes $w_i(n)$, com $i = 1,2,\ldots,N$, do filtro adaptativo.

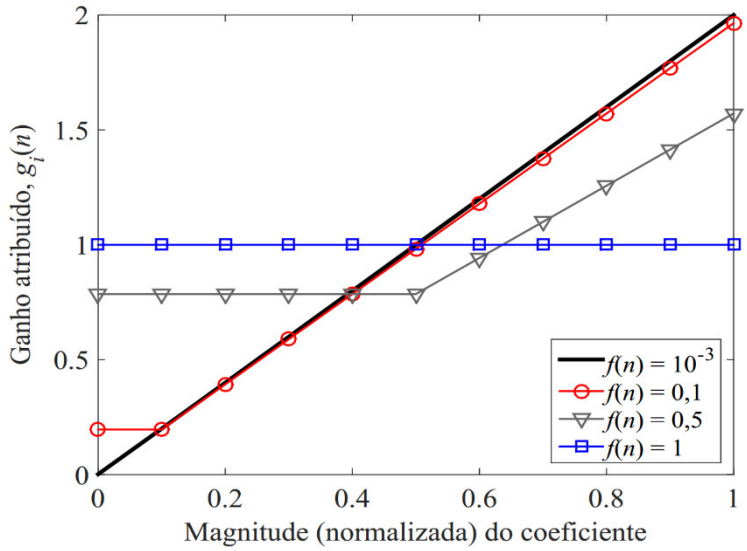



Figura 3.5. Ganho atribuído pelo algoritmo PNLMS padrão aos coeficientes de um filtro conforme a magnitude (normalizada) dos mesmos, considerando valores de $f(n)$, iguais a 0,001, 0,1, 0,5 e 1,0.

Os resultados das Figuras 3.4 e 3.5 podem ser melhor observados através da Tabela 3.4, a qual mostra as distribuições de ganho totais dos coeficientes ativos, além das médias dos ganhos totais dos coeficientes inativos, obtidas para os casos considerados na Figuras 3.4. Os valores da Tabela 3.4 são obtidos a partir de (3.12) e (3.13), sendo que $N_{\text{ativos}} = 4$ e $A = \{1, 30, 35, 85\}$, conforme os dados da planta esparsa **p**. Note dos valores da Tabela 3.4 que, assim como ocorre para o parâmetro de proporcionalidade, quanto menor o valor do fator de ativação, $f(n)$, maiores são os ganhos destinados aos coeficientes ativos e menores são os ganhos destinados aos coeficientes inativos.

Tabela 3.4. Distribuições de ganho totais dos coeficientes ativos e médias dos ganhos totais dos coeficientes inativos para 2000 iterações de identificação da planta esparsa **p,** usando o algoritmo PNLMS padrão e com $f(n)$ assumindo diversos valores, que vão de $10^{-7}$ a $10^{-2}$.

| **Parâmetros** | $\varphi_1$ | $\varphi_{30}$ | $\varphi_{35}$ | $\varphi_{85}$ | $\varphi_{\text{médio}}^{\text{inativos}}$ |
|---|---|---|---|---|---|
| $f(n) = 10^{-7}$ | 143,1 | 1153,0 | 562,5 | 99,4 | 0,4 |
| $f(n) = 10^{-6}$ | 140,6 | 1147,7 | 561,4 | 101,1 | 0,5 |
| $f(n) = 10^{-5}$ | 135,2 | 1138,8 | 558,3 | 101,4 | 0,7 |
| $f(n) = 10^{-4}$ | 117,5 | 1080,3 | 530,4 | 96,8 | 1,8 |
| $f(n) = 10^{-3}$ | 73,2 | 735,9 | 360,9 | 64,6 | 7,9 |
| $f(n) = 10^{-2}$ | 18,2 | 169,4 | 81,8 | 17,8 | 17,8 |

Note da Figura 3.5 e da Tabela 3.4 que um valor pequeno de $f(n)$ distribui melhor os ganhos de adaptação entre os coeficientes do filtro quando a planta é esparsa. Os resultados mostrados nas Figuras 3.4 e 3.5 e na Tabela 3.4 demonstram que um procedimento para otimizar a escolha de $f(n)$ pode melhorar o desempenho do algoritmo PNLMS para a identificação de plantas esparsas.

Os resultados das Figuras de 3.1 a 3.5 e das Tabelas de 3.2 a 3.4 mostram que tanto o parâmetro de proporcionalidade, $\rho$, quanto o fator de ativação, $f(n)$, causam um impacto semelhante no comportamento do algoritmo PNLMS padrão. Isto pode ser explicado pela análise de (3.7) e (3.8), referentes à função de proporcionalidade, $\phi_i(n)$, com $i = 1,2,\dots,N$, e ao fator de ativação, $f(n)$, do algoritmo PNLMS padrão, respectivamente. Note que o valor atribuído para o parâmetro de proporcionalidade, $\rho$, impacta diretamente no valor do fator de ativação, $f(n)$. Este último, por sua vez, impacta no valor da função de proporcionalidade e, consequentemente, no ganho de adaptação de cada coeficiente do filtro.

Apesar de atingir uma velocidade de convergência superior à dos algoritmos LMS e NLMS ao lidar com plantas esparsas, o desempenho do algoritmo PNLMS decai para plantas de esparsidade média ou com baixo grau de esparsidade (Benesty & Gay, 2002). Além disso, a rápida velocidade inicial de convergência deste algoritmo não é mantida em todo o processo de adaptação (Deng & Doroslovacky, 2006). Para superar estes problemas, versões aprimoradas do algoritmo PNLMS, tais como os algoritmos PNLMS++ (Gay, 1998), IPNLMS (Benesty & Gay, 2002), MPNLMS (Deng & Doroslovacky, 2006), SC-PNLMS (Loganathan, Khong & Naylor, 2008) e IAF-PNLMS (Souza, Tobias, Seara & Morgan, 2010), foram desenvolvidas. A seguir, uma das mais importantes versões aprimoradas do algoritmo PNLMS é descrito a seguir em detalhes, a saber, o algoritmo IPNLMS.

## 3.3 Algoritmo IPNLMS

O algoritmo IPNLMS, assim como o algoritmo PNLMS, deu origem a outros algoritmos adaptativos proporcionais, tais como o IIPNLMS (Cui, Naylor & Brown, 2004) e o SC-IPNLMS (Khong & Naylor, 2006). Tal algoritmo foi construído a partir de modificações na função de proporcionalidade do PNLMS. O principal objetivo dessas modificações foi tornar mais "suave" a escolha do ganho de adaptação individual para cada coeficiente do filtro adaptativo e, consequentemente, explorar melhor a filosofia proporcional originária do algoritmo PNLMS (Benesty & Gay, 2002). Nesta seção são apresentadas a estrutura e as principais características do algoritmo IPNLMS. Primeiramente, é mostrado um procedimento para derivar o algoritmo IPNLMS a partir do algoritmo PNLMS padrão. Logo após, é feito um estudo sobre o efeito do parâmetro de proporcionalidade do algoritmo IPNLMS sobre o comportamento do mesmo.

### *3.3.1 Derivação do Algoritmo IPNLMS a partir do PNLMS Padrão*

O algoritmo IPNLMS pode ser derivado a partir do PNLMS padrão (Souza, 2012). Para tal, em (3.8), substitui-se $\|\mathbf{w}(n)\|_\infty$ por $\|\mathbf{w}(n)\|_1$ (norma-1 do vetor de coeficientes do filtro adaptativo) e adota-se $\rho = 1/N$. Assim, tem-se um novo fator de ativação dado por

$$f(n) = \begin{cases} \frac{1}{N}\max[\delta, \|\mathbf{w}(n)\|_1], & n = 0 \\ \frac{\|\mathbf{w}(n)\|_1}{N}, & n \geq 1 \end{cases}. \tag{3.19}$$

Desta forma, para $n \geq 1$, a função de proporcionalidade dada por (3.7) pode ser reescrita como sendo

$$\phi_i(n) = \max\left[\frac{\|\mathbf{w}(n)\|_1}{N}, |w_i(n)|\right]. \tag{3.20}$$

Agora, para a função de proporcionalidade dada por (3.20), uma média ponderada é usada no lugar do operador max[·]. Logo, a nova função de proporcionalidade é (Benesty & Gay, 2002)

$$\phi_i(n) = (1-\alpha)\frac{\|\mathbf{w}(n)\|_1}{N} + (1+\alpha)|w_i(n)| \tag{3.21}$$

onde $-1 \leq \alpha < 1$ é um fator de ponderação. Note que a primeira parcela do lado direito de (3.21) é comum a todos os coeficientes do filtro, enquanto que a segunda é proporcional à magnitude do $i$-ésimo coeficiente.

Agora, considerando (3.21), desenvolve-se o denominador de (3.6) para obter

$$\begin{aligned} \sum_{i=1}^{N}\phi_i(n) &= \sum_{i=1}^{N}\left[(1-\alpha)\frac{\|\mathbf{w}(n)\|_1}{N} + (1+\alpha)|w_i(n)|\right] \\ &= \left[\frac{1-\alpha}{N}\sum_{i=1}^{N}\|\mathbf{w}(n)\|_1\right] + \left[(1+\alpha)\sum_{i=1}^{N}|w_i(n)|\right] \\ &= \frac{(1-\alpha)}{N}N\|\mathbf{w}(n)\|_1 + (1+\alpha)\|\mathbf{w}(n)\|_1 \\ &= (1-\alpha)\|\mathbf{w}(n)\|_1 + (1+\alpha)\|\mathbf{w}(n)\|_1 \\ &= 2\|\mathbf{w}(n)\|_1. \end{aligned} \tag{3.22}$$

Então, substituindo (3.21) e (3.22) em (3.6), obtém-se o ganho individual do algoritmo IPNLMS como sendo

$$g_i(n) = (1-\alpha)\frac{1}{2N} + (1+\alpha)\frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varsigma} \tag{3.23}$$

onde $\varsigma > 0$ é um parâmetro de regularização usado para evitar divisão por zero. O fator $\alpha$ é denominado parâmetro de proporcionalidade do algoritmo IPNLMS.

O algoritmo IPNLMS usa a regra de atualização dos coeficientes de (3.1), com os elementos de $\mathbf{G}(n)$ sendo calculados a partir de (3.23). A Tabela 3.5 contém o sumário do algoritmo IPNLMS.

Tabela 3.5. Sumário do algoritmo adaptativo IPNLMS.

1. Inicialização

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

e parâmetros

$$0 < \mu < 2$$

$$\varepsilon > 0$$

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

$$\varsigma > 0$$

2. Obtenção dos dados de entrada e saída da planta

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

e do filtro adaptativo

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

3. Cálculo do sinal de erro

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

4. Ganho individual dos coeficientes do filtro adaptativo

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

5. Matriz de ganhos individuais ($N$ x $N$)

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

6. Atualização dos coeficientes do filtro adaptativo

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

Para efeitos de análise comparativa, o parâmetro de regularização do algoritmo PNLMS deve ser escolhido como sendo

$$\varepsilon_{\mathrm{IPNLMS}} = \frac{1-\alpha}{2N}\varepsilon_{\mathrm{NLMS}} \tag{3.24}$$

Embora também possa ser apresentado como uma alternativa eficiente para plantas de esparsidade média, o algoritmo IPNLMS não é capaz de atingir a mesma velocidade inicial de convergência do PNLMS para plantas com resposta ao impulso altamente esparsas (Deng & Doroslovacky, 2006). Além disso, seu desempenho depende da escolha do parâmetro de proporcionalidade $-1 \leq \alpha < 1$, o qual é constante durante todo o processo de adaptação. Resultados experimentais demonstram que $0.0$, $-0.5$ e $-0.75$ são boas escolhas para tal parâmetro (Khong & Naylor, 2006). Note que, para $\alpha = -1$, o algoritmo IPNLMS reduz-se ao algoritmo NLMS. Em contrapartida, para $\alpha = 1$, o IPNLMS comporta-se de forma semelhante

ao PNLMS (Wagner & Doroslovacky, 2013). A seguir, é feito um estudo a respeito do efeito do parâmetro $\alpha$ no desempenho do algoritmo IPNLMS.

### *3.3.2 Análise do Comportamento do Algoritmo IPNLMS*

Nesta seção é estudado o efeito do parâmetro de proporcionalidade, $\alpha$, no desempenho do algoritmo adaptativo IPNLMS. Para tal, são realizadas simulações de Monte Carlo (média de 100 realizações independentes) para um problema de identificação de sistemas. Os valores das variáveis avaliadas são obtidos a partir de (3.14). O cenário de todas as simulações é o mesmo considerado na subseção 3.2.1. A figura de mérito utilizada para avaliar o desempenho do algoritmo IPNLMS é o desalinhamento normalizado em dB dado por (2.106).

Para avaliar o impacto de $\alpha$ no comportamento do algoritmo IPNLMS, a Figura 3.6 mostra curvas de desalinhamento para o problema de identificação da planta esparsa $\mathbf{p}$, considerando diversos valores para o parâmetro $\alpha$, que vão de $-1$ até $0{,}99$. O parâmetro de passo adotado é $\mu = 0{,}5$. Note das curvas da Figura 3.6 que o valor escolhido para o parâmetro de proporcionalidade impacta diretamente na velocidade de convergência durante todo o processo de adaptação do algoritmo IPNLMS.

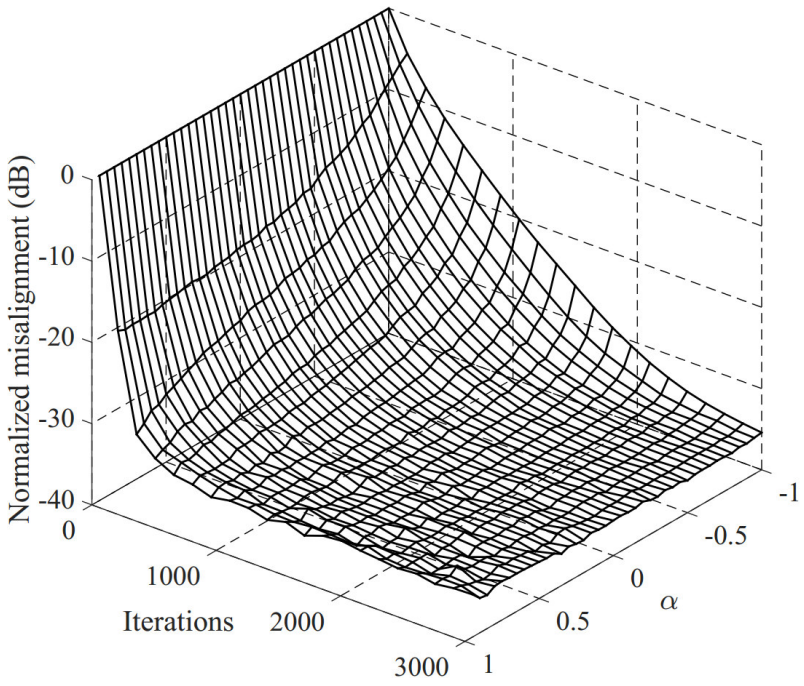



Figura 3.6. Desempenho do algoritmo IPNLMS para a identificação da planta esparsa $\mathbf{p}$, considerando diversos valores para $\alpha$ que vão de $-1$ a $0{,}99$, e um parâmetro de passo igual a $\mu = 0{,}5$.

A Figura 3.7 mostra o ganho de adaptação atribuído ao $i$-ésimo coeficiente em função da sua respectiva magnitude (normalizada), considerando valores de $\alpha$ iguais a $-0{,}9$, $-0{,}5$, $0{,}0$, $0{,}3$, $0{,}7$ e $0{,}99$. As magnitudes normalizadas dos coeficientes do filtro adaptativo são obtidas a partir de (3.18). As curvas da Figura 3.7 mostram que, quanto mais próximo de 1 for o parâmetro de proporcionalidade, mais proporcional é a atualização dos coeficientes do filtro.

Em contrapartida, para um valor de $\alpha$ mais próximo de $-1$, mais plana ou igualitária é a distribuição dos ganhos de adaptação entre os coeficientes do filtro adaptativo, a despeito da magnitude dos mesmos. Note também das curvas da Figura 3.7 que, independentemente do valor escolhido para o parâmetro de proporcionalidade, $\alpha$, o processo de adaptação ocorre sem que ocorra estagnação dos coeficientes com magnitude muito inferior à do maior coeficiente.

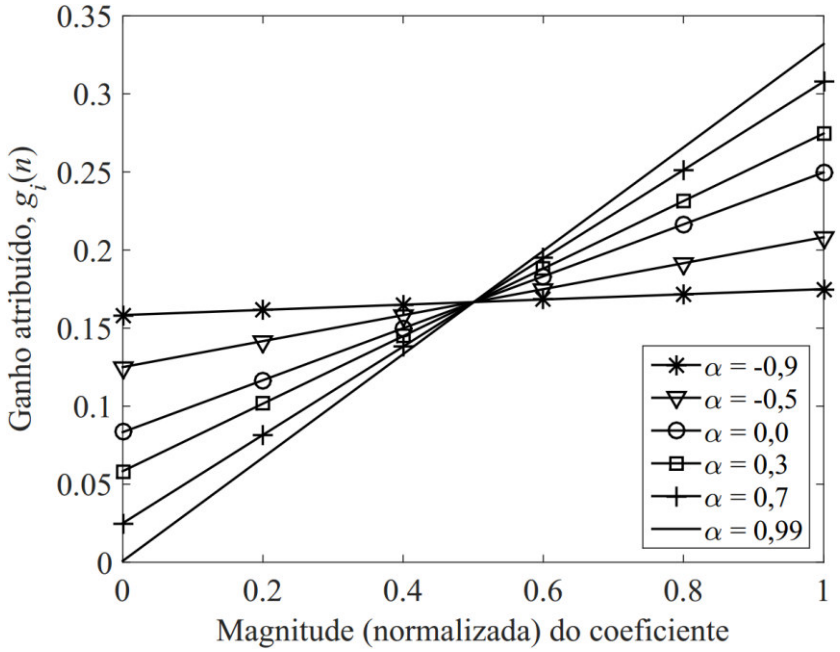



Figura 3.7. Ganho atribuído pelo algoritmo IPNLMS aos coeficientes de um filtro conforme a magnitude (normalizada) dos mesmos, considerando valores de α iguais a $-0{,}9$, $-0{,}5$, 0,0, 0,3, 0,7 e 0,99.

Os resultados mostrados nas Figuras 3.6 e 3.7 demonstram que um procedimento para otimizar a escolha do parâmetro de proporcionalidade, $\alpha$, pode melhorar o desempenho do algoritmo IPNLMS para a identificação de plantas esparsas.

Sabe-se que fatores como temperatura e pressão podem alterar a esparsidade de uma planta (Loganathan, Khong & Naylor, 2009). Um valor de $\alpha$ constante pode comprometer o desempenho do algoritmo IPNLMS ao lidar com uma planta de esparsidade variável. Além disso, o valor deste parâmetro deve ser escolhido de acordo com o grau de esparsidade da planta. Dessa forma, um valor inapropriado para tal parâmetro pode degradar o desempenho do IPNLMS. Para tentar contornar este problema, algoritmos tais como o IIPNLMS e o SC-IPNLMS foram desenvolvidos. Estes e outros algoritmos derivados dos algoritmos PNLMS e IPNLMS são mostrados na seção seguinte.

## 3.4 Outros Algoritmos da Classe PNLMS

Nesta seção são apresentados alguns dos algoritmos adaptativos proporcionais derivados dos algoritmos PNLMS e IPNLMS. Tais algoritmos foram desenvolvidos com o intuito de alcançar um desempenho superior ao dos algoritmos PNLMS e IPNLMS na identificação e controle de plantas esparsas. São eles os algoritmos PNLMS++, IIPNLMS, MPNLMS, SC-PNLMS, SC-IPNLMS, SC-MPNLMS e IAF-PNLMS.

*3.4.1 Algoritmo PNLMS++*

O algoritmo PNLMS++ foi introduzido por Steven L. Gay em 1998 e consiste de uma combinação dos algoritmos PNLMS e NLMS (Gay, 1998). Especificamente, o algoritmo PNLMS++ alterna, em cada iteração do processo de adaptação, entre as regras de adaptação dos algoritmos PNLMS e NLMS. O sumário do algoritmo PNLMS++ é mostrado na Tabela 3.6.

Tabela 3.6. Sumário do algoritmo adaptativo PNLMS++.

1. Inicialização

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

e parâmetros

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

2. Obtenção dos dados de entrada e saída

da planta

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

e do filtro adaptativo

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

3. Cálculo do sinal de erro

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

4. Se $n$ é ímpar:
   a. Fator de ativação
   $$f(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_\infty]$$
   b. Função de proporcionalidade
   $$\phi_i(n) = \max[f(n), |w_i(n)|], \qquad i = 1,2,3,\dots,N$$
   c. Ganho individual dos coeficientes do filtro adaptativo
   $$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N} \phi_j(n)}, \qquad i = 1,2,3,\dots,N$$
   d. Matriz de ganhos individuais ($N$ x $N$)
   $$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \dots \quad g_N(n)]$$

   Caso contrário, ou seja, se $n$ é par:
   a. Matriz de ganhos individuais ($N$ x $N$)
   $$\mathbf{G}(n) = \mathbf{I}$$

5. Atualização dos coeficientes do filtro adaptativo

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

A complexidade computacional e os requisitos de memória do algoritmo PNLMS++ correspondem, obviamente, às mesmas do algoritmo PNLMS (Gay, 1998). Uma implementação alternativa do algoritmo PNLMS++ consiste em alternar, a cada $I$ iterações, entre as regras de atualização dos algoritmos NLMS e PNLMS (Wagner & Doroslovacky, 2013).

### *3.4.2 Algoritmo IIPNLMS*

O algoritmo IIPNLMS foi proposto em 2004 com o objetivo principal de aprimorar o desempenho do algoritmo IPNLMS para problemas de cancelamento de eco de rede. Tal algoritmo busca explorar a característica esparsa dos caminhos de eco de rede, segmentando a resposta impulsiva destas plantas em regiões "ativas" e "inativas" (Cui, Naylor & Brown, 2004). Enquanto que, para o algoritmo IPNLMS, o parâmetro de proporcionalidade, $\alpha$, é o mesmo para todos os coeficientes do filtro adaptativo, para o algoritmo IIPNLMS tem-se um parâmetro de proporcionalidade variável definido por

$$\alpha_{i,\mathrm{IIPN}}(n) = \begin{cases} \alpha_1, & \text{se } \phi_i(n) > \gamma_{\mathrm{IIPN}} * \max[\boldsymbol{\phi}(n)] \\ \alpha_2, & \text{se } \phi_i(n) < \gamma_{\mathrm{IIPN}} * \max[\boldsymbol{\phi}(n)] \end{cases}, \qquad i = 1,2,3,\dots,N \quad (3.25)$$

onde

$$\phi_i(n) = \max[f(n), |w_i(n)|], \qquad i = 1,2,3,\dots,N \quad (3.26)$$

e

$$f(n) = \rho \max[\delta, \|\mathbf{w}(n)\|_\infty] \quad (3.27)$$

são a função de proporcionalidade e o fator de ativação do algoritmo PNLMS. Os possíveis valores que o parâmetro de proporcionalidade variável, $\alpha_{i,\mathrm{IIPN}}(n)$, do algoritmo IIPNLMS estão limitados em $-1 \leq \alpha_{1,2} < 1$. As variáveis $\rho$ e $\delta$ são os parâmetros de proporcionalidade e de inicialização, respectivamente. O vetor

$$\boldsymbol{\phi}(n) = [\phi_1(n)\ \ \phi_2(n) \dots\ \phi_N(n)] \quad (3.28)$$

é o vetor das funções de proporcionalidade dos coeficientes do filtro adaptativo na $n$-ésima iteração do processo de adaptação. A variável $\gamma_{\mathrm{IIPN}}$ é usada para controlar a fronteira entre as regiões de coeficientes ativos e inativos. Os ganhos de adaptação do algoritmo IIPNLMS são calculados a partir de

$$g_i(n) = \frac{\phi_i'(n)}{\frac{1}{N}\sum_{i=1}^{N}\phi_i'(n)}, \qquad i = 1,2,3,\dots,N \quad (3.29)$$

onde

$$\phi_i'(n) = \frac{1-\alpha_{i,\mathrm{IIPN}}(n)}{2} + \frac{1+\alpha_{i,\mathrm{IIPN}}(n)}{2}\phi_i(n), \qquad i = 1,2,3,\dots,N. \quad (3.30)$$

O algoritmo IIPNLMS apresenta-se como uma combinação ponderada dos algoritmos NLMS e PNLMS de forma que a parcela referente ao PNLMS exerce maior influência na adaptação dos coeficientes da região "inativa", enquanto que a parcela referente ao NLMS é

responsável, principalmente, pela convergência dos coeficientes da região "ativa". A Tabela 3.7 contém o sumário do algoritmo IIPNLMS.

Tabela 3.7. Sumário do algoritmo adaptativo IIPNLMS.

1. Inicialização

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

e parâmetros

$$0 < \mu < 2 \quad ; \quad \varepsilon > 0 \quad ; \quad -1 \leq \alpha_{1,2} < 1$$

2. Obtenção dos dados de entrada e saída

da planta

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

e do filtro adaptativo

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

3. Cálculo do sinal de erro

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

4. Fator de ativação

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

5. Função de proporcionalidade

$$\phi_i(n) = \max[f(n), |w_i(n)|], \qquad i = 1,2,3,\dots,N$$

6. Parâmetro de proporcionalidade variável

$$\alpha_{i,\mathrm{IIPN}}(n) = \begin{cases} \alpha_1, & \text{se } \phi_i(n) > \gamma_{\mathrm{IIPN}} * \max[\boldsymbol{\phi}(n)] \\ \alpha_2, & \text{se } \phi_i(n) < \gamma_{\mathrm{IIPN}} * \max[\boldsymbol{\phi}(n)] \end{cases}, \qquad i = 1,2,3,\dots,N$$

7. Ganho individual dos coeficientes do filtro adaptativo

$$\phi_i'(n) = \frac{1 - \alpha_{i,\mathrm{IIPN}}(n)}{2} + \frac{1 + \alpha_{i,\mathrm{IIPN}}(n)}{2}\phi_i(n), \qquad i = 1,2,3,\dots,N$$

$$g_i(n) = \frac{\phi_i'(n)}{\frac{1}{N}\sum_{i=1}^{N}\phi_i'(n)}, \qquad i = 1,2,3,\dots,N$$

8. Matriz de ganhos individuais ($N$ x $N$)

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

9. Atualização dos coeficientes do filtro adaptativo

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

A partir da abordagem proposta pelo algoritmo IIPNLMS, é possível explorar as vantagens de ambos os algoritmos NLMS e PNLMS em relação às características específicas de cada região (Cui, Naylor & Brown, 2004). A seguir é apresentado o algoritmo MPNLMS.

*3.4.3 Algoritmo MPNLMS*

O algoritmo MPNLMS foi desenvolvido em 2005 e possui uma regra de atualização que distribui os ganhos de adaptação de forma proporcional ao logaritmo da magnitude dos coeficientes do filtro adaptativo (Deng & Doroslovacky, 2005) e (Deng & Doroslovacky, 2006). O fator de ativação e a função de proporcionalidade do algoritmo MPNLMS são dados respectivamente por

$$f(n) = \rho \max[\delta, \ln(1+\beta|w_1(n)|), \ln(1+\beta|w_2(n)|), \dots, \ln(1+\beta|w_N(n)|)] \quad (3.31)$$

e

$$\phi_i(n) = \max[f(n), \ln(1+\beta|w_i(n)|)] \quad (3.32)$$

onde $\beta = 1/\kappa$, com $\kappa$ sendo um número positivo e pequeno. O *bias* unitário dos logaritmos de (3.31) e (3.32) evita instabilidade durante o período de inicialização, ou seja, quando $\mathbf{w}(0) = \mathbf{0}$ (Loganathan, Khong & Naylor, 2009). O sumário deste algoritmo é mostrado na Tabela 3.8.

Tabela 3.8. Sumário do algoritmo adaptativo MPNLMS.

1. Inicialização
   $$\mathbf{w}(0) = \mathbf{0}$$
   e parâmetros
   $$0 < \mu < 2 \quad ; \quad \delta > 0 \quad ; \quad \rho > 0 \quad ; \quad \varepsilon > 0$$
2. Obtenção dos dados de entrada e saída
   da planta
   $$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n)$$
   e do filtro adaptativo
   $$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$$
3. Cálculo do sinal de erro
   $$e(n) = d(n) - y(n) + z(n)$$
4. Fator de ativação
   $$f(n) = \rho \max[\delta, \ln(1+\beta|w_1(n)|), \dots, \ln(1+\beta|w_N(n)|)]$$
5. Função de proporcionalidade
   $$\phi_i(n) = \max[f(n), \ln(1+\beta|w_i(n)|)], \qquad i = 1,2,3,\dots,N$$
6. Ganho individual dos coeficientes do filtro adaptativo
   $$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N}\phi_j(n)}, \qquad i = 1,2,3,\dots,N$$
7. Matriz de ganhos individuais ($N$ x $N$)
   $$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \dots \quad g_N(n)]$$
8. Atualização dos coeficientes do filtro adaptativo
   $$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu\mathbf{G}(n)e(n)\mathbf{x}(n)}{\mathbf{x}^{\mathrm{T}}\mathbf{G}(n)\mathbf{x}(n) + \varepsilon}$$

A regra de atualização do algoritmo MPNLMS baseia-se no princípio da determinação dos ganhos de adaptação ótimos para cada coeficiente do filtro adaptativo (Deng & Doroslovacky, 2005). Tal algoritmo pode ser visto como uma derivação do algoritmo SD proporcional, uma vez busca manter a rápida velocidade inicial de convergência do algoritmo PNLMS durante todo o processo de adaptação (Deng & Doroslovacky, 2006). A velocidade de convergência uniforme é uma característica do algoritmo SD (Farhang-Boroujeny, 2013) e (Haykin, 1996).

### *3.4.4 Algoritmos SC-PNLMS, SC-IPNLMS e SC-MPNLMS*

Os algoritmos SC-PNLMS, SC-IPNLMS e SC-MPNLMS constituem uma classe de algoritmos proporcionais de esparsidade controlada (*sparseness-controlled proportionate algorithms*). Tais algoritmos foram desenvolvidos com o objetivo de aprimorar as características de convergência dos algoritmos PNLMS, IPNLMS e MPNLMS para plantas de esparsidade variável (Khong & Naylor, 2006), (Loganathan, Khong & Naylor, 2008) & (Loganathan, Khong & Naylor, 2009).

Sabe-se, de (1.2), que o grau de esparsidade de uma planta pode ser medido conforme a definição (Hoyer, 2004)

$$S(\mathbf{p}) = \frac{N}{\sqrt{N} - N}\left(1 - \frac{\|\mathbf{p}\|_1}{\sqrt{N}\|\mathbf{p}\|_2}\right) \tag{3.33}$$

onde $\mathbf{p}$ é o vetor de coeficientes da planta que deseja-se identificar. Como $0 \leq S(\mathbf{p}) \leq 1$, quanto mais próximo de zero for $S(\mathbf{p})$, mais esparsa será a planta em questão. Já que $\mathbf{p}$ não é conhecido durante o processo de adaptação, uma alternativa viável consiste em usar uma estimativa da esparsidade da planta baseada no vetor de coeficientes, $\mathbf{w}(n)$, do filtro adaptativo

$$\hat{S}[\mathbf{w}(n)] = \frac{N}{\sqrt{N} - N}\left(1 - \frac{\|\mathbf{w}(n)\|_1}{\sqrt{N}\|\mathbf{w}(n)\|_2}\right). \tag{3.34}$$

Os algoritmos SC-PNLMS, SC-IPNLMS e SC-MPNLMS são concebidos empregando (3.34) nas regas de atualização dos algoritmos PNLMS, IPNLMS e MPNLMS, respectivamente. Para os algoritmos SC-PNLMS e SC-MPNLMS, $\hat{S}[\mathbf{w}(n)]$ é aplicada no cálculo do parâmetro de proporcionalidade, $\rho$, da seguinte forma

$$\rho(n) = e^{-\lambda \hat{S}[\mathbf{w}(n)]} \tag{3.35}$$

onde $\lambda$ é uma constante positiva não-nula e $\rho(n)$ é o parâmetro de proporcionalidade da $n$-ésima iteração do processo de adaptação do algoritmo SC-PNLMS ou SC-MPNLMS. A escolha do fator $\lambda$ impacta diretamente na velocidade de convergência dos algoritmos SC-PNLMS e SC-MPNLMS, uma vez que este fator influencia diretamente no valor do parâmetro de proporcionalidade, $\rho(n)$. Resultados experimentais demonstram que 4,0, 6,0 e 8,0 são boas escolhas para tal fator (Loganathan, Khong & Naylor, 2009).

Como geralmente tem-se $\mathbf{w}(0) = \mathbf{0}$, para os algoritmos SC-PNLMS e SC-MPNLMS, a estimativa de esparsidade da planta dada por (3.34) só deve ser calculada para $n \geq N$, de forma

a evitar instabilidade uma vez que $\|\mathbf{w}(0)\|_2 = 0$. Assim, para os algoritmos SC-PNLMS e SC-MPNLMS, tem-se (Loganathan, Khong & Naylor, 2009)

$$\rho(n) = \frac{5}{N}\,, \qquad n < N. \tag{3.36}$$

O sumário dos algoritmos SC-PNLMS e SC-MPNLMS é mostrado na Tabela 3.9.

Tabela 3.9. Sumário dos algoritmos adaptativos SC-PNLMS e SC-MPNLMS.

1. Inicialização

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

   e parâmetros

   $0 < \mu < 2 \quad ; \quad \delta > 0 \quad ; \quad \rho > 0 \quad ; \quad \varepsilon > 0 \quad ; \quad \lambda > 0$

2. Dados de saída da planta e do filtro adaptativo

   $$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \quad \text{e} \quad y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$$

3. Cálculo do sinal de erro

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

4. Estimativa do grau de esparsidade da planta

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

5. Parâmetro de proporcionalidade

   $$\rho(n) = \begin{cases} \dfrac{5}{N}\,, & n < N \\ e^{-\lambda \hat{S}[\mathbf{w}(n)]}, & n \geq N \end{cases}$$

6. Fator de ativação e função de proporcionalidade

   i) Para o algoritmo SC-PNLMS

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

   $$\phi_i(n) = \max[f(n), |w_i(n)|]\,, \qquad i = 1,2,3,\ldots,N$$

   ii) Para o algoritmo SC-MPNLMS

   $$f(n) = \rho(n)\max[\delta, \ln(1 + \beta|w_1(n)|)\,, \ldots, \ln(1 + \beta|w_N(n)|)]$$

   $$\phi_i(n) = \max[f(n), \ln(1 + \beta|w_i(n)|)]\,, \qquad i = 1,2,3,\ldots,N$$

7. Ganho individual dos coeficientes do filtro adaptativo

   $$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N}\phi_j(n)}\,, \qquad i = 1,2,3,\ldots,N$$

8. Matriz de ganhos individuais ($N$ x $N$)

   $$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \ldots \quad g_N(n)]$$

9. Atualização dos coeficientes do filtro adaptativo

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

O algoritmo SC-IPNLMS emprega $\hat{S}[\mathbf{w}(n)]$ no cálculo dos ganhos de adaptação individuais do algoritmo IPNLMS (Khong & Naylor, 2006)

$$g_i(n) = \left\{\frac{1 - 0{,}5\hat{S}[\mathbf{w}(n)]}{N}\right\}\frac{(1-\alpha_{\mathrm{SC}})}{2N} + \left\{\frac{1 - 0{,}5\hat{S}[\mathbf{w}(n)]}{N}\right\}\frac{(1+\alpha_{\mathrm{SC}})|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varsigma} \quad (3.37)$$

onde $\alpha_{\mathrm{SC}}$ é o parâmetro de proporcionalidade do algoritmo SC-IPNLMS. Como resultado de (3.36), tem-se que o algoritmo SC-IPNLMS varia a contribuição dos termos proporcional e não-proporcional de acordo com a estimativa do grau de esparsidade da planta. Diferentemente do algoritmo IPNLMS, o qual usa apenas um parâmetro de proporcionalidade, $\alpha$, o qual é constante durante todo o processo de adaptação.

Analogamente ao caso dos algoritmos SC-PNLMS e SC-MPNLMS, para evitar instabilidade na estimativa da esparsidade da planta durante o início do processo de adaptação, adota-se para o algoritmo SC-IPNLMS (Khong & Naylor, 2006)

$$g_i(n) = \frac{(1-\alpha_{\mathrm{SC}})}{2N} + \frac{(1+\alpha_{\mathrm{SC}})|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varsigma}, \qquad n < N. \quad (3.38)$$

A Tabela 3.10 contém o sumário do algoritmo SC-IPNLMS.

Tabela 3.10. Sumário do algoritmo adaptativo SC-IPNLMS.

1. Inicialização e parâmetros

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

$$0 < \mu < 2 \quad ; \quad \varepsilon > 0 \quad ; \quad -1 \le \alpha < 1 \quad ; \quad \varsigma > 0$$

2. Dados de saída da planta e do filtro adaptativo

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \quad \text{e} \quad y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$$

3. Cálculo do sinal de erro

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

4. Estimativa do grau de esparsidade da planta

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

5. Ganho individual dos coeficientes do filtro adaptativo ($i = 1,2,3,\ldots,N$)

   i) Para $n < N$

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

   ii) Para $n \ge N$

$$g_i(n) = \left\{\frac{1 - 0{,}5\hat{S}[\mathbf{w}(n)]}{N}\right\}\frac{(1-\alpha_{\mathrm{SC}})}{2N} + \left\{\frac{1 - 0{,}5\hat{S}[\mathbf{w}(n)]}{N}\right\}\frac{(1+\alpha_{\mathrm{SC}})|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varsigma}$$

6. Matriz de ganhos individuais ($N$ x $N$)

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

7. Atualização dos coeficientes do filtro adaptativo

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

A seguir, é apresentado o algoritmo IAF-PNLMS.

*3.4.5 Algoritmo IAF-PNLMS*

Introduzido em 2010, o algoritmo IAF-PNLMS propõe o emprego de fatores de ativação individuais para cada coeficiente do filtro adaptativo (Souza, Tobias, Seara & Morgan, 2010). Tais fatores são determinados da seguinte maneira

$$f_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN \\ f_i(n-1), & \text{caso contrário} \end{cases} \tag{3.39}$$

com $i = 1,2,\dots,N$ e $m = 1,2,3,\dots$. A variável $f_i(n)$ é o fator de ativação do $i$-ésimo coeficiente do filtro na $n$-ésima iteração do processo de adaptação. Dessa forma, as funções de proporcionalidade para o algoritmo IAF-PNLMS são calculadas a partir de

$$\phi_i = \max[f_i(n), |w_i(n)|], \qquad i = 1,2,3,\dots,N. \tag{3.40}$$

Note de (3.39) e (3.40) que o algoritmo IAF-PNLMS requer uma inicialização do vetor $\mathbf{f}(n) = [f_1(n), f_2(n), \dots, f_N(n)]^{\mathrm{T}}$ de fatores de ativação individuais dos coeficientes do filtro. Normalmente, $\mathbf{f}(n)$ é inicializado com

$$f_i(0) = c, \qquad i = 1,2,\dots,N \tag{3.41}$$

onde $c$ é uma constante pequena e positiva. Resultados experimentais demonstram que $10^{-2}/N$ é uma boa escolha para tal constante (Souza, Tobias, Seara & Morgan, 2010).

O algoritmo IAF-PNLMS possui duas características interessantes que podem ser observadas em (3.39) e (3.40). A primeira delas está relacionada à memória dos fatores de ativação individuais, a qual está associada aos valores dos respectivos coeficientes do filtro. Note de (3.39) que o $i$-ésimo fator de ativação, $f_i(n)$, contém informação herdada da magnitude do $i$-ésimo coeficiente. A segunda diz respeito à convergência dos fatores de ativação individuais. Para demonstrar tal característica, assume-se convergência e apresenta-se as seguintes propriedades para o $i$-ésimo fator de ativação, $f_i(n)$, e para a $i$-ésima função de proporcionalidade, dados respectivamente por (3.39) e (3.40):

(P1) Se o $i$-ésimo coeficiente, $w_i(n)$, é ativo, tem-se, no instante $n = mN$

$$f_i(mN) = \frac{1}{2}|w_i(mN-1)| + \frac{1}{2}|w_i(mN)| \tag{3.42}$$

e

$$\phi_i(mN) = \max\left\{\frac{1}{2}[|w_i(mN-1)| + |w_i(mN)|], |w_i(mN)|\right\}. \tag{3.43}$$

(P2) Se o $i$-ésimo coeficiente, $w_i(n)$, é inativo, tem-se, no instante $n = mN$

$$f_i(n) = \frac{1}{2^m}f_i(0) + \frac{1}{2^m}|w_i(N)| + \frac{1}{2^{m-1}}|w_i(2N)| + \cdots$$
$$\cdots + \frac{1}{2^2}|w_i[(m-1)N]| + \frac{1}{2}|w_i(mN)| \tag{3.44}$$

e

$$\phi_i(mN) = \max\left\{\frac{1}{2^m} f_i(0) + \frac{1}{2^m}|w_i(N)| + \cdots + \frac{1}{2}|w_i(mN)|, |w_i(mN)|\right\} \quad (3.45)$$

Portanto, de P1) e P2) verifica-se que, conforme $m$ aumenta, $|w_i(mN-1)| \rightarrow |w_i(mN)|$ e, consequentemente, $\phi_i(mN) \rightarrow |w_i(mN)|$, para ambos coeficientes ativos e inativos. Desta forma, tem-se que $g_i(n)$ é proporcional a $|w_i(n)|$, independentemente se $w_i(n)$ for ativo ou inativo. Como resultado, tem-se para o algoritmo IAF-PNLMS que o $i$-ésimo fator de ativação, $f_i(n)$, converge para a magnitude do respectivo coeficiente, $|w_i(n)|$ (Souza, Tobias, Seara & Morgan, 2010)

$$\lim_{n\to\infty}[f_i(n) - |w_i(n)|], \qquad i = 1,2,\ldots,N. \quad (3.46)$$

Note de (3.45) que $\phi_i(mN)$ é sempre maior do que zero. Isto evita estagnação dos coeficientes do filtro adaptativo durante o processo de adaptação. Apesar de proporcionar um aumento na velocidade de convergência geral do algoritmo PNLMS, o algoritmo IAF-PNLMS provoca uma diminuição na velocidade de convergência dos coeficientes de menor magnitude. Isto se dá pela transferência dos de adaptação dos coeficientes inativos para os coeficientes ativos que ocorre no algoritmo IAF-PNLMS (Wagner & Doroslovacky, 2013). O sumário do algoritmo IAF-PNLMS é mostrado na Tabela 3.11.

Tabela 3.11. Sumário do algoritmo adaptativo IAF-PNLMS.

1. Inicialização
   $\mathbf{w}(0) = \mathbf{0} \quad ; \quad \boldsymbol{\phi}(0) = \mathbf{0} \quad ; \quad f_i(0) = c, \qquad i = 1,2,\ldots,N$
   e parâmetros
   $$0 < \mu < 2 \quad ; \quad \varepsilon > 0$$
2. Dados de saída da planta e do filtro adaptativo
   $$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \quad \text{e} \quad y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n)$$
3. Cálculo do sinal de erro
   $$e(n) = d(n) - y(n) + z(n)$$
4. Fatores de ativação individuais ($i = 1,2,\ldots,N$)
   $$f_i(n) = \begin{cases} \frac{1}{2}|w_i(n)| + \frac{1}{2}\phi_i(n-1), & n = mN \\ f_i(n-1), & \text{caso contrário} \end{cases}$$
5. Função de proporcionalidade
   $$\phi_i = \max[f_i(n), |w_i(n)|], \qquad i = 1,2,3,\ldots,N$$
6. Ganho individual dos coeficientes do filtro adaptativo
   $$g_i(n) = \frac{\phi_i(n)}{\sum_{j=1}^{N}\phi_j(n)}, \qquad i = 1,2,3,\ldots,N$$
7. Matriz de ganhos individuais ($N$ x $N$)
   $$\mathbf{G}(n) = \mathrm{diag}[g_1(n) \quad g_2(n) \quad \ldots \quad g_N(n)]$$
8. Atualização dos coeficientes do filtro adaptativo
   $$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu\mathbf{G}(n)e(n)\mathbf{x}(n)}{\mathbf{x}^{\mathrm{T}}\mathbf{G}(n)\mathbf{x}(n) + \varepsilon}$$

O principal objetivo do algoritmo IAF-PNLMS é obter uma adaptação verdadeiramente proporcional. Isto se dá pelo emprego de fatores de ativação individuais, em contraste com o algoritmo PNLMS, o qual possui ganhos de adaptação proporcionais apenas para os coeficientes considerados ativos (vide Figuras 3.3 e 3.5).

# Capítulo 4

# Métodos de Otimização

Frequentemente, profissionais de diversos ramos tais como engenheiros, gerentes de fábrica, planejadores e pesquisadores em geral lidam com problemas que necessitam de algum tipo de otimização. Tais problemas podem envolver, por exemplo, planejamento e alocação de recursos e mão de obra limitados para atingir um dado resultado, ou mesmo escolha da melhor combinação de valores e parâmetros que impactam na operação de um determinado dispositivo ou processo (Bazaraa, Sherali & Shetty, 2006). Para solucionar este tipo de problema, foram desenvolvidos métodos de otimização os quais são baseados nos mais diversos princípios matemáticos e filosofias de busca. O principal objetivo destes métodos é propor uma solução elegante e bem direcionada para o problema que se deseja solucionar. Em linhas gerais, os métodos de otimização podem ser entendidos como ferramentas de conceptualização e análise que buscam a solução correta para um dado problema. Tais métodos operam com base em uma gama de princípios que definem uma metodologia de execução (Luenberger & Ye, 2010).

Métodos clássicos de otimização empregam ferramentas de cálculo diferencial e conceitos envolvendo funções de uma ou várias variáveis. Exemplos de métodos clássicos incluem os baseados em gradiente e o método de Newton. Tais métodos são ditos de otimização local uma vez que, dependendo do ponto de partida e das direções de descida, podem levar a busca para ótimos locais, ou seja, pontos de máximo ou de mínimo que não são necessariamente os maximizadores ou minimizadores da função objetivo (Luenberguer & Ye, 2008). Os chamados métodos de otimização global empregam, em sua grande maioria, algoritmos de busca inspirados em situações encontradas na natureza. Exemplos de métodos de busca global incluem algoritmos genéticos, otimização por enxame de partículas (*particle swarm optimization*), lógica *fuzzy*, busca tabu, dentre outros (Parlados, Du & Graham, 2013). Estes métodos realizam buscas com certo grau de aleatoriedade apoiadas por mecanismos e estratégias de direcionamento que evitam estagnação em ótimos locais.

Este capítulo tem o objetivo de apresentar alguns conceitos fundamentais envolvendo teoria de otimização, além dos métodos de otimização empregados neste trabalho para otimizar o desempenho dos algoritmos adaptativos proporcionais para a identificação de plantas esparsas. Primeiramente, são mostradas as formas de modelagem e caracterização de problemas de otimização. Logo após, são discutidas as condições para identificação e verificação das soluções ótimas para um dado problema de otimização, também chamadas de condições de otimalidade. Posteriormente, são apresentados alguns dos principais métodos de otimização local e global desenvolvidos nas últimas décadas. Por fim, os métodos adotados por este

trabalho para otimizar a escolha dos parâmetros dos algoritmos PNLMS e IPNLMS são apresentados.

## 4.1 Caracterização de um Problema de Otimização

Problemas de otimização podem ser representados por modelos matemáticos tipicamente constituídos por três elementos:

1. Conjunto de variáveis de decisão: contém os parâmetros cujos respectivos valores levam a uma dada solução para o problema.
2. Função de custo: é uma função das variáveis de decisão do problema que se deseja minimizar ou maximizar.
3. Restrições: conjunto de funções que definem o espaço de busca das possíveis soluções para o problema. Soluções obtidas a partir de valores de variáveis de decisão que não pertencem ao conjunto de restrições devem ser descartadas.

A busca pela melhor solução deve então ser pautada pela determinação do minimizador ou maximizador global da função de custo.

Considere uma dada função de custo, $f(\mathbf{s})$, definida em $\mathbb{R}^m \rightarrow \mathbb{R}$, tal que $\mathbf{s}$ é um vetor de $m$ componentes, ou seja, $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_m]^{\mathrm{T}}$. Um problema geral de otimização pode ser caracterizado por uma notação matemática tal como

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{s}) \\ \text{sujeito a} \quad & h_i(\mathbf{s}) \leq 0, \quad i = 1,2,\dots,p \\ & l_j(\mathbf{s}) = 0, \quad j = 1,2,\dots,q \end{aligned} \tag{4.1}$$

onde o objetivo em (4.1) é encontrar um vetor $\mathbf{s} \in \mathbb{R}^m$ que minimize $f(\mathbf{s})$ e que satisfaça as condições $h_i(\mathbf{s}) \leq 0$, com $i = 1,2,\dots,p$ e $l_j(\mathbf{s}) = 0$, com $j = 1,2,\dots,q$. O vetor $\mathbf{s}$ é formado pelas variáveis de decisão $s_1$, $s_2$, ... , $s_m$ e as funções $f$, $h_i$ e $l_j$ são definidas em $\mathbb{R}^m \rightarrow \mathbb{R}$. As inequações $h_i(\mathbf{s}) \leq 0$, com $i = 1,2,\dots,p$, e as equações as equações $l_j(\mathbf{s}) = 0$, com $j = 1,2,\dots,q$, são chamadas de restrições de desigualdade e de igualdade, respectivamente. As variáveis $p$ e $q$ representam a quantidade de restrições de desigualdade e igualdade, respectivamente. Um problema que possua restrições de igualdade ou de desigualdade (ou ambas) é dito de otimização restrita ou com restrições. Em contrapartida, um problema descrito da forma

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{s}) \\ \text{sujeito a} \quad & \mathbf{s} \in \mathbb{R}^m \end{aligned} \tag{4.2}$$

ou seja, sem restrições de igualdade ou desigualdade, é chamado de problema de otimização irrestrita (Luenberger & Ye, 2010). Um problema de otimização restrita também pode ser representado da forma

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{s}) \\ \text{sujeito a} \quad & \mathbf{s} \in X \end{aligned} \tag{4.3}$$

ou

$$\mathbf{s}^* = \arg\min_{\mathbf{s} \in X} f(\mathbf{s}) \tag{4.4}$$

onde $X$ é um subconjunto de $\mathbb{R}^m$. Tal conjunto, comumente chamado de região factível, é constituído pela intersecção entre as restrições de igualdade e desigualdade. Cada elemento $\mathbf{s} \in X$ é chamado de solução factível. Obviamente, qualquer solução $\mathbf{s} \notin X$ deve ser descartada, uma vez que não atende às restrições do problema.

Uma das mais importantes características de um problema de otimização está relacionada com o caráter do conjunto de soluções factíveis. Especificamente, problemas de otimização podem ser classificados como contínuos ou discretos. Problemas contínuos são aqueles nos quais o conjunto $X$ possui infinitos elementos, ou seja, apresenta um caráter contínuo. Em contrapartida, problemas discretos são aqueles nos quais o conjunto $X$ possui um número finito de elementos. Exemplos típicos de problemas discretos são problemas de otimização combinatória, tais como planejamento de rotas, alocação ótima de recursos e mão de obra, dentre outros (Bertsekas, 1999).

### *4.1.1 Busca Unidimensional e Multidimensional*

Outra característica importante de problemas de otimização está relacionada com a quantidade de variáveis de decisão. Especificamente, quando $m = 1$, ou seja, quando o conjunto de soluções factíveis, $X$, é constituído por elementos do tipo $\mathbf{s} \in \mathbb{R}$, tem-se um problema de otimização unidimensional. Neste caso, a função de custo, $f$, e as restrições $h_i$, com $i = 1,2, \ldots, p$, e $l_j$, com $j = 1,2, \ldots, q$, são todas definidas em $\mathbb{R} \rightarrow \mathbb{R}$. Uma alternativa para resolver este tipo de problema é o uso de métodos iterativos de busca unidimensional, também chamados de métodos de busca linear. Tais métodos de busca têm como ponto de partida uma solução candidata inicial, $\mathbf{s}^0$, e a cada iteração, $k$, com $k = 0,1,2, \ldots$, é gerada uma nova solução $\mathbf{s}^k$ candidata à ótima do problema. A solução candidata da próxima iteração, $\mathbf{s}^{k+1}$, depende da solução da iteração anterior, $\mathbf{s}^k$, e de informações relacionadas com a função de custo, $f$. Podem ser usados valores de $f$ em pontos específicos ou talvez a primeira e segunda derivadas ($f'$ e $f''$, respectivamente) da função de custo na busca por novas soluções. Exemplos de métodos de busca unidimensional incluem os métodos da razão áurea, de Fibonacci, da bisseção, da secante, dentre outros (Bazaraa, Sherali & Shetty, 2006) e (Luenberguer & Ye, 2010). Cada método possui critérios de convergência próprios que definem quando a busca deve ser interrompida.

Métodos de busca linear são de particular interesse para o ramo da otimização, pois podem ser vistos como casos especiais de métodos de otimização multidimensional, os quais são aplicados em problemas que possuem mais de uma variável de decisão ($m > 1$). Além disso, um método de busca unidimensional pode ser empregado como parte integrante de métodos de busca multidimensional (Chong & Zak, 2013). Em particular, algoritmos usados por métodos iterativos de busca multidimensional geralmente realizam uma busca linear em cada iteração. Como exemplo, considere uma função de custo $f$, definida em $\mathbb{R}^m \rightarrow \mathbb{R}$, que se

deseja minimizar. O algoritmo usado por um método de busca multidimensional para determinação do minimizador de $f$ possui a forma

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \mu_k \Delta \mathbf{s}^k \tag{4.5}$$

onde o vetor $\Delta \mathbf{s}^k$ é chamado de direção de busca e $\mu_k > 0$ é o parâmetro de passo. Dado um ponto inicial, $\mathbf{s}^0$, deve-se escolher $\mu_k$ de forma que a função

$$o(\mu) = f(\mathbf{s}^k + \mu \Delta \mathbf{s}^k) \tag{4.6}$$

seja minimizada. Dessa forma, um método de busca unidimensional, tal como o método da razão áurea ou o método da bisseção, é usado para determinar o valor adequado para o parâmetro de passo, $\mu_k$, de (4.5). Uma condição apropriada para tal escolha é

$$f(\mathbf{s}^{k+1}) < f(\mathbf{s}^k). \tag{4.7}$$

A seguir são discutidos os requisitos necessários para que uma dada solução possa ser considerada uma minimizadora (ou maximizadora) local ou global de um problema de otimização. Tais requisitos são chamados de condições de otimalidade.

## 4.2 Condições de Otimalidade

Tendo posse do modelo matemático para o problema que se deseja solucionar, verifica-se que o termo "otimizar" consiste basicamente em encontrar o conjunto de valores para as variáveis de decisão que minimizam (ou maximizam) a função de custo. Isto deve ocorrer sem que nenhuma das restrições de igualdade ou desigualdade do problema sejam violadas (Hagan, Demuth, Beale & de Jesus, 2015). Porém, antes de realizar a busca pelo minimizador (ou maximizador) da função de custo, deve-se definir os requisitos que uma determinada solução candidata deve atender para ser considerada como ótima. Para tal, considere uma função de custo, $f(\mathbf{s})$, definida em $\mathbb{R}^m \to \mathbb{R}$, tal que $\mathbf{s} \in \mathbb{R}^m$. Diz-se que $\mathbf{s}^* \in X$ é um mínimo local de $f$ se existe $\vartheta > 0$ tal que

$$f(\mathbf{s}^*) \leq f(\mathbf{s}), \qquad \forall \mathbf{s} \text{ com } \|\mathbf{s} - \mathbf{s}^*\| < \vartheta \tag{4.8}$$

onde $\|\cdot\|$ representa a norma euclidiana do vetor em questão. Analogamente, $\mathbf{s}^*$ é dito um máximo local de $f$ se

$$f(\mathbf{s}^*) \geq f(\mathbf{s}), \qquad \forall \mathbf{s} \text{ com } \|\mathbf{s} - \mathbf{s}^*\| < \vartheta. \tag{4.9}$$

O vetor $\mathbf{s}^*$ é dito mínimo global de $f$ se

$$f(\mathbf{s}^*) \leq f(\mathbf{s}), \qquad \forall \mathbf{s} \in \mathbb{R}^m. \tag{4.10}$$

Analogamente, $\mathbf{s}^*$ é dito máximo global de $f$ se

$$f(\mathbf{s}^*) \geq f(\mathbf{s}), \qquad \forall \mathbf{s} \in \mathbb{R}^m. \tag{4.11}$$

Em ambos os casos descritos em (4.10) e (4.11), $\mathbf{s}^*$ é denominado solução ótima do problema. A função de custo $f$ é comumente chamada de função objetivo (Bazaraa, Sherali & Shetty, 2006). Em particular, se $\mathbf{s}^*$ é um mínimo local (ou global) de $f$, tem-se também que $\mathbf{s}^*$ é um mínimo local (ou global) de $-f$ (Bertsekas, 1999). Para verificar se uma dada solução $\mathbf{s}^* \in X$ atende os requisitos de (4.8) e (4.9), pode-se fazer uso de ferramentas de cálculo tais

com expansões em série de Taylor e vetores gradiente. Então, com o auxílio de tais ferramentas, pode-se definir algumas condições que $\mathbf{s}^*$ deve atender para ser considerada um ótimo local de $f$. Estas condições são apresentadas a seguir.

*4.2.1 Condições Necessárias de Otimalidade*

Caso a função objetivo $f: \mathbb{R}^m \to \mathbb{R}$ seja duplamente diferenciável em, é possível usar ferramentas como gradientes, expansões em série de Taylor e matrizes Hessianas para verificar se uma dada solução candidata é mais vantajosa em relação a outras localizadas em uma vizinhança próxima. Especificamente, considerando pequenas variações, $\Delta\mathbf{s}$, de um dado vetor $\mathbf{s}^*$, tem-se que a variação no valor de função objetivo em função de $\Delta\mathbf{s}$ pode ser estimada a partir uma aproximação de primeira ordem dada por

$$f(\mathbf{s}^* + \Delta\mathbf{s}) - f(\mathbf{s}^*) \approx \nabla f(\mathbf{s}^*)^{\mathrm{T}} \Delta\mathbf{s} \tag{4.12}$$

onde o termo $\nabla f(\mathbf{s}^*)$ é o vetor gradiente de $f$ calculado no ponto $\mathbf{s}^*$. A variação no valor de função objetivo em função de $\Delta\mathbf{s}$ também pode ser estimada a partir de uma aproximação de segunda ordem dada por

$$f(\mathbf{s}^* + \Delta\mathbf{s}) - f(\mathbf{s}^*) \approx \nabla f(\mathbf{s}^*)^{\mathrm{T}} \Delta\mathbf{s} + \frac{1}{2} \Delta\mathbf{s}^{\mathrm{T}} \nabla^2 f(\mathbf{s}^*) \Delta\mathbf{s} \tag{4.13}$$

onde o termo $\nabla^2 f(\mathbf{s}^*)$ é a matriz Hessiana de $f$ calculada no ponto $\mathbf{s}^*$. Caso $\mathbf{s}^*$ seja um mínimo local irrestrito de $f$, a variação de primeira ordem dada por (4.12) é não-negativa, ou seja

$$\nabla f(\mathbf{s}^*)^{\mathrm{T}} \Delta\mathbf{s} = \sum_{i=1}^{m} \frac{\partial f(\mathbf{s}^*)}{\partial s_i} \Delta s_i \geq 0 \,. \tag{4.14}$$

Como $\Delta\mathbf{s}$ é arbitrário, para que $\mathbf{s}^*$ seja um minimizador local de $f$, deve-se ter

$$\frac{\partial f(\mathbf{s}^*)}{\partial s_i} \geq 0 \tag{4.15}$$

para $i = 1,2, \ldots, m$. Em outras palavras, a taxa de variação de $f$ em $\mathbf{s}^*$ em qualquer direção $\Delta\mathbf{s}$ deve ser não-negativa. Logo, tem-se

$$\nabla f(\mathbf{s}^*) = 0 \,. \tag{4.16}$$

O resultado de (4.16) corresponde à condição de otimalidade necessária de primeira ordem.

Considerando agora a aproximação de segunda ordem de (4.13), para que $\mathbf{s}^*$ seja um mínimo local irrestrito de $f$, também deve-se ter uma variação não-negativa no valor de função objetivo em função de $\Delta\mathbf{s}$, ou seja

$$\nabla f(\mathbf{s}^*)^{\mathrm{T}} \Delta\mathbf{s} + \frac{1}{2} \Delta\mathbf{s}^{\mathrm{T}} \nabla^2 f(\mathbf{s}^*) \Delta\mathbf{s} \geq 0 \,. \tag{4.17}$$

Uma vez que, de (4.14) e (4.16), tem-se

$$\nabla f(\mathbf{s}^*)^{\mathrm{T}} \Delta\mathbf{s} = 0 \,. \tag{4.18}$$

Então, (4.17) passa a ser constituída somente pela forma quadrática

$$\Delta\mathbf{s}^{\mathrm{T}}\nabla^2 f(\mathbf{s}^*)\Delta\mathbf{s} \geq 0\,. \tag{4.19}$$

Logo, $\nabla^2 f(\mathbf{s}^*)$ deve ser positiva semidefinida em qualquer direção $\Delta\mathbf{s}$. Esta definição corresponde à condição de otimalidade necessária de segunda ordem.

Um candidato a mínimo local irrestrito, $\mathbf{s}^*$, de $f$ precisa atender as condições de otimalidade descritas por (4.16) e (4.19). Porém, tais condições não são suficientes para que $\mathbf{s}^*$ seja realmente um mínimo local de $f$. As condições suficientes de otimalidade são apresentadas a seguir.

### *4.2.2 Condições Suficientes de Otimalidade*

Considere um vetor $\mathbf{s}^* \in X$ que satisfaz a condição necessária de primeira ordem

$$\nabla f(\mathbf{s}^*) = 0 \tag{4.20}$$

e que também satisfaz a condição necessária de segunda ordem da seguinte forma

$$\Delta\mathbf{s}^{\mathrm{T}}\nabla^2 f(\mathbf{s}^*)\Delta\mathbf{s} > 0 \tag{4.21}$$

ou seja, a Hessiana de $f$ é positiva definida (ao invés de positiva semidefinida) em $\mathbf{s}^*$ para qualquer direção $\Delta\mathbf{s}$. Então, tem-se que o valor de $f$ tende a crescer para pequenas trajetórias partindo de $\mathbf{s}^*$. Logo, as condições descritas por (4.20) e (4.21) são suficientes para que $\mathbf{s}^*$ seja um minimizador local de $f$.

Mínimos locais que não satisfazem (4.20) e (4.21) são chamados de singulares. Caso contrário, são chamados de não-singulares. A otimalidade de mínimos singulares não pode ser verificada se $f$ não possuir caráter convexo. Além disso, o comportamento de métodos de otimização clássicos na vizinhança deste tipo de mínimo tende a ser lento e possivelmente errático. Tais características fazem dos mínimos singulares os mais difíceis de serem analisados (Bertsekas, 1999).

### *4.2.3 Condições de Karush-Kuhn-Tucker*

As condições necessárias e suficientes de otimalidade apresentadas nas subseções 4.2.1 e 4.2.2 são aplicáveis a problemas gerais de otimização não-linear irrestrita. Entretanto, para problemas de otimização com restrições, uma solução candidata a ótima local (ou global) deve atender a outra série de requisitos (Chong & Zak, 2013). Para verificar tais condições, considere o seguinte problema de otimização restrita

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{s}) \\ \text{sujeito a} \quad & h_i(\mathbf{s}) \leq 0, \quad i = 1,2,\dots,p \\ & l_j(\mathbf{s}) = 0, \quad j = 1,2,\dots,q \end{aligned} \tag{4.22}$$

onde $f(\mathbf{s})$, $h_i(\mathbf{s}) \leq 0$, com $i = 1,2,\dots,p$, e $l_j(\mathbf{s}) = 0$, com $j = 1,2,\dots,q$, são a função objetivo e as restrições de desigualdade e igualdade, respectivamente, todas definidas em $\mathbb{R}^m \to \mathbb{R}$. O vetor $\mathbf{s}$ pertence ao conjunto de soluções factíveis definido por

$$X = \{\mathbf{s} \in \mathbb{R}^m \mid h_i(\mathbf{s}) \leq 0, i = 1,2,\dots,p \text{ e } l_i(\mathbf{s}) = 0, j = 1,2,\dots,q\}. \quad (4.23)$$

Uma restrição de desigualdade, $h_i(\mathbf{s}) \leq 0$, é dita ativa no ponto $\mathbf{s}^*$ se

$$h_i(\mathbf{s}^*) = 0 \quad (4.24)$$

caso contrário, $h_i(\mathbf{s})$ é dita inativa em $\mathbf{s}^*$. Por convenção, uma restrição de igualdade $l_j(\mathbf{s}) = 0$ é sempre considerada como sendo ativa para qualquer $\mathbf{s} \in X$. O conjunto das restrições de desigualdade ativas em $\mathbf{s}^*$ é definido como

$$J(\mathbf{s}^*) \triangleq \{ i \mid h_i(\mathbf{s}^*) = 0\}. \quad (4.25)$$

Diz-se que $\mathbf{s}^*$ é um ponto regular se os vetores $\nabla h_i(\mathbf{s}^*)$, com $i \in J(\mathbf{s}^*)$, e $\nabla l_j(\mathbf{s}^*)$, com $j = 1,2,\dots,q$, são linearmente independentes.

A função Lagrangeana (ou simplesmente o Lagrangeano) para o problema de (4.22) é dada por

$$\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = f(\mathbf{s}) + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(\mathbf{s}) + \boldsymbol{\sigma}^{\mathrm{T}}\mathbf{l}(\mathbf{s}) \quad (4.26)$$

onde $\boldsymbol{\lambda} = \left[\lambda_1\ \lambda_2\ \dots \lambda_p\right]^{\mathrm{T}}$ e $\boldsymbol{\sigma} = \left[\sigma_1\ \sigma_2\ \dots\ \sigma_q\right]^{\mathrm{T}}$ são vetores constituídos pelos multiplicadores de Lagrange, e $\mathbf{h}(\mathbf{s}) = \left[h_1(\mathbf{s})\ h_2(\mathbf{s})\ \dots h_p(\mathbf{s})\right]^{\mathrm{T}}$ e $\mathbf{l}(\mathbf{s}) = \left[l_1(\mathbf{s})\ l_2(\mathbf{s})\ \dots l_q(\mathbf{s})\right]^{\mathrm{T}}$ são vetores contendo as restrições de desigualdade e igualdade, respectivamente.

Com base nas definições de (4.24) a (4.26), pode-se estabelecer as condições Karush-Kuhn-Tucker (KKT) necessárias de primeira ordem. Especificamente, considerando que $f(\mathbf{s})$, $\mathbf{h}(\mathbf{s})$ e $\mathbf{l}(\mathbf{s})$ sejam diferenciáveis em $\mathbb{R}^m$, e que $\mathbf{s}^*$ é um ponto regular e um minimizador local do problema dado por (4.22), então existem $\boldsymbol{\lambda} \in \mathbb{R}^p$ e $\boldsymbol{\sigma} \in \mathbb{R}^q$ que satisfazem as seguintes condições

$$\begin{aligned} \nabla_{\mathbf{s}}\mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) &= \mathbf{0} \\ \boldsymbol{\lambda} &\geq \mathbf{0} \\ \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(\mathbf{s}^*) &= 0 \\ \mathbf{h}(\mathbf{s}^*) &\leq \mathbf{0} \\ \mathbf{l}(\mathbf{s}^*) &= \mathbf{0} \end{aligned} \quad (4.27)$$

onde $\nabla_{\mathbf{s}}\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\sigma})$ é o vetor gradiente do Lagrangeano do problema em relação a $\mathbf{s}$. Pela condição de complementaridade, $\boldsymbol{\lambda}^{\mathrm{T}}\mathbf{h}(\mathbf{s}^*) = 0$, tem-se

$$\lambda_i = 0, \qquad \forall h_i(\mathbf{s}^*) < 0 \quad (4.28)$$

ou seja, os operadores de Lagrange relacionados às restrições de desigualdade inativas são nulos. As condições KKT necessárias de primeira ordem dadas por (4.27) referem-se a um problema de minimização semelhante ao de (4.22). Para o caso de um problema de maximização dado por

$$\begin{aligned} &\text{maximizar} \quad f(\mathbf{s}) \\ &\text{sujeito a} \quad h_i(\mathbf{s}) \leq 0, \qquad i = 1,2,\dots,p \end{aligned} \quad (4.29)$$

$$l_j(\mathbf{s}) = 0, \qquad j = 1,2,\dots,q$$

estas mesmas condições podem ser aplicadas se a função objetivo, $f(\mathbf{s})$, a ser maximizada for multiplicada por $-1$.

Para definir as condições KKT necessárias de segunda ordem, considere a matriz

$$\nabla^2 \mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \boldsymbol{\sigma}) = \nabla^2 f(\mathbf{s}) + \sum_{i=1}^{p} \lambda_i \nabla^2 h_i(\mathbf{s}) + \sum_{j=1}^{q} \sigma_j \nabla^2 l_j(\mathbf{s}) \tag{4.30}$$

onde $\nabla^2 h_i(\mathbf{s})$ e $\nabla^2 l_j(\mathbf{s})$ são as matrizes Hessianas de $h_i(\mathbf{s})$ e $l_j(\mathbf{s})$, respectivamente. Dessa forma, considerando que $f(\mathbf{s})$, $\mathbf{h}(\mathbf{s})$ e $\mathbf{l}(\mathbf{s})$ sejam duplamente diferenciáveis em $\mathbb{R}^m$, e que $\mathbf{s}^*$ é um ponto regular e um minimizador local do problema de minimização dado por (4.22), então existem $\boldsymbol{\lambda} \in \mathbb{R}^p$ e $\boldsymbol{\sigma} \in \mathbb{R}^q$ tais que

$$\begin{aligned} \nabla_{\mathbf{s}} \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) &= \mathbf{0} \\ \boldsymbol{\lambda} &\geq \mathbf{0} \\ \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{h}(\mathbf{s}^*) &= 0 \end{aligned} \tag{4.31}$$

e que, para todo $\mathbf{y}$ definido em

$$T(\mathbf{s}^*) = \left\{ \mathbf{y} \in \mathbb{R}^m \;\middle|\; \nabla h_i(\mathbf{s}^*)^{\mathrm{T}} \mathbf{y} = 0, \;\; \nabla l_j(\mathbf{s}^*)^{\mathrm{T}} \mathbf{y} = 0, \;\; i \in J(\mathbf{s}^*) \right\} \tag{4.32}$$

tem-se

$$\mathbf{y}^{\mathrm{T}} \nabla_{\mathbf{s}}^2 \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) \mathbf{y} \geq 0. \tag{4.33}$$

As condições KKT necessárias de segunda ordem são dadas por (4.31) e (4.33).

Para definir as condições KKT suficientes de segunda ordem, considere o conjunto

$$\tilde{T}(\mathbf{s}^*, \boldsymbol{\lambda}) = \left\{ \mathbf{y} \mid \nabla h_i(\mathbf{s}^*)^{\mathrm{T}} \mathbf{y} = 0, \;\; \nabla l_j(\mathbf{s}^*)^{\mathrm{T}} \mathbf{y} = 0, \;\; i \in \tilde{J}(\mathbf{s}^*, \boldsymbol{\lambda}) \right\} \tag{4.34}$$

onde

$$\tilde{J}(\mathbf{s}^*, \boldsymbol{\lambda}) = \{ i \mid h_i(\mathbf{s}^*) = 0, \;\; \lambda_i > 0 \}. \tag{4.35}$$

Note que

$$\tilde{J}(\mathbf{s}^*, \boldsymbol{\lambda}) \subset J(\mathbf{s}^*). \tag{4.36}$$

Logo, tem-se que

$$T(\mathbf{s}^*) \subset \tilde{T}(\mathbf{s}^*, \boldsymbol{\lambda}). \tag{4.37}$$

Dessa forma, considerando que $f(\mathbf{s})$, $\mathbf{h}(\mathbf{s})$ e $\mathbf{l}(\mathbf{s})$ sejam duplamente diferenciáveis em $\mathbb{R}^m$, e que existam uma solução factível, $\mathbf{s}^* \in \mathbb{R}^m$, para o problema de minimização dado por (4.22), e vetores $\boldsymbol{\lambda} \in \mathbb{R}^p$ e $\boldsymbol{\sigma} \in \mathbb{R}^q$ tais que

$$\begin{aligned} \nabla_{\mathbf{s}} \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) &= \mathbf{0} \\ \boldsymbol{\lambda} &\geq \mathbf{0} \\ \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{h}(\mathbf{s}^*) &= 0 \end{aligned} \tag{4.38}$$

e que, para todo $\mathbf{y} \in \tilde{T}(\mathbf{s}^*, \boldsymbol{\lambda})$, tenha-se

$$\mathbf{y}^{\mathrm{T}} \nabla_{\mathbf{s}}^2 \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) \mathbf{y} > 0 \tag{4.39}$$

então $\mathbf{s}^*$ é um minimizador local de $f(\mathbf{s})$. Analogamente, considerando um problema de maximização semelhante ao de (4.29), tem-se que $\mathbf{s}^*$ é um maximizador local de $f(\mathbf{s})$ se existirem $\boldsymbol{\lambda} \in \mathbb{R}^p$ e $\boldsymbol{\sigma} \in \mathbb{R}^q$ tais que

$$\begin{aligned} \nabla_{\mathbf{s}} \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) &= \mathbf{0} \\ \boldsymbol{\lambda} &\leq \mathbf{0} \\ \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{h}(\mathbf{s}^*) &= 0 \end{aligned} \tag{4.40}$$

e que para todo $\mathbf{y} \in \tilde{T}(\mathbf{s}^*, \boldsymbol{\lambda})$ tenha-se

$$\mathbf{y}^{\mathrm{T}} \nabla_{\mathbf{s}}^2 \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma}) \mathbf{y} < 0 \tag{4.41}$$

ou seja, se os multiplicadores de Lagrange referentes às restrições de desigualdade forem não-positivos e se $\nabla_{\mathbf{s}}^2 \mathcal{L}(\mathbf{s}^*, \boldsymbol{\lambda}, \boldsymbol{\sigma})$ for negativa definida em $\tilde{T}(\mathbf{s}^*, \boldsymbol{\lambda})$.

Para que as condições de otimalidade apresentadas acima possam ser verificadas é necessário que a função objetivo e as restrições sejam diferenciáveis. Este requisito é atendido por funções de caráter suave. Uma função suave é multiplamente diferenciável em qualquer ponto de seu domínio. Tal característica garante à função um comportamento previsível, pelo menos localmente, permitindo que algoritmos usados por métodos de busca escolham corretamente as direções de busca (Sun & Yuan, 2006). Tipicamente, fronteiras não-suaves do conjunto de soluções factíveis podem ser descritas por um conjunto de restrições suaves. Porém, para alguns problemas cuja função objetivo e restrições possuem caráter não-suave, tais como alguns problemas de otimização combinatória, não é possível realizar tais adaptações.

Métodos de otimização clássicos, tais como o método de descida mais íngreme (SD – *steepest descent method*) e o método de Newton, operam satisfatoriamente em problemas cuja função objetivo e respectivas restrições são suaves. Tais métodos usam ferramentas de cálculo como derivadas e matrizes Hessianas, e podem ser entendidos como métodos de busca local (Chong & Zak, 2013). Em contrapartida, métodos de busca aleatória direcionada e/ou inspirados em situações encontradas na natureza, tais como os algoritmos genéticos, a busca tabu e o recozimento simulado (*simulated annealing*), demonstram-se mais adequados para lidar com problemas de natureza discreta ou cujas funções envolvidas são de caráter não-suave. Estes métodos, frequentemente chamados de métodos de otimização global, usam apenas valores de função objetivo de diversos pontos pertencentes ou não ao conjunto de soluções factíveis. A seguir, são apresentados alguns dos principais métodos de busca local e global empregados na resolução de problemas de otimização.

## 4.3 Métodos de Otimização Local

Métodos clássicos de otimização, tais como o método SD e o método de Newton, fazem uso de algoritmos iterativos cujo objetivo é minimizar uma função objetivo $f\colon \mathbb{R}^m \to \mathbb{R}$. A ideia

fundamental de tais algoritmos reside em realizar uma busca de descida iterativa (*iterative descent*) que funciona de acordo com os seguintes passos:

1. A busca tem como ponto de partida uma estimativa inicial $\mathbf{s}^0 \in \mathbb{R}^m$.
2. A cada nova iteração, $k$, é gerado um novo vetor, $\mathbf{s}^k \in \mathbb{R}^m$, de forma que o valor de $f$ decresça iterativamente, ou seja

$$f(\mathbf{s}^{k+1}) < f(\mathbf{s}^k), \qquad k = 0,1,2,\dots \tag{4.42}$$

onde $\mathbf{s}^k$ e $\mathbf{s}^{k+1}$ são as estimativas das iterações anterior e atual, respectivamente. Desta forma, a estimativa da solução ótima é sucessivamente melhorada ao passo que $f$ caminha passo a passo em direção a um ponto de mínimo. Um algoritmo de busca empregado por um método clássico de otimização pode ser descrito por

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \mu_k \Delta\mathbf{s}^k \tag{4.43}$$

onde $\mu_k > 0$ e $\Delta\mathbf{s}^k \neq \mathbf{0}$ são, respectivamente, o parâmetro de passo e a direção de busca adotados. O parâmetro $\mu_k$ é tipicamente escolhido com o auxílio de um método de busca linear, tal como o método da razão áurea ou o método de Fibonacci, ou segundo um conjunto de princípios que garantem estabilidade e convergência do método. A direção de busca, $\Delta\mathbf{s}^k$, adotada varia conforme o método empregado (Bazaraa, Sherali & Shetty, 2006). Note que o desempenho dos métodos de busca local depende principalmente da estimativa inicial e da direção de busca escolhida (Chong & Zak, 2013). Esta seção apresenta alguns dos principais métodos de otimização local desenvolvidos nas últimas décadas. O primeiro deles é o método SD.

### *4.3.1 Método de Descida mais Íngreme*

O método SD é um dos mais simples e fundamentais métodos de minimização empregado em problemas de otimização irrestrita. Também comumente chamado de método do gradiente, este método caracteriza-se principalmente por adotar o valor negativo do gradiente da função objetivo como direção de descida (Sun & Yuan, 2006).

O gradiente de uma função, $f\colon \mathbb{R}^m \to \mathbb{R}$, em um dado ponto, $\mathbf{s}^m \in \mathbb{R}$, dado por

$$\nabla f(\mathbf{s}) = \begin{bmatrix} \dfrac{\partial f(\mathbf{s})}{\partial s_1} \\ \dfrac{\partial f(\mathbf{s})}{\partial s_2} \\ \vdots \\ \dfrac{\partial f(\mathbf{s})}{\partial s_m} \end{bmatrix} \tag{4.44}$$

onde $\partial f(\mathbf{s})/\partial s_i$ é a derivada parcial de $f$ em relação a $s_i$, com $i = 1,2,\dots,m$, pode ser definido como sendo o vetor que indica a direção da máxima taxa de crescimento de $f$ a partir de $\mathbf{s}$. Logo, a direção de máximo decréscimo de $f$ em $\mathbf{s}$ será $-\nabla f(\mathbf{s})$. Assim, o algoritmo que descreve o método SD pode ser descrito por

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \mu_k \nabla f(\mathbf{s}^k) \tag{4.45}$$

onde $\nabla f(\mathbf{s}^k)$ é o vetor gradiente de $f$ na estimativa da iteração atual $k$. Sabe-se que o método SD possui a seguinte propriedade (Chong & Zak, 2013):

$$f(\mathbf{s}^{k+1}) < f(\mathbf{s}^k) \tag{4.46}$$

se

$$\nabla f(\mathbf{s}^k) \neq \mathbf{0}. \tag{4.47}$$

Então, caso exista algum ponto $\mathbf{s}^k$ tal que $\nabla f(\mathbf{s}^k) = \mathbf{0}$, tem-se que $\mathbf{s}^k$ satisfaz a condição de otimalidade necessária de primeira ordem. Neste caso, $\mathbf{s}^{k+1} = \mathbf{s}^k$. Logo, um critério de parada ou de convergência para o algoritmo SD é

$$\nabla f(\mathbf{s}^{k+1}) = \mathbf{0}. \tag{4.48}$$

Porém, em aplicações práticas, a condição (4.48) não é viável pelo fato de que o cômputo numérico do gradiente raramente será igual a zero. Uma alternativa para a definição de tal critério pode ser dada por

$$|f(\mathbf{s}^{k+1}) - f(\mathbf{s}^k)| < \vartheta \tag{4.49}$$

onde $\vartheta > 0$ é um limiar previamente especificado, geralmente escolhido como sendo $0 < \vartheta \ll 1$. Este limiar também é comumente chamado de tolerância. Outro critério de convergência adequado é dado por

$$\|\mathbf{s}^{k+1} - \mathbf{s}^k\| < \vartheta. \tag{4.50}$$

A seguir é apresentado o método de Newton.

### *4.3.2 Método de Newton*

O método SD usa apenas derivadas de primeira ordem para minimizar a função objetivo. Porém, tal estratégia nem sempre é a mais efetiva. Caso derivadas de ordem superior sejam usadas, o método de busca resultante pode alcançar um desempenho superior ao do método SD (Chong & Zak, 2013).

A ideia básica do método de Newton consiste em minimizar de forma iterativa a aproximação quadrática da função objetivo em um dado ponto. Tal método usa derivadas de primeira e segunda ordem da função objetivo e, caso o ponto inicial escolhido estiver próximo de um minimizador local, o método de Newton alcança de fato um desempenho superior ao do método SD (Bazaraa, Sherali & Shetty, 2006).

Considere uma função objetivo $f(\mathbf{s})\colon \mathbb{R} \to \mathbb{R}$. A aproximação quadrática desta função em um dado ponto $\mathbf{s}^k \in \mathbb{R}$ é dada por

$$v = f(\mathbf{s}^k) + f'(\mathbf{s}^k)(\mathbf{s} - \mathbf{s}^k) + \frac{1}{2} f''(\mathbf{s}^k)(\mathbf{s} - \mathbf{s}^k)^2 \tag{4.51}$$

onde $f'(\mathbf{s}^k)$ e $f''(\mathbf{s}^k)$ são as derivadas de primeira e segunda ordem de $f$ em $\mathbf{s}^k$, respectivamente. Assim, toma-se $\mathbf{s}^{k+1}$ como sendo o ponto onde a derivada de $v$ é igual a zero. Logo, tem-se

$$f'(\mathbf{s}^k) + f''(\mathbf{s}^k)(\mathbf{s}^{k+1} - \mathbf{s}^k) = 0 \tag{4.52}$$

o que leva a

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \frac{f'(\mathbf{s}^k)}{f''(\mathbf{s}^k)}. \tag{4.53}$$

O procedimento iterativo dado por (4.53) possui velocidade de convergência superior à do método SD. Para uma função objetivo $f(\mathbf{s})$: $\mathbb{R}^m \to \mathbb{R}$, a aproximação quadrática de (4.51) passa a ser

$$v = f(\mathbf{s}^k) + \nabla f(\mathbf{s}^k)(\mathbf{s} - \mathbf{s}^k) + \frac{1}{2}\nabla^2 f(\mathbf{s}^k)(\mathbf{s} - \mathbf{s}^k)^2. \tag{4.54}$$

Aplicando a condição de otimalidade necessária de primeira ordem em (4.54), tem-se

$$\begin{aligned} \nabla v &= \mathbf{0} \\ &= \nabla f(\mathbf{s}^k) + \nabla^2 f(\mathbf{s}^k)(\mathbf{s} - \mathbf{s}^k). \end{aligned} \tag{4.55}$$

Assumindo que $\nabla^2 f(\mathbf{s}^k)$ seja positiva definida, tem-se que o mínimo de $v$ é alcançado em

$$\mathbf{s}^{k+1} = \mathbf{s}^k - [\nabla^2 f(\mathbf{s}^k)]^{-1} \nabla f(\mathbf{s}^k). \tag{4.56}$$

Inserindo um parâmetro de passo, $\mu_k$, no termo de correção de (4.56) obtém-se a regra recursiva do método de Newton

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \mu_k [\nabla^2 f(\mathbf{s}^k)]^{-1} \nabla f(\mathbf{s}^k). \tag{4.57}$$

Para o critério de convergência do método de Newton, pode-se adotar as mesmas condições para o método SD dadas por (4.49) ou (4.50). Outro critério de convergência aplicável ao método de Newton é

$$\|\nabla f(\mathbf{s}^{k+1})\| < \vartheta. \tag{4.58}$$

Note que, para $\mu_k = 1$, o método de Newton encontra o mínimo global de uma função quadrática positiva definida em uma única iteração. O método de Newton converge mais rapidamente do que o método SD. Porém, a carga computacional requerida pelo método de Newton é significativamente superior ao do método SD. Enquanto que o método SD requer apenas o cálculo do gradiente, o método de Newton também exige o cômputo da inversa da matriz Hessiana da função objetivo em cada iteração (Bertsekas, 1996).

## 4.4 Métodos de Otimização Global

Os métodos de otimização apresentados na seção anterior convergem, na melhor das hipóteses, para um ótimo local. Portanto, quando do uso de tais métodos, é desejável que o ponto de partida esteja próximo de um minimizador global. Além disso, há a necessidade de cálculos de derivadas de primeira ou mesmo de segunda ordem. Daí o porquê de serem chamados de métodos de otimização local. Outra desvantagem de métodos clássicos, tais como o método SD ou o de Newton, é que estes não operam adequadamente quando perturbações aleatórias (tal como um ruído de medição) são impostas à função objetivo (Fogel, 1994).

Em contrapartida, os métodos apresentados nessa seção são considerados de otimização global, uma vez que buscam percorrer todo o conjunto de soluções factíveis. Tais métodos não requerem cálculo de derivadas já que só usam valores de função objetivo. Como consequência disso, podem ser empregados em uma maior classe de problemas, incluindo otimização combinatória, problemas discretos e envolvendo funções de caráter não-suave (Chong & Zak, 2013). Outra vantagem dos métodos de otimização global é que estes também podem ser usados como ferramentas para escolha adequada do ponto de partida de um método de busca local. A seguir são apresentados dois métodos de otimização cujas técnicas de busca são inspiradas em situações encontradas na natureza. São eles, respectivamente, os algoritmos genéticos e os métodos de otimização por enxame de partículas.

### *4.4.1 Algoritmos Genéticos*

Evolução natural é um processo de otimização baseado em populações. Algoritmos evolutivos vêm sendo empregados em vários problemas de engenharia e ciência da computação. As primeiras ideias relacionadas com a computação evolutiva surgiram na década de 1950 com a proposta de implementar o processo de evolução natural em computador para resolver problemas. A principal vantagem destes algoritmos reside no uso de uma representação matemática das variáveis de interesse do sistema alvo. Isto confere robustez e flexibilidade para lidar com problemas de auto grau de complexidade, seguindo um conjunto de procedimentos genéricos capazes de serem adaptados para atuar em vários cenários.

A computação evolutiva engloba uma família de algoritmos inspirados na teoria da evolução das espécies de Darwin. Historicamente, foram desenvolvidos independentemente três tipos de algoritmos evolutivos (Fogel, 1994) e (Back, 1996):

- Algoritmos genéticos: desenvolvidos por John Henry Holland no final da década de 1960, com o intuito de propor uma caracterização matemática para processos de adaptação presentes em sistemas naturais, além de desenvolver sistemas artificiais computacionais com mecanismos semelhantes aos dos processos naturais (Holland, 1975).
- Programação evolutiva: proposta inicialmente por Lawrence J. Fogel na década de 1960 como uma técnica para criar inteligência artificial a partir da evolução de máquinas de estado finito (Fogel, 1964).
- Estratégias evolutivas: propostas por Schwefel e Rechenberg para solucionar problemas de otimização de parâmetros (Schwefel, 1965) e (Rechenberg, 1973).

Os algoritmos genéticos pertencem à classe das técnicas baseadas em inteligência artificial ou metaheurísticas. Tais algoritmos baseiam-se na codificação genética para simular o processo de evolução. A grande vantagem deste método está na relativa simplicidade, robustez, flexibilidade e capacidade de encontrar a solução ótima global. Isto se dá pela aplicação de procedimentos de busca aleatória direcionada com características não lineares e com múltiplos picos e descontinuidades. Dessa forma, o espaço de soluções factíveis é percorrido com certa aleatoriedade, porém com um objetivo bem definido. Graças a estas

peculiaridades, os algoritmos genéticos evitam convergência em ótimos locais. Sua generalidade e simplicidade também lhes confere utilidade destacada na resolução de problemas onde outros métodos de otimização encontram dificuldades.

Todavia, a maior desvantagem dos algoritmos genéticos é o elevado tempo de processamento. Por ser um método aleatório de base evolutiva (e a evolução é naturalmente um processo lento), a demora para encontrar a solução ótima é quase inevitável. Uma codificação eficiente das soluções e a redução do espaço de busca podem ajudar a atenuar este problema. Os algoritmos genéticos também podem atuar em conjunto com outras técnicas, formando um algoritmo híbrido que proporcione uma maior eficiência computacional e robustez.

Um algoritmo genético básico é construído a partir de uma população inicial aleatória e da execução iterativa de um ciclo composto por três estágios:

- Avaliação de cada indivíduo da população;
- Seleção de indivíduos para reprodução, e;
- Manipulação genética para criar uma nova população.

Cada vez que este ciclo for completado, diz-se que uma geração ocorreu. Cada um dos indivíduos avaliados durante o processo evolutivo também é chamado de cromossomo (*string*). Cada cromossomo possui genes em diferentes posições (*locus*) e cada gene contém informações acerca das características do indivíduo. A relação entre essas características e o meio no qual os indivíduos estão inseridos é medida pelo índice de adaptação (*fitness function*). Tal índice está diretamente relacionado com a função objetivo do problema.

Após a criação da população inicial aleatória e avaliação de seus respectivos indivíduos, ocorre a evolução das populações em gerações com a ajuda de três operadores:

- Reprodução: cópia de alguns indivíduos para a geração futura em razão do seu índice de adaptação.
- Cruzamento: atua sobre um par de indivíduos escolhidos aleatoriamente, combinando suas características (também de forma aleatória) para formar um novo indivíduo que fará parte da próxima geração.
- Mutação: modificação aleatória dos valores contidos nos genes, ou seja, das características de um dado indivíduo.

Cruzamento e mutação são os dois operadores fundamentais de um algoritmo genético. Enquanto que os cruzamentos aceleram o processo de evolução, as mutações produzem um tipo de perturbação que pode levar o processo evolutivo a um melhor domínio de busca. O percentual de cruzamentos e mutações é definido por índices probabilísticos específicos de cada operador. Apenas uma parcela dos indivíduos de uma geração é escolhida para o cruzamento, e também um pequeno número dos novos indivíduos gerados sofre mutação.

Considere o seguinte problema de otimização que deseja-se solucionar usando um algoritmo genético

$$\begin{aligned} \text{maximizar} \quad & f(\mathbf{s}) \\ \text{sujeito a} \quad & \mathbf{s} \in X \end{aligned} \tag{4.59}$$

onde $f: \mathbb{R}^m \to \mathbb{R}$. Note que (4.59) corresponde a uma maximização, já que este tipo de problema é mais conveniente para descrever o funcionamento dos algoritmos genéticos (Chong & Zak, 2013). Dessa forma, um algoritmo genético para resolver (4.59) pode ser descrito da seguinte forma:

1. Faça $k = 0$. Determine um índice de adaptação que possua relação direta com $f(\mathbf{s})$ e gere uma população inicial aleatória $P^k$.
2. Avalie os indivíduos de $P^k$.
3. Se o critério de convergência (ou de parada) for satisfeito, finalize o algoritmo e retorne a solução ótima, $\mathbf{s}^*$, a qual é obtida a partir do indivíduo mais bem adaptado. Caso contrário vá para o passo 4.
4. Selecione um conjunto, $I^k$, de indivíduos de $P^k$ que passarão pelos processos de reprodução, cruzamento e mutação para gerar $P^{k+1}$.
5. Faça $k = k + 1$ e vá para o passo 2.

É importante salientar que os algoritmos genéticos não trabalham diretamente com parâmetros do sistema, mas com uma codificação dos mesmos. Os objetos de análise são um conjunto ou uma população de soluções alternativas (não uma única alternativa), que são avaliadas cada uma pelo seu índice de adaptação ao meio. As regras de busca no espaço de soluções factíveis são de caráter probabilístico. A seguir é apresentado o método de otimização por enxame de partículas.

### *4.4.2 Otimização por Enxame de Partículas*

A otimização por enxame de partículas é um método de busca aleatória baseado em populações. Este método, o qual foi proposto por James Kennedy e Russel C. Eberhart em 1995, é inspirado em princípios de interação social (Kennedy & Eberhart, 1995). A busca pela solução ótima é baseada nas interações entre indivíduos pertencentes a um bando de pássaros ou cardume de peixes. Tal ideia é proveniente do trabalho desenvolvido por Craig Reynolds, cujo objetivo era simular padrões de movimento para animação computacional (Reynolds, 1987).

Baseados no modelo de Reynolds, Kennedy e Eberhart definiram três componentes que influenciam no deslocamento de uma partícula inserida em um enxame:

- Componente de inércia: preserva uma memória dos últimos deslocamentos da partícula. Previne mudanças drásticas de direção.
- Componente cognitiva: índice que quantifica desempenho da partícula no ciclo atual em relação ao dos ciclos anteriores. Devido à memória de experiências vivenciadas em ciclos anteriores, as partículas são atraídas para a melhor posição já visitada até o ciclo atual.
- Componente social: quantifica o desempenho de uma partícula em relação às demais do enxame. Devido a esta componente, as partículas também são atraídas pela melhor posição encontrada por partículas vizinhas.

Cada partícula do enxame representa uma solução candidata. Um enxame, por sua vez, pode ser visto como uma população de indivíduos que se movem de maneira aparentemente desorganizada. O método de otimização por enxame de partículas busca imitar o comportamento de grupos de animais e insetos, tais como um enxame de abelhas ou uma manada de gnus.

Considere o seguinte problema

$$\begin{aligned} &\text{minimizar} \quad f(\mathbf{s}) \\ &\text{sujeito a} \quad \mathbf{s} \in X \end{aligned} \tag{4.60}$$

onde $f: \mathbb{R}^m \to \mathbb{R}$. Para resolver este problema através do método de otimização por enxame de partículas, parte-se de uma população inicial aleatória de pontos em $\mathbb{R}^m$. Então, associa-se a cada ponto nesta população um vetor de velocidade. Dessa forma, cada ponto corresponde à posição de uma partícula, que está se movendo a uma determinada velocidade. Deve-se então avaliar a função objetivo de cada ponto da população para criar uma nova população de pontos, cada um com sua respectiva velocidade associada. A criação de novos pontos e velocidades se dá por meio de certas operações que envolvem as relações existentes entre as partículas do enxame.

Cada partícula mantém memória da sua melhor posição já visitada (*best-so-far position*). Esta posição é um valor relacionado com $f(\mathbf{s})$ e é representada por $p_{best}$. Em contrapartida, a melhor posição dentre as já visitadas por toda a população é chamada de melhor global (*global best*) e é representada por $g_{\text{best}}$. As partículas interagem entre si pela atualização de suas respectivas velocidades em relação à melhor pessoal e à melhor global. A velocidade de cada partícula é ponderada por um termo aleatório, com números aleatórios distintos sendo gerados por velocidades em direção a $p_{\text{best}}$ e $g_{\text{best}}$. Para o critério de parada, pode-se finalizar a busca após atingir um número determinado de ciclos ou após um determinado valor de função objetivo ser alcançado.

Considere o problema de minimização de (4.60). Seja $M$ o tamanho da população. A posição da partícula $i$, com $1 \leq i \leq M$, é representada por $\mathbf{b}_i \in \mathbb{R}^m$, e a sua velocidade associada por $\mathbf{c}_i \in \mathbb{R}^m$. Seja também $p_{i,\text{best}}$ a $p_{\text{best}}$ da partícula $i$. O algoritmo de busca usado pelo método de otimização por enxame de partículas para solucionar (4.60) pode ser descrito da seguinte forma:

1. Faça $k = 0$. Gere $\mathbf{b}_i^k \in \mathbb{R}^m$ e $\mathbf{c}_i^k \in \mathbb{R}^m$, com $i = 1,2, \ldots, M$, e faça

$$p_{i,\text{best}} = \mathbf{b}_i^k \tag{4.61}$$

e

$$g_{\text{best}} = \arg \min_{\mathbf{b} \in \{\mathbf{b}_1^k, \ldots, \mathbf{b}_M^k\}} f(\mathbf{s}). \tag{4.62}$$

2. Gere sequências aleatórias $\mathbf{r}_i^k$ e $\mathbf{o}_i^k \in \mathbb{R}^m$ com componentes uniformemente distribuídas no intervalo $U = [0, 1]$ e faça

$$\mathbf{c}_i^{k+1} = \omega \mathbf{c}_i^k + c_{\text{cog}} \mathbf{r}_i^k \circ \left(p_{i,\text{best}} - \mathbf{b}_i^k\right) + c_{\text{soc}} \mathbf{o}_i^k \circ \left(g_{\text{best}} - \mathbf{b}_i^k\right) \tag{4.63}$$

e

$$\mathbf{b}_i^{k+1} = \mathbf{b}_i^k + \mathbf{c}_i^{k+1}. \tag{4.64}$$

3. Se existir $\mathbf{b}_i^{k+1}$, com $1 \leq i \leq M$, tal que

$$f\left(\mathbf{b}_i^{k+1}\right) < f(g_{\text{best}}) \tag{4.65}$$

então faça

$$g_{\text{best}} = \mathbf{b}_i^{k+1}. \tag{4.66}$$

4. Se o critério de para for satisfeito, finalize a busca e retorne $\mathbf{s}^* = g_{\text{best}}$. Caso contrário, faça $k = k + 1$ e vá para o passo 2.

As constantes $c_{\text{cog}}$, $c_{\text{soc}}$ e $\omega$ são as componentes cognitiva, social e de inércia, respectivamente. Recomenda-se um valor próximo porém menor do que 1 para $\omega$. As componentes $c_{\text{cog}}$ e $c_{\text{soc}}$ estabelecem o quão bem direcionada uma partícula está em relação às melhores posições. Valores recomendados para tais constantes são $c_{\text{cog}}, c_{\text{soc}} \approx 2$ (Chong & Zak, 2013). O operador "$\circ$" corresponde à multiplicação entre os elementos de duas matrizes de mesma dimensão. Por exemplo, tomando-se duas matrizes $\mathbf{A}$ e $\mathbf{B}$ de ordem $I$ x $J$, o resultado de $\mathbf{A} \circ \mathbf{B}$ é uma matriz $\mathbf{C}$ com as mesmas dimensões de $\mathbf{A}$ e $\mathbf{B}$, cujos elementos são dados por $c_{i,j} = a_{i,j} * b_{i,j}$, com $1 \leq i \leq I$ e $1 \leq j \leq J$.

Os métodos de otimização por enxame de partículas vêm sendo desenvolvidos desde o trabalho proposto por Kennedy e Eberhart em 1995. Uma variante do algoritmo mostrado acima foi proposta por Maurice Clerc em 1999, na qual as velocidades são atualizadas por (Clerc, 1999)

$$\mathbf{c}_i^{k+1} = \kappa\left[\mathbf{c}_i^k + c_{\text{cog}}\mathbf{r}_i^k \circ \left(p_{i,\text{best}} - \mathbf{b}_i^k\right) + c_{\text{soc}}\mathbf{o}_i^k \circ \left(g_{\text{best}} - \mathbf{b}_i^k\right)\right] \tag{4.67}$$

onde $\kappa$ é chamado de fator de constrição, calculado a partir de

$$\kappa = \frac{2}{\left|2 - \varpi - \sqrt{\varpi^2 - 4\varpi}\right|} \tag{4.68}$$

sendo $\varpi = c_{\text{cog}} + c_{\text{soc}}$ e $\varpi > 4$.

O método de otimização por enxame de partículas difere de outros métodos de busca aleatória, tais como o método simplex e o de recozimento simulado, pelo fato de atualizar não somente a solução candidata a ótima do problema, mas também uma população de soluções candidatas, representadas pelas partículas do enxame.

## 4.5 Métodos Empregados na Otimização dos Algoritmos PNLMS e IPNLMS

Esta seção apresenta os dois métodos de busca empregados por este trabalho para otimizar a escolha de parâmetros dos algoritmos PNLMS e IPNLMS. A saber, o método da razão áurea e o método da busca tabu.

### *4.5.1 Método da Razão Áurea*

Métodos de otimização clássicos buscam a minimização da função objetivo a partir de uma direção de descida. Porém, alguns problemas envolvendo funções de caráter não-linear ou não-suave não podem ser resolvidos analiticamente. Alternativas para tais casos são métodos que buscam de forma inteligente o ponto de mínimo da função objetivo. Uma classe de métodos que usam tal estratégia é a dos métodos de busca linear.

Busca linear ou unidimensional consiste em um processo para determinação do ponto ótimo de uma curva. Isto é equivale à minimização de uma função de uma única variável. Métodos populares de busca linear incluem os de Fibonacci, da bisseção e da razão áurea (Bazaraa, Sherali & Shetty, 2006).

A relação de seção áurea, dada pela constante $\tau = \left(\sqrt{5}+1\right)/2 \cong 1.618$, foi considerada pelos gregos o valor mais esteticamente correto para a razão entre dois lados adjacentes de um mesmo retângulo (Luenberguer & Ye, 2008). O método da razão áurea, o qual é baseado nesta constante, é uma técnica de otimização linear que dispensa o cálculo de derivadas. O objetivo de tal método é minimizar (ou maximizar) uma função objetivo unimodal, ou seja, com apenas um único ponto de mínimo (ou de máximo) em torno de um dado intervalo de incerteza (Bazaraa, Sherali & Shetty, 2006). Este intervalo consiste de um conjunto de soluções factíveis, onde é sabido que o mesmo contém o minimizador (ou maximizador) da função objetivo. A Figura 4.1 mostra um exemplo de uma função $f(s)\colon \mathbb{R} \to \mathbb{R}$ de caráter unimodal em um dado intervalo $[a, b] \subset \mathbb{R}$.

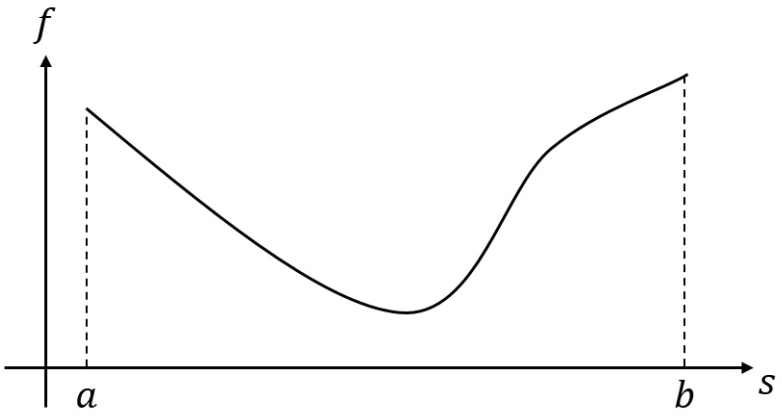



Figura 4.1. Exemplo de uma função unimodal em um intervalo $[a, b]$.

Considere o seguinte problema de otimização

$$\begin{aligned} \text{minimizar} \quad & f(s) \\ \text{sujeito a} \quad & s \in [a, b] \end{aligned} \tag{4.69}$$

com $[a, b] \subset \mathbb{R}$. O algoritmo empregado pelo o método da razão áurea para resolver (4.69) é descrito a seguir em detalhes:

1. Faça $k = 0$. Inicialize o método da razão áurea com os limites $a$ e $b$ do intervalo de incerteza $[a, b]$ e com a tolerância, $tol > 0$, adotada para convergência.
2. Faça $k = k + 1$ e calcule
$$\lambda_1^k = a(1 - \tau) + b\tau \tag{4.70}$$
$$\lambda_2^k = a\tau + b(1 - \tau). \tag{4.71}$$
3. Obtenha $f(\lambda_1^k)$ e $f(\lambda_2^k)$ a partir de $\lambda_1^k$ e $\lambda_2^k$, respectivamente.
4. Se
$$\left|f(\lambda_1^k) - f(\lambda_2^k)\right| < tol \tag{4.72}$$
vá para o passo 6. Caso contrário, vá para o passo 5.
5. Se
$$f(\lambda_1^k) > f(\lambda_2^k) \tag{4.73}$$
faça
$$k = k + 1 \tag{4.74}$$
$$a = \lambda_1^{k-1} \tag{4.75}$$
$$\lambda_1^k = \lambda_2^{k-1} \tag{4.76}$$
$$\lambda_2^k = a\tau + b(1 - \tau) \tag{4.77}$$
e retorne ao passo 3. Caso contrário, faça
$$k = k + 1 \tag{4.78}$$
$$b = \lambda_2^{k-1} \tag{4.79}$$
$$\lambda_2^k = \lambda_1^{k-1} \tag{4.80}$$
$$\lambda_1^k = b\tau + a(1 - \tau) \tag{4.81}$$
e retorne ao passo 3.
6. Razão áurea convergiu. Retornar
$$s^* = \lambda_1^k \tag{4.82}$$
ou
$$s^* = \lambda_2^k. \tag{4.83}$$

Sendo $s^*$ o minimizador de $f$ dentro do intervalo $[a, b]$. As variáveis $\lambda_1^k$ e $\lambda_2^k$ são os candidatos a $s^*$ do ciclo $k$ do método da razão áurea.

Apesar da fácil implementação e rápida convergência, para que o método da razão áurea tenha desempenho satisfatório é necessário que a função objetivo seja unimodal, ou seja, possua apenas um único minimizador dentro do intervalo de incerteza. Outra característica importante deste método está diretamente relacionada com o intervalo $[a, b]$, o qual diminui de tamanho a

cada ciclo. A Figura 4.2 mostra um exemplo de como este intervalo diminui do ciclo anterior, $k$, para o ciclo seguinte, $k+1$, de execução do método da razão áurea.

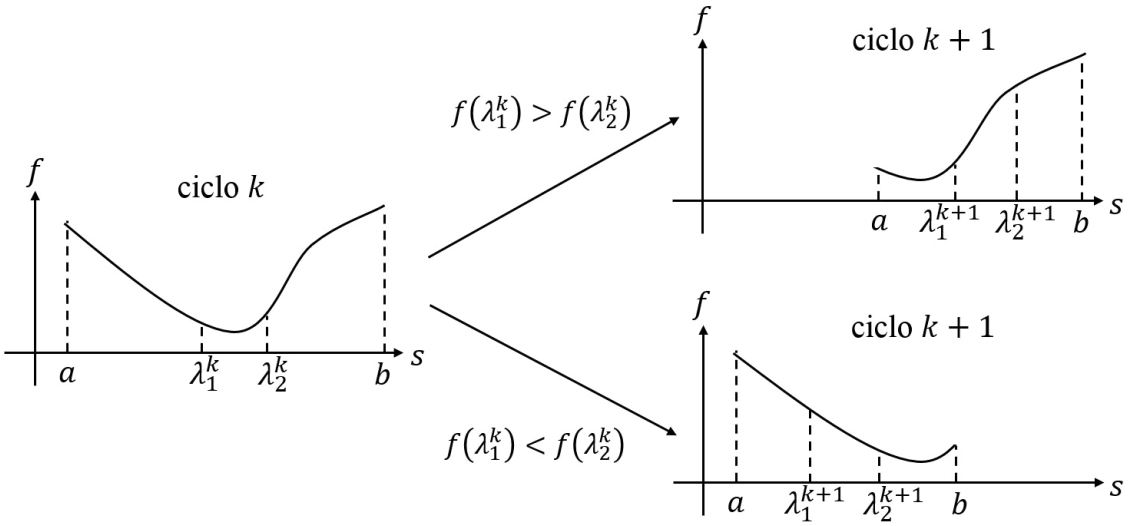



Figura 4.2. Redução no tamanho do intervalo $[a, b]$ durante a execução do método da razão áurea.

Seja o tamanho inicial do intervalo de incerteza $T_1 = b - a$, para $k = 1$, tem-se que o tamanho de $[a, b]$ após $k$ ciclos de execução do método razão áurea é dado por (Luenberguer & Ye, 2008)

$$T_k = \left(\frac{1}{\tau}\right)^{k-1} T_1 \tag{4.84}$$

e de (4.84) tem-se que

$$\frac{T_{k+1}}{T_k} = \frac{1}{\tau} = 0{,}618. \tag{4.85}$$

Portanto, de (4.85) conclui-se que o método da razão áurea converge linearmente para o minimizador de $f$ em $[a, b]$, com uma taxa de convergência igual a $1/\tau = 0{,}618$. A seguir é apresentado o método da busca tabu.

*4.5.2 Método da Busca Tabu*

O método da busca tabu, assim como os algoritmos genéticos e o método de otimização por enxame de partículas, é uma metaheurística que usa uma técnica de busca local com movimentos simples e mecanismos de memória que evitam a estagnação em ótimos locais (Glover, 1989). Tendo suas origens no final da década de 70, sendo empregada em problemas de engenharia da computação e de otimização combinatória, o método da busca tabu tem ampliado seu ramo de aplicação nas últimas décadas, atingindo diversas áreas de áreas que vão da microeletrônica aos sistemas elétricos de potência (Glover, 1990), (Zuo, Murray & Smith, 2014) e (Junior, Cossi, Contreras & Mantovani, 2013). Exemplos mais recentes de aplicação

do método da busca tabu incluem treinamento de redes neurais artificiais, projeto de filtros FIR passa-baixas para implementação física (*hardware*), conformação de feixe em arranjos de antenas adaptativas, dentre outros (Glover & Laguna, 1997), (Pham & Karaboga, 2000) e (Gao, Dai, Yuen & Wang, 2016).

O termo "tabu" (ou *taboo*), segundo dicionários internacionais, possui significado semelhante a algo proibido de ser mencionado por seu caráter sagrado, atribuído por determinada crença (Laguna, 1994). Baseada nesta definição, a filosofia da busca tabu consiste na diversificação do espaço de busca, explorando uma gama de princípios pré-definidos. Para tal, usa-se uma estrutura de memória flexível capaz de aceitar movimentos que não tragam necessariamente uma melhora direta. O objetivo é prevenir repetições e explorar várias regiões do espaço de soluções factíveis, promovendo assim a evolução na tomada de decisão a partir do aprendizado adquirido com experiências passadas. Os componentes básicos de uma busca tabu são:

- Vizinhança de possíveis soluções;
- Movimentos adotados para a obtenção de novas soluções;
- Mecanismos de memória, e;
- Critérios de aspiração e convergência.

O ponto de partida de toda busca tabu é uma solução inicial escolhida aleatoriamente. Então, a partir desta solução inicial e dos movimentos adotados, cria-se um conjunto aleatório de soluções vizinhas. Estas soluções vizinhas são então comparadas com a solução inicial para determinar a melhor solução do ciclo atual da busca. Então, inicia-se um novo ciclo e cria-se um novo conjunto aleatório de soluções vizinhas a partir da melhor solução do ciclo anterior. As novas soluções vizinhas são então comparadas com a melhor solução do ciclo anterior para determinar a melhor do ciclo atual. Este processo se repete até que o critério de convergência seja satisfeito.

A principal ferramenta usada pela busca tabu para evitar repetições é um mecanismo de memória de curto prazo denominado lista tabu. Nesta lista são armazenados movimentos realizados recentemente, tornando-os proibidos de serem executados durante um dado intervalo de ciclos denominado período tabu. Outra ferramenta chave é o critério de aspiração, o qual anula a condição tabu de determinado movimento caso este leve a uma solução com ganhos significativos para a busca. Tal critério pode se basear nos valores de função objetivo das soluções já visitadas, em direções de busca definidas previamente ou durante a execução da busca, ou por influência de regiões do espaço de soluções factíveis (Glover & Laguna, 1997).

Considere um problema de otimização semelhante ao de (4.60). O algoritmo empregado pelo método da busca tabu para resolver este problema é descrito a seguir em detalhes:

1. Faça $k = 0$. Escolha uma solução inicial aleatória $\mathbf{s}^0 \in X$ e calcule $f(\mathbf{s}^0)$.
2. Faça $k = k + 1$. Gere um conjunto $\mathbf{V}^k$ de $M$ soluções vizinhas a partir de $\mathbf{s}^{k-1}$ e verifique se cada solução vizinha $\mathbf{v}_i^k \in \mathbf{V}^k$ $(i = 1, 2, \dots, M)$ é "tabu".

3. Calcule $f\left(\mathbf{v}_i^k\right)$ $(i = 1, 2, \dots, M)$ e verifique se existem soluções vizinhas $\mathbf{v}_i^k$, geradas a partir de movimentos "tabu", que satisfazem o critério de aspiração.
4. Se existe $\mathbf{v}_i^k \in \mathbf{V}^k$ $(i = 1, 2, \dots, M)$, tal que $\mathbf{v}_i^k$ é melhor do que $\mathbf{s}^{k-1}$ e $\mathbf{v}_i^k$ não é "tabu" ou satisfaz o critério de aspiração, faça

$$\mathbf{s}^k = \mathbf{v}_i^k. \tag{4.86}$$

   Caso contrário, faça

$$\mathbf{s}^k = \mathbf{s}^{k-1}. \tag{4.87}$$

5. Atualize a lista tabu.
6. Se o critério de convergência é satisfeito, vá para o passo 7. Caso contrário, retorne ao passo 2.
7. Busca tabu convergiu. Retorne

$$\mathbf{s}^* = \mathbf{s}^k. \tag{4.88}$$

   Sendo $\mathbf{s}^*$ a solução ótima do problema.

Para entender melhor o funcionamento do método da busca tabu, pode-se recorrer a um exemplo clássico de otimização combinatória, tal como o problema das $m$ rainhas. Este problema consiste na alocação de $m$ rainhas dispostas em um tabuleiro de xadrez de dimensão $m$ x $m$. Esta alocação deve ser feita de tal forma que nenhuma das rainhas seja capaz de capturar outra com apenas um movimento, ou seja, sem que ocorram colisões entre as rainhas do tabuleiro (Laguna, 1994). Dessa forma, o problema das $m$ rainhas pode ser tratado como um problema de permutação, onde a $i$-ésima rainha, $\mathrm{Q}_i$, com $i = 1,2, \dots, m$, está localizada na posição correspondente à $i$-ésima linha e à $\pi(i)$-ésima coluna. Então, uma solução candidata pode ser representada por uma permutação do tipo

$$\Pi = \{\pi(1), \pi(2), \dots, \pi(m)\} \tag{4.89}$$

que especifica a posição exata das $m$ rainhas no tabuleiro de xadrez. Note que este tipo de configuração evita que uma rainha tenha a chance de capturar outra localizada na mesma linha ou na mesma coluna. Assim, o problema resume-se em minimizar o número total de colisões nas diagonais do tabuleiro. A Figura 4.3 mostra um tabuleiro de tamanho 6 x 6 para o qual a disposição das rainhas (representadas por $\mathrm{Q}_i$, com $i = 1,2, \dots, m$) corresponde à permutação $\Pi = \{4, 5, 1, 6, 2, 3\}$. Esta permutação também pode ser representada na forma de um vetor, como mostrado na Figura 4.4. Cada elemento deste vetor contém um índice de coluna do tabuleiro referente à posição de uma das rainhas. A numeração da Figura 4.4 significa que a rainha $\mathrm{Q}_1$ está na coluna 4, a rainha $\mathrm{Q}_2$ está na coluna 5 e assim por diante. Note que, para a configuração da Figura 4.3, tem-se um total de três possíveis colisões entre as rainhas, conforme a indicação das setas duplas. Especificamente, tem-se uma possível colisão entre $\mathrm{Q}_1$ e $\mathrm{Q}_2$, outra entre $\mathrm{Q}_2$ e $\mathrm{Q}_5$ e outra entre $\mathrm{Q}_5$ e $\mathrm{Q}_6$.

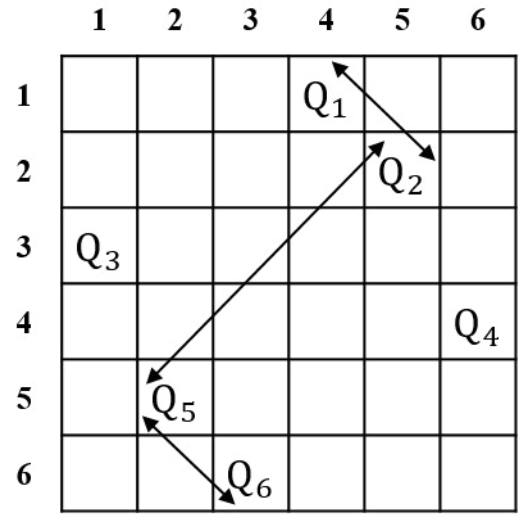



Figura 4.3. Configuração de $m = 6$ rainhas em um tabuleiro de tamanho 6 x 6.

| 4 | 5 | 1 | 6 | 2 | 3 |
|---|---|---|---|---|---|

Figura 4.4. Permutação inicial para o problema da Figura 4.3.

Para solucionar o problema da Figura 4.3 usando o método da busca tabu, toma-se o vetor da Figura 4.4 como sendo a solução inicial, $\mathbf{s}^0$, e define-se um movimento para a criação do conjunto de soluções vizinhas, $\mathbf{V}^k$. Como o problema das $m$ rainhas pode ser tratado como um problema de permutação, um movimento adequado consiste na troca entre os valores de duas posições escolhidas aleatoriamente. A Figura 4.5 mostra um exemplo de execução deste movimento.

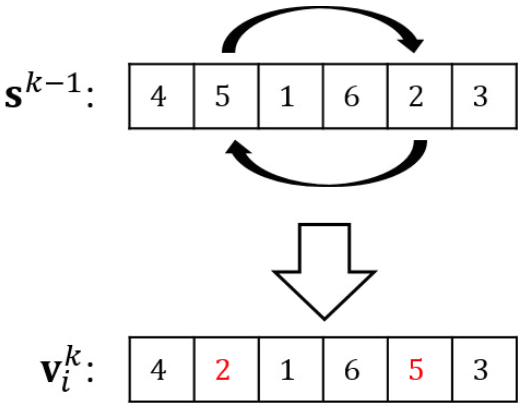



Figura 4.5. Movimento adotado para a criação das soluções vizinhas.

Note que, no exemplo da Figura 4.5, as rainhas $Q_2$ e $Q_5$ são escolhidas aleatoriamente para trocar de posição no tabuleiro, dando origem a uma solução vizinha, $\mathbf{v}_i^k \in \mathbf{V}^k$, com $1 \leq$

$i \leq M$. Assim, o conjunto $\mathbf{V}^k$ pode ser formado a partir de $M$ movimentos semelhantes ao da Figura 4.5, conforme exemplo mostrado na Figura 4.6. Então, em cada ciclo, $k$, os elementos de $\mathbf{V}^k$ são comparados com a melhor solução do ciclo anterior, $\mathbf{s}^{k-1}$, em relação ao valor de função objetivo, ou seja, o número total de possíveis colisões que cada solução candidata possui.

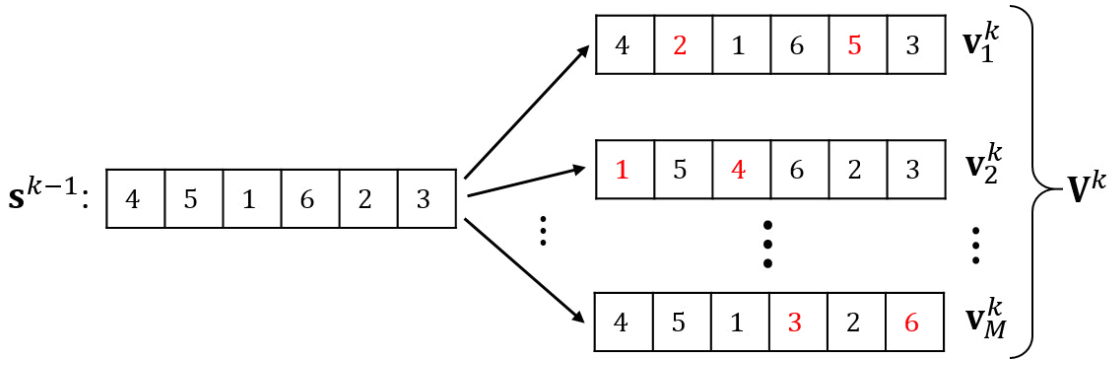



Figura 4.6. Exemplo de criação do conjunto $\mathbf{V}^k$ para o problema da Figura 4.3 usando movimentos semelhantes ao da Figura 4.5.

Para prevenir repetições de movimentos realizados em um passado recente, os quais podem resultar em soluções já visitadas anteriormente, usa-se a lista tabu. Para o problema em questão, pode-se construir uma estrutura semelhante a uma matriz triangular superior, para a qual apenas os elementos acima da diagonal principal são relevantes. Tal estrutura é mostrada na Figura 4.7. Note que os números armazenados dentro desta estrutura indicam o último ciclo em que determinado movimento possuirá a condição de proibido ou "tabu". Por exemplo, a troca entre $Q_2$ e $Q_5$ permanece como "tabu" até o ciclo $k = 4$. A quantidade de ciclos que um movimento é considerado "tabu" depende do valor do período tabu. Quando um dado movimento deixar de ser "tabu", o número que marca tal condição na lista tabu deve ser apagado da mesma. Então, considerando o exemplo da Figura 4.7, quando a busca atingir o ciclo $k = 5$, o número "4", que marca uma condição "tabu" para a troca entre $Q_2$ e $Q_5$, deve ser apagado da lista tabu.

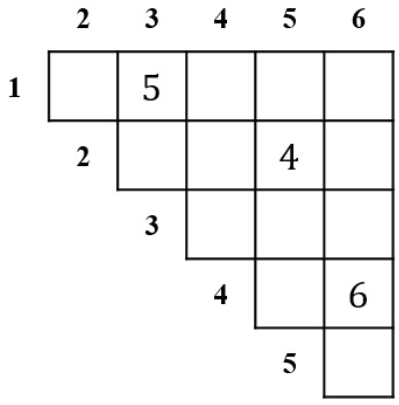



Figura 4.7. Lista tabu para o problema da Figura 4.3.

Em relação ao critério de aspiração, pode-se definir a seguinte condição: se uma determinada solução candidata do ciclo atual proveniente de um movimento "tabu" levar a um número total de colisões menor do que a da melhor solução do ciclo anterior (além de um número total de colisões menor do que as demais soluções de $\mathbf{V}^k$), então esta solução tem sua condição "tabu" revogada e passa a vigorar como a melhor do ciclo atual. Já para o critério de convergência, pode-se parar a busca caso seja encontrada uma solução para a qual o valor de função objetivo seja igual a zero, ou seja, não há colisão entre as rainhas. Outro critério de convergência aplicável consiste em parar a busca após uma determinada quantidade de ciclos ter sido alcançada.

Conforme os objetivos pretendidos e as características do problema, o método busca tabu ainda pode valer-se de outros recursos para aprimorar seu desempenho. Estratégias de diversificação e intensificação, assim como outros mecanismos de memória além da lista tabu, são exemplos de tais recursos. Variações no período tabu e no tamanho do conjunto $\mathbf{V}^k$ são perfeitamente aplicáveis, bem como o uso de uma lista de soluções candidatas proveniente de execuções anteriores da busca. Tais candidatos podem figurar em novas execuções, com o intuito de acelerar o processo de busca ou mesmo encontrar soluções ainda melhores. Pode-se também recorrer a outras estratégicas de cunho "genético", tais como primeiro de melhora, aspiração plus, busca paralela, dentre outros (Glover & Laguna, 1997) e (Pham & Karaboga, 2000).

# Capítulo 5

# Algoritmos Adaptativos Proporcionais com Escolha de Parâmetros baseada em Métodos de Otimização

Neste capítulo são propostas versões melhoradas dos algoritmos PNLMS e IPNLMS. Nessas propostas é considerado o uso de métodos de otimização para determinar os valores dos parâmetros dos algoritmos PNLMS e IPNLMS. Primeiramente, é apresentada a função objetivo a ser minimizada pelas metodologias adotadas, juntamente com uma análise do efeito dos parâmetros $\rho$ e $f(n)$ do algoritmo PNLMS, e do parâmetro $\alpha$ do algoritmo IPNLMS, no comportamento da função objetivo adotada. Então, os algoritmos PNLMS e IPNLMS com otimização de parâmetros propostos são obtidos.

## 5.1 Função Objetivo Adotada

Com base na análise do comportamento dos algoritmos PNLMS e IPNLMS realizada no capítulo 3, verifica-se que é possível aprimorar o desempenho destes algoritmos a partir da escolha correta dos parâmetros dos mesmos. Essa escolha pode ser feita com a ajuda de um método de otimização. Porém, conforme visto no capítulo 4, tais métodos operam sob o princípio da minimização ou maximização de uma função de custo, também chamada de função objetivo. Conforme visto nos capítulos 2 e 3, o principal objetivo de um algoritmo adaptativo é minimizar iterativamente o erro entre a saída desejada (ou da planta) e a saída do filtro. Então, uma função objetivo baseada neste erro mostra-se uma escolha coerente. Em outras palavras, quanto mais próximo de zero o erro, menor é a diferença entre os coeficientes do filtro e os da planta e, consequentemente, mais rápida é a velocidade de convergência do algoritmo adaptativo.

Sabe-se do capítulo 3 que em cada iteração dos algoritmos PNLMS e IPNLMS é calculado o valor do erro de estimação *a priori* dado por

$$\begin{aligned} e(n) &= d(n) - y(n) + v(n) \\ &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n) + v(n) \end{aligned} \tag{5.1}$$

onde $d(n)$ e $y(n)$ são as saídas da planta e do filtro adaptativo, respectivamente. As variáveis $\mathbf{x}(n)$, $\mathbf{w}(n)$ e $v(n)$ são os vetores de entrada e de coeficientes do filtro adaptativo, e o ruído aditivo de medição, respectivamente. Sabe-se também do capítulo 2 que o algoritmo NLMS foi concebido na tentativa de aumentar a velocidade de convergência do algoritmo LMS a partir da

minimização do erro de estimação *a posteriori* dado por (Sayed, 2003) e (Nascimento & Silva, 2014)

$$e_p(n) = d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n) \tag{5.2}$$

Note que a única diferença entre (5.1) e (5.2) é a estimativa dos coeficientes do filtro adaptativo adotada para o cálculo do erro. Então, tem-se que (5.2) pode ser obtida a partir de (5.1) apenas substituindo a estimativa anterior, $\mathbf{w}(n)$, pela estimativa atual, $\mathbf{w}(n+1)$, dos coeficientes do filtro adaptativo. Logo, há uma relação direta entre os erros de estimação *a priori* e *a posteriori*. Dessa forma, a função objetivo adotada pelas metodologias propostas para otimizar a escolha dos parâmetros dos algoritmos PNLMS e IPNLMS é

$$\psi(n) = 10\log_{10}\left\{\left[e_p(n)\right]^2\right\} \tag{5.3}$$

onde $\psi(n)$ é o erro quadrático *a posteriori* em dB.

Assim, o problema de otimização a ser solucionado pelos métodos de otimização adotados em cada iteração do processo de adaptação do algoritmo PNLMS é dado por

$$\rho_{\mathrm{opt}}(n) = \arg \min_{0<\rho(n)<1} \psi(n) \tag{5.4}$$

ou

$$f_{\mathrm{opt}}(n) = \arg \min_{0<f(n)<1} \psi(n) \tag{5.5}$$

sendo $\rho_{\mathrm{opt}}(n)$ [ou $f_{\mathrm{opt}}(n)$] é o parâmetro de proporcionalidade [ou o fator de ativação] ótimo da $n$-ésima iteração do processo de adaptação do algoritmo PNLMS. Note, de (3.1), (3.6), (3.7) e (3.8), que $\rho$ [ou $f(n)$] influencia diretamente no valor de $\mathbf{w}(n+1)$ e este, por sua vez, impacta no valor de (5.3).

Analogamente, o problema de otimização a ser solucionado pelos métodos de otimização adotados em cada iteração do processo de adaptação do algoritmo IPNLMS é encontrar

$$\alpha_{\mathrm{opt}}(n) = \arg \min_{-1\leq\alpha(n)<1} \psi(n) \tag{5.6}$$

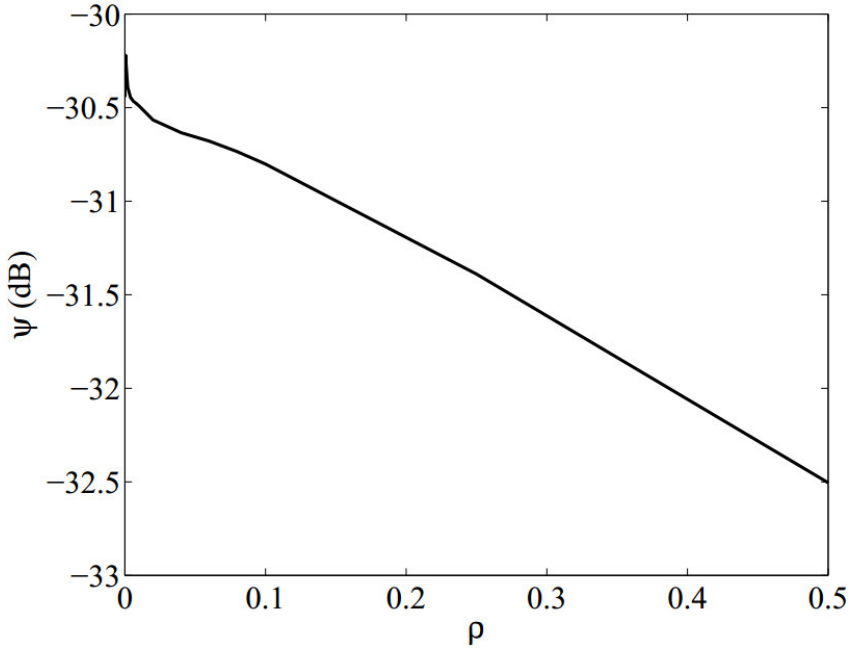
onde $\alpha_{\mathrm{opt}}(n)$ é o parâmetro de proporcionalidade ótimo da $n$-ésima iteração do processo de adaptação do algoritmo IPNLMS. Note novamente, de (3.1) e (3.23), que $\alpha$ influencia diretamente no valor de $\mathbf{w}(n+1)$ e este, por sua vez, impacta no valor de (5.3).

A seguir é feita uma breve análise do impacto dos parâmetros $\rho$ e $f(n)$ do algoritmo PNLMS, e do parâmetro $\alpha$ do algoritmo IPNLMS, sobre a função objetivo adotada.

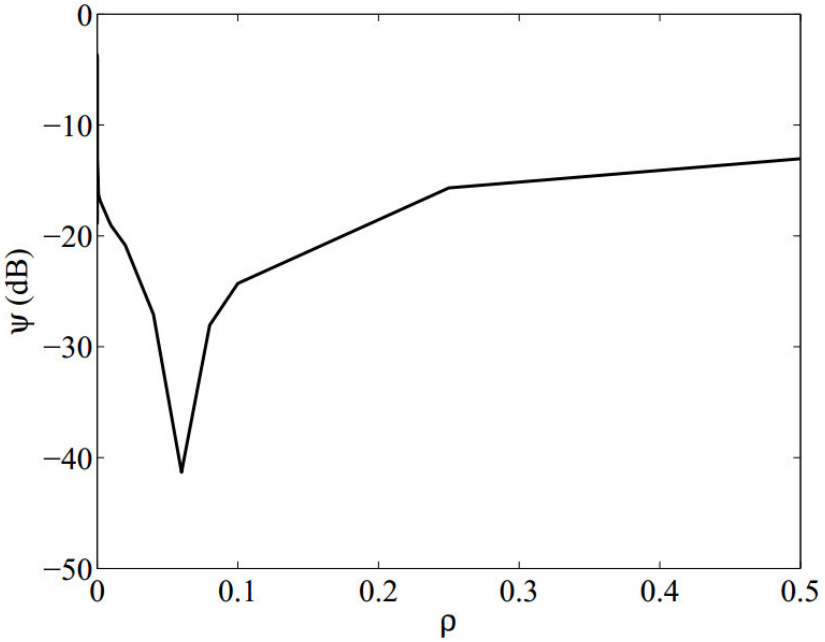
### *5.1.1 Impacto do Parâmetro de Proporcionalidade do Algoritmo PNLMS no Comportamento do Erro a posteriori*

Para verificar o efeito de $\rho$ no comportamento de (5.3), são realizadas simulações de Monte Carlo (média de 100 realizações independentes) para um problema de identificação de sistemas. O cenário das simulações é o mesmo apresentado na subseção 3.2.1.

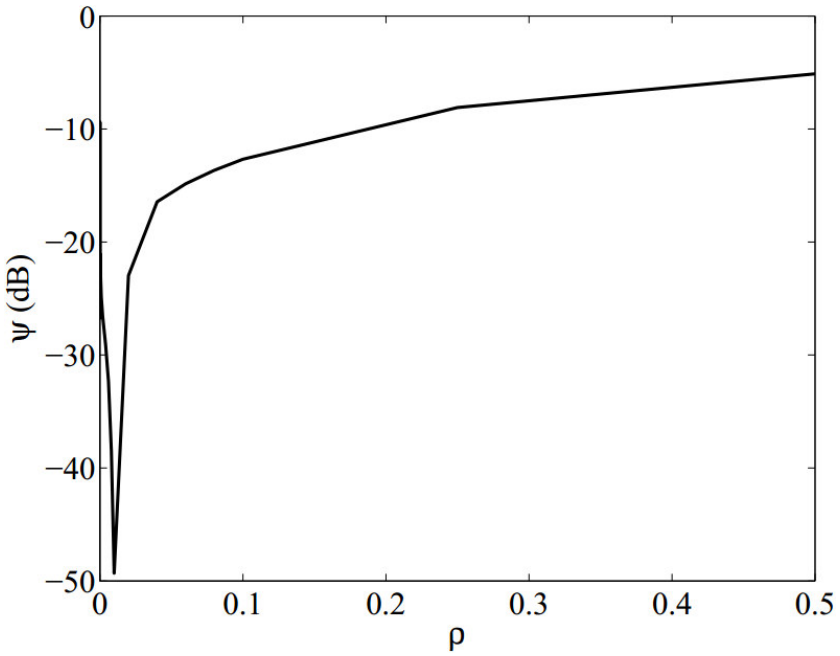
Com o objetivo de ilustrar o impacto de $\rho$ em (5.3), a Figura 5.1 mostra exemplos do comportamento do erro quadrático *a posteriori* em dB nas iterações (a) 25, (b) 50 e (c) 100 do processo de adaptação do algoritmo PNLMS para vários valores de $\rho$, que vão de $10^{-9}$ a 0,5. Os parâmetros de passo e inicialização considerados são $\mu = 0{,}5$ e $\delta = 0{,}01$, respectivamente.




(a)




(b)




(c)

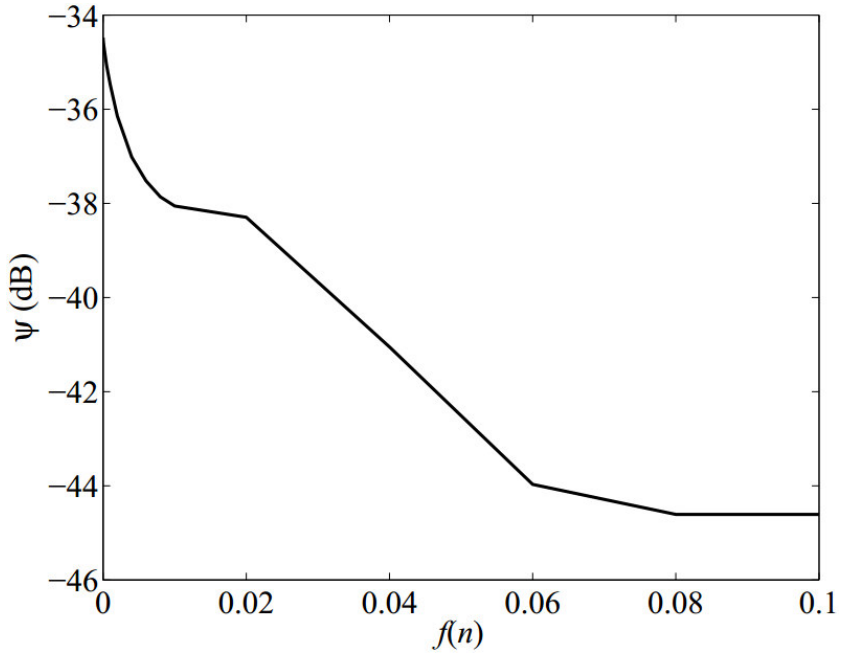
Figura 5.1. Comportamento do erro quadrático *a posteriori* em dB, $\psi(n)$, em uma dada iteração, $n$, do processo de adaptação do algoritmo PNLMS para a identificação de uma planta de esparsidade $S(\mathbf{p}) = 0{,}9435$, com $\mu = 0{,}5$, $\delta = 0{,}01$, e considerando vários valores de $\rho$ que vão de $10^{-9}$ a 0,5. (a) Iteração $n = 25$. (b) Iteração $n = 50$. (c) Iteração $n = 100$.

Note das curvas da Figura 5.1 que o valor do parâmetro de proporcionalidade, $\rho$, do algoritmo PNLMS influencia diretamente no comportamento do erro de estimação *a posteriori*. Então, tem-se que a velocidade de convergência do algoritmo PNLMS pode ser aprimorada com o uso de um método de otimização para determinar $\rho_{\text{opt}}(n)$ que solucione o problema de (5.4). Note também, do comportamento unimodal das curvas da Figura 5.1, que o método da razão áurea é aplicável para determinar $\rho_{\text{opt}}(n)$.

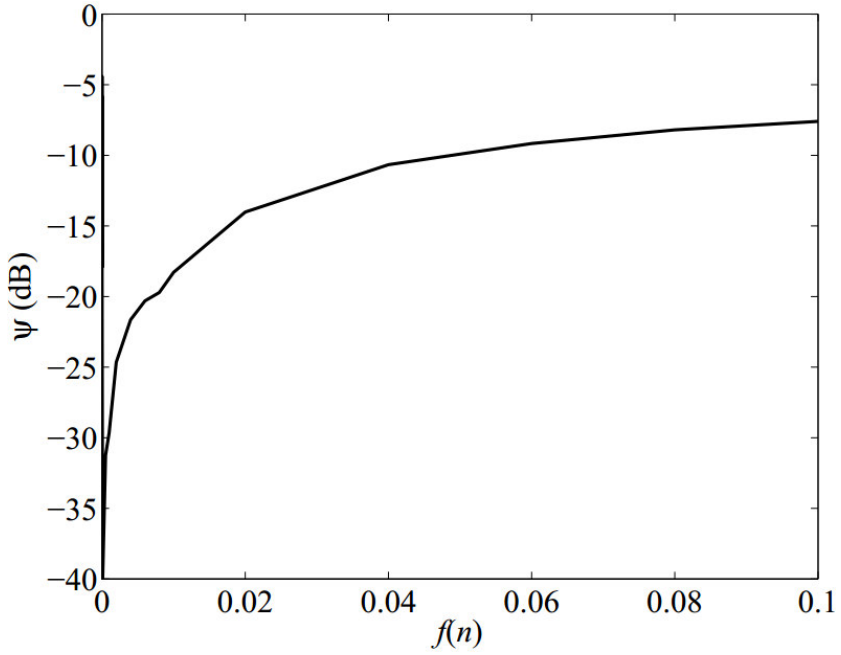
### *5.1.2 Impacto do Fator de Ativação do Algoritmo PNLMS no Comportamento do Erro a posteriori*

Similarmente à análise realizada na subseção anterior, o efeito de $f(n)$ no comportamento de (5.3) também pode ser verificado a partir de simulações de Monte Carlo (média de 100 realizações independentes), considerando um problema de identificação de sistemas. Para tal, considera-se novamente o cenário apresentado na subseção 3.2.1 para as simulações.

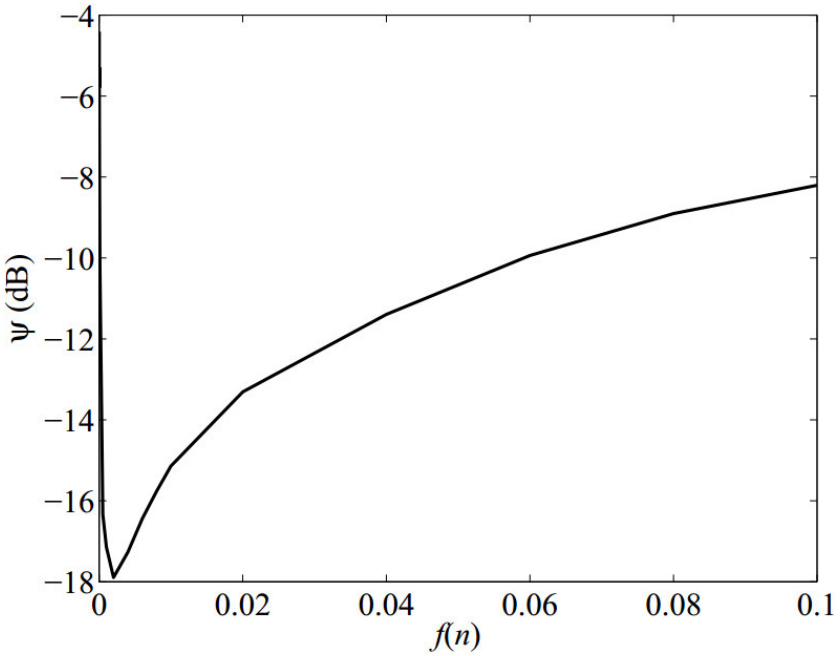
Com o intuito de mostrar o efeito de $f(n)$ em (5.3), a Figura 5.2 mostra exemplos do comportamento do erro quadrático *a posteriori* em dB nas iterações (a) 25, (b) 50 e (c) 100 do processo de adaptação do algoritmo PNLMS, considerando vários valores de $f(n)$, que vão de $10^{-9}$ a 0,1. O parâmetro de passo considerado é $\mu = 0{,}5$.


−34
−36
−38
−40
−42
−44
−46
ψ (dB)
0
0.02
0.04
0.06
0.08
0.1
f(n)

(a)


0
−5
−10
−15
−20
−25
−30
−35
−40
ψ (dB)
0
0.02
0.04
0.06
0.08
0.1
f(n)

(b)




(c)

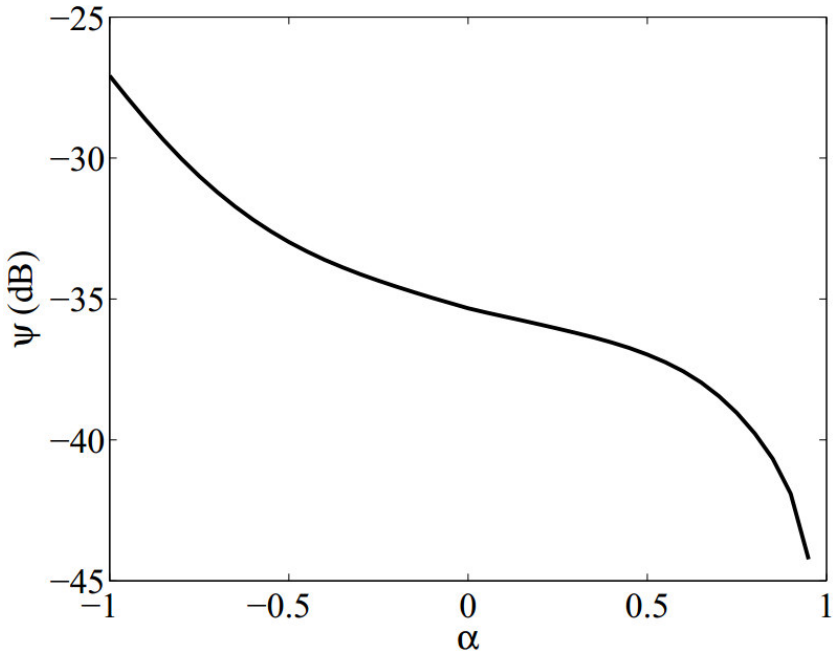
Figura 5.2. Comportamento do erro quadrático *a posteriori* em dB, $\psi(n)$, em uma dada iteração, $n$, do processo de adaptação do algoritmo PNLMS para a identificação de uma planta de esparsidade $S(\mathbf{p}) = 0{,}9435$, com $\mu = 0{,}5$, e considerando vários valores de $f(n)$ que vão de $10^{-9}$ a 0,1. (a) Iteração $n = 25$. (b) Iteração $n = 50$. (c) Iteração $n = 100$.

Note das curvas da Figura 5.2 que o valor do fator de ativação, $f(n)$, do algoritmo PNLMS influencia diretamente no valor do erro de estimação *a posteriori*. Então, tem-se que a velocidade de convergência do algoritmo PNLMS pode ser melhorada a partir do uso de um método de otimização para determinar $f_{\text{opt}}(n)$ que solucione o problema de (5.5). Note também que o comportamento unimodal das curvas da Figura 5.2 torna o método da razão áurea aplicável para a determinação de $f_{\text{opt}}(n)$.

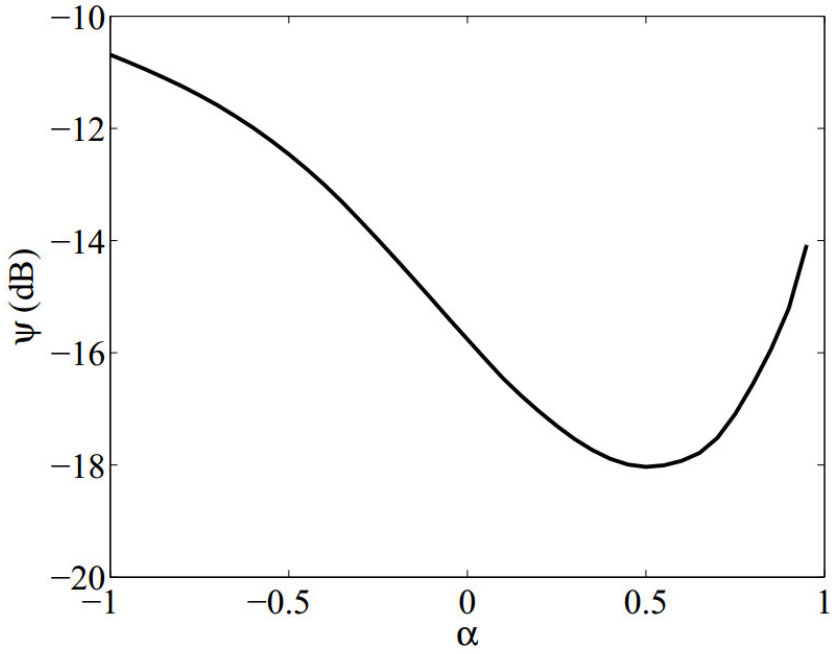
### *5.1.2 Impacto do Parâmetro de Proporcionalidade do Algoritmo IPNLMS sobre o Erro a posteriori*

Similarmente à análise realizada nas subseções anteriores, o efeito de $\alpha$ no comportamento de (5.3) é verificado a partir de simulações de Monte Carlo (média de 100 realizações independentes), considerando um problema de identificação de sistemas usando o algoritmo IPNLMS. O cenário das simulações é o mesmo apresentado na subseção 2.3.2.

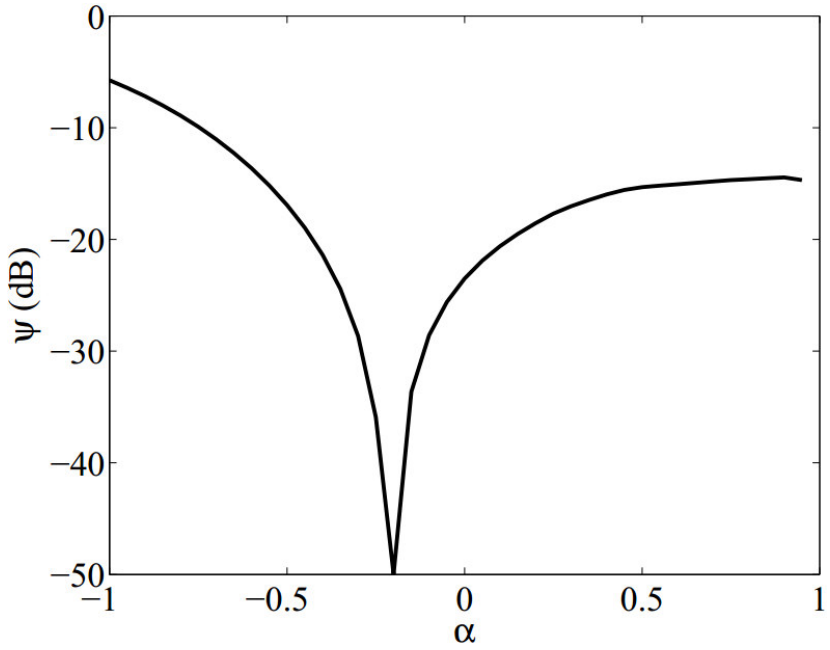
Com o objetivo de ilustrar o impacto de $\alpha$ em (5.3), a Figura 5.3 mostra exemplos do comportamento do erro quadrático *a posteriori* em dB nas iterações (a) 25, (b) 50 e (c) 100 do processo de adaptação do algoritmo PNLMS, considerando vários valores de $\alpha$ que vão de $-1{,}0$ a $0{,}99$. O parâmetro de passo considerado é $\mu = 0{,}5$.




(a)




(b)




(c)

Figura 5.3. Comportamento do erro quadrático *a posteriori* em dB, $\psi(n)$, em uma dada iteração, $n$, do processo de adaptação do algoritmo PNLMS para a identificação de uma planta de esparsidade $S(\mathbf{p}) = 0{,}9435$, com $\mu = 0{,}5$, e considerando vários valores de $\alpha$ que vão de $-1{,}0$ a $0{,}99$. (a) Iteração $n = 25$. (b) Iteração $n = 50$. (c) Iteração $n = 100$.

Note, da Figura 5.3, que o valor do parâmetro de proporcionalidade, $\alpha$, do algoritmo IPNLMS impacta diretamente no valor de (5.3). Então, tem-se que a velocidade de convergência do algoritmo IPNLMS pode ser melhorada como emprego de um método de otimização para determinar $\alpha_{\text{opt}}(n)$ que solucione o problema de (5.6). Note também que o comportamento unimodal da curva da Figura 5.3 torna o método da razão áurea aplicável para resolver o problema de (5.6), em cada iteração, $n$, do processo de adaptação do algoritmo IPNLMS, a partir da escolha de $\alpha_{\text{opt}}(n)$.

A seguir são apresentadas as metodologias propostas para otimizar a escolha dos parâmetros dos algoritmos PNLMS e IPNLMS.

## 5.2 Metodologias Propostas

O algoritmo PNLMS adota valores constantes para o parâmetro de proporcionalidade, $\rho$, e para o fator de ativação, $f(n)$, em sua respectiva regra de atualização. Analogamente, o algoritmo IPNLMS também emprega um parâmetro $\alpha$ constante durante todo o processo de adaptação. Em oposição às regras de atualização adotadas por estes algoritmos, as metodologias propostas por este trabalho de pesquisa fazem uso de métodos de otimização para determinar parâmetros ótimos para cada iteração, $n$, do processo de adaptação dos novos algoritmos propostos. São eles:

- Algoritmos $\rho$GS-PNLMS ($\rho$-*golden section* PNLMS) e $f$GS-PNLMS ($f$-*golden section* PNLMS): ambos empregam o método da razão áurea para determinar o $\rho_{\text{opt}}(n)$ [ou o

$f_{\text{opt}}(n)$] que solucione (5.4) [ou (5.5)] em cada iteração, $n$, do processo de adaptação do algoritmo PNLMS.

- Algoritmos $\rho$TS-PNLMS ($\rho$-*tabu search* PNLMS) e $f$TS-PNLMS ($f$-*tabu search* PNLMS): ambos empregam o método da busca tabu para determinar $\rho_{\text{opt}}(n)$ [ou o $f_{\text{opt}}(n)$] que solucione (5.4) [ou (5.5)] em cada iteração, $n$, do processo de adaptação do algoritmo PNLMS.
- Algoritmo GS-IPNLMS (*golden section* IPNLMS): usa o método da razão áurea para determinar $\alpha_{\text{opt}}(n)$ que solucione (5.6) em cada iteração, $n$, do processo de adaptação do algoritmo IPNLMS.
- Algoritmo TS-IPNLMS (*tabu search* IPNLMS): usa o método da busca tabu para determinar $\alpha_{\text{opt}}(n)$ que solucione (5.6) em cada iteração, $n$, do processo de adaptação do algoritmo IPNLMS.

Estes algoritmos são obtidos a seguir.

### *5.2.1 Algoritmo PNLMS com Otimização do Parâmetro de Proporcionalidade baseada no Método da Razão Áurea*

O algoritmo $\rho$GS-PNLMS usa o método da razão áurea para determinar o valor ótimo do parâmetro de proporcionalidade, $\rho$, de cada iteração do processo de adaptação do algoritmo PNLMS.

O algoritmo $\rho$GS-PNLMS é descrito a seguir em detalhes:

1. Inicialize o vetor de coeficientes do filtro adaptativo ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.7}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.8}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.9}$$

3. Calcule o erro *a priori*

$$e(n) = d(n) - y(n) + v(n). \tag{5.10}$$

4. Atualize o vetor de coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$\rho_{\text{opt}}(n) = \arg \min_{0<\rho(n)<1} \psi(n) \tag{5.11}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.12}$$

Para tal, utilize o seguinte procedimento baseado no método da razão áurea:

I. Faça

$$k = 1 \tag{5.13}$$

$$0 < a \ll 1 \tag{5.14}$$

$$a < b < 1 \tag{5.15}$$

defina o valor da tolerância, $tol > 0$, para a convergência da razão áurea e calcule

$$\rho_1^k(n) = a(1-\tau) + b\tau \tag{5.16}$$

$$\rho_2^k(n) = a\tau + b(1-\tau). \tag{5.17}$$

II. A partir de (3.1), (3.6), (3.7) e (3.8), obtenha $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$ usando $\rho_1^k(n)$ e $\rho_2^k(n)$, respectivamente.

III. A partir de (5.12), obtenha $\psi[\rho_1^k(n)]$ e $\psi[\rho_2^k(n)]$ usando $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$, respectivamente. Se

$$\left|\psi[\rho_1^k(n)] - \psi[\rho_2^k(n)]\right| < tol \tag{5.18}$$

vá para V. Caso contrário, vá para IV.

IV. Se

$$\psi[\rho_1^k(n)] > \psi[\rho_2^k(n)] \tag{5.19}$$

faça

$$k = k + 1 \tag{5.20}$$

$$a = \rho_1^{k-1}(n) \tag{5.21}$$

$$\rho_1^k(n) = \rho_2^{k-1}(n) \tag{5.22}$$

$$\rho_2^k(n) = a\tau + b(1-\tau) \tag{5.23}$$

e retorne para II. Caso contrário, faça

$$k = k + 1 \tag{5.24}$$

$$b = \rho_2^{k-1}(n) \tag{5.25}$$

$$\rho_2^k(n) = \rho_1^{k-1}(n) \tag{5.26}$$

$$\rho_1^k(n) = a(1-\tau) + b\tau \tag{5.27}$$

e retorne para II.

V. Razão áurea convergiu. Faça

$$\rho_{\text{opt}}(n) = \rho_1^k(n) \tag{5.28}$$

ou

$$\rho_{\text{opt}}(n) = \rho_2^k(n). \tag{5.29}$$

Em seguida,obtenha $\mathbf{w}(n+1)$ usando

$$f(n) = \rho_{\text{opt}}(n) \max[\delta, \|\mathbf{w}(n)\|_\infty]. \tag{5.30}$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo $\rho$GS-PNLMS.

Sendo $\rho_1^k(n)$ e $\rho_2^k(n)$ candidatos a $\rho_{\text{opt}}(n)$ do ciclo $k$ da razão áurea.

A seguir, o algoritmo $f$GS-PNLMS é obtido.

### *5.2.2 Algoritmo PNLMS com Otimização do Fator de Ativação baseada no Método da Razão Áurea*

O algoritmo $f$GS-PNLMS usa o método da razão áurea para determinar o valor ótimo para o fator de ativação, $f(n)$, em cada iteração do processo de adaptação do algoritmo PNLMS.

O algoritmo $f$GS-PNLMS é descrito a seguir em detalhes:

1. Inicialize o vetor de coeficientes do filtro adaptativo ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.31}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.32}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.33}$$

3. Calcule o erro *a priori*

$$e(n) = d(n) - y(n) + v(n). \tag{5.34}$$

4. Atualize o vetor de coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$f_{\text{opt}}(n) = \arg \min_{0<f(n)<1} \psi(n) \tag{5.35}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.36}$$

Para tal, utilize o seguinte procedimento baseado no método da razão áurea:

I. Faça

$$k = 1 \tag{5.37}$$

$$0 < a \ll 1 \tag{5.38}$$

$$a < b < 1 \tag{5.39}$$

defina o valor da tolerância, $tol > 0$, para a convergência da razão áurea e calcule

$$f_1^k(n) = a(1-\tau) + b\tau \tag{5.40}$$

$$f_2^k(n) = a\tau + b(1-\tau). \tag{5.41}$$

II. A partir de (3.1), (3.6) e (3.7), obtenha $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$ usando $f_1^k(n)$ e $f_2^k(n)$, respectivamente.

III. A partir de (5.12), obtenha $\psi[f_1^k(n)]$ e $\psi[f_2^k(n)]$ usando $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$, respectivamente. Se

$$\left|\psi[f_1^k(n)] - \psi[f_2^k(n)]\right| < tol \tag{5.42}$$

vá para V. Caso contrário, vá para IV.

IV. Se

$$\psi[f_1^k(n)] > \psi[f_2^k(n)] \tag{5.43}$$

faça

$$k = k + 1 \tag{5.44}$$

$$a = f_1^{k-1}(n) \tag{5.45}$$

$$f_1^k(n) = f_2^{k-1}(n) \tag{5.46}$$

$$f_2^k(n) = a\tau + b(1-\tau) \tag{5.47}$$

e retorne para II. Caso contrário, faça

$$k = k + 1 \tag{5.48}$$

$$b = f_2^{k-1}(n) \tag{5.49}$$

$$f_2^k(n) = f_1^{k-1}(n) \tag{5.50}$$

$$f_1^k(n) = a(1-\tau) + b\tau \tag{5.51}$$

e retorne para II.

V. Razão áurea convergiu. Faça

$$f_{\text{opt}}(n) = f_1^k(n) \tag{5.52}$$

ou

$$f_{\text{opt}}(n) = f_2^k(n). \tag{5.53}$$

Em seguida, calcule $\mathbf{w}(n+1)$ usando

$$\phi_i(n) = \max\left[f_{\text{opt}}(n), |w_i(n)|\right], \qquad i = 1,2,\dots,N. \tag{5.54}$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo $f$GS-PNLMS.

Sendo $f_1^k(n)$ e $f_2^k(n)$ candidatos a $f_{\text{opt}}(n)$ do ciclo $k$ da razão áurea.

Note que o algoritmo $f$GS-PNLMS não necessita de (3.8) e, consequentemente, dispensa uso dos parâmetros $\rho$ e $\delta$ do algoritmo PNLMS original.

### *5.2.3 Algoritmo PNLMS com Otimização do Parâmetro de Proporcionalidade baseada no Método da Busca Tabu*

O algoritmo $\rho$TS-PNLMS usa o método da busca tabu para determinar $\alpha_{\text{opt}}(n)$ que solucione (5.4) em cada iteração do processo de adaptação do algoritmo PNLMS. Para tal, faz-se necessário definir a forma dos elementos da busca tabu, tais como o conjunto de soluções vizinhas e a lista tabu, para o problema em questão. Estes elementos são descritos a seguir em detalhes.

Para a busca tabu empregada pelo algoritmo $\rho$TS-PNLMS, considere a melhor solução do ciclo atual, $k$, como sendo $\rho^k(n)$. O conjunto de soluções vizinhas, $\mathbf{V}^k$, consiste de um vetor de $M$ elementos $v_i^k \in \mathbf{V}^k$, com $i = 1, 2, \ldots, M$, onde cada elemento é um possível valor para $\rho^k(n)$. A Figura 5.4 mostra um exemplo de $\mathbf{V}^k$ formado por $M$ possíveis $\rho^k(n)$.

| $v_1^k$ | $v_2^k$ | $v_3^k$ | | $v_{M-1}^k$ | $v_M$ |
|---|---|---|---|---|---|
| $5 * 10^{-8}$ | $10^{-4}$ | $5 * 10^{-6}$ | ... | $5 * 10^{-5}$ | $10^{-3}$ |

Figura 5.4. Exemplo de $\mathbf{V}^k$ adotado para o algoritmo $\rho$TS-PNLMS.

Em relação ao movimento adotado para criação das novas soluções vizinhas, tem-se que cada elemento de $\mathbf{V}^k$ deve ser aleatoriamente escolhido de um conjunto $\mathbf{H}_\rho$, cujos elementos pertencem a um intervalo $[L_\rho, U_\rho] \subset \mathbb{R}$. Os limites deste intervalo são tais que

$$L_\rho = 10^{-t} \tag{5.55}$$

$$L_\rho < U_\rho \tag{5.56}$$

e

$$U_\rho = 5 * 10^{-4} \tag{5.57}$$

onde $t > 0$ é um número inteiro.

Assim, os elementos $h_j \in \mathbf{H}_\rho$, com $j = 1,2, \ldots, M_\rho$, onde $M_\rho$ é o número de elementos de $\mathbf{H}_\rho$, são obtidos a partir de

$$h_1 = L_\rho \tag{5.58}$$

$$h_{M_\rho} = U_\rho \tag{5.59}$$

e, se $j$ é par e $j < M_\rho$, tem-se

$$h_j = 5 * h_{j-1} \tag{5.60}$$

ou, se $j$ é ímpar e $j > 1$, tem-se

$$h_j = 2 * h_{j-1}. \tag{5.61}$$

Note que $\mathbf{V}^k \subset \mathbf{H}_\rho$. A solução inicial, $\rho^0(n)$, também deve ser escolhida aleatoriamente de $\mathbf{H}_\rho$, de forma que a melhor solução do ciclo anterior, $\rho^{k-1}(n)$, e os elementos de $\mathbf{V}^k$ possuam sempre esta característica comum. Para reduzir o custo computacional de cada ciclo, $k$, da busca tabu, adota-se

$$M \leq \frac{1}{2} M_\rho. \tag{5.62}$$

A estrutura adotada para a lista tabu é a de uma matriz de ordem 2 x $(\mathrm{PT} + 1)$, onde PT é o período tabu. Os movimentos "tabu" são registrados a partir do armazenamento de dois valores em uma dada coluna, $c_{\mathrm{LT}}$, da lista tabu. O primeiro deles é o movimento realizado, ou seja, a melhor solução obtida no ciclo atual, $\rho^k(n)$, escolhida aleatoriamente de $\mathbf{H}_\rho$, que deve ser armazenado na primeira linha de $c_{\mathrm{LT}}$. O segundo valor é igual à soma "$k +$ PT" e deve ser armazenado na segunda linha de $c_{\mathrm{LT}}$. A Figura 5.5 mostra a estrutura adotada para a lista tabu pelo algoritmo $\rho$TS-PNLMS. Neste exemplo, tem-se, no $k$-ésimo ciclo da busca pelo $\rho_{\mathrm{opt}}(n)$ da $n$-ésima iteração do processo de adaptação do algoritmo $\rho$TS-PNLMS, dois movimentos tabu registrados. Especificamente, o movimento caracterizado pela escolha de $v_i^k = 5 * 10^{-9}$ $(1 \leq i \leq M)$ possui condição "tabu" até o ciclo $k = 3$, conforme indicado pelos dados armazenados da Coluna 1. Analogamente para os dados armazenados na Coluna 2. As Colunas 3 a PT + 1 estão disponíveis para armazenar novos movimentos "tabu".

| Coluna 1 | Coluna 2 | Coluna 3 | ⋯ | Coluna PT + 1 |
|---|---|---|---|---|
| $5 * 10^{-9}$ | $10^{-4}$ | 0 | ⋯ | 0 |
| 3 | 5 | 0 | ⋯ | 0 |

Figura 5.5. Estrutura adotada pelo algoritmo $\rho$TS-PNLMS para a lista tabu.

Obviamente, um novo movimento só deve ser armazenado na lista tabu caso ocorra mudança da melhor solução encontrada do ciclo anterior, $k - 1$, para o ciclo atual, $k$, ou seja, se

$$\psi[\rho^k(n)] < \psi[\rho^{k-1}(n)]. \tag{5.63}$$

Note que, se um movimento deixa de ser considerado "tabu" em um dado ciclo da busca, seu respectivo registro deve ser apagado da lista tabu.

Em relação ao critério de aspiração adotado, considere uma dada solução vizinha $v_i^k$ criada a partir de um movimento "tabu". Assim, se

$$\psi\left[v_i^k\right] < \psi\left[v_l^k\right], \text{ com } 1 \leq i, l \leq M \text{ e } i \neq l \tag{5.64}$$

e

$$\psi\left[v_i^k\right] < \psi[\rho^{k-1}(n)] \tag{5.65}$$

então $v_i^k$ tem sua condição "tabu" anulada e passa a vigorar como a melhor solução do ciclo atual $k$, ou seja, $\rho^k(n) = v_i^k$.

Para o critério de convergência, considera-se que o método da busca tabu convergiu se uma mesma solução permanecer como a melhor solução atual durante três ciclos consecutivos. Em outras palavras, caso tenha-se no ciclo atual $k$

$$\rho^k(n) = \rho^{k-1}(n) = \rho^{k-2}(n) \tag{5.66}$$

diz-se que o critério de convergência é satisfeito, ou seja

$$\rho_{\text{opt}}(n) = \rho^k(n). \tag{5.67}$$

O algoritmo $\rho$TS-PNLMS é descrito a seguir em detalhes:

1. Inicialize o vetor de coeficientes do filtro adaptativo ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.68}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.69}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.70}$$

3. Calcule o erro *a priori*

$$e(n) = d(n) - y(n) + v(n). \tag{5.71}$$

4. Atualize o vetor de coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$\rho_{\text{opt}}(n) = \arg \min_{0<\rho(n)<1} \psi(n) \tag{5.72}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.73}$$

Para tal, utilize o seguinte procedimento baseado no método da busca tabu:

I. Faça $k = 0$, escolha uma solução inicial aleatória, $\rho^0(n) \in \mathbf{H}_\rho$ e, a partir de (5.73), calcule $\psi[\rho^0(n)]$.

II. Faça $k = k + 1$, gere $\mathbf{V}^k$ a partir de $\rho^{k-1}(n)$ e verifique quais soluções vizinhas, $v_i^k \in \mathbf{V}^k$, com $i = 1, 2, \ldots, M$, são "tabu".

III. A partir de (5.73), calcule $\psi\left[v_i^k\right]$, com $i = 1, 2, \ldots, M$.

IV. Se existe $v_i^k$ tal que

$$\psi\left[v_i^k\right] < \psi\left[v_j^k\right], \text{ com } 1 \leq i, j \leq M \text{ e } i \neq j \tag{5.74}$$

e

$$\psi\left[v_i^k\right] < \psi[\rho^{k-1}(n)] \tag{5.75}$$

faça

$$\rho^k(n) = v_i^k. \tag{5.76}$$

Caso contrário, faça

$$\rho^k(n) = \rho^{k-1}(n). \tag{5.77}$$

V. Atualize a lista tabu.

VI. Se o critério de convergência de (5.66) for satisfeito, vá para VII. Caso contrário, retorne a II.

VII. Busca tabu convergiu. Retorne

$$\rho_{\text{opt}}(n) = \rho^k(n) \tag{5.78}$$

e, em seguida, calcule $\mathbf{w}(n+1)$ usando

$$f(n) = \rho_{\text{opt}}(n) \max[\delta, \|\mathbf{w}(n)\|_\infty]. \tag{5.79}$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo $\rho$TS-PNLMS.

A seguir, o algoritmo $f$TS-PNLMS é obtido.

### *5.2.4 Algoritmo PNLMS com Otimização do Fator de Ativação baseada no Método da Busca Tabu*

O algoritmo $f$TS-PNLMS usa o método da busca tabu para determinar $f_{\text{opt}}(n)$ que solucione (5.5) em cada iteração do processo de adaptação do algoritmo PNLMS. Os elementos da busca tabu adotados para este algoritmo, tais como o conjunto de soluções vizinhas, $\mathbf{V}^k$, a lista tabu, e os critérios de aspiração e de convergência, são os mesmos do algoritmo $\rho$TS-PNLMS, com algumas modificações.

Ao invés de um conjunto $\mathbf{H}_\rho$, tem-se agora um conjunto $\mathbf{H}_f$, cujos elementos pertencem a um intervalo $\left[L_f, U_f\right] \subset \mathbb{R}$. Os limites deste intervalo são tais que

$$L_f = 10^{-t} \tag{5.80}$$

$$L_f < U_f \tag{5.81}$$

e

$$U_f = 10^{-4} \tag{5.82}$$

sendo $t > 0$ um número inteiro. Os elementos $h_j \in \mathbf{H}_f$, com$j = 1,2,..,M_f$, onde $M_f$ é o total de elementos de $\mathbf{H}_f$, são obtidos a partir de

$$h_1 = L_f \tag{5.83}$$

$$h_{M_f} = U_f \tag{5.84}$$

e, se $j$ é par e $j < M_f$, tem-se

$$h_j = 5 * h_{j-1} \tag{5.85}$$

ou, se $j$ é ímpar e $j > 1$, tem-se

$$h_j = 2 * h_{j-1}. \tag{5.86}$$

Analogamente ao caso do algoritmo $\rho$TS-PNLMS, para o algoritmo $f$TS-PNLMS tem-se que $\mathbf{V}^k \subset \mathbf{H}_f$. A solução inicial, $f^0(n)$, também deve ser escolhida aleatoriamente de $\mathbf{H}_f$, de forma que a melhor solução do ciclo anterior, $f^{k-1}(n)$, e os elementos de $\mathbf{V}^k$ possuam sempre esta característica comum. Para reduzir o custo computacional de cada ciclo, $k$, da busca tabu, adota-se

$$M \leq \frac{1}{2} M_f. \tag{5.87}$$

O algoritmo $f$TS-PNLMS é descrito a seguir em detalhes:

1. Inicialize o vetor de coeficientes do filtro adaptativo ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.88}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.89}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.90}$$

3. Calcule o erro *a priori*

$$e(n) = d(n) - y(n) + v(n). \tag{5.91}$$

4. Atualize o vetor de coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$f_{\mathrm{opt}}(n) = \arg \min_{0<f(n)<1} \psi(n) \tag{5.92}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.93}$$

Para tal, utilize o seguinte procedimento baseado no método da busca tabu:

I. Faça $k = 0$, escolha uma solução inicial aleatória, $f^0(n) \in \mathbf{H}_f$ e, a partir de (5.93), calcule $\psi[f^0(n)]$.

II. Faça $k = k + 1$, gere $\mathbf{V}^k$ a partir de $f^{k-1}(n)$ e verifique quais soluções vizinhas, $v_i^k \in \mathbf{V}^k$, com $i = 1, 2, \ldots, M$, são "tabu".

III. A partir de (5.93), calcule $\psi\left[v_i^k\right]$, com $i = 1, 2, \ldots, M$.

IV. Se existe $v_i^k$ tal que

$$\psi\left[v_i^k\right] < \psi\left[v_j^k\right], \text{ com } 1 \leq i, j \leq M \text{ e } i \neq j \tag{5.94}$$

e

$$\psi\left[v_i^k\right] < \psi[f^{k-1}(n)] \tag{5.95}$$

faça

$$f^k(n) = v_i^k. \tag{5.96}$$

Caso contrário, faça

$$f^k(n) = f^{k-1}(n). \tag{5.97}$$

V. Atualize a lista tabu.

VI. Se o critério de convergência dado por

$$f^k(n) = f^{k-1}(n) = f^{k-2}(n) \tag{5.98}$$

for satisfeito, vá para VII. Caso contrário, retorne a II.

VII. Busca tabu convergiu. Retorne

$$f_{\text{opt}}(n) = f^k(n) \tag{5.99}$$

e, em seguida, calcule $\mathbf{w}(n + 1)$ usando

$$\phi_i(n) = \max\left[f_{\text{opt}}(n), |w_i(n)|\right], \qquad i = 1,2, \ldots, N. \tag{5.100}$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo $f$TS-PNLMS.

A seguir, o algoritmo GS-IPNLMS é obtido.

### *5.2.5 Algoritmo IPNLMS com Otimização do Parâmetro de Proporcionalidade baseada no Método da Razão Áurea*

O algoritmo GS-IPNLMS usa o método da razão áurea para determinar $\alpha_{\text{opt}}(n)$ que solucione (5.6) em cada instante, $n$, do processo de adaptação do algoritmo IPNLMS. Este algoritmo pode ser executado conforme os seguintes passos:

1. Inicialize o vetor de coeficientes do filtro ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.101}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.102}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.103}$$

3. Calcule o sinal de erro

$$e(n) = d(n) - y(n) + v(n). \tag{5.104}$$

4. Atualize o vetor de coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$\alpha_{\mathrm{opt}}(n) = \arg \min_{-1 \le \alpha(n) < 1} \psi(n) \tag{5.105}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{T}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.106}$$

Para tal, utilize o seguinte procedimento baseado no método da razão áurea:

I. Faça

$$k = 1 \tag{5.107}$$

$$a = -1 \tag{5.108}$$

$$b = +1 \tag{5.109}$$

defina a tolerância $tol$ para convergência da razão áurea e calcule

$$\alpha_1^k(n) = a(1-\tau) + b\tau \tag{5.110}$$

$$\alpha_2^k(n) = a\tau + b(1-\tau) \tag{5.111}$$

II. A partir de (3.1) e (3.23), obtenha $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$ usando $\alpha_1^k(n)$ e $\alpha_2^k(n)$, respectivamente.

III. A partir de (5.106), obtenha $\psi[\alpha_1^k(n)]$ e $\psi[\alpha_2^k(n)]$ usando $\mathbf{w}_1^k(n+1)$ e $\mathbf{w}_2^k(n+1)$, respectivamente. Se

$$\left|\psi[\alpha_1^k(n)] - \psi[\alpha_2^k(n)]\right| < tol \tag{5.112}$$

Vá para V. Caso contrário, vá para IV.

IV. Se

$$\psi[\alpha_1^k(n)] > \psi[\alpha_2^k(n)] \tag{5.113}$$

faça

$$k = k + 1 \tag{5.114}$$

$$a = \alpha_1^{k-1}(n) \tag{5.115}$$

$$\alpha_1^k(n) = \alpha_2^{k-1}(n) \tag{5.116}$$

$$\alpha_2^k(n) = a\tau + b(1-\tau) \tag{5.117}$$

e retorne para II. Caso contrário, faça

$$k = k + 1 \tag{5.118}$$

$$b = \alpha_2^{k-1}(n) \tag{5.119}$$

$$\alpha_2^k(n) = \alpha_1^{k-1}(n) \tag{5.120}$$

$$\alpha_1^k(n) = a(1-\tau) + b\tau \tag{5.121}$$

e retorne para II.

V. Razão áurea convergiu. Faça

$$\alpha_{\text{opt}}(n) = \alpha_1^k(n) \tag{5.122}$$

ou

$$\alpha_{\text{opt}}(n) = \alpha_2^k(n). \tag{5.123}$$

Em seguida, calcule $\mathbf{w}(n+1)$ usando

$$g_i(n) = \left[1 - \alpha_{\text{opt}}(n)\right]\frac{1}{2N} + \left[1 + \alpha_{\text{opt}}(n)\right]\frac{|w_i(n)|}{2\|\mathbf{w}(n)\|_1 + \varepsilon}. \tag{5.124}$$

com $i = 1,2,\ldots,N$.

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo GS-IPNLMS.

Sendo $\alpha_1^k(n)$ e $\alpha_2^k(n)$ candidatos a $\alpha_{\text{opt}}(n)$ no ciclo $k$ do método da razão áurea.

A seguir, o algoritmo TS-IPNLMS é obtido.

### *5.2.6 Algoritmo IPNLMS com Otimização do Parâmetro de Proporcionalidade baseada no Método da Busca Tabu*

O algoritmo TS-IPNLMS usa o método da busca tabu para determinar $\alpha_{\text{opt}}(n)$ que solucione (5.6) em cada iteração do processo de adaptação do algoritmo IPNLMS. Para tal, faz-se necessário definir as estruturas dos elementos integrantes da busca tabu, como o conjunto de soluções vizinhas, a lista tabu e o critério de aspiração, para o problema em questão. Estes elementos são descritos a seguir em detalhes.

Para a metodologia proposta, considere o conjunto das soluções vizinhas, $\mathbf{V}^k$, e a melhor solução, $\alpha^k(n)$, do ciclo atual, $k$, da busca tabu. O conjunto $\mathbf{V}^k$ pode ser representado por um vetor de $M$ elementos, cada um contendo um possível valor para $\alpha^k(n)$. Assim, uma solução

vizinha $v_i^k \in \mathbf{V}^k$, com $i = 1, 2, \ldots, M$, é um número decimal com $D$ casas decimais. Um exemplo de $\mathbf{V}^k$, formado por $M$ possíveis $\alpha^k(n)$ com duas casas decimais cada ($D = 2$), é mostrado na Figura 5.6.

| $v_1^k$ | $v_2^k$ | $v_3^k$ | | $v_{M-1}^k$ | $v_M$ |
|---|---|---|---|---|---|
| 0,72 | −0,15 | 0,0 | ... | 0,97 | −1,0 |

Figura 5.6. Exemplo de um conjunto $\mathbf{V}^k$ de soluções vizinhas adotado para o algoritmo TS-IPNLMS.

O movimento utilizado para criar as novas soluções vizinhas consiste de uma soma simples. Esta soma tem como parcelas a melhor solução do ciclo anterior, $\alpha^{k-1}(n)$, e um número escolhido aleatoriamente de um conjunto $\mathbf{H}_\alpha$ de números decimais, com $D$ casas decimais cada, pertencentes ao intervalo $[L_\alpha, U_\alpha] \subset \mathbb{R}$. Como exemplo, toma-se $L_\alpha = -0{,}2$, $U_\alpha = -0{,}2$ e $D = 1$. Então, tem-se que $\mathbf{H}_\alpha = \{-0{,}2\ , -0{,}1\ , 0\ , 0{,}1\ , 0{,}2\}$. Considerando $\alpha^{k-1}(n) = 0{,}7$, duas possíveis novas soluções vizinhas são

$$v_j^k = 0{,}7 + 0{,}1 = 0{,}8 \tag{5.125}$$

e

$$v_l^k = 0{,}7 + (-0{,}2) = 0{,}5 \tag{5.126}$$

com $1 \leq j, l \leq M$.

A estrutura adotada para a lista tabu é a de uma matriz de ordem 2 x (PT + 1), onde PT é o período tabu. Os movimentos "tabu" Os movimentos "tabu" são registrados a partir do armazenamento de três valores em uma dada coluna, $c_{\mathrm{LT}}$, da lista tabu. Os dois primeiros identificam o movimento realizado, ou seja, $\alpha^{k-1}(n)$ e o valor escolhido aleatoriamente de $\mathbf{H}_\alpha$. Estes valores são gravados na primeira e segunda linhas de $c_{\mathrm{LT}}$, respectivamente. Já o terceiro valor corresponde à soma "$k$ + PT", e deve ser armazenado na terceira linha de $c_{\mathrm{LT}}$. Se um dado movimento deixa de ser considerado "tabu" em um ciclo $k$ da busca, seu respectivo registro deve ser apagado da lista tabu. A Figura 5.7 mostra a estrutura adotada para a lista tabu do algoritmo TS-IPNLMS. Neste exemplo, tem-se, no $k$-ésimo ciclo da busca pelo $\alpha_{\mathrm{opt}}(n)$ da $n$-ésima iteração do processo de adaptação do algoritmo TS-IPNLMS, dois movimentos tabu registrados. Especificamente, o movimento caracterizado pela soma de $\alpha^{k-1}(n) = +0{,}7$ com $v_i^k = -0{,}1$ ($1 \leq i \leq M$) possui condição "tabu" até o ciclo $k = 4$, conforme indicado pelos dados armazenados da Coluna 1. Analogamente para os dados armazenados na Coluna 2. As Colunas 3 a PT + 1 estão disponíveis para armazenar novos movimentos "tabu".

| Coluna 1 | Coluna 2 | Coluna 3 | ⋯ | Coluna PT + 1 |
|---|---|---|---|---|
| +0,7 | −0,3 | 0 | ⋯ | 0 |
| −0,1 | +0,5 | 0 | ⋯ | 0 |
| 4 | 7 | 0 | ⋯ | 0 |

Figura 5.7. Estrutura adotada para a lista tabu do algoritmo TS-IPNLMS.

Em relação ao critério de aspiração adotado, considere uma dada solução vizinha $v_i^k$ criada a partir de um movimento "tabu". Assim, se

$$\psi\left[v_i^k\right] < \psi\left[v_j^k\right], \text{ com } 1 \leq i,j \leq M \text{ e } i \neq j \tag{5.127}$$

e

$$\psi\left[v_i^k\right] < \psi[\alpha^{k-1}(n)] \tag{5.128}$$

então $v_i^k$ tem sua condição "tabu" anulada e passa a vigorar como a melhor solução do ciclo atual $k$, ou seja, $\alpha^k(n) = v_i^k$.

Para o critério de convergência, considera-se que o método da busca tabu converge quando uma mesma solução permanece como a melhor solução atual durante três ciclos consecutivos. Em outras palavras, caso tenha-se no ciclo atual $k$ a seguinte condição atendida

$$\alpha^k(n) = \alpha^{k-1}(n) = \alpha^{k-2}(n) \tag{5.129}$$

diz-se que o critério de convergência é satisfeito, ou seja

$$\alpha_{\text{opt}}(n) = \alpha^k(n). \tag{5.130}$$

A seguir, o algoritmo TS-IPNLMS é descrito em detalhes:

1. Inicialize o vetor de coeficientes do filtro ($n = 0$)

$$\mathbf{w}(0) = \mathbf{0}. \tag{5.131}$$

2. Obtenha os dados de entrada e saída da planta

$$d(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{p}(n) \tag{5.132}$$

e do filtro adaptativo

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n). \tag{5.133}$$

3. Calcule o sinal de erro

$$e(n) = d(n) - y(n) + v(n). \tag{5.134}$$

4. Atualize dos coeficientes do filtro adaptativo.

Nesta etapa, obtenha

$$\alpha_{\text{opt}}(n) = \arg \min_{-1 \leq \alpha(n) < 1} \psi(n) \tag{5.135}$$

onde

$$\psi(n) = 10 \log_{10}\{[d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n+1) + v(n)]^2\}. \tag{5.136}$$

Para tal, utilize o seguinte procedimento baseado no método da busca tabu:

I. Faça $k = 0$, escolha uma solução inicial aleatória, $-1 \leq \alpha^0(n) < 1$, e a partir de (5.136), calcule $\psi[\alpha^0(n)]$.

II. Faça $k = k + 1$, gere $\mathbf{V}^k$ a partir de $\alpha^{k-1}(n)$ e verifique quais soluções vizinhas $v_i^k \in \mathbf{V}^k$, com $i = 1, 2, \ldots, M$, são "tabu".

III. A partir de (5.136), calcule $\psi\left[v_i^k\right]$, com $i = 1, 2, \ldots, M$.

IV. Se existe $v_i^k$ tal que

$$\psi\left[v_i^k\right] < \psi\left[v_j^k\right], \text{ com } 1 \leq i, j \leq M \text{ e } i \neq j \tag{5.137}$$

e

$$\psi\left[v_i^k\right] < \psi[\alpha^{k-1}(n)] \tag{5.138}$$

faça

$$\alpha^k(n) = v_i^k. \tag{5.139}$$

Caso contrário, faça

$$\alpha^k(n) = \alpha^{k-1}(n). \tag{5.140}$$

V. Atualize a lista tabu.

VI. Se o critério de convergência de (5.129) for satisfeito, vá para VII. Caso contrário, retorne a II.

VII. Busca tabu convergiu. Retorne

$$\alpha_{\text{opt}}(n) = \alpha^k(n) \tag{5.141}$$

e, em seguida, calcule $\mathbf{w}(n+1)$ usando

$$g_l(n) = \left[1 - \alpha_{\text{opt}}(n)\right]\frac{1}{2N} + \left[1 + \alpha_{\text{opt}}(n)\right]\frac{|w_l(n)|}{2\|\mathbf{w}(n)\|_1 + \varepsilon}. \tag{5.142}$$

5. Vá para a iteração seguinte ($n = n + 1$) e repita os passos 2, 3 e 4 até o fim do processo de adaptação do algoritmo TS-IPNLMS.

A seguir, é feita uma breve discussão a respeito da complexidade computacional dos algoritmos propostos.

## 5.3 Complexidade Computacional dos Algoritmos Propostos

A complexidade computacional dos algoritmos propostos neste trabalho de pesquisa está diretamente ligada aos métodos de otimização adotados, os quais são executados em cada iteração do processo de adaptação. O algoritmo $\rho$GS-PNLMS, por exemplo, requer $(4N + 2)K_{\text{G}} + 4N - 1$ operações de adição, $(5N + 4)(K_{\text{G}} + 1)$ operações de multiplicação, $(N + 1)(K_{\text{G}} + 1)$ operações de divisão e $(2N + 1)K_{\text{G}} + 2N$ operações de comparação a mais do que o algoritmo PNLMS padrão em cada iteração. A variável $K_{\text{G}}$ é o total de ciclos do método da razão áurea em uma dada iteração do algoritmo $\rho$GS-IPNLMS. Já o algoritmo TS-IPNLMS requer $(5N + 2)(MK_{\text{T}} + 1)$ adições, $(6N + 4)(MK_{\text{T}} + 1)$ multiplicações, $(N + 2)(MK_{\text{T}} + 1)$ divisões e $(2M + 1)K_{\text{T}}$ comparações a mais do que o algoritmo IPNLMS original. A variável $K_{\text{T}}$ é o total de ciclos do método da busca tabu executados em uma dada iteração do algoritmo TS-IPNLMS. Por outro lado, os algoritmos $f$GS-PNLMS e $f$TS-PNLMS não necessitam da determinação de $\rho \max[\delta, \|\mathbf{w}(n)\|_\infty]$, resultando assim em uma economia

de $N$ operações de comparação e uma operação de multiplicação, se comparado com os algoritmos PNLMS e $\rho$GS-PNLMS. A Tabela 5.1 mostra as complexidades computacionais dos algoritmos NLMS, PNLMS, IPNLMS e dos algoritmos propostos neste trabalho.

Tabela 5.1. Complexidade computacional dos algoritmos propostos.

| **Algoritmo** | **Adições** | **Multiplicações** | **Divisões** | **Comparações** |
|---|---|---|---|---|
| NLMS | $3N$ | $3N+1$ | $1$ | $0$ |
| PNLMS | $4N$ | $5N+2$ | $N+1$ | $2N$ |
| IPNLMS | $5N+2$ | $6N+3$ | $N+2$ | $0$ |
| $\rho$GS-PNLMS | $(4N+2)K_{\mathrm{G}}+7N$ | $(5N+4)K_{\mathrm{G}}+10N$ | $(N+1)K_{\mathrm{G}}+2N$ | $(2N+1)K_{\mathrm{G}}+4N$ |
| $\rho$TS-PNLMS | $4N(MK_{\mathrm{T}}+2)$ | $(5N+3)MK_{\mathrm{T}}+5N$ | $(N+1)(MK_{\mathrm{T}}+2)$ | $(N+2)MK_{\mathrm{T}}+5N$ |
| $f$GS-PNLMS | $(4N+1)K_{\mathrm{G}}+8N$ | $(5N+3)K_{\mathrm{G}}+8N$ | $(N+1)(K_{\mathrm{G}}+2)$ | $(N+1)K_{\mathrm{G}}+2N$ |
| $f$TS-PNLMS | $4N(MK_{\mathrm{T}}+2)$ | $(5N+2)MK_{\mathrm{T}}+10N$ | $(N+1)(MK_{\mathrm{T}}+2)$ | $(N+2)MK_{\mathrm{T}}+2N$ |
| GS-IPNLMS | $(4N+5)K_{\mathrm{G}}+8N$ | $(5N+6)K_{\mathrm{G}}+10N$ | $3K_{\mathrm{G}}+3$ | $2K_{\mathrm{G}}-1$ |
| TS-IPNLMS | $(5N+2)MK_{\mathrm{T}}+10N$ | $(6N+4)MK_{\mathrm{T}}+12N$ | $(N+2)(MK_{\mathrm{T}}+2)$ | $(2M+1)K_{\mathrm{T}}$ |

No capítulo seguinte, o desempenho dos novos algoritmos propostos é avaliado.

# Capítulo 6

# Avaliação de Desempenho dos Algoritmos Propostos

Para avaliar o desempenho dos algoritmos propostos, simulações de Monte Carlo (média de 100 realizações independentes) são realizadas, considerando um problema de identificação de sistemas. Tais simulações têm como objetivos comparar o desempenho dos algoritmos propostos com o dos algoritmos PNLMS, IPNLMS e SC-PNLMS em termos de velocidade de convergência, resposta a perturbações e rastreamento dos coeficientes da planta a ser identificada. São considerados dois cenários para a realização das simulações numéricas. O primeiro deles é o mesmo especificado no Capítulo 3 (veja subseção 3.2.1), consistindo de uma planta esparsa, $\mathbf{p}$, de esparsidade $S(\mathbf{p}) = 0{,}9435$, com coeficientes ativos iguais a $\{0{,}1\ , 1{,}0\ , -0{,}5\ , 0{,}1\}$ localizados nas posições $\{1, 30, 35, 85\}$, respectivamente. O segundo consiste de uma planta de esparsidade variável constituída a partir de dados da planta $\mathbf{p}$ e de um processo markoviano de 1ª ordem (Loganathan, Habets & Naylor, 2010). Para a entrada, $x(n)$, considera-se o sinal correlacionado AR(2), com $\chi = 10$, dado por (3.15). O desempenho de cada um dos algoritmos propostos é avaliado considerando exemplos comparativos envolvendo a planta $\mathbf{p}$. Primeiramente, os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS são avaliados. Logo após, é verificado o desempenho dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS. Por fim, avalia-se o desempenho dos algoritmos GS-IPNLMS e TS-IPNLMS.

## 6.1 Algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS

Nesta seção, o desempenho dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS é comparado com o dos algoritmos PNLMS e IPNLMS em termos da velocidade de convergência. Para tal, dois exemplos comparativos são considerados a seguir. O primeiro considera uma inversão nos coeficientes da planta esparsa, $\mathbf{p}$, no instante $n = 2000$. Já o segundo considera um deslocamento dos coeficientes de $\mathbf{p}$ em $n = 2000$.

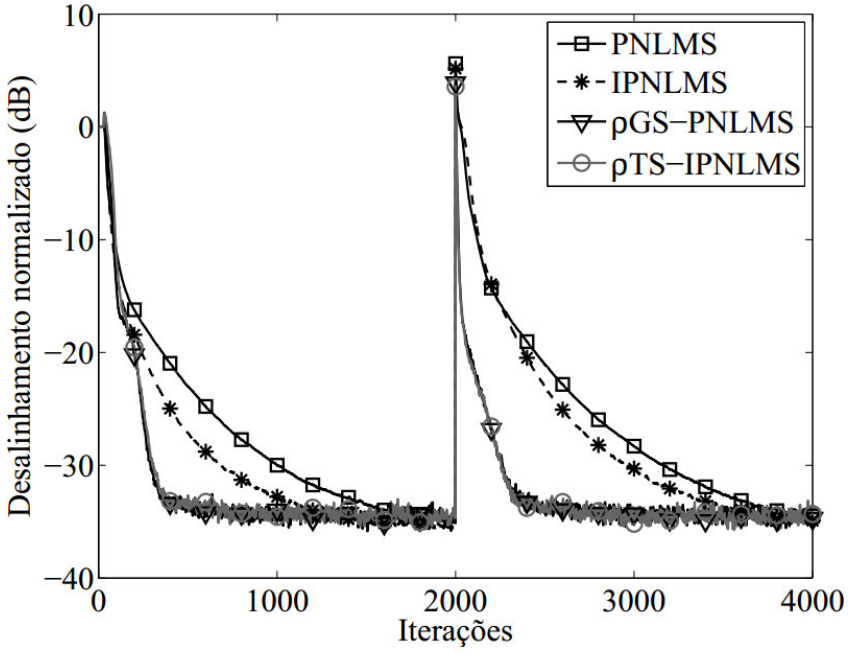
### *6.1.1 Exemplo 1*

Este exemplo tem o objetivo de comparar o desempenho dos algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS em termos de velocidade de convergência e resposta a uma perturbação ocorrida no instante $n = 2000$, quando o vetor de coeficientes da planta muda de $\mathbf{p}$ para $-\mathbf{p}$. Esta inversão na resposta impulsiva da não altera o seu grau de esparsidade.

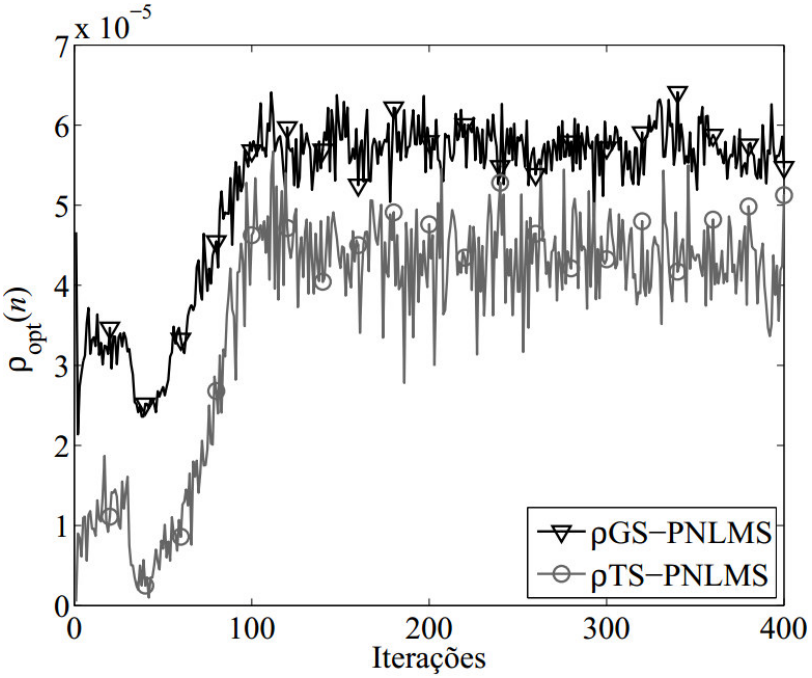
A Figura 6.1 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\rho_{\text{opt}}(n)$, para os algoritmos $\rho$GS-

PNLMS e $\rho$TS-PNLMS, e a quantidade de ciclos necessária para os métodos da razão áurea e da busca tabu atingirem a convergência em cada iteração do processo de adaptação dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS, respectivamente. Já a Figura 6.2 mostra o comportamento dos coeficientes ativos e inativos do algoritmo $\rho$GS-PNLMS, enquanto que a Figura 6.3 mostra o comportamento dos coeficientes ativos e inativos do algoritmo $\rho$TS-PNLMS. As Figuras 6.4 e 6.5 mostram o comportamento dos coeficientes $w_{35}(n)$ e $w_{85}(n)$, $w_2(n)$ e $w_{97}(n)$, respectivamente, antes e após ocorrer a perturbação em $\mathbf{p}$, para os algoritmos considerados. Adota-se um parâmetro de passo igual a $\mu = 0{,}5$ para os algoritmos considerados neste exemplo, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para o algoritmo PNLMS, considera-se $\delta = 0{,}01$ e $\rho = 0{,}05$, e para o algoritmo IPNLMS, adota-se $\alpha = 0{,}0$ (Benesty & Gay, 2002), (Souza, Tobias, Seara & Morgan, 2010). Para o algoritmo $\rho$GS-PNLMS, considera-se $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$. Já para o algoritmo $\rho$TS-PNLMS, adota-se $L_\rho = 10^{-15}$, $M = 6$ e PT $= 3$.

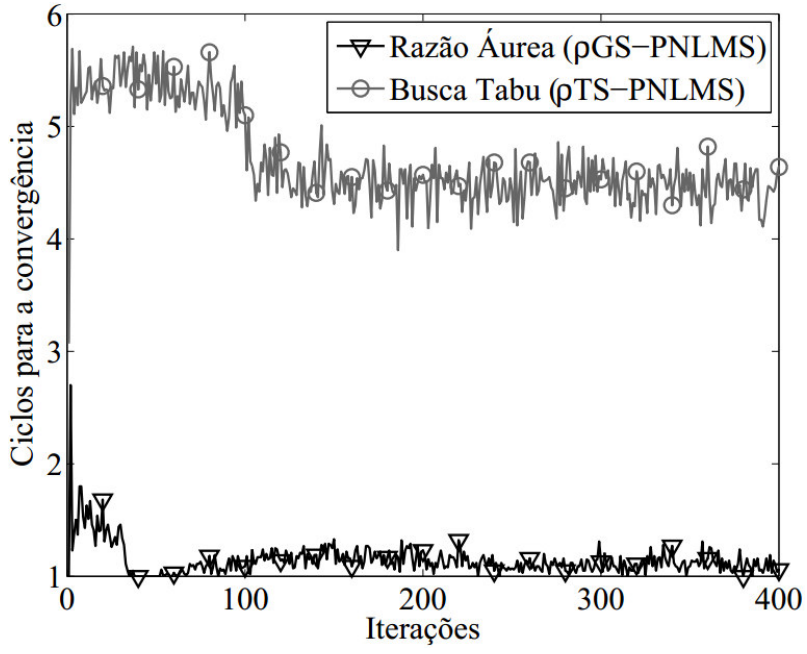
Note das curvas da Figura 6.1(a) que os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS alcançam velocidades de convergência mais rápidas do que as dos algoritmos PNLMS e IPNLMS, mesmo após ocorrer a perturbação na planta. Note também, das curvas da Figura 6.1(b), que $\rho_{\text{opt}}(n)$ apresenta comportamento semelhante para os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS. Para o algoritmo $\rho$GS-PNLMS, $\rho_{\text{opt}}(n)$ inicia em $4{,}7 * 10^{-5}$ no instante $n = 1$ e, em menos de 200 iterações, estabiliza-se em torno de $5{,}9 * 10^{-5}$, mesmo após a perturbação. Já para o algoritmo $\rho$TS-PNLMS, após iniciar em $5{,}0 * 10^{-7}$ no instante $n = 1$, $\rho_{\text{opt}}(n)$ estabiliza-se, em menos de 200 iterações, em torno de $4{,}4 * 10^{-5}$, mesmo após ocorrer a inversão nos coeficientes da planta. Observe das curvas da Figura 6.1(c) que, após $n = 100$, os métodos da razão áurea (algoritmo $\rho$GS-PNLMS) e da busca tabu (algoritmo $\rho$TS-PNLMS) alcançam médias inferiores a 2 e 5 ciclos, respectivamente, para atingir a convergência em cada iteração do processo de adaptação.




(a)

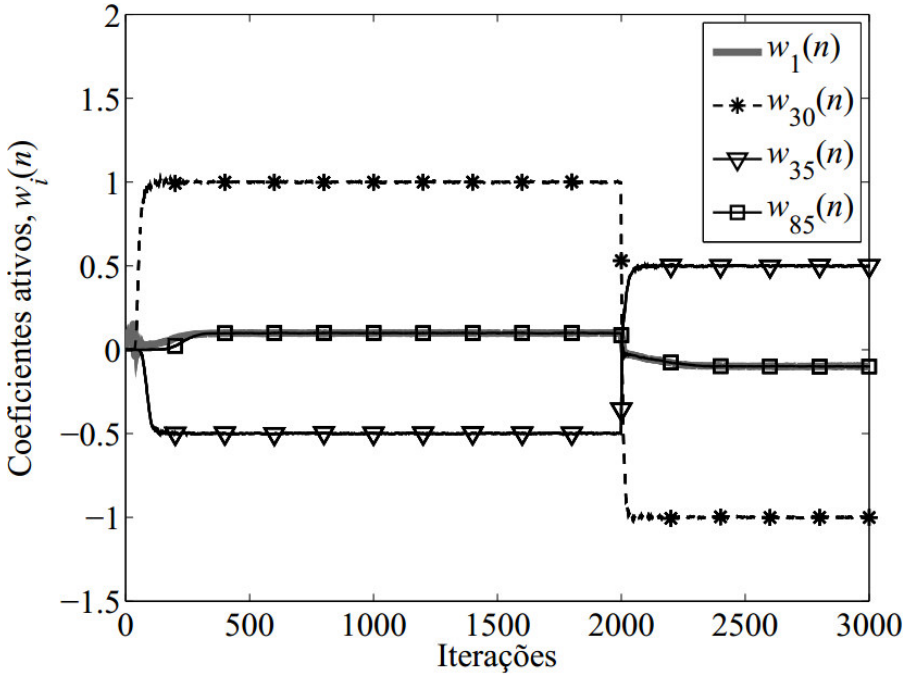



(b)

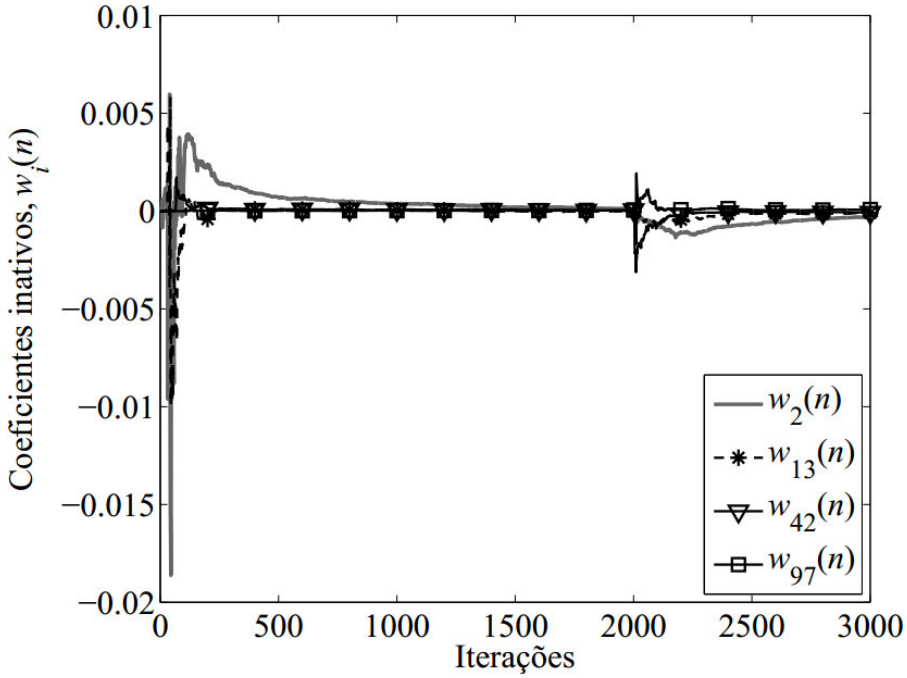



(c)

Figura 6.1. Comportamento dos algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS considerando uma inversão nos coeficientes da planta $\mathbf{p}$ em $n = 2000$. (a) Desalinhamento normalizado dos algoritmos PNLMS (com $\delta = 0{,}01$ e $\rho = 0{,}05$), IPNLMS (com $\alpha = 0{,}0$), $\rho$GS-PNLMS (com $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$) e $\rho$TS-PNLMS (com $L_\rho = 10^{-15}$, $M = 6$ e PT $= 3$). (b) Parâmetro de proporcionalidade ótimo, $\rho_{\text{opt}}(n)$, dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS. (c) Quantidade de ciclos para os métodos da razão áurea ($\rho$GS-PNLMS) e da busca tabu ($\rho$TS-PNLMS) atingirem a convergência.

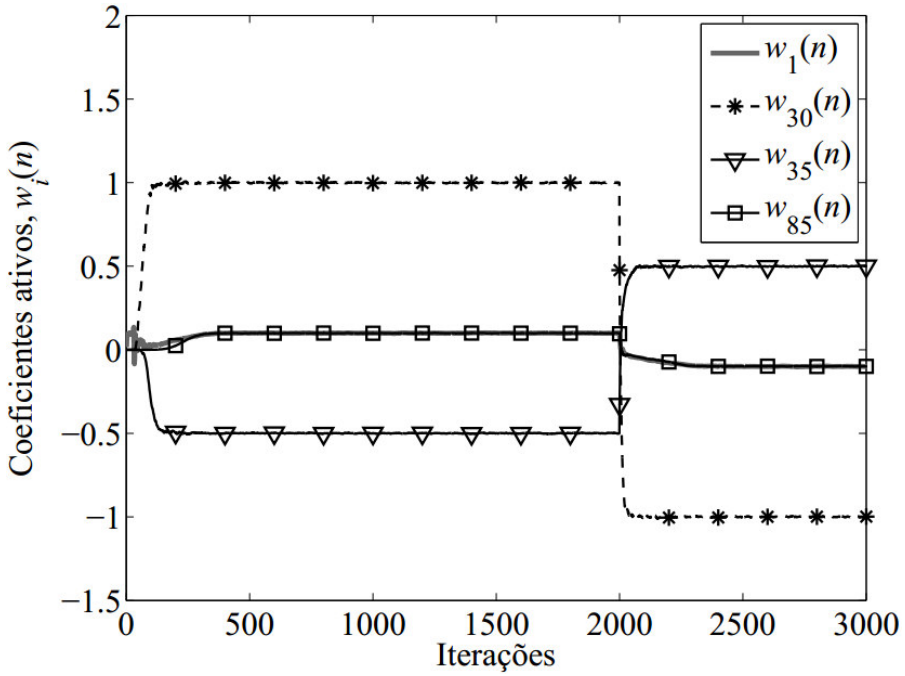



(a)

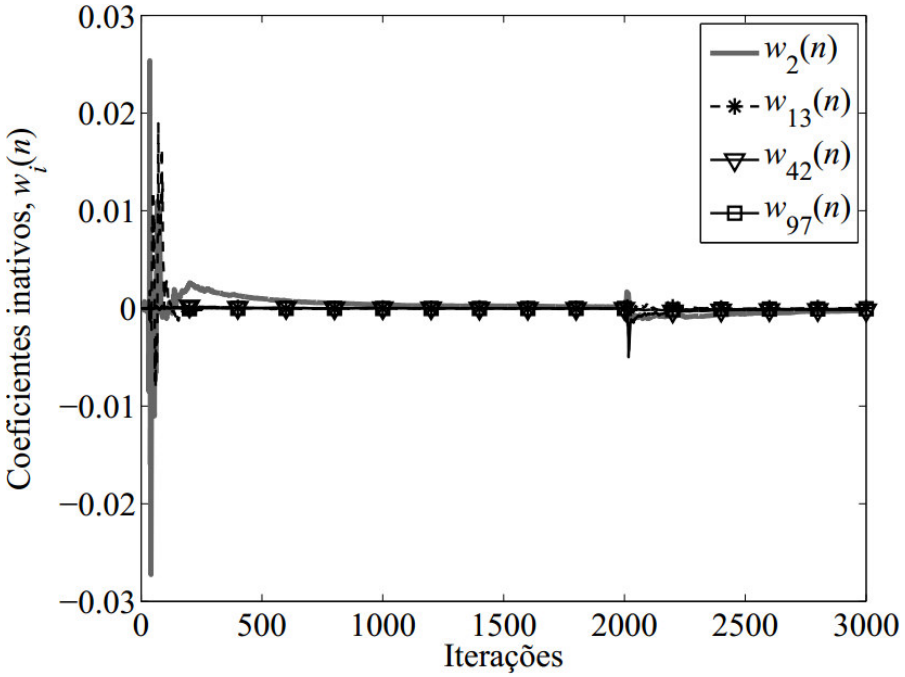



(b)

Figura 6.2. Comportamento dos coeficientes do algoritmo $\rho$GS-PNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. (a) Coeficientes ativos. (b) Coeficientes inativos.

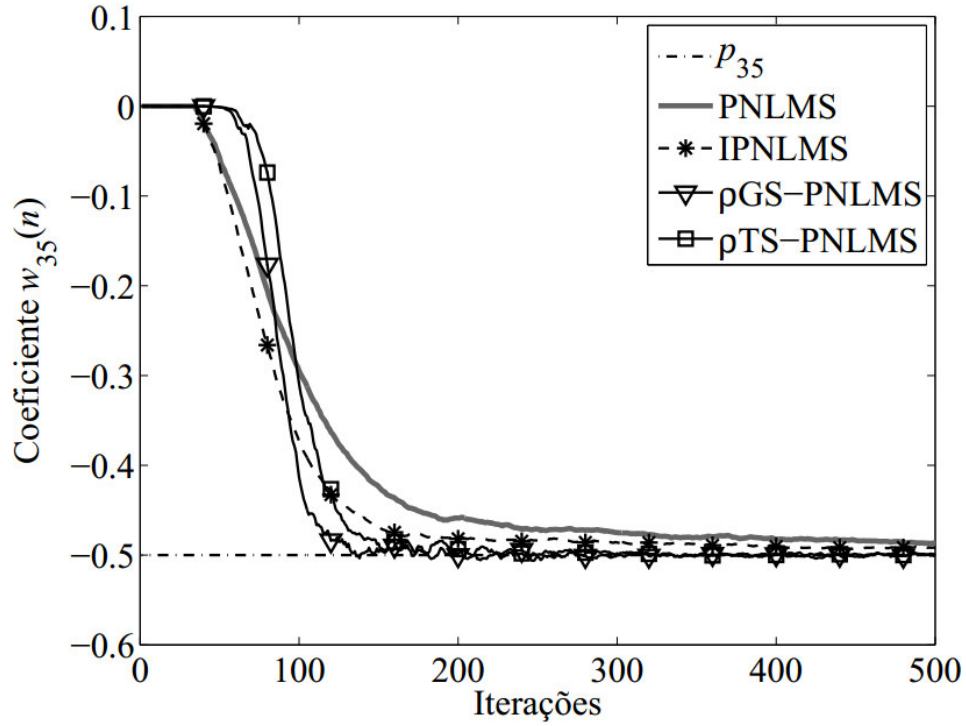



(a)

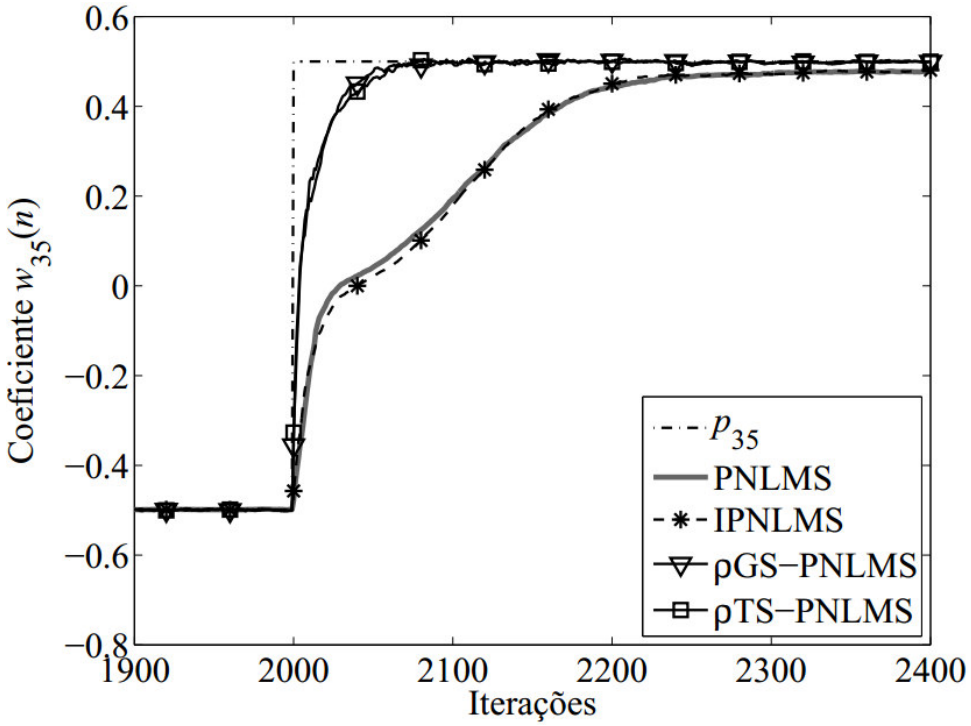



(b)

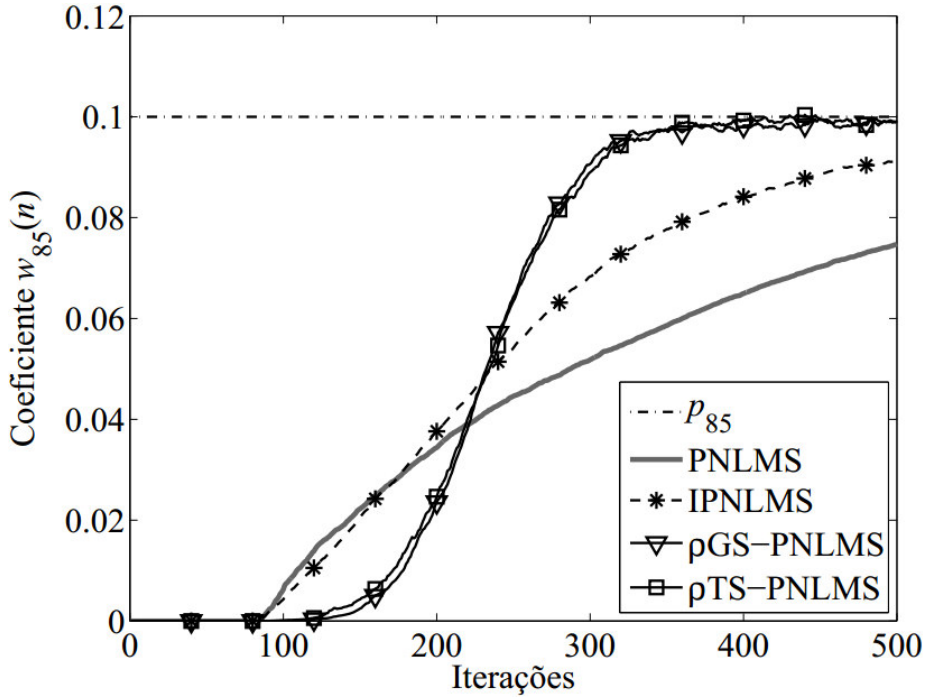
Figura 6.3. Comportamento dos coeficientes do algoritmo $\rho$TS-PNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. (a) Coeficientes ativos. (b) Coeficientes inativos.


0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
0
100
200
300
400
500
Iterações
Coeficiente $w_{35}(n)$
$p_{35}$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

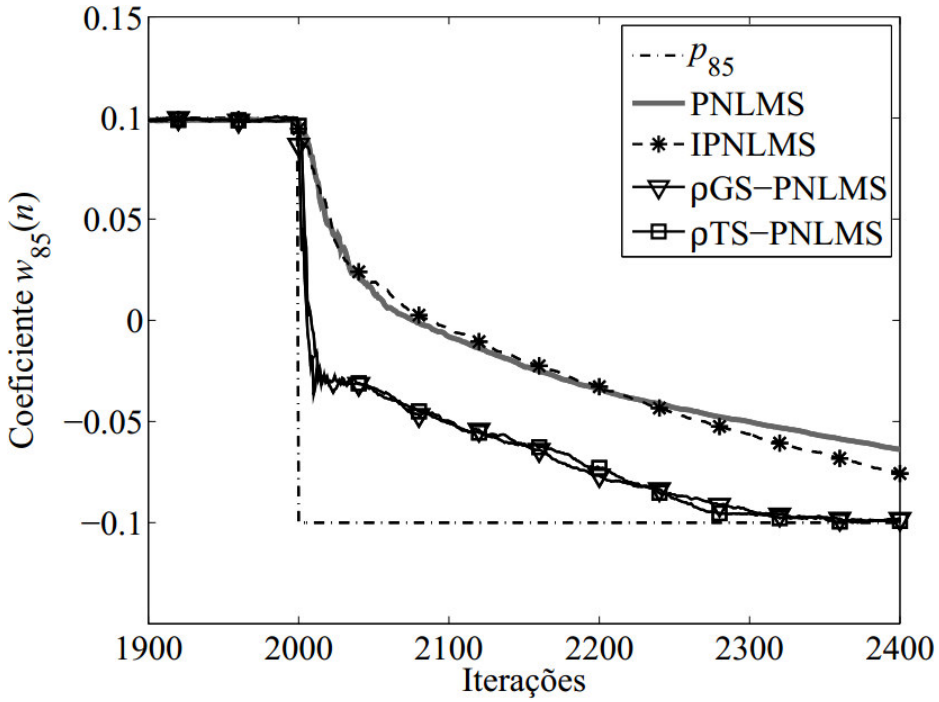
(a)


0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{35}(n)$
$p_{35}$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

(b)

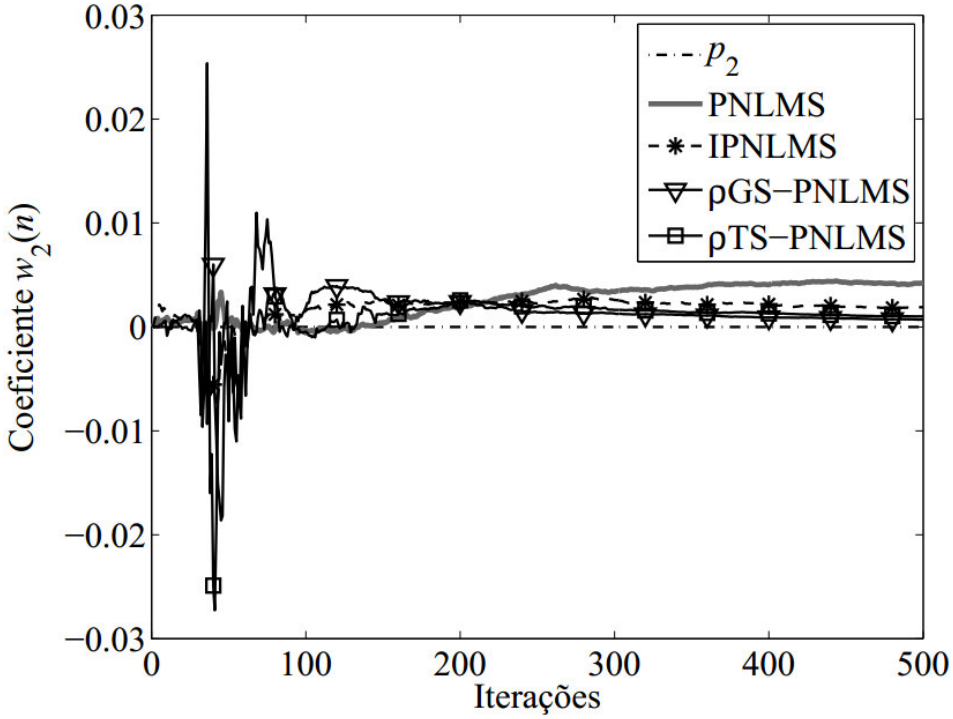



(c)

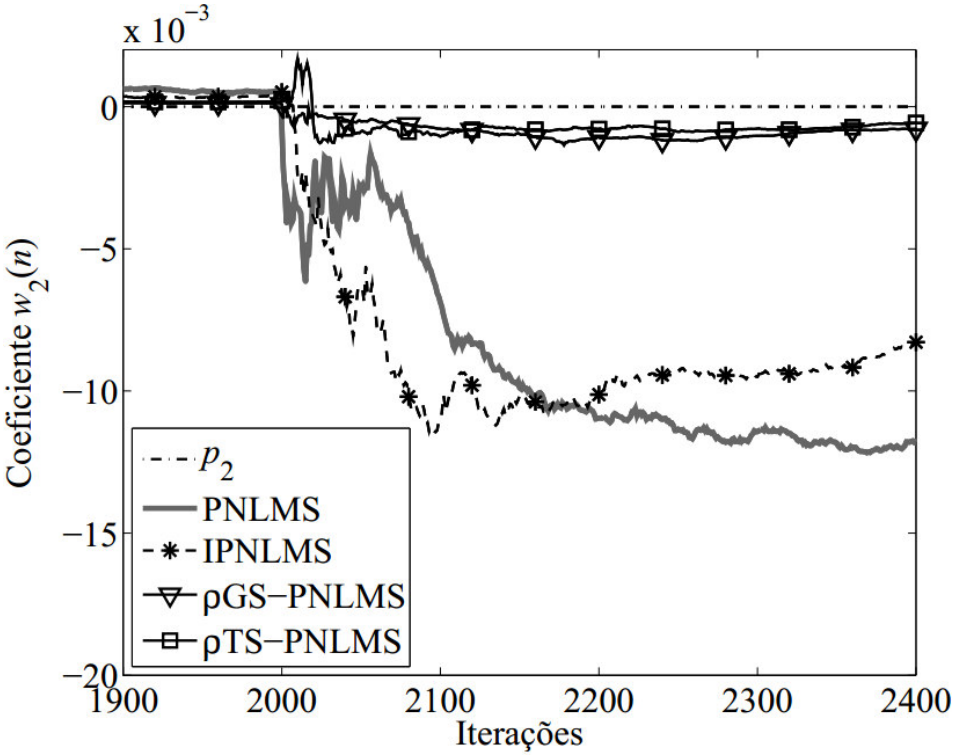



(d)

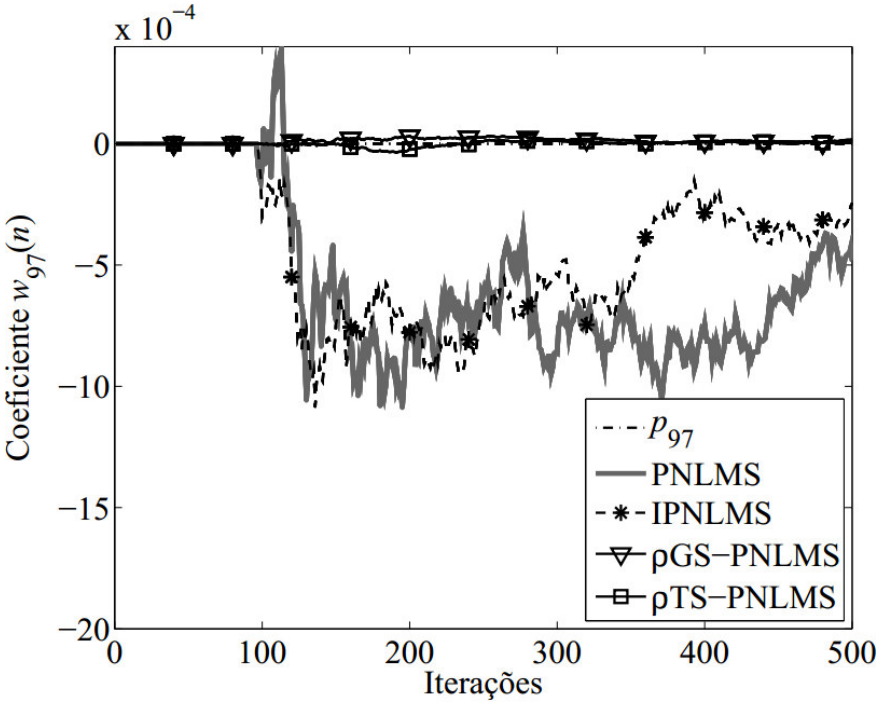
Figura 6.4. Comportamento dos coeficientes ativos $w_{35}(n)$ e $w_{85}(n)$ para os algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS, considerando uma perturbação na planta $\mathbf{p}$ em $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. (a) Coeficiente $w_{35}(n)$ antes da perturbação. (b) Coeficiente $w_{35}(n)$ após a perturbação. (c) Coeficiente $w_{85}(n)$ antes da perturbação. (d) Coeficiente $w_{85}(n)$ após a perturbação.


0.03
0.02
0.01
0
−0.01
−0.02
−0.03
0
100
200
300
400
500
Iterações
Coeficiente $w_2(n)$
$p_2$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

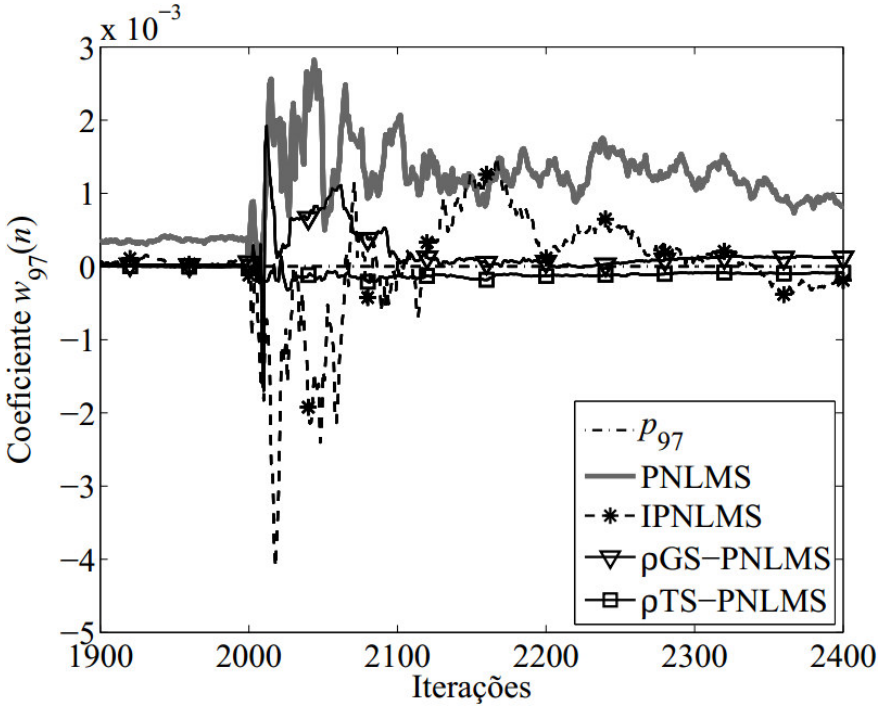
(a)


x 10^−3
0
−5
−10
−15
−20
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_2(n)$
$p_2$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

(b)




(c)




(d)

Figura 6.5. Comportamento dos coeficientes inativos $w_2(n)$ e $w_{97}(n)$ para os algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS, considerando uma perturbação na planta $\mathbf{p}$ em $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. (a) Coeficiente $w_2(n)$ antes da perturbação. (b) Coeficiente $w_2(n)$ após a perturbação. (c) Coeficiente $w_{97}(n)$ antes da perturbação. (d) Coeficiente $w_{97}(n)$ após a perturbação.

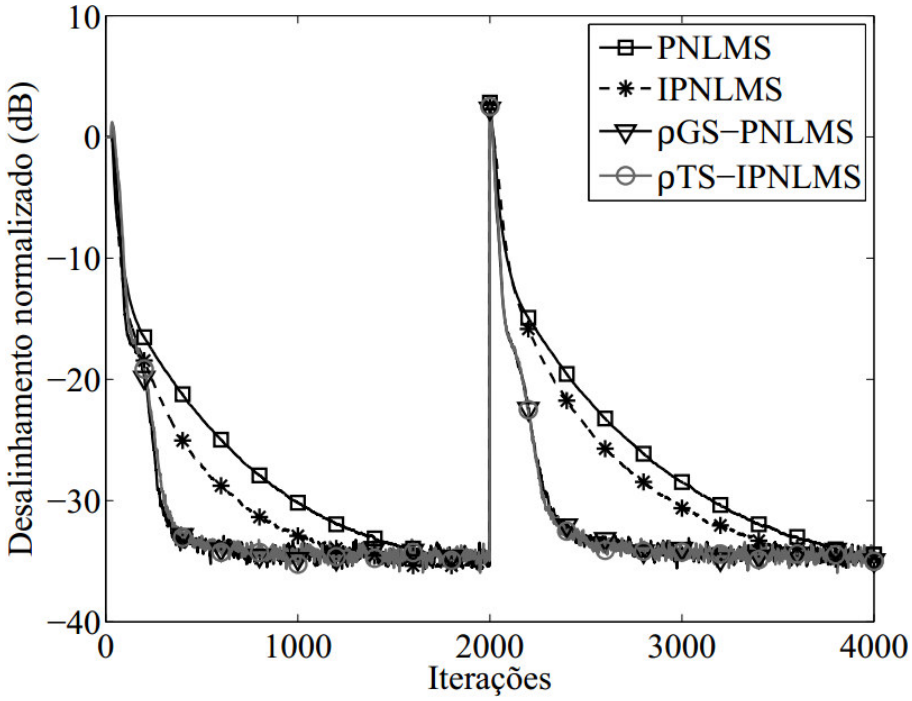
Note das curvas das Figuras 6.2 e 6.3 que os coeficientes ativos e inativos dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS convergem rapidamente para os valores de referência da planta $\mathbf{p}$, mesmo após ocorrer a perturbação. Note também, das curvas das Figuras 6.4 e 6.5, que os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS demonstram capacidade de rastreamento superior à dos algoritmos PNLMS e IPNLMS, tanto para coeficientes ativos como inativos, mesmo após a inversão nos coeficientes de $\mathbf{p}$.

*6.1.2 Exemplo 2*

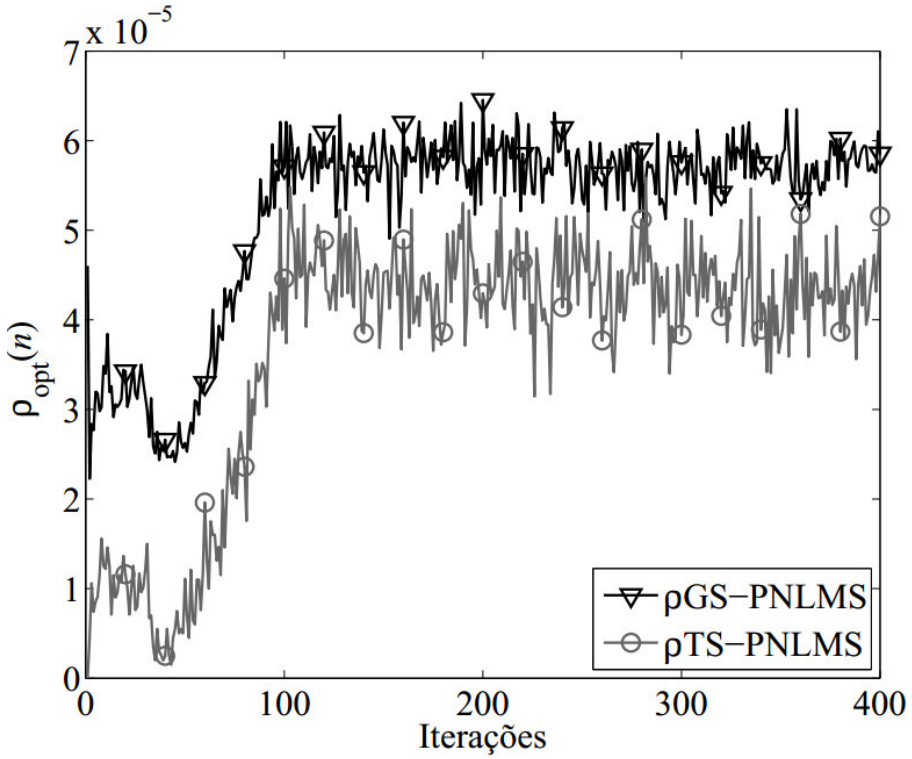
Neste exemplo, os algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS são novamente comparados em termos de velocidade de convergência e resposta a uma perturbação que não altera o grau de esparsidade da planta. Desta vez, no instante $n = 2000$, o vetor de coeficientes de $\mathbf{p}$ é deslocado de 12 amostras para a direita, alterando-se, dessa forma, as posições de todos os coeficientes ativos. Assim, os coeficientes ativos da planta esparsa $\mathbf{p}$, cujos valores são iguais a $\{0{,}1\ ,\ 1{,}0\ ,\ -0{,}5\ ,\ 0{,}1\}$, são movidos das posições $\{1, 30, 35, 85\}$ para as posições $\{13, 42, 47, 97\}$, respectivamente. Os valores para os parâmetros dos algoritmos considerados são os mesmos do exemplo anterior.

A Figura 6.6 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\rho_{\text{opt}}(n)$, para os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS, e a quantidade de ciclos necessária para os métodos da razão áurea e da busca tabu atingirem a convergência em cada iteração do processo de adaptação dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS, respectivamente. Já as Figuras 6.7 e 6.8 mostram o comportamento dos coeficientes $w_1(n)$, $w_{13}(n)$, $w_{30}(n)$, $w_{42}(n)$, $w_{35}(n)$, $w_{47}(n)$, $w_{85}(n)$ e $w_{97}(n)$ para os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS, respectivamente. A Figura 6.9 mostra o comportamento dos coeficientes $w_{30}(n)$, $w_{42}(n)$, e $w_{85}(n)$ e $w_{97}(n)$ após ocorrer a perturbação em $\mathbf{p}$, para os algoritmos considerados.

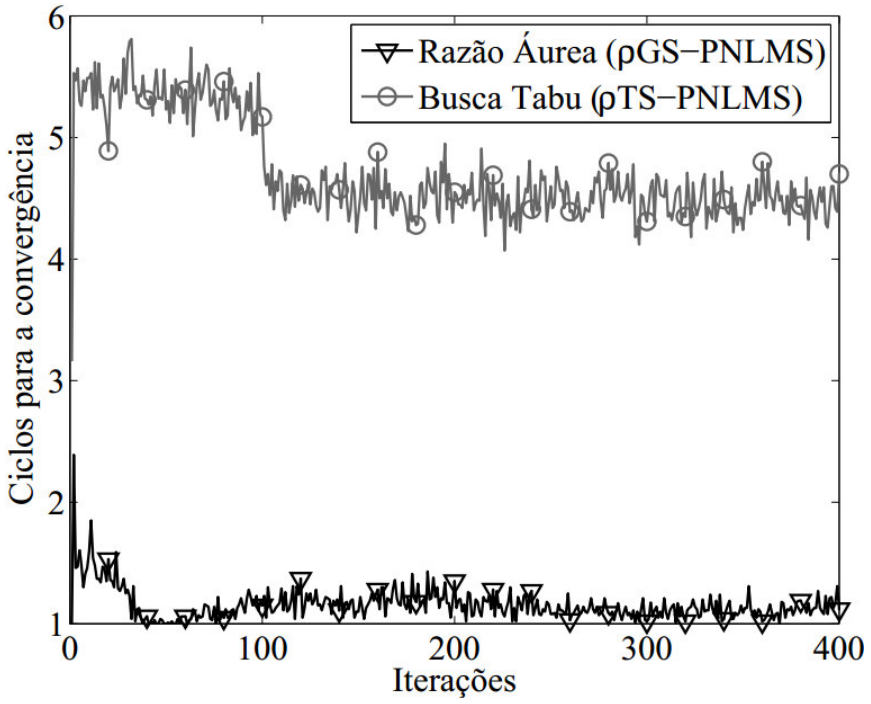
Note das curvas da Figura 6.6(a) que os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS alcançam novamente a mais rápida velocidade de convergência, mesmo após ocorrer o deslocamento dos coeficientes da planta. Note também, das curvas da Figura 6.6(b) que $\rho_{\text{opt}}(n)$ novamente apresenta comportamento semelhante para ambos os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS. Para o algoritmo $\rho$GS-PNLMS, $\rho_{\text{opt}}(n)$ inicia em $4{,}6 * 10^{-5}$ no instante $n = 1$ e, em menos de 200 iterações, estabiliza-se em torno de $5{,}7 * 10^{-5}$, mesmo após a perturbação. Já para o algoritmo $\rho$TS-PNLMS, após iniciar em $4{,}0 * 10^{-6}$ no instante $n = 1$, $\rho_{\text{opt}}(n)$ estabiliza-se, em menos de 200 iterações, em torno de $4{,}5 * 10^{-5}$, mesmo após ocorrer o deslocamento nos coeficientes da planta. Observe das curvas da Figura 6.6(c) que, após $n = 100$, os métodos da razão áurea (algoritmo $\rho$GS-PNLMS) e da busca tabu (algoritmo $\rho$TS-PNLMS) alcançam novamente médias inferiores a 2 e 5 ciclos, respectivamente, para atingir a convergência em cada iteração do processo de adaptação.


PNLMS
IPNLMS
ρGS−PNLMS
ρTS−IPNLMS
Desalinhamento normalizado (dB)
Iterações
10
0
−10
−20
−30
−40
0
1000
2000
3000
4000

(a)

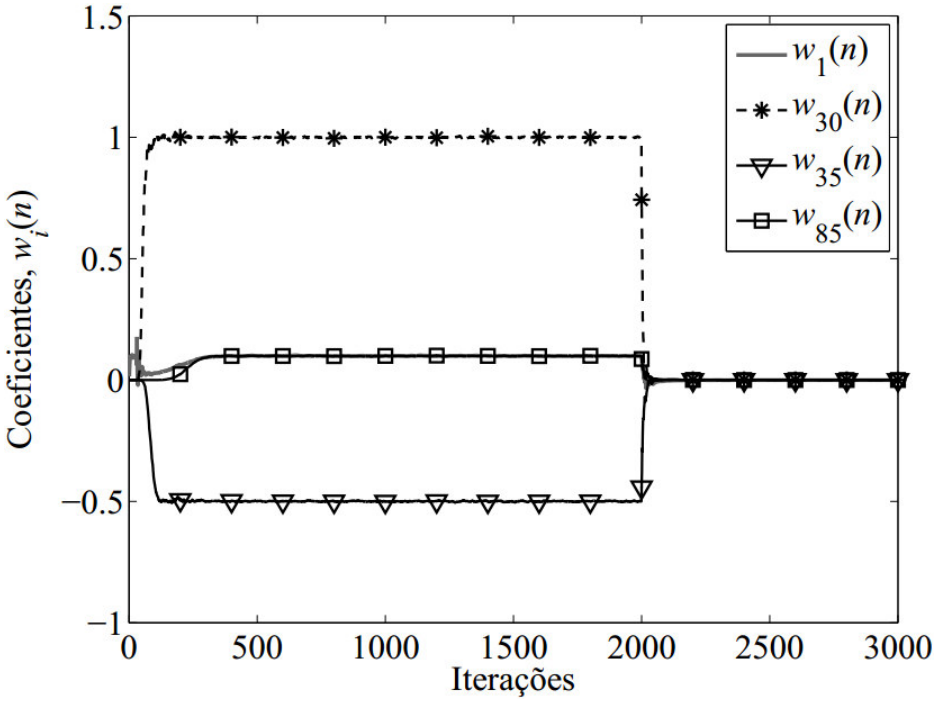

x 10^−5
ρGS−PNLMS
ρTS−PNLMS
ρopt(n)
Iterações
7
6
5
4
3
2
1
0
0
100
200
300
400

(b)

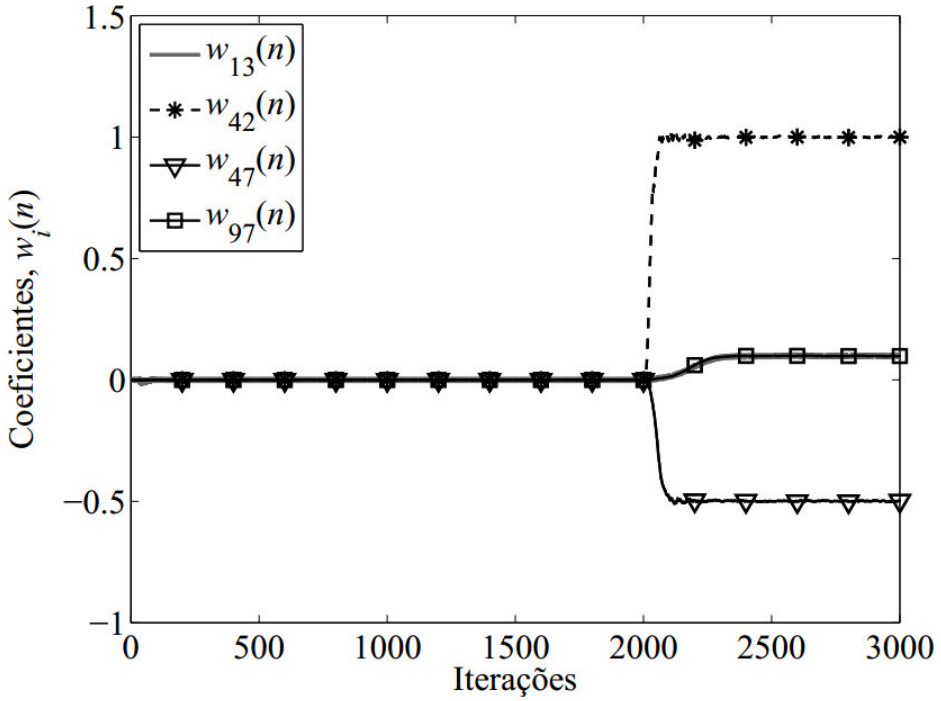



(c)

Figura 6.6. Comportamento dos algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS considerando um deslocamento de 12 amostras à direita dos coeficientes da planta $\mathbf{p}$ em $n = 2000$. (a) Desalinhamento normalizado dos algoritmos PNLMS (com $\delta = 0{,}01$ e $\rho = 0{,}05$), IPNLMS (com $\alpha = 0{,}0$), $\rho$GS-PNLMS (com $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$) e $\rho$TS-PNLMS (com $L_\rho = 10^{-15}$, $M = 6$ e PT $= 3$). (b) Parâmetro de proporcionalidade ótimo, $\rho_{\text{opt}}(n)$, dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS. (c) Quantidade de ciclos para os métodos da razão áurea ($\rho$GS-PNLMS) e da busca tabu ($\rho$TS-PNLMS) atingirem a convergência.

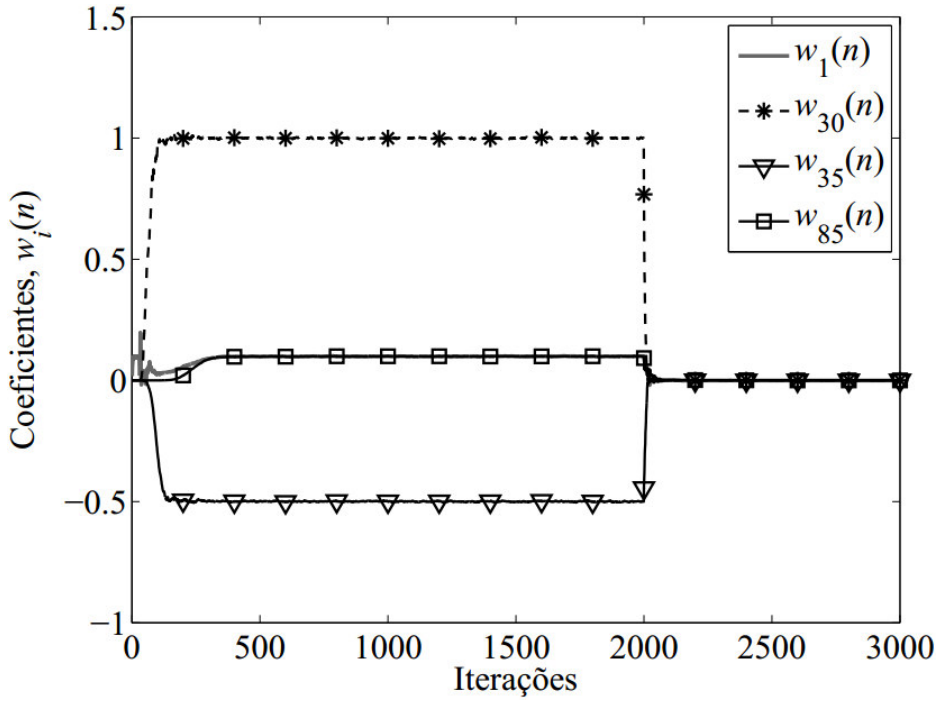



(a)

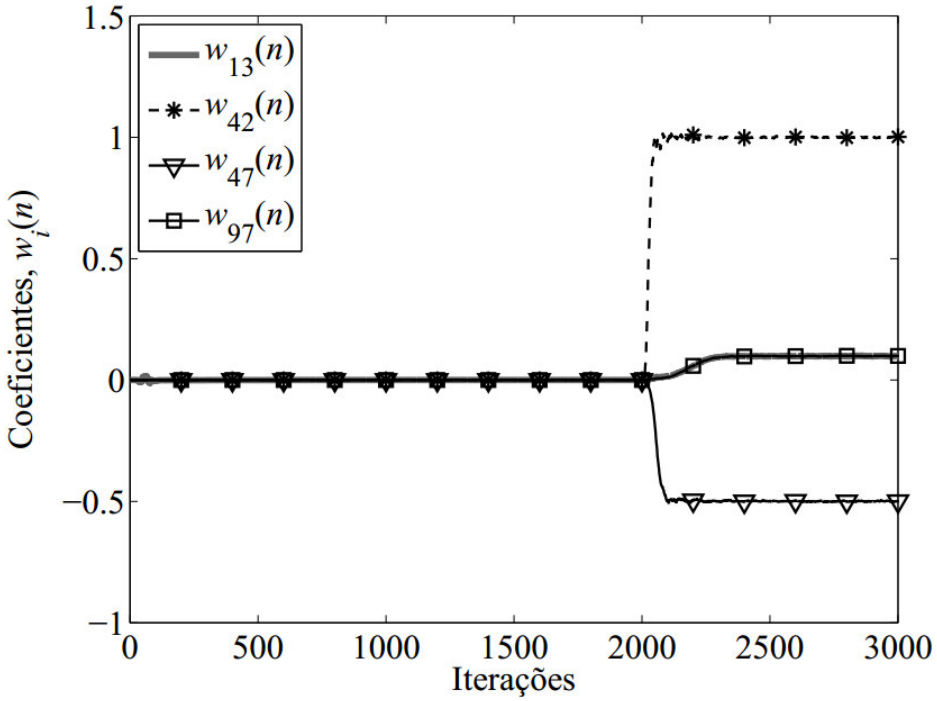



(b)

Figura 6.7. Comportamento dos coeficientes do algoritmo $\rho$GS-PNLMS considerando uma perturbação na planta esparsa **p** em $n = 2000$, quando os coeficientes de **p** são deslocados de 12 amostras para a direita. (a) Coeficientes $w_1(n)$, $w_{30}(n)$, $w_{35}(n)$ e $w_{85}(n)$. (b) Coeficientes $w_{13}(n)$, $w_{42}(n)$, $w_{47}(n)$ e $w_{97}(n)$.

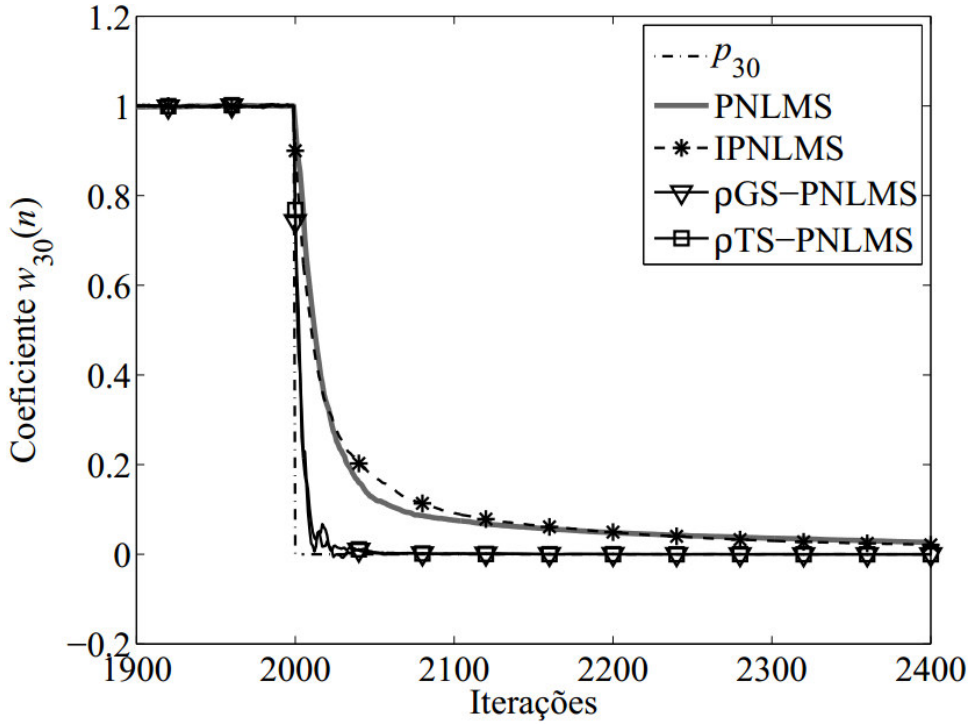



(a)

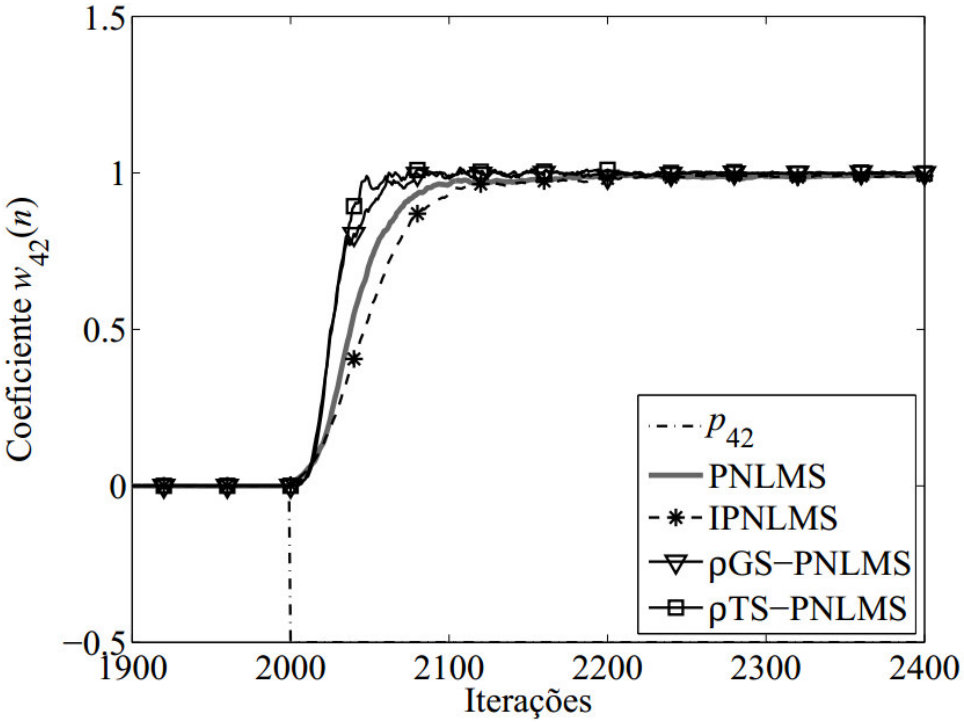



(b)

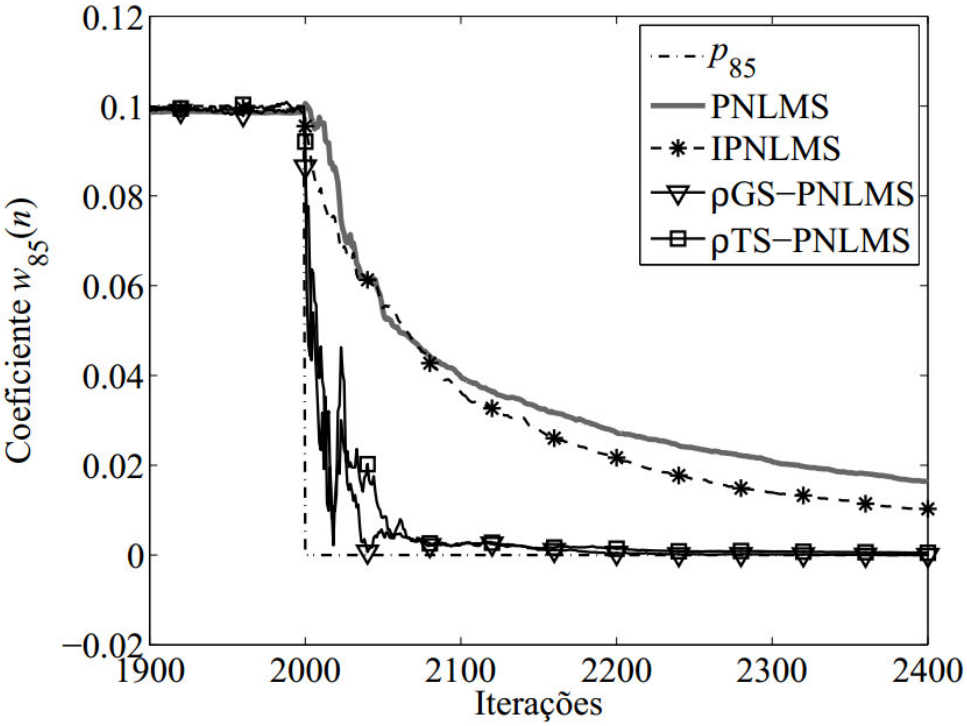
Figura 6.8. Comportamento dos coeficientes do algoritmo $\rho$TS-PNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando os coeficientes de $\mathbf{p}$ são deslocados de 12 amostras para a direita. (a) Coeficientes $w_1(n)$, $w_{30}(n)$, $w_{35}(n)$ e $w_{85}(n)$. (b) Coeficientes $w_{13}(n)$, $w_{42}(n)$, $w_{47}(n)$ e $w_{97}(n)$.



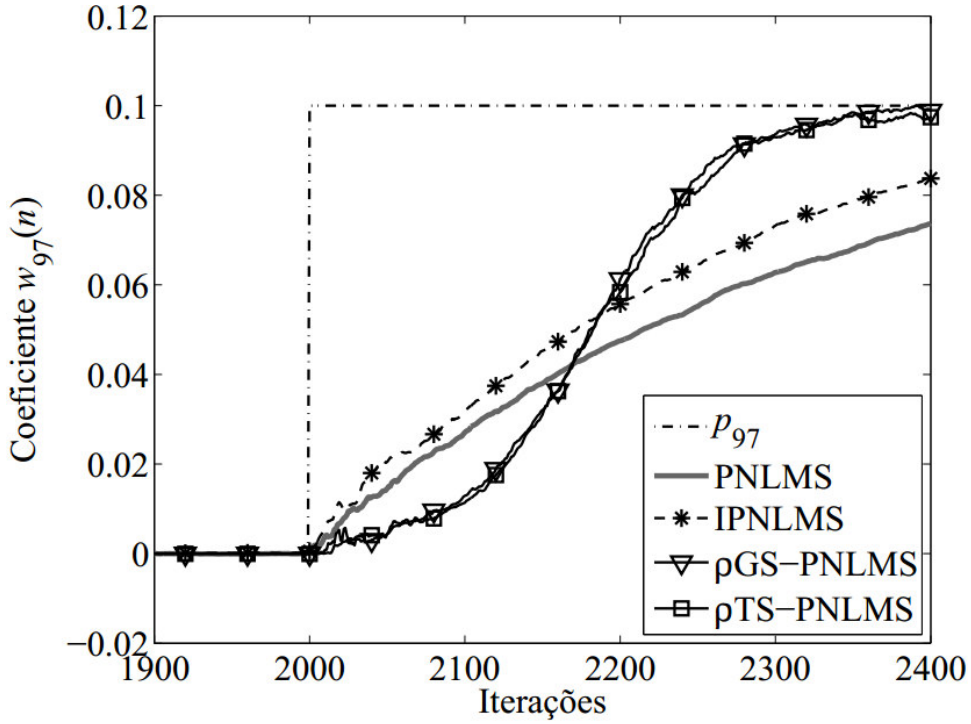

(a)


1.5
1
0.5
0
−0.5
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{42}(n)$
$p_{42}$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

(b)


0.12
0.1
0.08
0.06
0.04
0.02
0
−0.02
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{85}(n)$
$p_{85}$
PNLMS
IPNLMS
ρGS−PNLMS
ρTS−PNLMS

(c)




(d)

Figura 6.9. Comportamento dos coeficientes $w_{30}(n)$ $w_{42}(n)$, e $w_{85}(n)$ e $w_{97}(n)$ para os algoritmos PNLMS, IPNLMS, $\rho$GS-PNLMS e $\rho$TS-PNLMS, após o deslocamento de 12 amostras para a direita de $\mathbf{p}$, ocorrido no instante $n = 2000$. (a) Coeficiente $w_{30}(n)$. (b) Coeficiente $w_{42}(n)$. (c) Coeficiente $w_{85}(n)$. (d) Coeficiente $w_{97}(n)$.

Note, das curvas das Figuras 6.7 e 6.8 que, novamente, os coeficientes ativos e inativos dos algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS convergem rapidamente para os valores desejados da planta $\mathbf{p}$, mesmo após ocorrer a perturbação. Além disso, tem-se que os coeficientes ativos de maior magnitude como, por exemplo, $w_{30}(n)$ antes de $n = 2000$ e $w_{42}(n)$ após $n = 2000$, convergem mais rapidamente do que os coeficientes ativos de menor magnitude, como $w_1(n)$ antes de $n = 2000$ e $w_{97}(n)$ após $n = 2000$. Note também, das curvas da Figura 6.9, que os algoritmos $\rho$GS-PNLMS e $\rho$TS-PNLMS demonstram novamente maior capacidade de rastreamento, tanto para coeficientes ativos como inativos, mesmo após o deslocamento nos coeficientes de $\mathbf{p}$.

## 6.2 Algoritmos $f$GS-PNLMS e $f$TS-PNLMS

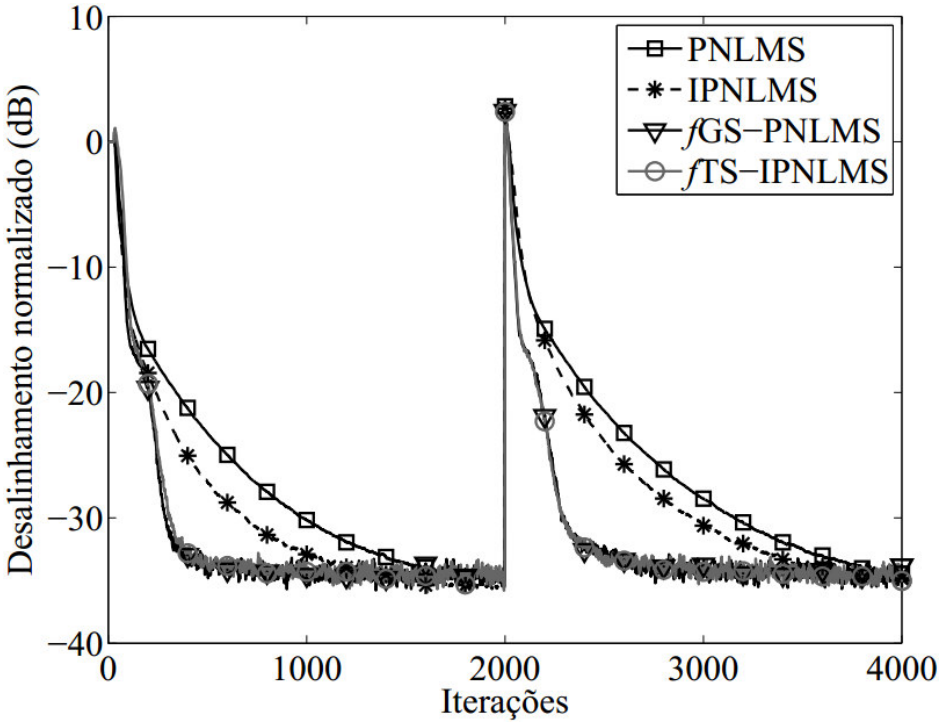
Nesta seção, o desempenho dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS é comparado com o dos algoritmos PNLMS e IPNLMS, em termos da velocidade de convergência. Para tal, dois exemplos comparativos são considerados a seguir. O primeiro considera um deslocamento dos coeficientes da planta esparsa, $\mathbf{p}$, em $n = 2000$. Já o segundo considera uma planta de esparsidade variável no tempo obtida a partir de dados da planta $\mathbf{p}$ e do uso de um processo markoviano de primeira ordem (Loganathan, Habets & Naylor, 2010).

*6.2.1 Exemplo 3*

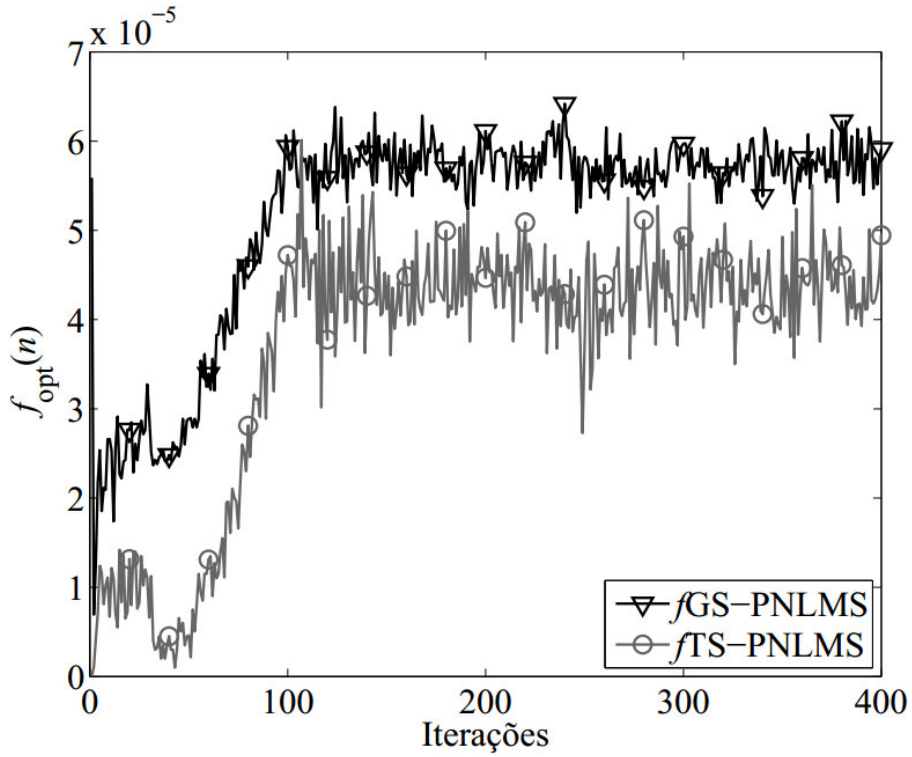
Este exemplo tem o objetivo de comparar o desempenho dos algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS em termos de velocidade de convergência e resposta a uma perturbação ocorrida no instante $n = 2000$, quando o vetor de coeficientes da planta $\mathbf{p}$ é deslocado de 12 amostras para a direita, alterando as posições dos coeficientes ativos. Dessa forma, os coeficientes ativos de $\mathbf{p}$, cujos valores são iguais a $\{0{,}1\ ,\ 1{,}0\ ,\ -0{,}5\ ,\ 0{,}1\}$, são movidos das posições $\{1, 30, 35, 85\}$ para as posições $\{13, 42, 47, 97\}$, respectivamente.

A Figura 6.10 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do fator de ativação ótimo, $f_{\text{opt}}(n)$, para os algoritmos $f$GS-PNLMS e $f$TS-PNLMS, e a quantidade de ciclos necessária para os métodos da razão áurea e da busca tabu atingirem a convergência em cada iteração do processo de adaptação dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS, respectivamente. Já as Figuras 6.11 e 6.12 mostram o comportamento dos coeficientes $w_1(n)$, $w_{13}(n)$, $w_{30}(n)$, $w_{42}(n)$, $w_{35}(n)$, $w_{47}(n)$, $w_{85}(n)$ e $w_{97}(n)$ para os algoritmos $f$GS-PNLMS e $f$TS-PNLMS, respectivamente. A Figura 6.13 mostra o comportamento dos coeficientes $w_1(n)$, $w_{13}(n)$, e $w_{35}(n)$ e $w_{47}(n)$ para os algoritmos considerados após ocorrer a perturbação em $\mathbf{p}$. Adota-se um parâmetro de passo igual a $\mu = 0{,}5$ para os algoritmos considerados neste exemplo, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para o algoritmo PNLMS, considera-se $\delta = 0{,}01$ e $\rho = 0{,}05$, e para o algoritmo IPNLMS, adota-se $\alpha = 0{,}0$ (Benesty & Gay, 2002) e (Souza, Tobias, Seara & Morgan, 2010). Para o algoritmo $f$GS-PNLMS, considera-se $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$. Já para o algoritmo $f$TS-PNLMS, adota-se $L_f = 10^{-15}$, $M = 6$ e PT $= 3$.

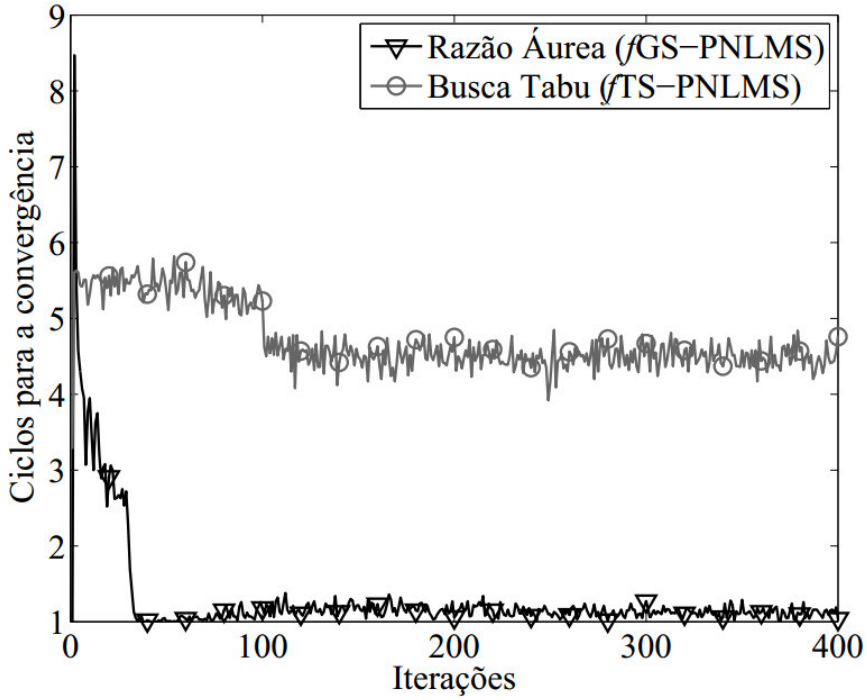
Note, das curvas da Figura 6.10(a), que os algoritmos $f$GS-PNLMS e $f$TS-PNLMS alcançam velocidades de convergência mais rápidas do que as dos algoritmos PNLMS e IPNLMS, mesmo após ocorrer o deslocamento dos coeficientes da planta. Note também, das curvas da Figura 6.10(b), que $f_{\text{opt}}(n)$ apresenta comportamento semelhante para ambos os algoritmos $f$GS-PNLMS e $f$TS-PNLMS. Para o algoritmo $f$GS-PNLMS, após iniciar em $5{,}9 * 10^{-5}$ no instante $n = 1$, em menos de 200 iterações, $f_{\text{opt}}(n)$ estabiliza-se em torno de $5{,}6 * 10^{-5}$, mesmo após a perturbação. Já para o algoritmo $f$TS-PNLMS, $f_{\text{opt}}(n)$ inicia em $3{,}1 * 10^{-8}$ no instante $n = 1$, e estabiliza-se, em menos de 200 iterações, em torno de $4{,}6 * 10^{-5}$, mesmo após ocorrer o deslocamento nos coeficientes da planta. Observe das curvas da Figura 6.10(c) que, após $n = 100$, os métodos da razão áurea (algoritmo $f$GS-PNLMS) e da busca tabu (algoritmo $f$TS-PNLMS) alcançam médias inferiores a 2 e 5 ciclos, respectivamente, para atingir a convergência em cada iteração do processo de adaptação.


PNLMS
IPNLMS
fGS−PNLMS
fTS−IPNLMS
Desalinhamento normalizado (dB)
Iterações
10
0
−10
−20
−30
−40
0
1000
2000
3000
4000

(a)

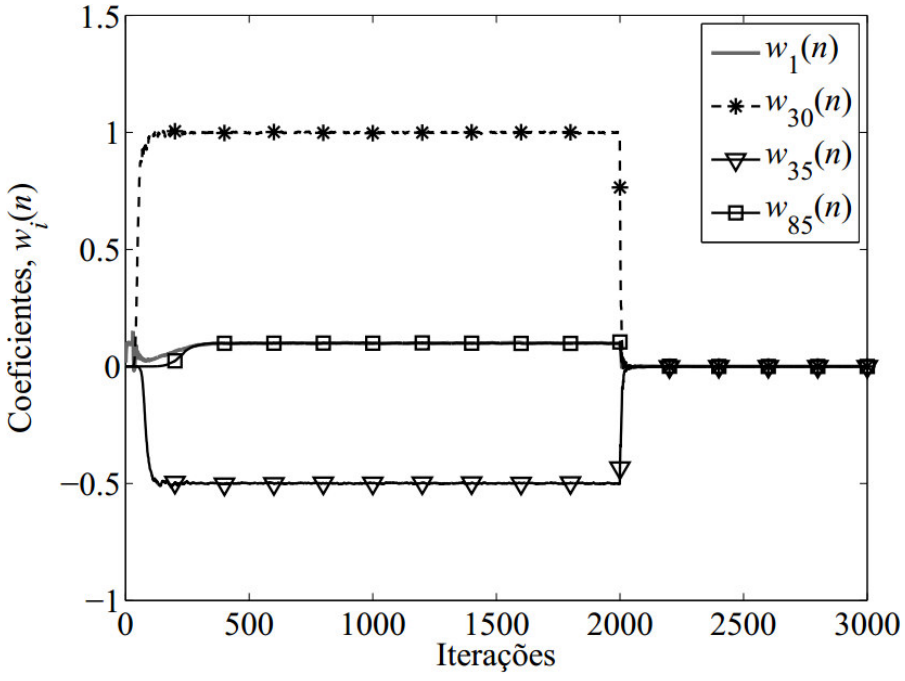

x 10−5
fGS−PNLMS
fTS−PNLMS
Iterações
7
6
5
4
3
2
1
0
0
100
200
300
400

(b)

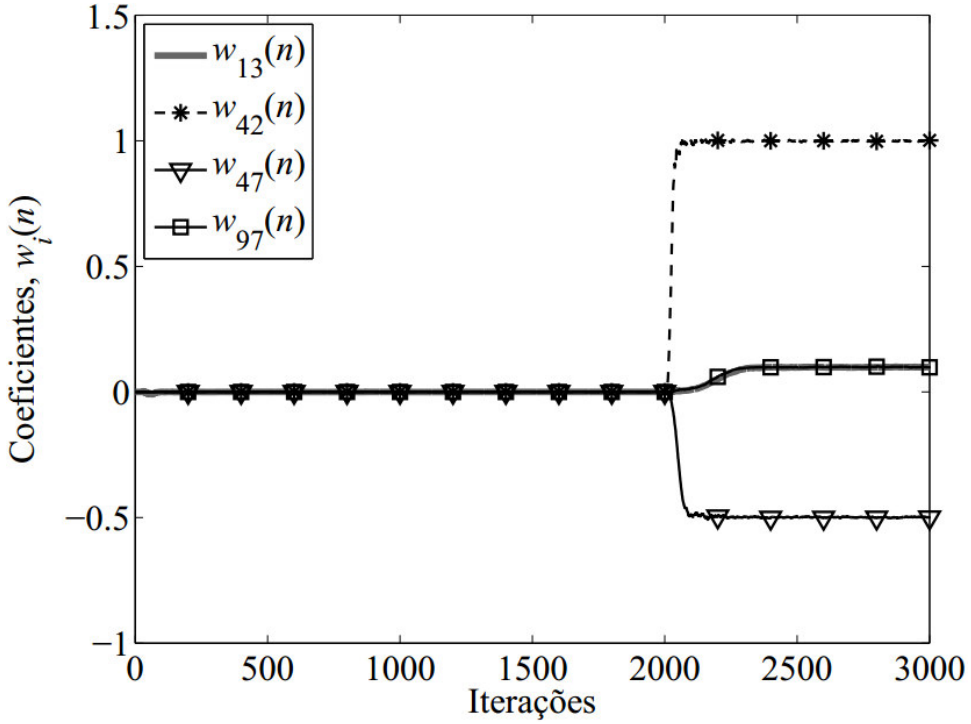



(c)

Figura 6.10. Comportamento dos algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS considerando um deslocamento de 12 amostras à direita dos coeficientes da planta $\mathbf{p}$ em $n = 2000$. (a) Desalinhamento normalizado dos algoritmos PNLMS (com $\delta = 0{,}01$ e $\rho = 0{,}05$), IPNLMS (com $\alpha = 0{,}0$), $f$GS-PNLMS (com $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$) e $f$TS-PNLMS (com $L_f = 10^{-15}$, $M = 6$ e PT = 3). (b) Fator de ativação ótimo, $f_{\text{opt}}(n)$, dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS. (c) Quantidade de ciclos para os métodos da razão áurea ($f$GS-PNLMS) e da busca tabu ($f$TS-PNLMS) atingirem a convergência.

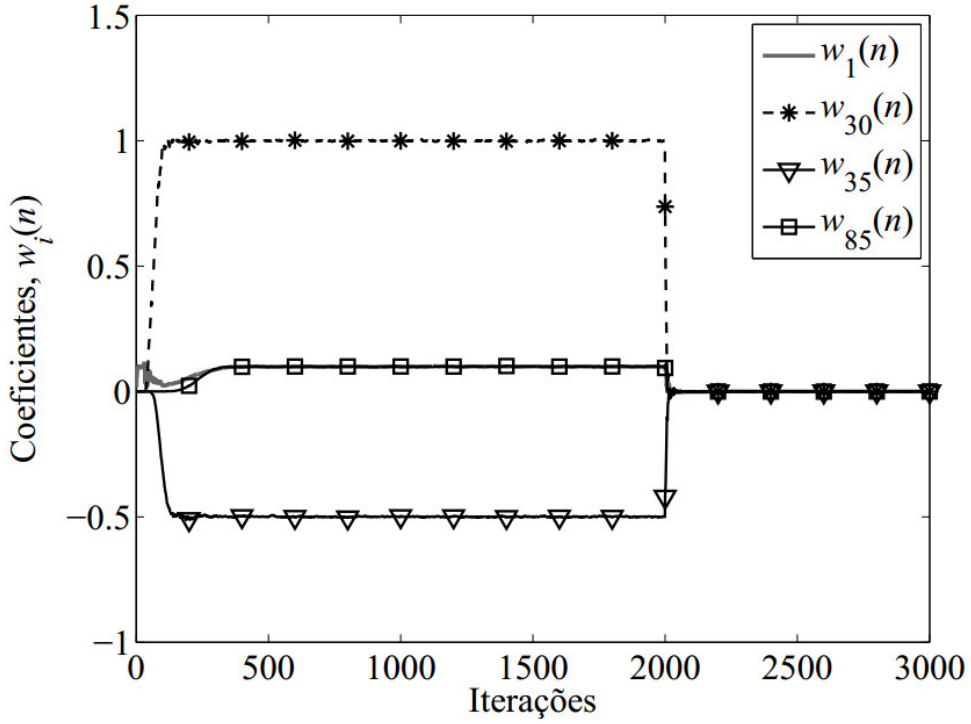



(a)

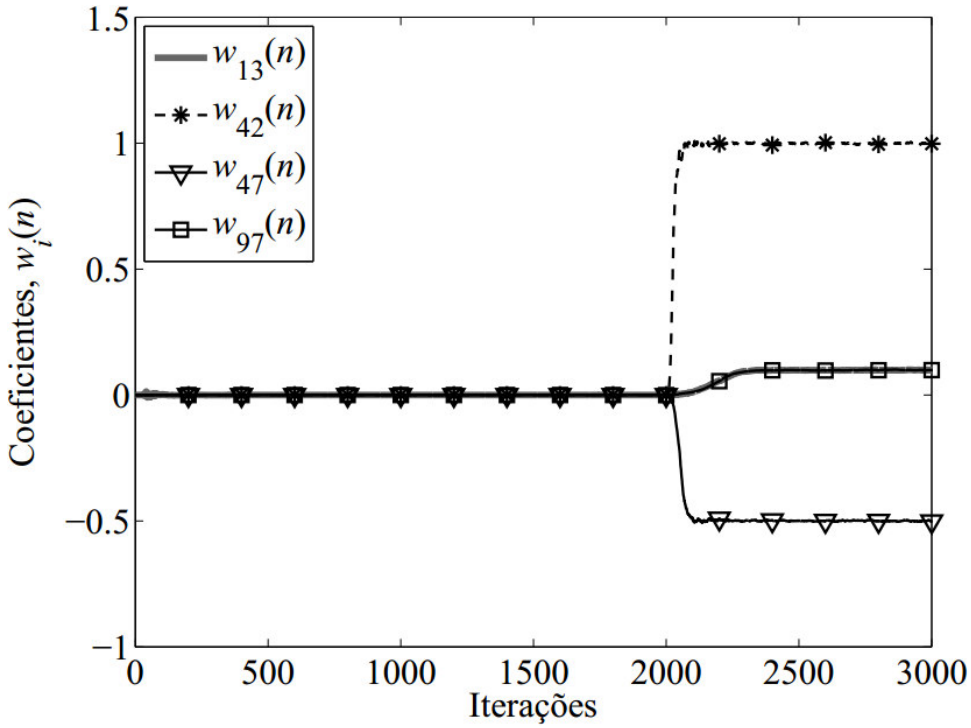



(b)

Figura 6.11. Comportamento dos coeficientes do algoritmo $f$GS-PNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando os coeficientes de $\mathbf{p}$ são deslocados de 12 amostras para a direita. (a) Coeficientes $w_1(n)$, $w_{30}(n)$, $w_{35}(n)$ e $w_{85}(n)$. (b) Coeficientes $w_{13}(n)$, $w_{42}(n)$, $w_{47}(n)$ e $w_{97}(n)$.

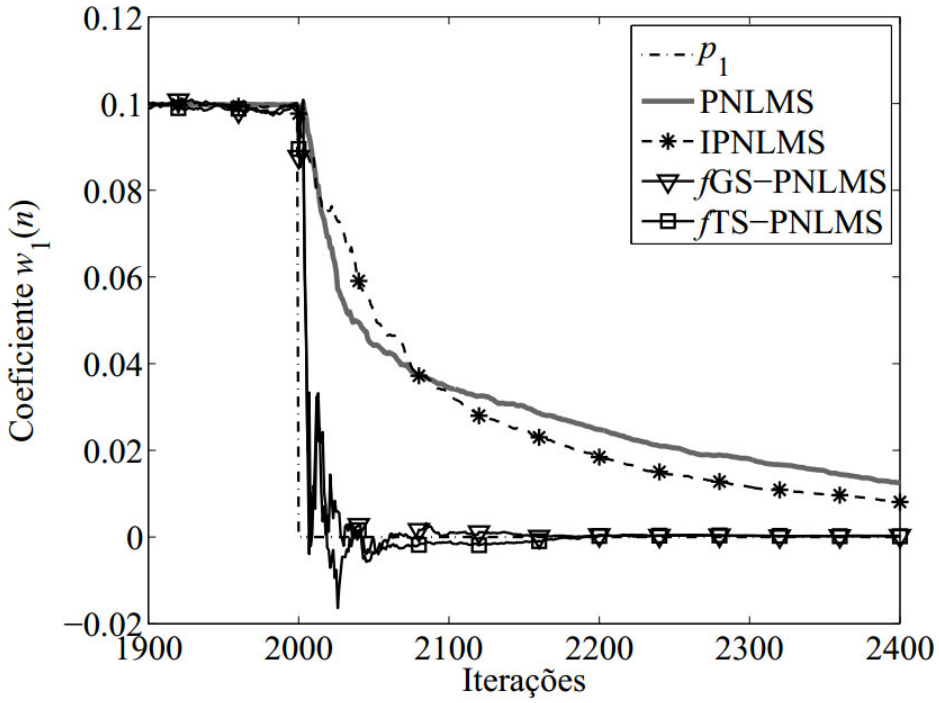



(a)

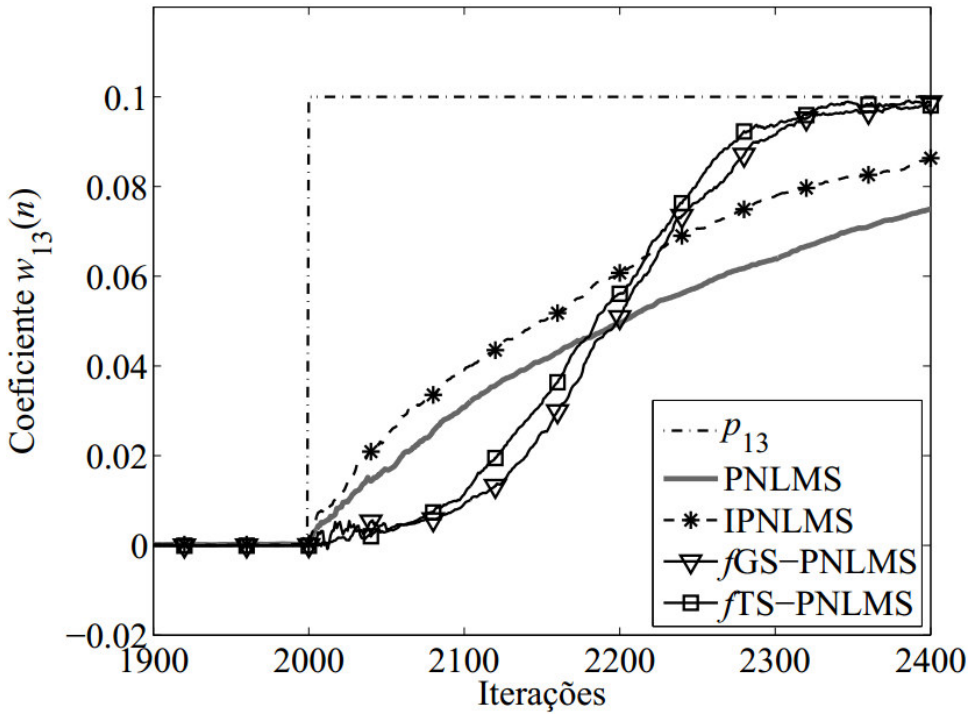



(b)

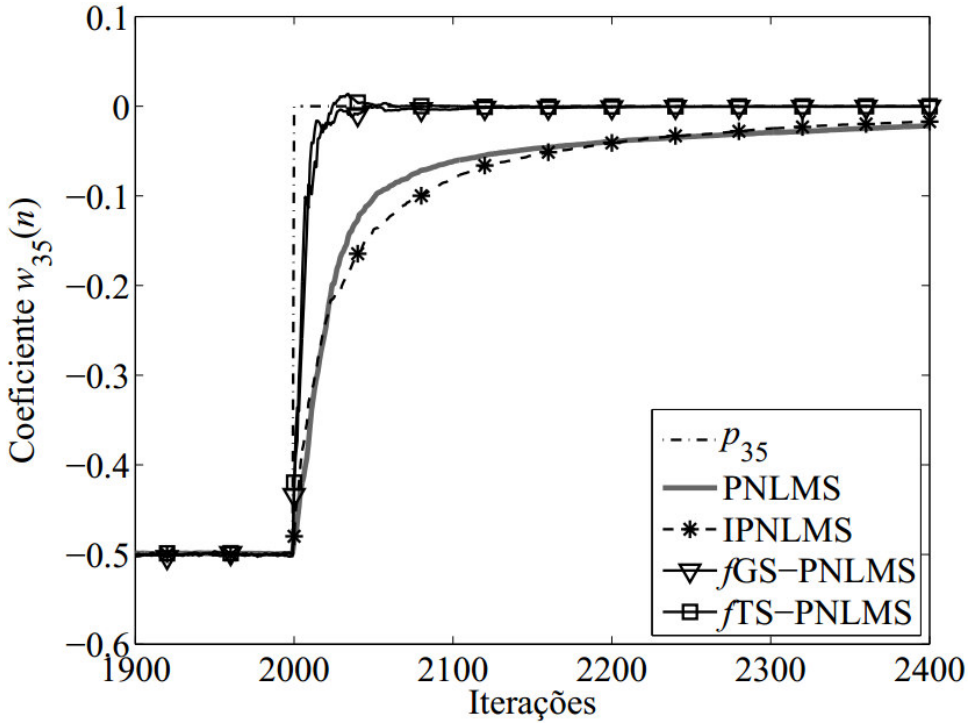
Figura 6.12. Comportamento dos coeficientes do algoritmo $f$TS-PNLMS considerando uma perturbação na planta esparsa **p** em $n = 2000$, quando os coeficientes de **p** são deslocados de 12 amostras para a direita. (a) Coeficientes $w_1(n)$, $w_{30}(n)$, $w_{35}(n)$ e $w_{85}(n)$. (b) Coeficientes $w_{13}(n)$, $w_{42}(n)$, $w_{47}(n)$ e $w_{97}(n)$.



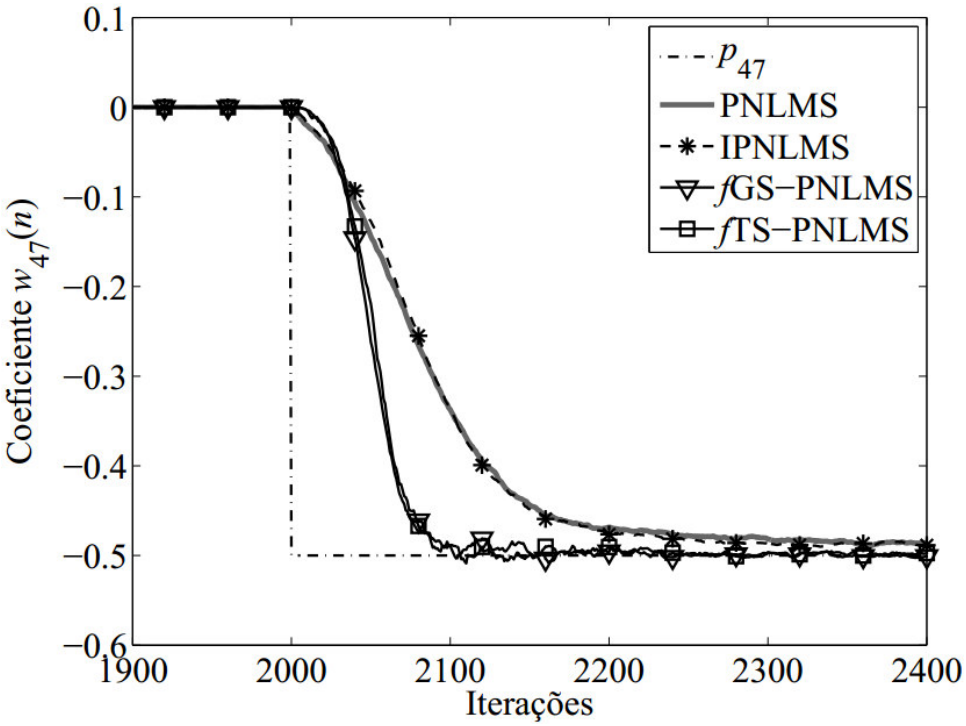

(a)


0.1
0.08
0.06
0.04
0.02
0
−0.02
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{13}(n)$
$p_{13}$
PNLMS
IPNLMS
$f$GS−PNLMS
$f$TS−PNLMS

(b)


0.1
0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{35}(n)$
$p_{35}$
PNLMS
IPNLMS
$f$GS−PNLMS
$f$TS−PNLMS

(c)




(d)

Figura 6.13. Comportamento dos coeficientes $w_1(n)$ $w_{13}(n)$, e $w_{35}(n)$ e $w_{47}(n)$ para os algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS, após o deslocamento de 12 amostras para a direita de $\mathbf{p}$, ocorrido no instante $n = 2000$. (a) Coeficiente $w_1(n)$. (b) Coeficiente $w_{13}(n)$. (c) Coeficiente $w_{35}(n)$. (d) Coeficiente $w_{47}(n)$.

Note, das curvas das Figuras 6.11 e 6.12 que tanto os coeficientes ativos como os coeficientes inativos dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS convergem rapidamente para os valores desejados da planta $\mathbf{p}$, mesmo após ocorrer a perturbação. Além disso, tem-se que os coeficientes ativos de maior magnitude como, por exemplo, $w_{35}(n)$ antes de $n = 2000$ e $w_{47}(n)$ após $n = 2000$, convergem mais rapidamente do que os coeficientes ativos de menor magnitude, como $w_{85}(n)$ antes de $n = 2000$ e $w_{13}(n)$ após $n = 2000$. Note também, das curvas da Figura 6.13, que os algoritmos $f$GS-PNLMS e $f$TS-PNLMS apresentam maior capacidade de rastreamento de coeficientes ativos e inativos do que os algoritmos PNLMS e IPNLMS, mesmo após ocorrer o deslocamento em $\mathbf{p}$.

### *6.2.2 Exemplo 4*

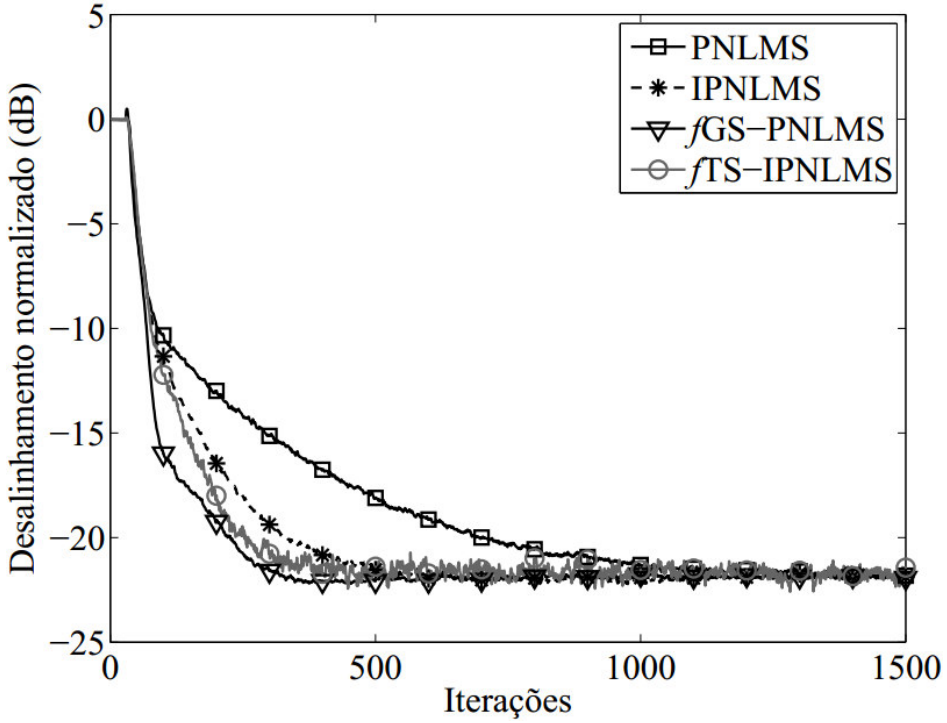
Neste exemplo, a velocidade de convergência dos algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-IPNLMS é comparada considerando a identificação de uma planta com esparsidade variável no tempo. Para tal, é utilizado o processo markoviano de primeira ordem dado por (Loganathan, Habets & Naylor, 2010)

$$\mathbf{p}(n) = \lambda\mathbf{p}(n) + \sqrt{1-\lambda^2}\mathbf{v}(n) \tag{6.1}$$

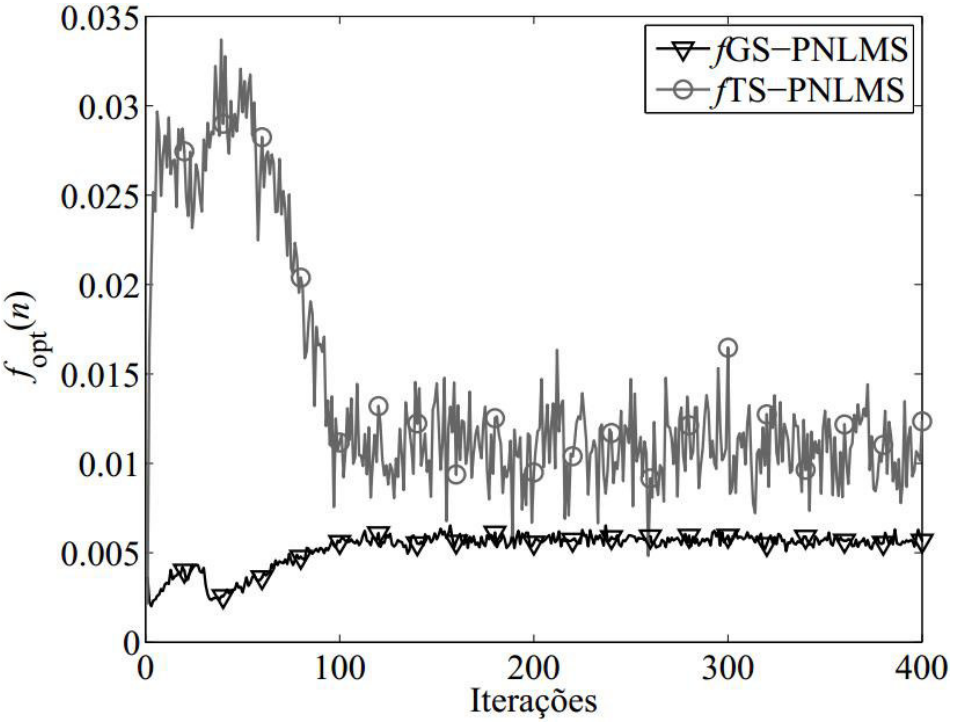
onde $\mathbf{v}(n)$ é uma sequência aleatória de tamanho $N$, formada a partir de uma distribuição gaussiana de média zero e variância $\sigma_{\mathrm{v}}^2$. O fator $0 < \lambda < 1$ controla as contribuições da "memória" e das "inovações" referentes à resposta impulsiva da planta variante $\mathbf{p}(n)$. Note que, para $\lambda = 1$, tem-se uma planta invariante no tempo. Neste exemplo, $\mathbf{p}(n)$ é inicializada (instante $n = 0$) com os valores da planta esparsa $\mathbf{p}$, e considera-se $\sigma_{\mathrm{v}}^2 = 1$ e $\lambda = 1 - 10^{-7}$.

A Figura 6.14 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do fator de ativação ótimo, $f_{\mathrm{opt}}(n)$, para os algoritmos $f$GS-PNLMS e $f$TS-PNLMS, a variação no grau de esparsidade da planta $\mathbf{p}(n)$, e a quantidade de ciclos necessária para os métodos da razão áurea e da busca tabu atingirem a convergência em cada iteração do processo de adaptação dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS, respectivamente. Já as Figuras 6.15 e 6.16 mostram, a partir do instante $n = 1000$, o comportamento dos coeficientes $w_{30}(n)$ e $w_{97}(n)$, respectivamente, para cada um dos algoritmos considerados em relação aos valores correspondentes de $\mathbf{p}(n)$. Adota-se parâmetros de passo iguais a $\mu_{\mathrm{PN}} = \mu_{\mathrm{IPN}} = 1{,}6$, $\mu_{f\mathrm{GS}} = 0{,}3$ e $\mu_{f\mathrm{TS}} = 1{,}7$ para os algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS, respectivamente, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Os demais parâmetros adotados para os algoritmos considerados são os mesmos do exemplo anterior, exceto que, para o algoritmo $f$GS-PNLMS considera-se agora $tol = 10^{-1}$.

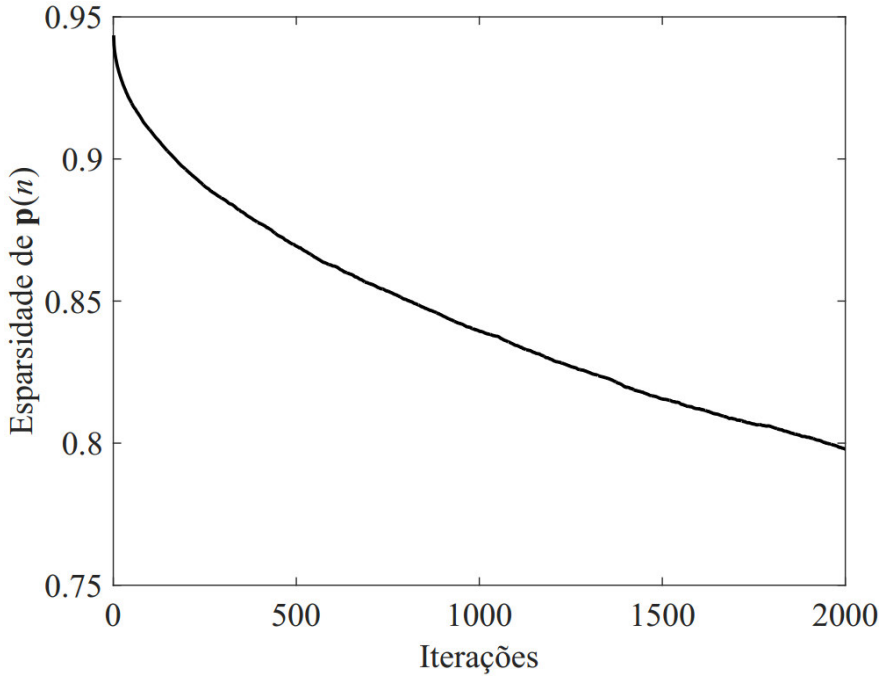
Note, das curvas da Figura 6.10(a), que os algoritmos $f$GS-PNLMS e $f$TS-PNLMS alcançam velocidades de convergência mais rápidas do que as dos algoritmos PNLMS e IPNLMS, mesmo ocorrendo uma redução gradual da esparsidade da planta $\mathbf{p}(n)$ de $S[\mathbf{p}(n)] = 0{,}9435$ em $n = 0$ para $S[\mathbf{p}(n)] = 0{,}7979$ em $n = 2000$, conforme a definição de (1.2). Além disso, tem-se que o algoritmo $f$GS-PNLMS alcança a mais rápida velocidade de convergência, superando o desempenho do algoritmo $f$TS-PNLMS. Note também, das curvas da Figura 6.10(b) que, para o algoritmo $f$GS-PNLMS, $f_{\mathrm{opt}}(n)$ inicia em $3{,}7 * 10^{-3}$ no instante $n = 1$ e, em menos de 200 iterações, estabiliza-se em torno de $5{,}6 * 10^{-3}$. Já para o algoritmo $f$TS-PNLMS, após iniciar em $2{,}1 * 10^{-3}$ no instante $n = 1$, em menos de 200 iterações, $f_{\mathrm{opt}}(n)$ mantém-se oscilando em torno de $1{,}1 * 10^{-3}$. Observe das curvas da Figura 6.10(d) que o método da razão áurea (algoritmo $f$GS-PNLMS), a partir de $n = 13$, necessita de apenas 1 ciclo para atingir a convergência em cada iteração. Já o método da busca tabu (algoritmo $f$TS-PNLMS) alcança, após $n = 100$, uma média inferior a 5 ciclos para atingir a convergência em cada iteração do processo de adaptação.


PNLMS
IPNLMS
fGS−PNLMS
fTS−IPNLMS
Desalinhamento normalizado (dB)
Iterações
5
0
−5
−10
−15
−20
−25
0
500
1000
1500

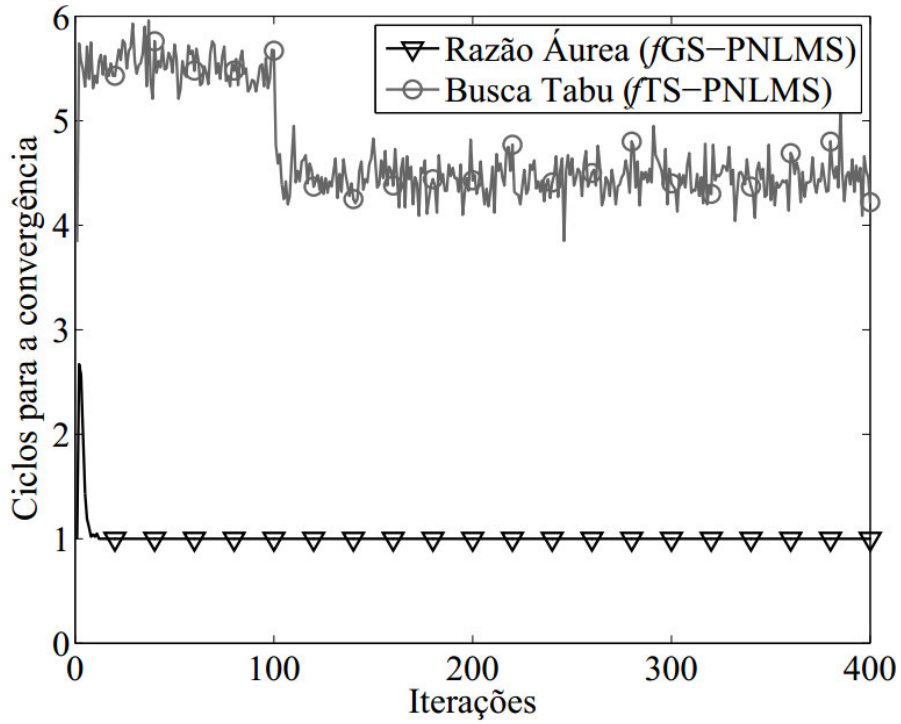
(a)


fGS−PNLMS
fTS−PNLMS
$f_{opt}(n)$
Iterações
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
100
200
300
400

(b)

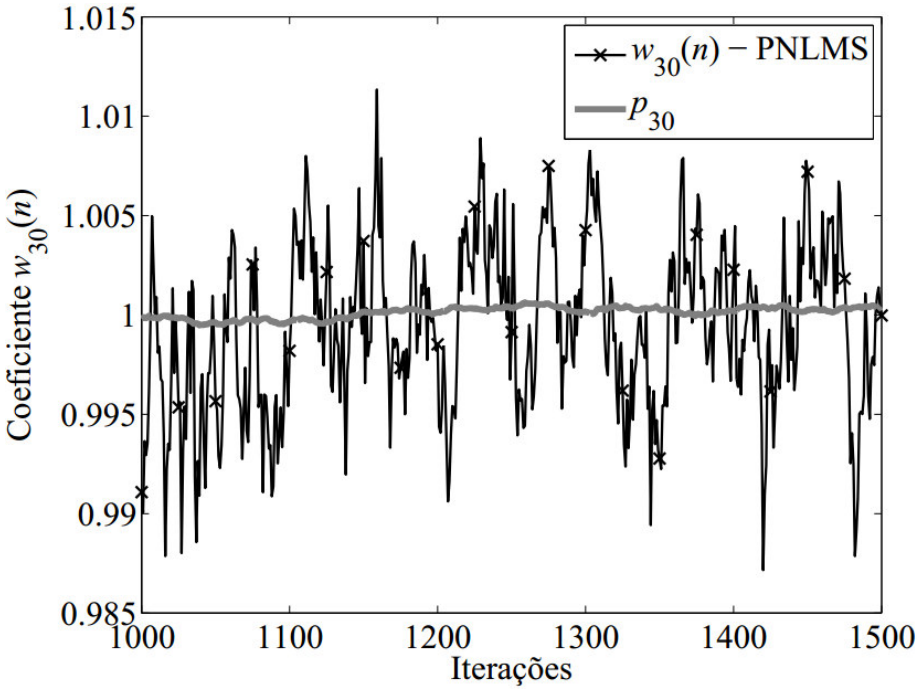



(c)

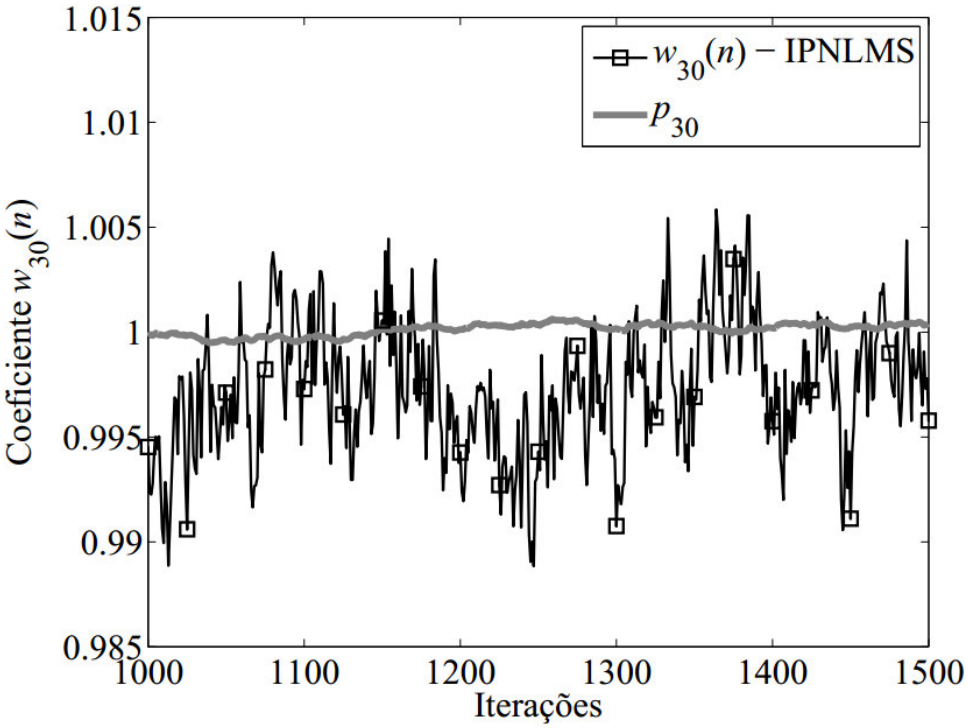



(d)

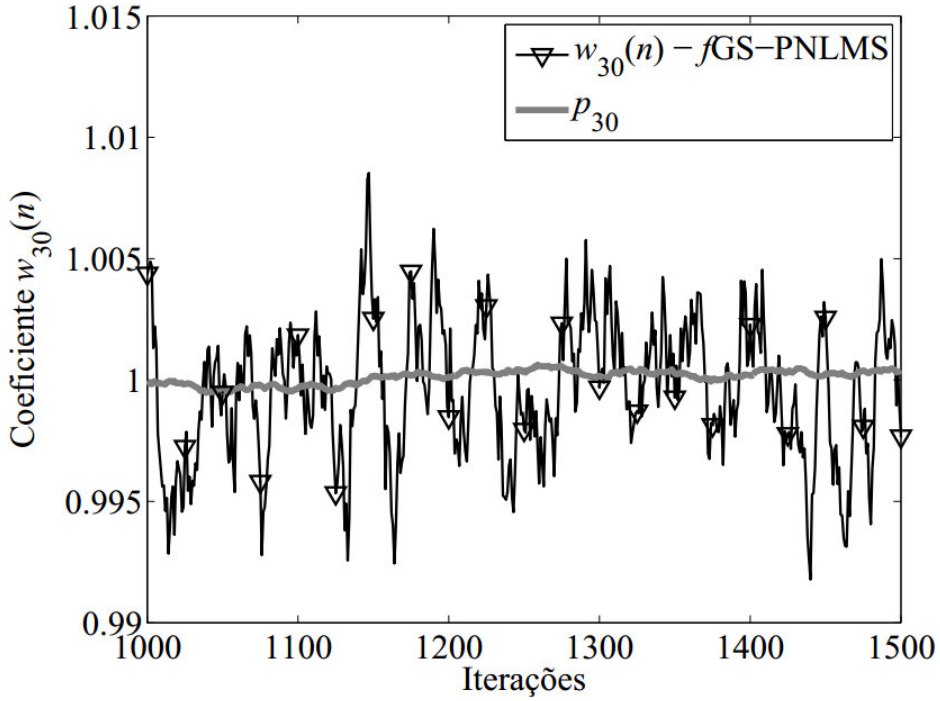
Figura 6.14. Comportamento dos algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS para a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Desalinhamento normalizado dos algoritmos PNLMS (com $\mu_{\mathrm{PN}} = 1{,}6$, $\delta = 0{,}01$ e $\rho = 0{,}05$), IPNLMS (com $\mu_{\mathrm{IPN}} = 1{,}6$ e $\alpha = 0{,}0$), $f$GS-PNLMS (com $\mu_{f\mathrm{GS}} = 1{,}7$, $a = 10^{-15}$, $b = 10^{-4}$ e $tol = 10^{-3}$) e $f$TS-PNLMS (com $\mu_{f\mathrm{TS}} = 0{,}3$, $L_f = 10^{-15}$, $M = 6$ e PT = 3). (b) Fator de ativação ótimo, $f_{\mathrm{opt}}(n)$, dos algoritmos $f$GS-PNLMS e $f$TS-PNLMS. (c) Variação da esparsidade de $\mathbf{p}(n)$. (d) Quantidade de ciclos para os métodos da razão áurea ($f$GS-PNLMS) e da busca tabu ($f$TS-PNLMS) atingirem a convergência.


1.015
1.01
1.005
1
0.995
0.99
0.985
1000
1100
1200
1300
1400
1500
Iterações
Coeficiente $w_{30}(n)$
$w_{30}(n)$ − PNLMS
$p_{30}$

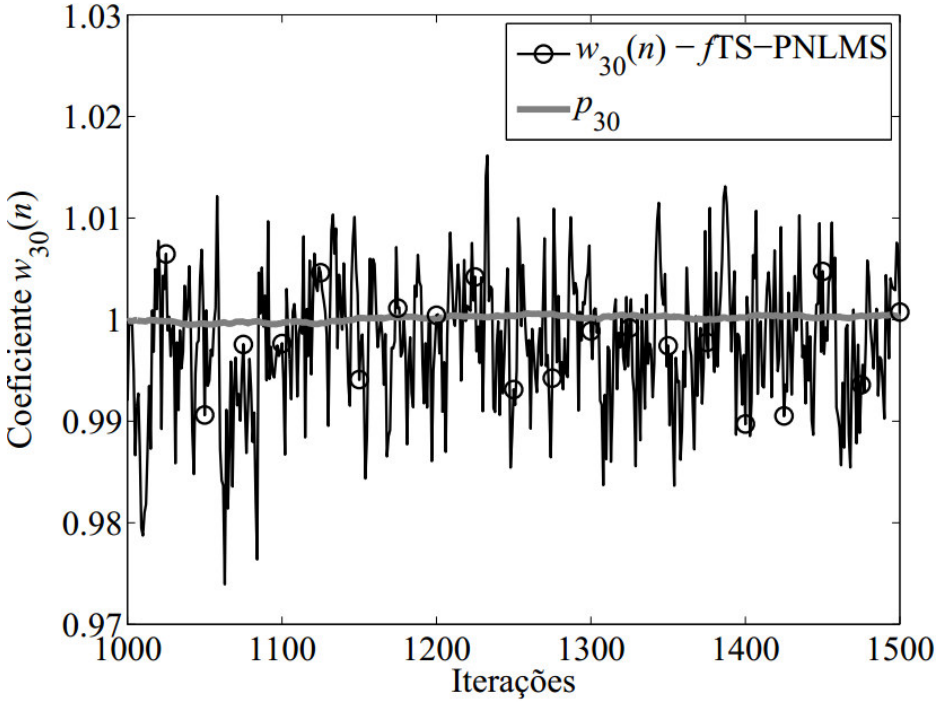
(a)


1.015
1.01
1.005
1
0.995
0.99
0.985
1000
1100
1200
1300
1400
1500
Iterações
Coeficiente $w_{30}(n)$
$w_{30}(n)$ − IPNLMS
$p_{30}$

(b)

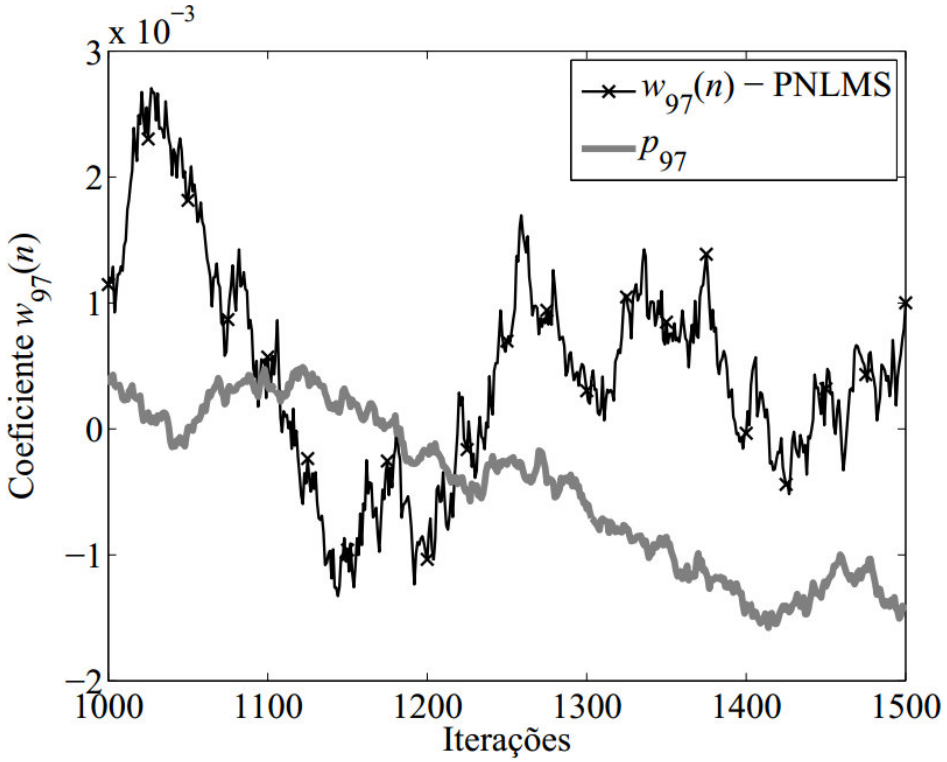



(c)

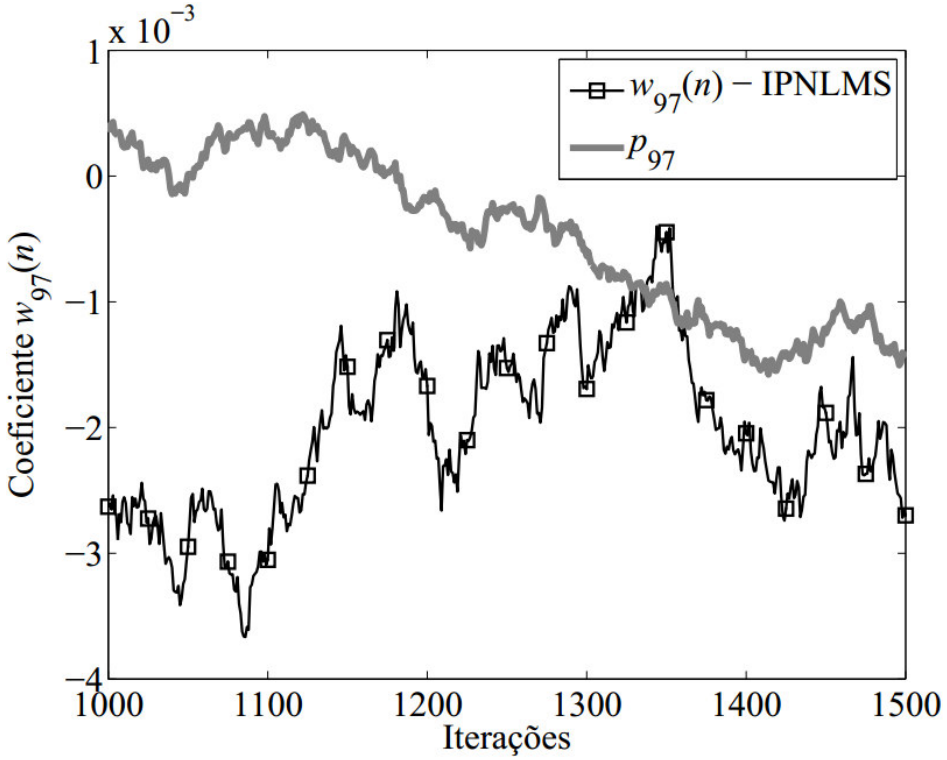



(d)

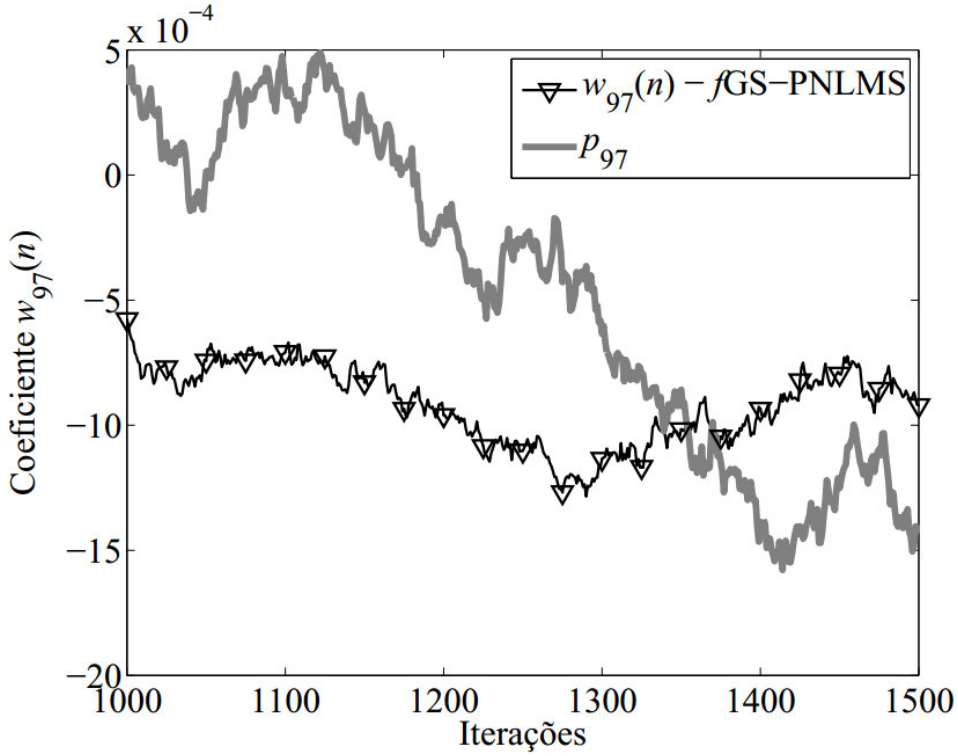
Figura 6.15. Comportamento do coeficiente $w_{30}(n)$, a partir do instante $n = 1000$, para os algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS, considerando a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Algoritmo PNLMS (b) Algoritmo IPNLMS. (c) Algoritmo $f$GS-PNLMS. (d) Algoritmo $f$TS-PNLMS.


x 10^−3
3
2
1
0
−1
−2
1000
1100
1200
1300
1400
1500
Iterações
Coeficiente w97(n)
w97(n) − PNLMS
p97

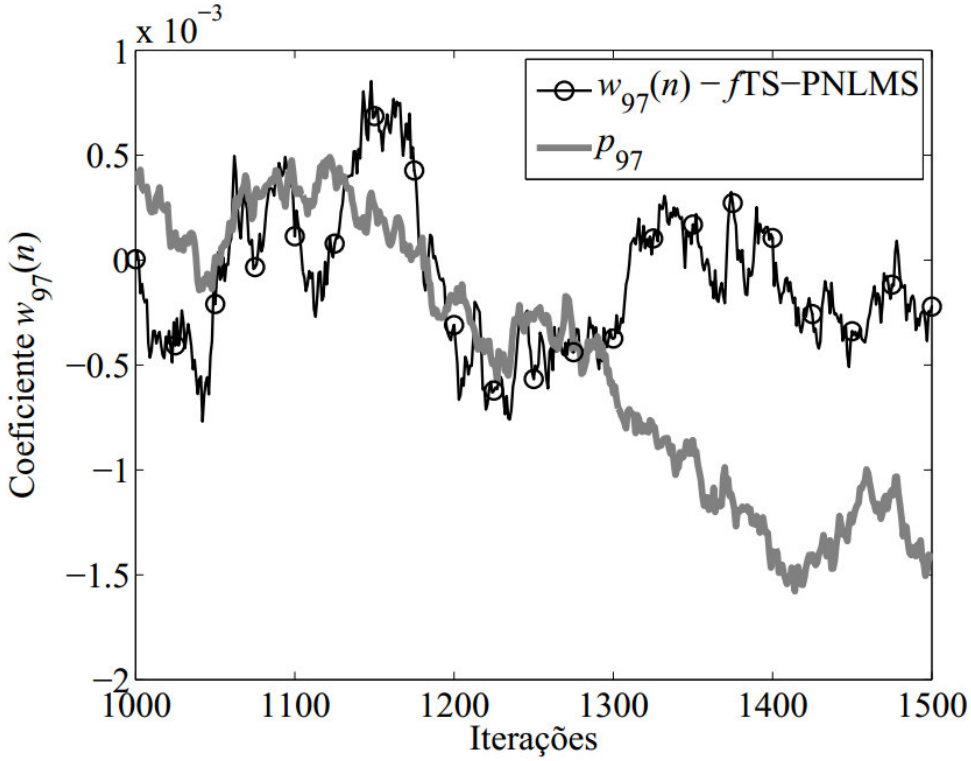
(a)


x 10^−3
1
0
−1
−2
−3
−4
1000
1100
1200
1300
1400
1500
Iterações
Coeficiente w97(n)
w97(n) − IPNLMS
p97

(b)




(c)




(d)

Figura 6.16. Comportamento do coeficiente $w_{97}(n)$, a partir do instante $n = 1000$, para os algoritmos PNLMS, IPNLMS, $f$GS-PNLMS e $f$TS-PNLMS, considerando a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Algoritmo PNLMS (b) Algoritmo IPNLMS. (c) Algoritmo $f$GS-PNLMS. (d) Algoritmo $f$TS-PNLMS.

Note, das curvas das Figuras 6.15 e 6.16 que, apesar da variação aleatória a cada iteração nos valores dos coeficientes da planta de esparsidade variável, $\mathbf{p}(n)$, os algoritmos $f$GS-PNLMS e $f$TS-PNLMS conseguem manter os valores de seus coeficientes, tanto de maior como de menor magnitude, próximos dos valores desejados.

## 6.3 Algoritmo GS-IPNLMS

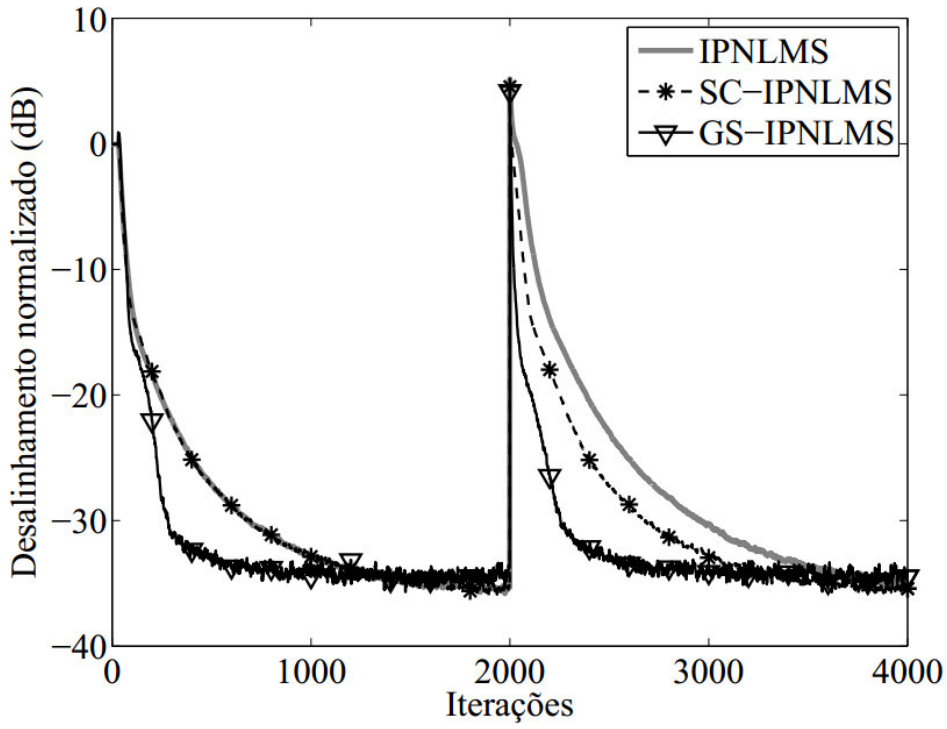
Nesta seção, o desempenho do algoritmo GS-IPNLMS é comparado com o dos algoritmos IPNLMS e SC-IPNLMS, em termos de velocidade de convergência e de resposta a perturbações na planta. Para tal, dois exemplos comparativos são considerados. O primeiro exemplo considera uma inversão nos coeficientes da planta esparsa $\mathbf{p}$. Já o segundo exemplo considera a identificação da planta de esparsidade variável $\mathbf{p}(n)$ dada por (6.1).

### *6.3.1 Exemplo 5*

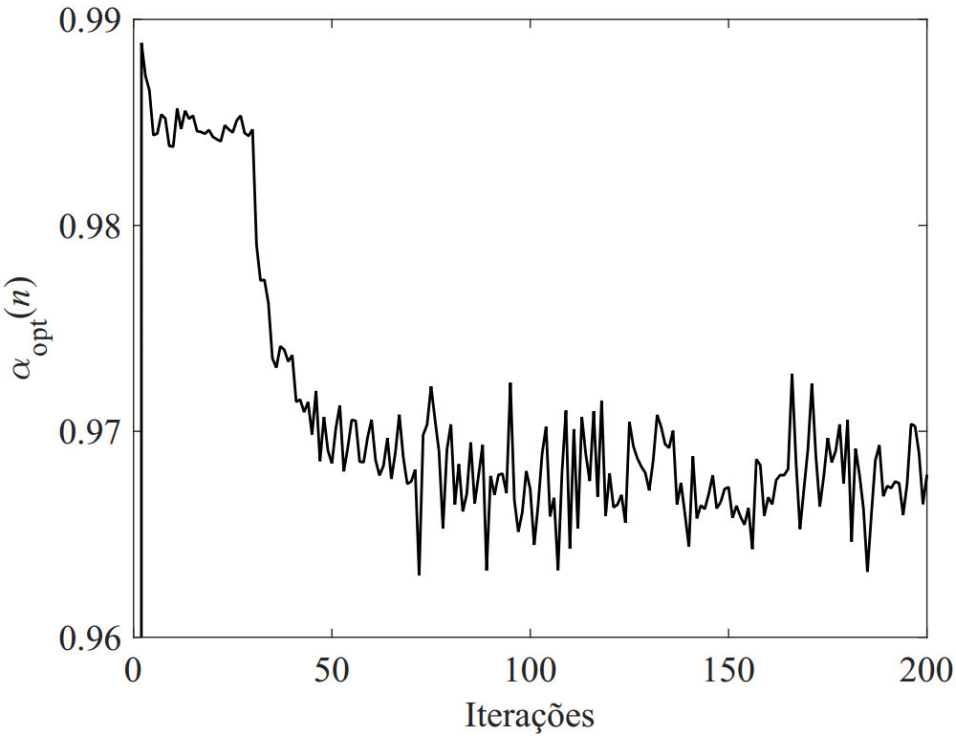
Este exemplo tem como objetivo comparar o desempenho dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS em termos de velocidade de convergência e de resposta a uma perturbação ocorrida na planta esparsa $\mathbf{p}$ no instante $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. Esta inversão não altera o grau de esparsidade da planta.

A Figura 6.17 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, para o algoritmo GS-IPNLMS, e a quantidade de ciclos necessária para o método da razão áurea atingir a convergência em cada iteração do processo de adaptação do algoritmo GS-IPNLMS. Já a Figura 6.18 mostra o comportamento dos coeficientes ativos e inativos para o algoritmo GS-IPNLMS. A Figura 6.19 mostra o comportamento dos coeficientes $w_{30}(n)$, $w_{35}(n)$, $w_{42}(n)$ e $w_{97}(n)$ após ocorrer a perturbação em $\mathbf{p}$, para os algoritmos considerados. Adota-se um parâmetro de passo igual a $\mu = 0{,}5$ para os algoritmos considerados neste exemplo, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para o algoritmo PNLMS, considera-se $\delta = 0{,}01$ e $\rho = 0{,}05$, e para o algoritmo IPNLMS, adota-se $\alpha = 0{,}0$ (Benesty & Gay, 2002) e (Souza, Tobias, Seara & Morgan, 2010). Já para o algoritmo GS-IPNLMS, considera-se $tol = 10^{-3}$.

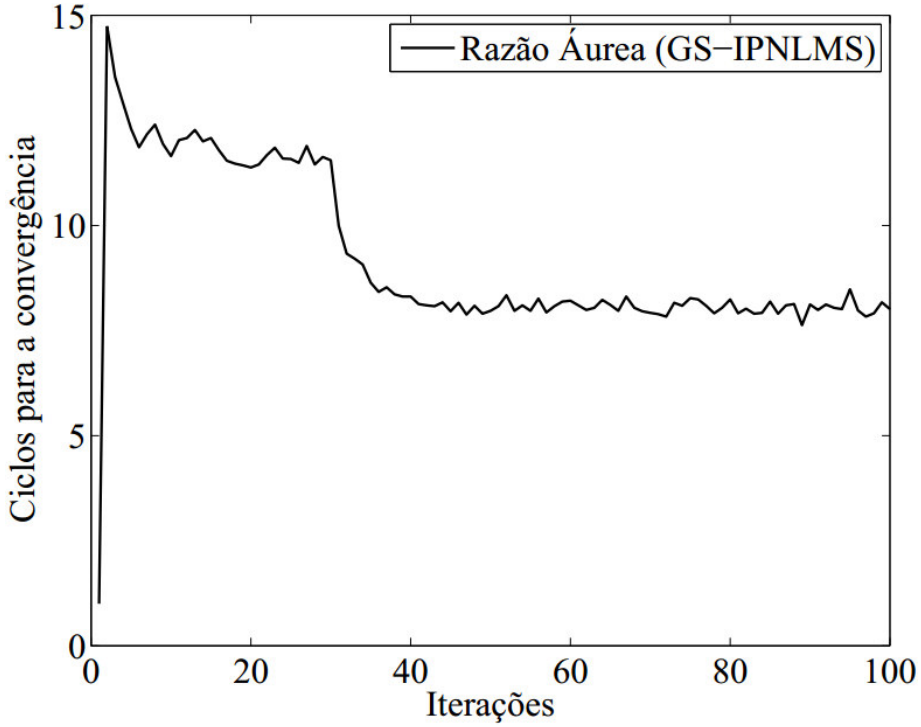
Note das curvas da Figura 6.17(a) que o algoritmo GS-IPNLMS alcança a mais rápida velocidade de convergência entre os algoritmos considerados, mesmo após ocorrer a perturbação em $\mathbf{p}$. Note também, da Figura 6.17(b), que $\alpha_{\text{opt}}(n)$ inicia com o valor $-0{,}230$ no instante $n = 1$, muda para $0{,}984$ em $n = 2$ e, em menos de 100 iterações, estabiliza-se em torno de $0{,}965$, mesmo após a inversão nos coeficientes da planta $\mathbf{p}$. Observe da Figura 6.17(c) que o método da razão áurea converge em apenas 1 ciclo na iteração $n = 1$ e alcança, após $n = 40$, uma média inferior a 9 ciclos para atingir a convergência em cada iteração do processo de adaptação do algoritmo GS-IPNLMS.


IPNLMS
SC−IPNLMS
GS−IPNLMS
Desalinhamento normalizado (dB)
Iterações
10
0
−10
−20
−30
−40
0
1000
2000
3000
4000

(a)

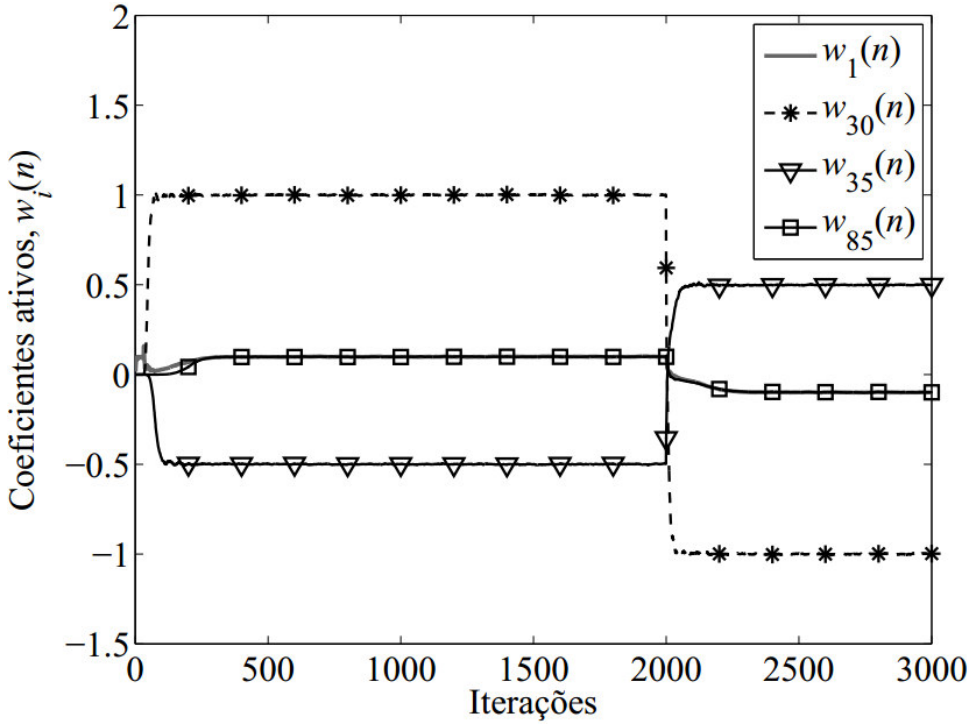

$\alpha_{opt}(n)$
Iterações
0.99
0.98
0.97
0.96
0
50
100
150
200

(b)

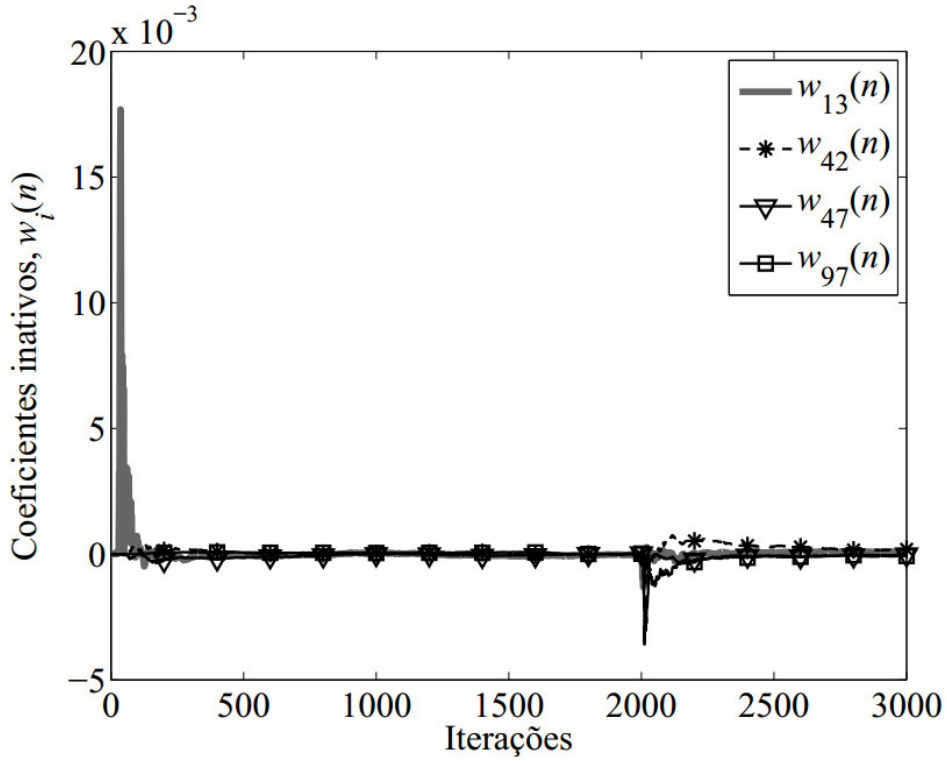



(c)

Figura 6.17. Comportamento dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS, considerando uma inversão nos coeficientes da planta $\mathbf{p}$ em $n = 2000$. (a) Desalinhamento normalizado dos algoritmos IPNLMS (com $\alpha = 0{,}05$), SC-IPNLMS (com $\alpha = 0{,}0$) e GS-IPNLMS (com $tol = 10^{-3}$). (b) Parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, do algoritmo GS-IPNLMS. (c) Quantidade de ciclos para o método da razão áurea atingir a convergência.

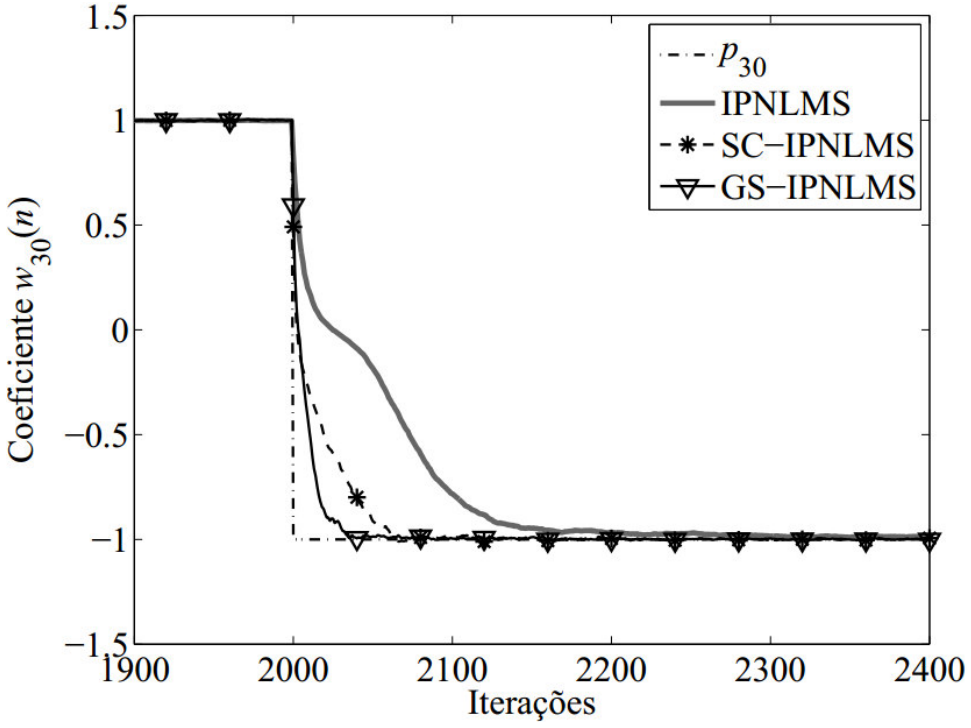



(a)

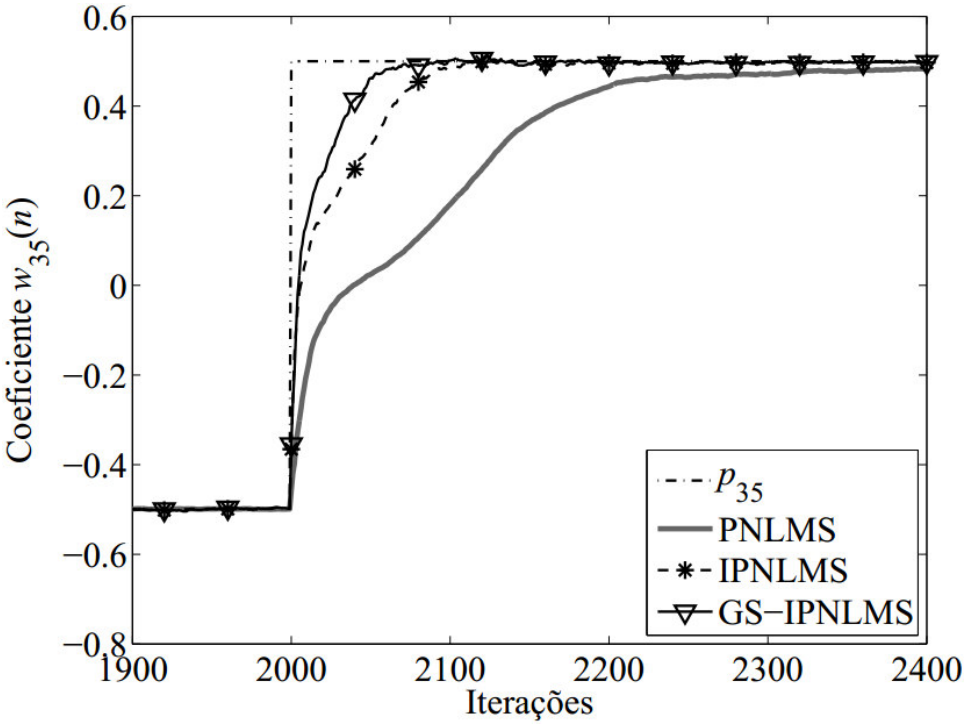



(b)

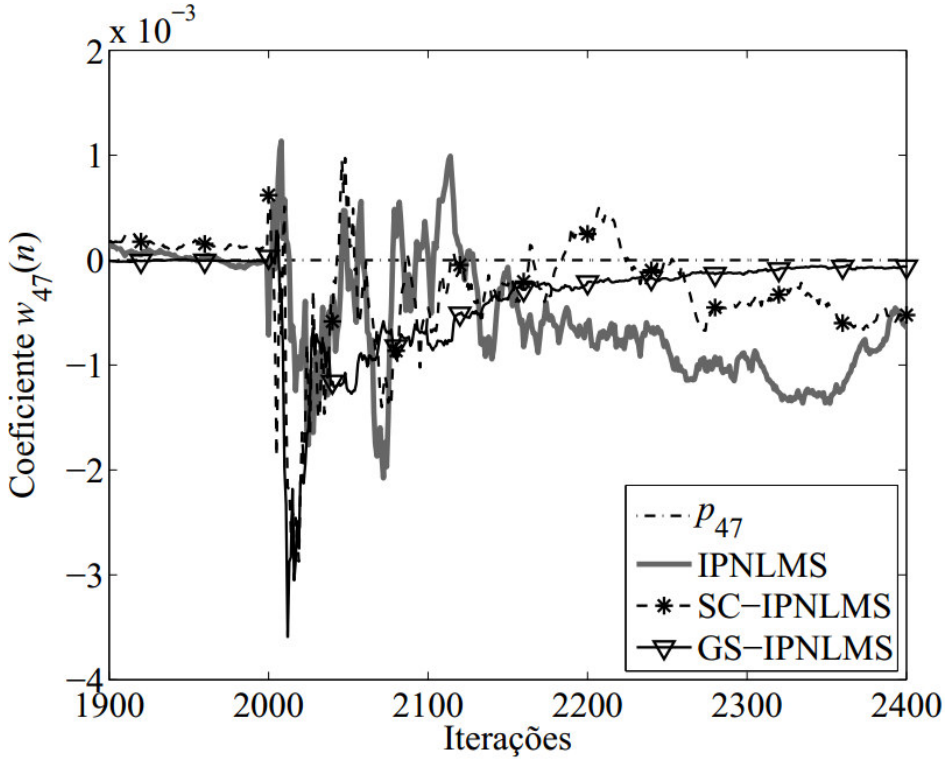
Figura 6.18. Comportamento dos coeficientes do algoritmo GS-IPNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando $\mathbf{p}$ é mudada para $-\mathbf{p}$. (a) Coeficientes ativos. (b) Coeficientes inativos.



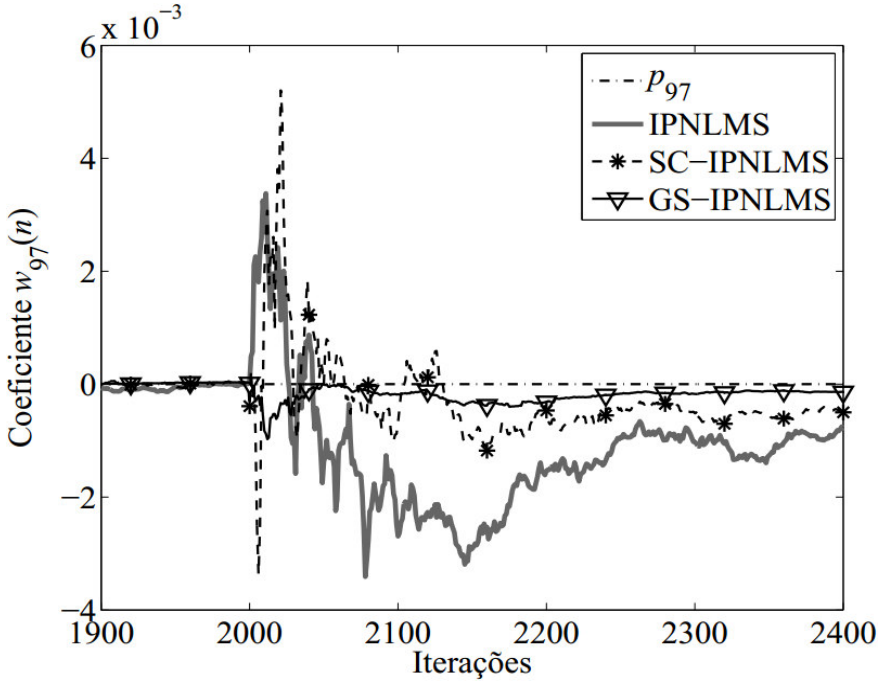

(a)


0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{35}(n)$
$p_{35}$
PNLMS
IPNLMS
GS−IPNLMS

(b)


x 10^−3
2
1
0
−1
−2
−3
−4
1900
2000
2100
2200
2300
2400
Iterações
Coeficiente $w_{47}(n)$
$p_{47}$
IPNLMS
SC−IPNLMS
GS−IPNLMS

(c)




(d)

Figura 6.19. Comportamento dos coeficientes $w_{30}(n)$ $w_{35}(n)$, e $w_{47}(n)$ e $w_{97}(n)$ para os algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS, após a inversão dos coeficientes ativos de $\mathbf{p}$, ocorrida no instante $n = 2000$. (a) Coeficiente $w_{30}(n)$. (b) Coeficiente $w_{35}(n)$. (c) Coeficiente $w_{42}(n)$. (d) Coeficiente $w_{97}(n)$.

Note, das curvas da Figura 6.18, que os coeficientes ativos e inativos do algoritmo GS-IPNLMS convergem rapidamente para os valores de referência da planta $\mathbf{p}$, mesmo após ocorrer a perturbação. Além disso, tem-se que os coeficientes ativos de maior magnitude, $w_{30}(n)$ e $w_{35}(n)$, convergem mais rapidamente do que os coeficientes ativos de menor magnitude, $w_1(n)$ e $w_{85}(n)$. Note também, das curvas da Figura 6.19, que o algoritmo GS-IPNLMS demonstra capacidade de rastreamento de coeficientes ativos e inativos superior à dos algoritmos IPNLMS e SC-IPNLMS, mesmo após a inversão nos coeficientes de $\mathbf{p}$.

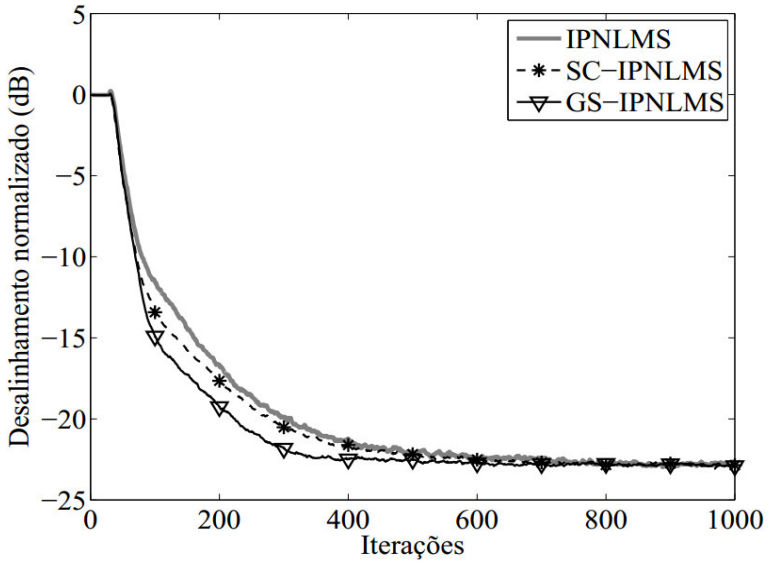
*6.3.2 Exemplo 6*

Neste exemplo a velocidade de convergência dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS é comparada considerando a identificação da planta de esparsidade variável, $\mathbf{p}(n)$, dada por (6.1). Dessa forma, $\mathbf{p}(n)$ é inicializada novamente com os dados da planta de esparsidade $S(\mathbf{p}) = 0{,}9435$ do exemplo anterior, ou seja, $\mathbf{p}(0) = \mathbf{p}$. Além disso, considera-se também $\sigma_v^2 = 1$ e $\lambda = 1 - 10^{-7}$ (Loganathan, Habets & Naylor, 2010).

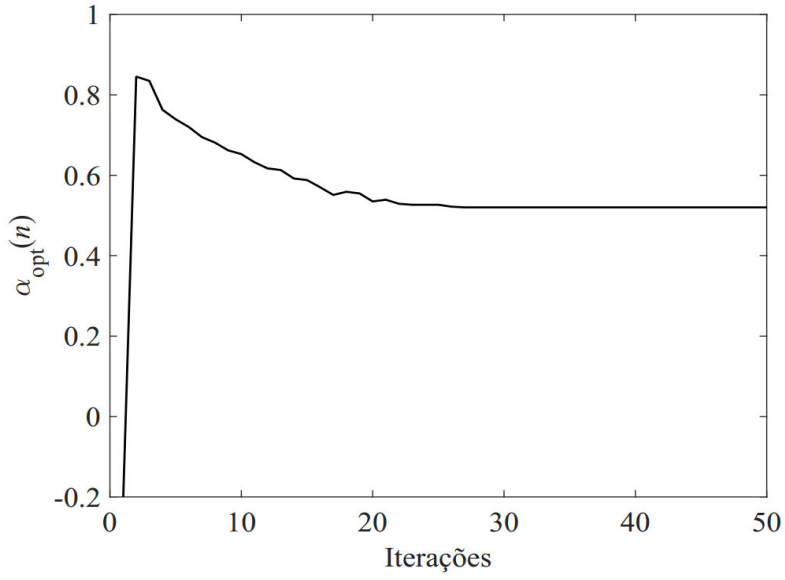
A Figura 6.20 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, para o algoritmo GS-IPNLMS, a variação na esparsidade de $\mathbf{p}(n)$, e a quantidade de ciclos necessária para o método da razão áurea atingir a convergência em cada iteração do processo de adaptação do algoritmo GS-IPNLMS. Já a Figura 6.21 mostra, a partir do instante $n = 500$, o comportamento dos coeficientes $w_{13}(n)$, $w_{35}(n)$, $w_{85}(n)$ e $w_{97}(n)$, para cada um dos algoritmos considerados em relação aos valores correspondentes de $\mathbf{p}(n)$. Os parâmetros de

passo adotados para os algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS são $\mu_{\text{IPN}} = 1{,}5$, $\mu_{\text{SC-IPN}} = 0{,}9$ e $\mu_{\text{GS-IPN}} = 0{,}5$, respectivamente, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para os algoritmos IPNLMS e SC-IPNLMS também adota-se $\alpha_{\text{IPN}} = \alpha_{\text{SC-IPN}} = 0{,}0$, e para o algoritmo GS-IPNLMS considera-se $tol = 10^{-1}$.

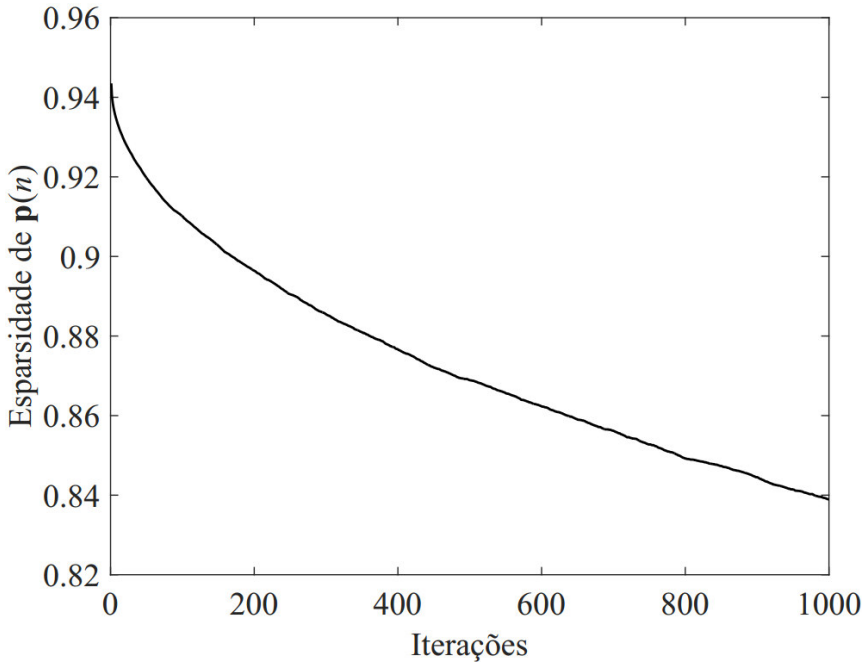
Note das curvas da Figura 6.20(a) que o algoritmo GS-IPNLMS alcança mais uma vez a maior velocidade de convergência, mesmo ocorrendo uma redução gradual na esparsidade da planta $\mathbf{p}(n)$, de $S[\mathbf{p}(n)] = 0{,}9435$ em $n = 0$ para $S[\mathbf{p}(n)] = 0{,}8389$ em $n = 1000$, como mostra a Figura 6.20(c). Note também, da Figura 6.20(b), que $\alpha_{\text{opt}}(n)$ inicia com o valor de $-0{,}215$ em $n = 1$ e, logo na iteração seguinte, muda para $0{,}846$, estabilizando-se em $0{,}520$ após 40 iterações do processo de adaptação do algoritmo GS-IPNLMS. Observe da Figura 6.20(d) que, após $n = 30$, o método da razão áurea necessita de apenas 1 ciclo para atingir a convergência em cada iteração do processo de adaptação do algoritmo GS-IPNLMS.



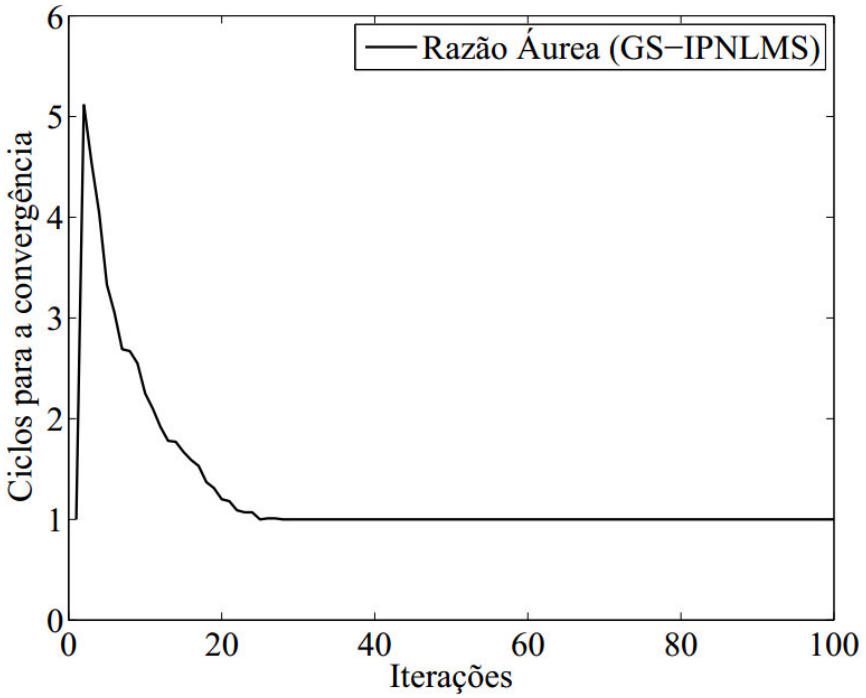

(a)




(b)

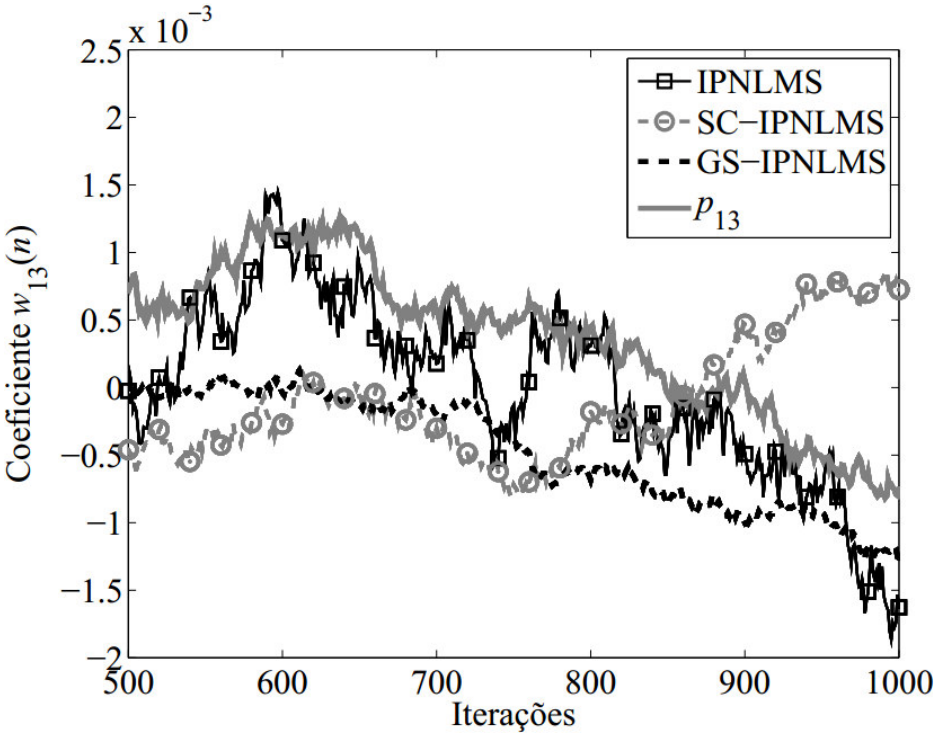



(c)

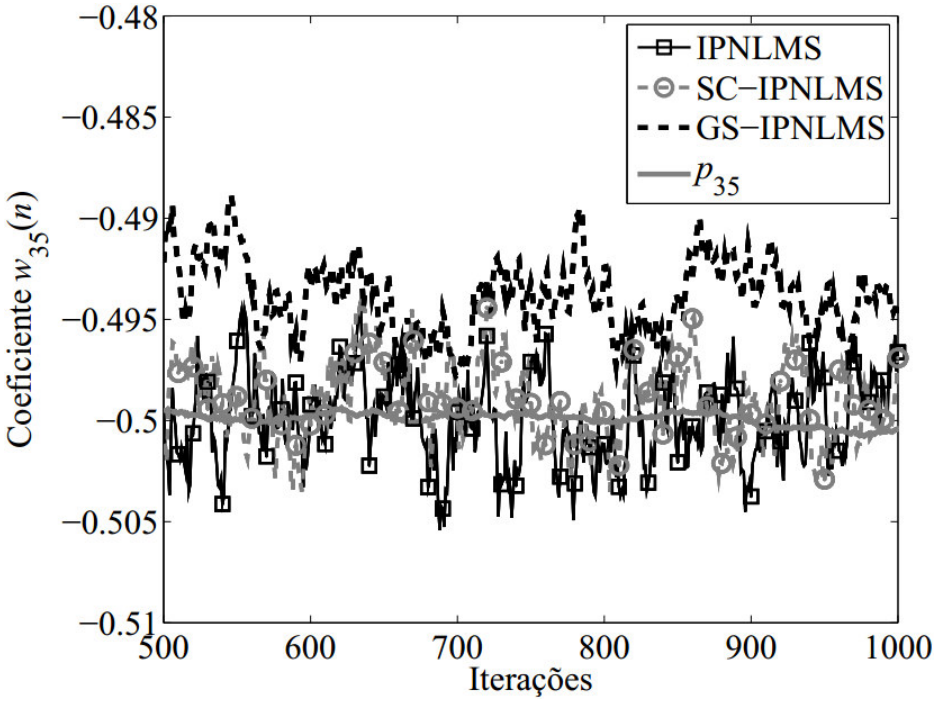



(d)

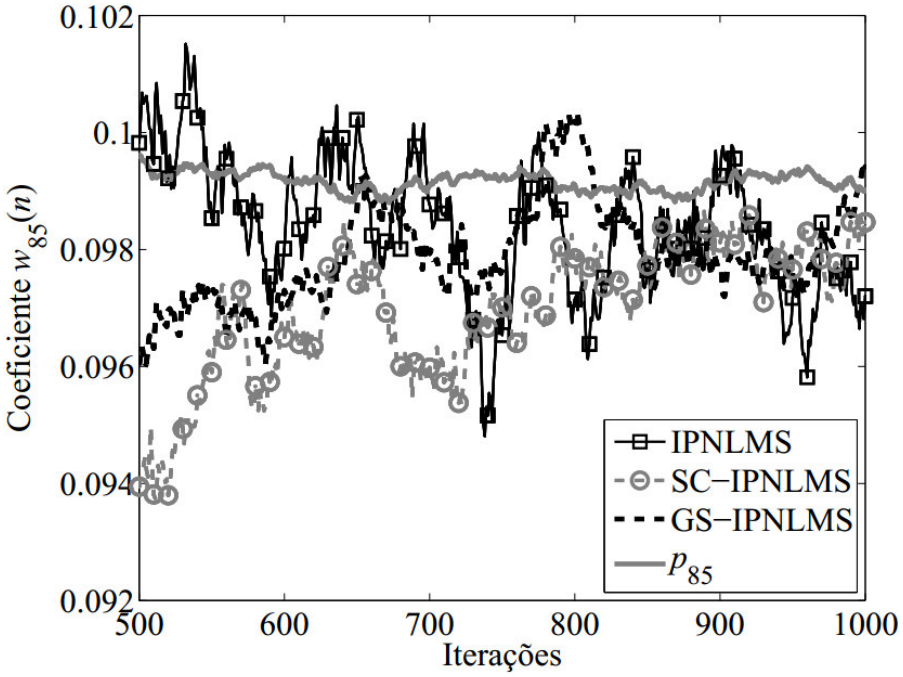
Figura 6.20. Comportamento dos algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS para a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Desalinhamento normalizado dos algoritmos IPNLMS (com $\mu_{\mathrm{IPN}} = 1{,}5$ e $\alpha_{\mathrm{IPN}} = 0{,}0$), SC-IPNLMS (com $\mu_{\mathrm{SC-IPN}} = 0{,}9$ e $\alpha_{\mathrm{SC-IPN}} = 0{,}0$) e GS-IPNLMS (com $tol = 10^{-3}$). (b) Parâmetro de proporcionalidade ótimo, $\alpha_{\mathrm{opt}}(n)$, para o algoritmo GS-IPNLMS. (c) Variação no grau de esparsidade de $\mathbf{p}(n)$. (d) Quantidade de ciclos para o método da razão áurea atingir a convergência.


x 10^−3
2.5
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
500
600
700
800
900
1000
Iterações
Coeficiente $w_{13}(n)$
IPNLMS
SC−IPNLMS
GS−IPNLMS
$p_{13}$

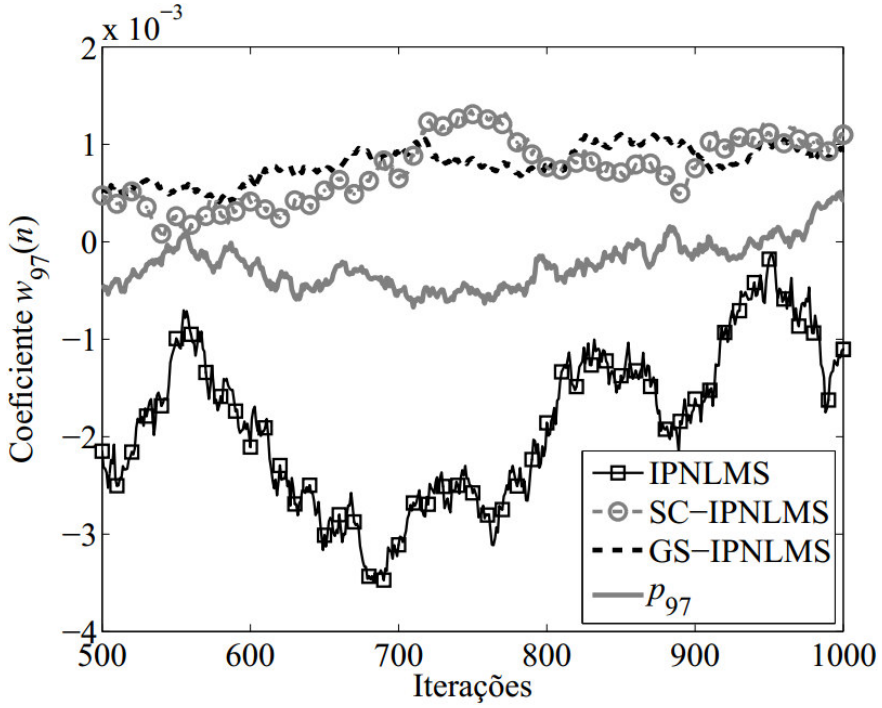
(a)


−0.48
−0.485
−0.49
−0.495
−0.5
−0.505
−0.51
500
600
700
800
900
1000
Iterações
Coeficiente $w_{35}(n)$
IPNLMS
SC−IPNLMS
GS−IPNLMS
$p_{35}$

(b)




(c)




(d)

Figura 6.21. Comportamento dos coeficientes $w_{13}(n)$, $w_{35}(n)$, $w_{85}(n)$ e $w_{97}(n)$, a partir do instante $n = 500$, para os algoritmos IPNLMS, SC-IPNLMS e GS-IPNLMS, considerando a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Coeficiente $w_{13}(n)$. (b) Coeficiente $w_{35}(n)$. (c) Coeficiente $w_{85}(n)$. (d) Coeficiente $w_{97}(n)$.

Note das curvas da Figura 6.21 que, apesar da variação aleatória a cada iteração nos valores dos coeficientes da planta de esparsidade variável, o algoritmo GS-IPNLMS consegue

manter os valores de seus coeficientes, tanto de maior como de menor magnitude, próximos dos valores desejados de $\mathbf{p}(n)$.

## 6.4 TS-IPNLMS

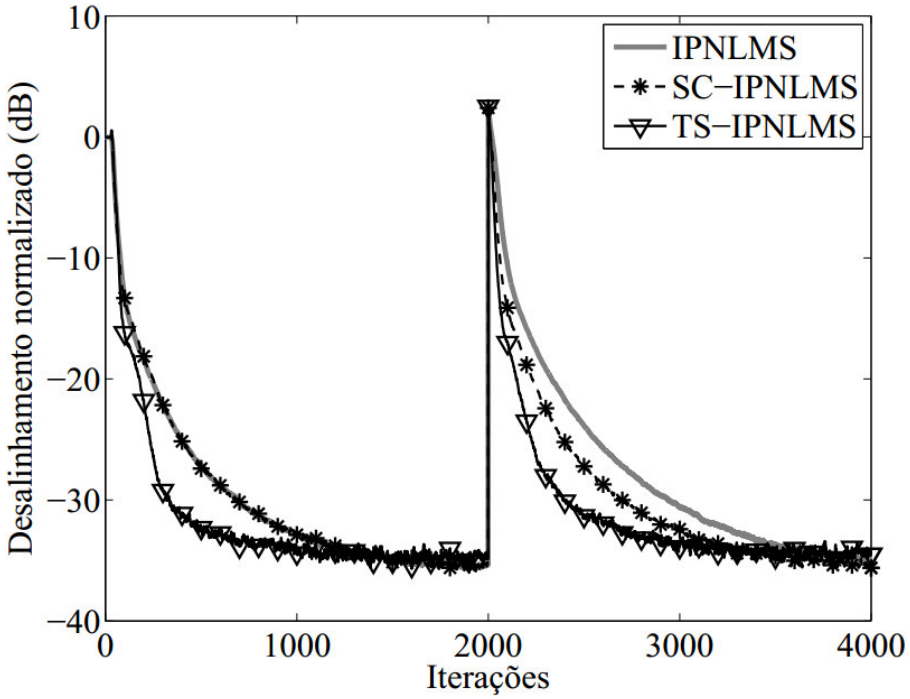
Nesta seção, o desempenho do algoritmo TS-IPNLMS é comparado com o dos algoritmos IPNLMS e SC-IPNLMS em termos de velocidade de convergência. Para tanto, dois exemplos comparativos são mostrados a seguir. O primeiro considera um deslocamento nos coeficientes da planta esparsa $\mathbf{p}$. Já o segundo considera a identificação da planta de esparsidade variável, $\mathbf{p}(n)$, dada por (6.1). Para a execução do método da busca tabu, em todos os exemplos considera-se $D = 2$, $L_\alpha = -2.0$, $U_\alpha = +2.0$, $M = 5$ e PT $= 3$.

### *6.4.1 Exemplo 7*

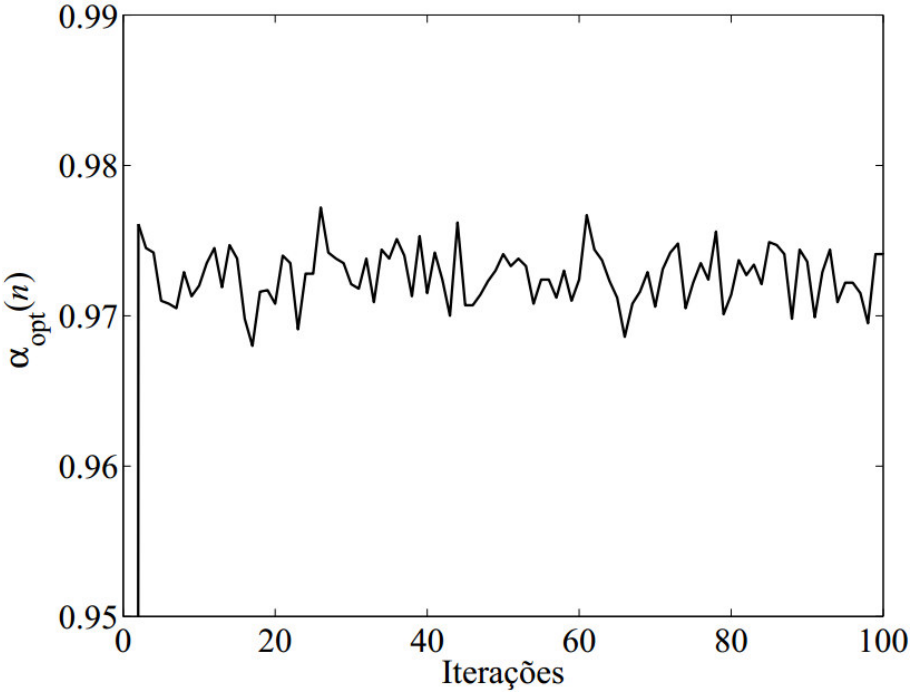
Este exemplo tem o objetivo de comparar o desempenho dos algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS, em termos de velocidade de convergência e resposta a uma perturbação ocorrida no instante $n = 2000$, quando o vetor de coeficientes da planta $\mathbf{p}$ é deslocado de 12 amostras para a direita, alterando as posições dos coeficientes ativos. Dessa forma, os coeficientes ativos de $\mathbf{p}$, cujos valores são iguais a $\{0{,}1\ ,\ 1{,}0\ ,\ -0{,}5\ ,\ 0{,}1\}$, são movidos das posições $\{1, 30, 35, 85\}$ para as posições $\{13, 42, 47, 97\}$, respectivamente.

A Figura 6.22 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, para o algoritmo TS-IPNLMS, e a quantidade de ciclos necessária para o método da busca tabu atingir a convergência em cada iteração do processo de adaptação do algoritmo TS-IPNLMS, respectivamente. Já a Figura 6.23 mostra o comportamento dos coeficientes $w_1(n)$, $w_{13}(n)$, $w_{30}(n)$, $w_{42}(n)$, $w_{35}(n)$, $w_{47}(n)$, $w_{85}(n)$ e $w_{97}(n)$ para o algoritmo TS-IPNLMS. A Figura 6.24 mostra o comportamento dos coeficientes $w_1(n)$, $w_{13}(n)$, e $w_{35}(n)$ e $w_{47}(n)$ para os algoritmos considerados após ocorrer a perturbação em $\mathbf{p}$. Adota-se um parâmetro de passo igual a $\mu = 0{,}5$ para os algoritmos considerados neste exemplo, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para os algoritmos IPNLMS e SC-IPNLMS, também considera-se $\alpha = 0{,}0$ (Benesty & Gay, 2002) e (Souza, Tobias, Seara & Morgan, 2010).

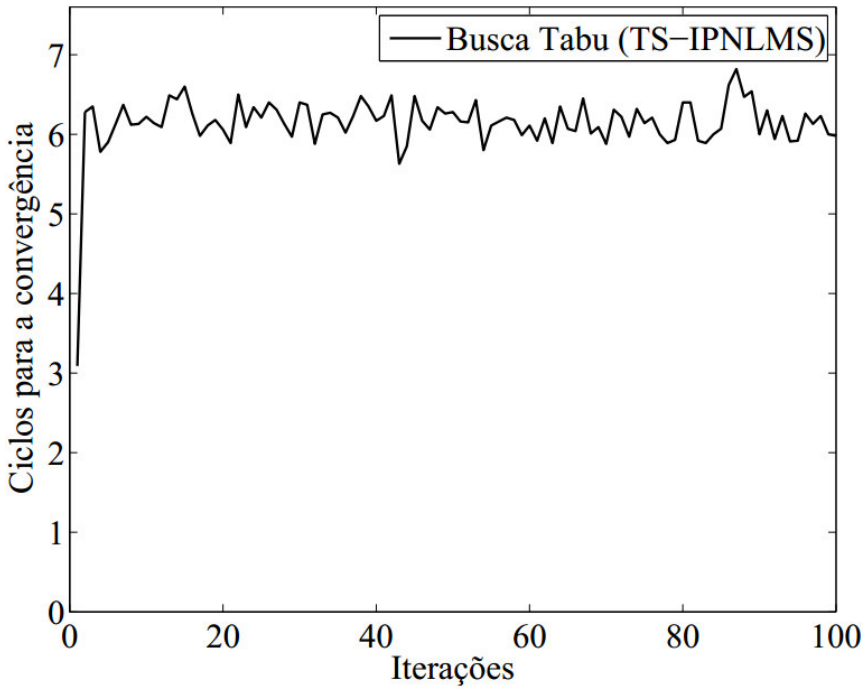
Note das curvas da Figura 6.22(a) que o algoritmo TS-IPNLMS alcança velocidade de convergência superior às dos algoritmos IPNLMS e SC-IPNLMS, mesmo após ocorrer a perturbação na planta. Note também, da Figura 6.22(b), que o $\alpha_{\text{opt}}(n)$ inicia com o valor $-0{,}031$ na primeira iteração ($n = 1$), muda para $0{,}940$ na segunda iteração ($n = 2$) e, em menos de 20 iterações do processo de adaptação do algoritmo TS-IPNLMS, estabiliza-se em torno de $0{,}944$, mesmo após o deslocamento dos coeficientes de $\mathbf{p}$. Observe da Figura 6.22(c) que o método da busca tabu alcança, em menos de 20 iterações, uma média inferior a 7 ciclos para atingir a convergência em cada iteração do processo de adaptação do algoritmo TS-IPNLMS.


IPNLMS
SC−IPNLMS
TS−IPNLMS
Desalinhamento normalizado (dB)
Iterações
10
0
−10
−20
−30
−40
0
1000
2000
3000
4000

(a)

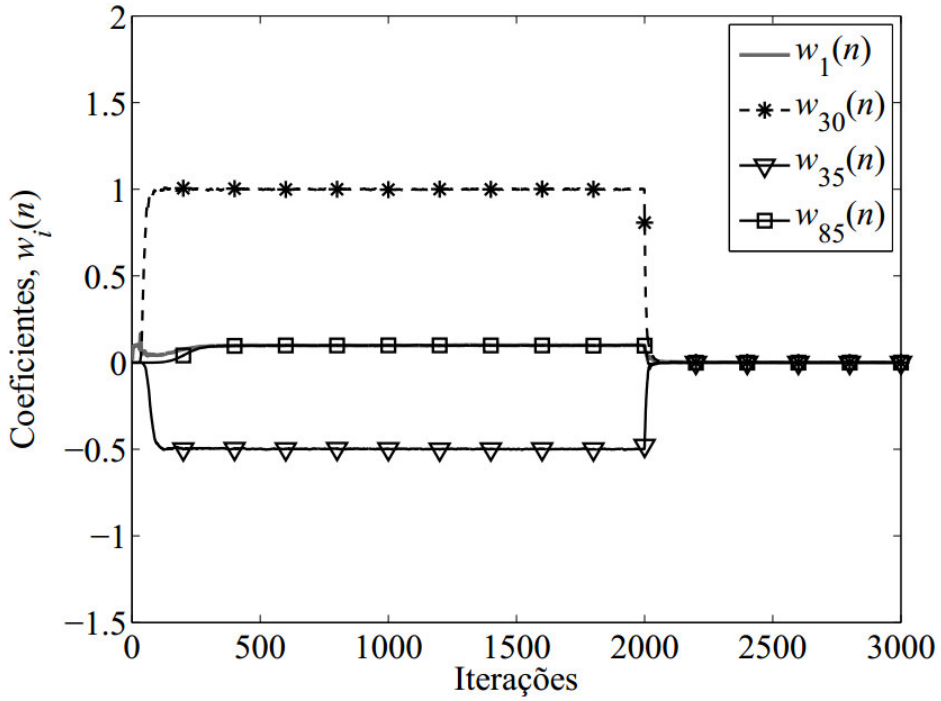

$\alpha_{opt}(n)$
Iterações
0.99
0.98
0.97
0.96
0.95
0
20
40
60
80
100

(b)

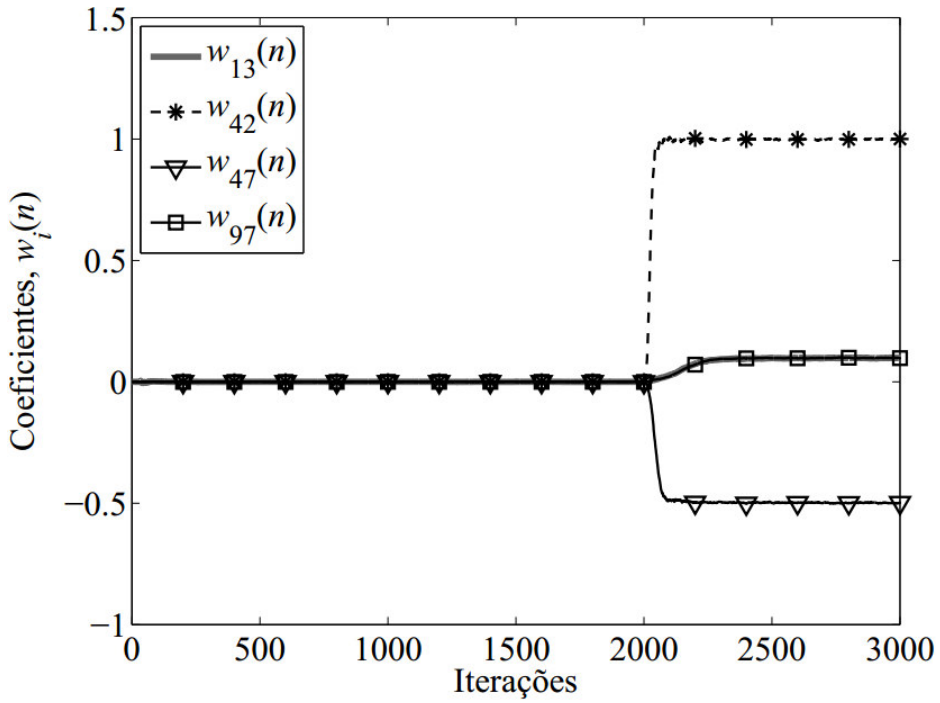



(c)

Figura 6.22. Comportamento dos algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS considerando um deslocamento de 12 amostras à direita dos coeficientes da planta **p** em $n = 2000$. (a) Desalinhamento normalizado dos algoritmos IPNLMS (com $\alpha = 0{,}0$), SC-IPNLMS (com $\alpha = 0{,}0$) e TS-IPNLMS (com $L_\alpha = -2{,}0$, $U_\alpha = +2{,}0$, $M = 5$ e PT $= 3$). (b) Parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, para o algoritmo TS-IPNLMS. (c) Quantidade de ciclos para o método da busca tabu atingir a convergência.

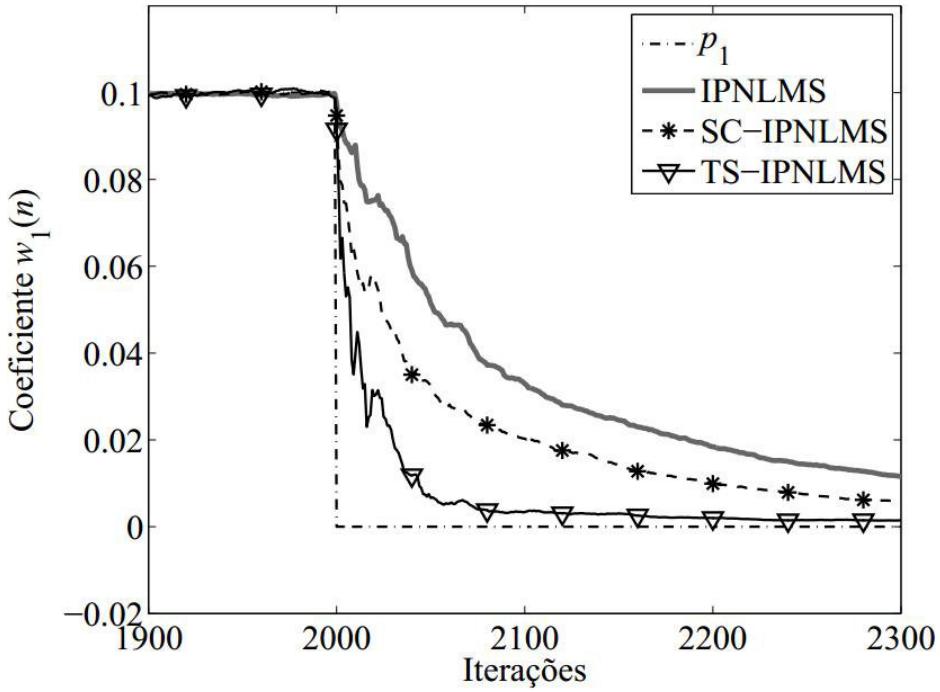



(a)

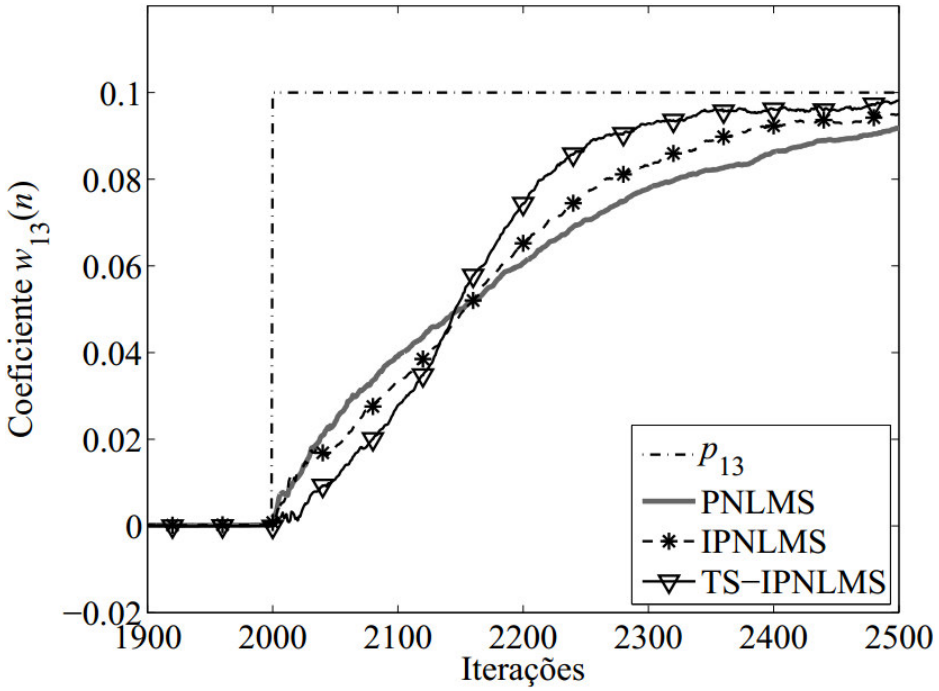



(b)

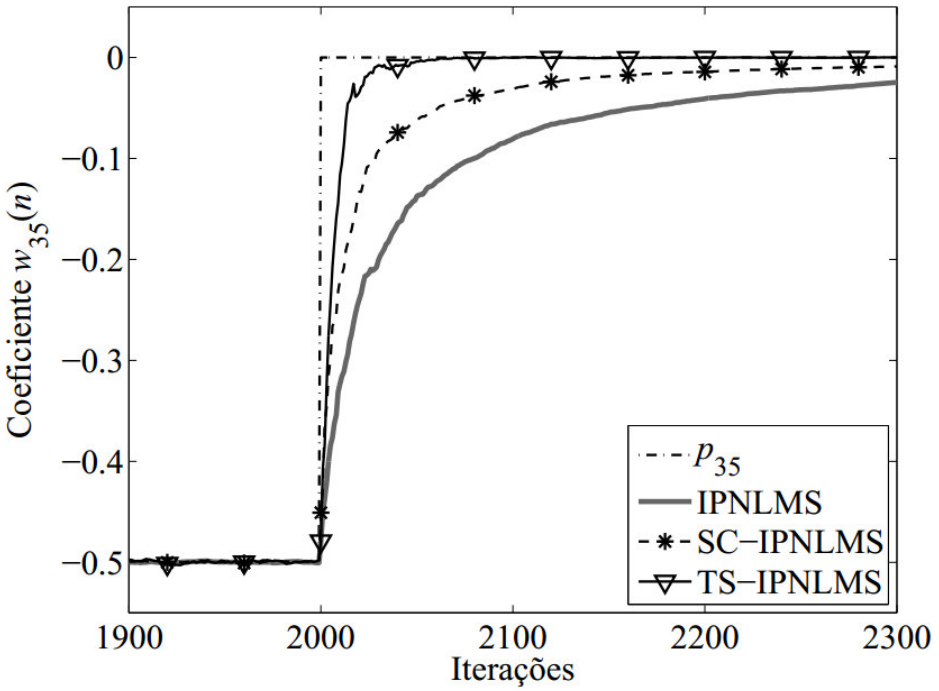
Figura 6.23. Comportamento dos coeficientes do algoritmo TS-IPNLMS considerando uma perturbação na planta esparsa $\mathbf{p}$ em $n = 2000$, quando os coeficientes de $\mathbf{p}$ são deslocados de 12 amostras para a direita. (a) Coeficientes $w_1(n)$, $w_{30}(n)$, $w_{35}(n)$ e $w_{85}(n)$. (b) Coeficientes $w_{13}(n)$, $w_{42}(n)$, $w_{47}(n)$ e $w_{97}(n)$.



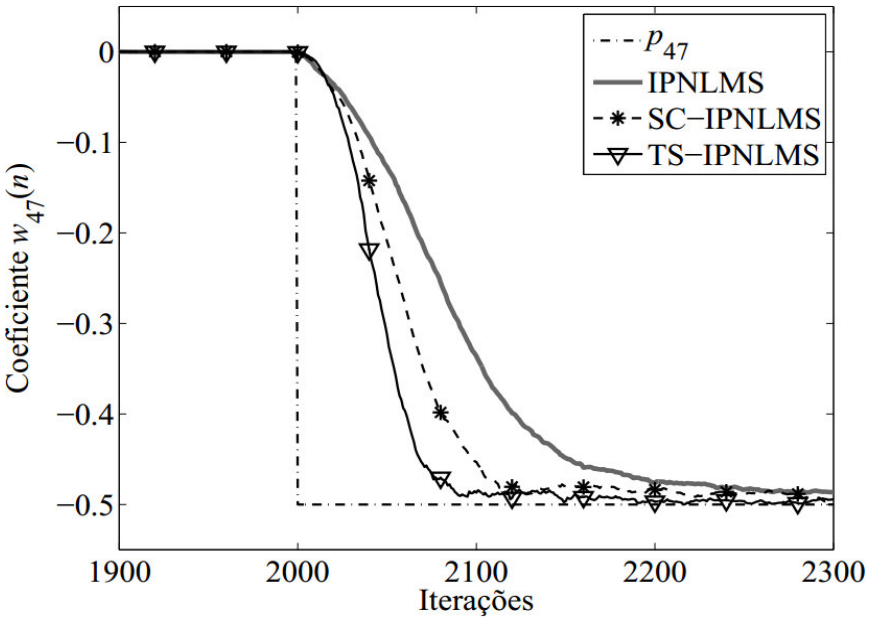

(a)


0.1
0.08
0.06
0.04
0.02
0
−0.02
1900
2000
2100
2200
2300
2400
2500
Iterações
Coeficiente $w_{13}(n)$
$p_{13}$
PNLMS
IPNLMS
TS−IPNLMS

(b)


0
−0.1
−0.2
−0.3
−0.4
−0.5
1900
2000
2100
2200
2300
Iterações
Coeficiente $w_{35}(n)$
$p_{35}$
IPNLMS
SC−IPNLMS
TS−IPNLMS

(c)



(d)

Figura 6.24. Comportamento dos coeficientes $w_1(n)$ $w_{13}(n)$, e $w_{35}(n)$ e $w_{47}(n)$ para os algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS, após o deslocamento de 12 amostras para a direita de $\mathbf{p}$, ocorrido no instante $n = 2000$. (a) Coeficiente $w_1(n)$. (b) Coeficiente $w_{13}(n)$. (c) Coeficiente $w_{35}(n)$. (d) Coeficiente $w_{47}(n)$.

Note, das curvas das Figuras 6.23 e 6.24 que os coeficientes ativos e inativos do algoritmo TS-IPNLMS convergem rapidamente para os valores correspondentes da planta $\mathbf{p}$, mesmo após ocorrer a perturbação. Além disso, tem-se que os coeficientes ativos de maior magnitude como, por exemplo, $w_{35}(n)$ antes de $n = 2000$ e $w_{47}(n)$ após $n = 2000$, convergem mais rapidamente do que os coeficientes ativos de menor magnitude, como $w_1(n)$ antes de $n = 2000$ e $w_{13}(n)$ após $n = 2000$. Note também, das curvas da Figura 6.24, que o algoritmo TS-IPNLMS demonstra capacidade de rastreamento de coeficientes superior à dos algoritmos IPNLMS e SC-IPNLMS, mesmo após ocorrer o deslocamento em $\mathbf{p}$.

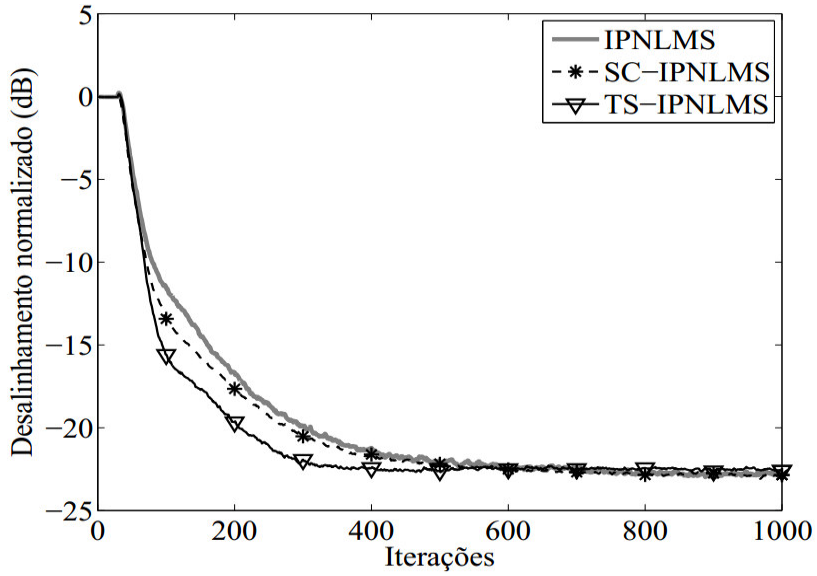
*6.4.2 Exemplo 8*

Neste exemplo, a velocidade de convergência dos algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS é comparada considerando a identificação da planta de esparsidade variável, $\mathbf{p}(n)$, dada por (6.1). Dessa forma, $\mathbf{p}(n)$ é inicializada novamente com os dados da planta de esparsidade $S(\mathbf{p}) = 0{,}9435$ do exemplo anterior, ou seja, $\mathbf{p}(0) = \mathbf{p}$. Além disso, também considera-se $\sigma_v^2 = 1$ e $\lambda = 1 - 10^{-7}$ (Loganathan, Habets & Naylor, 2010).

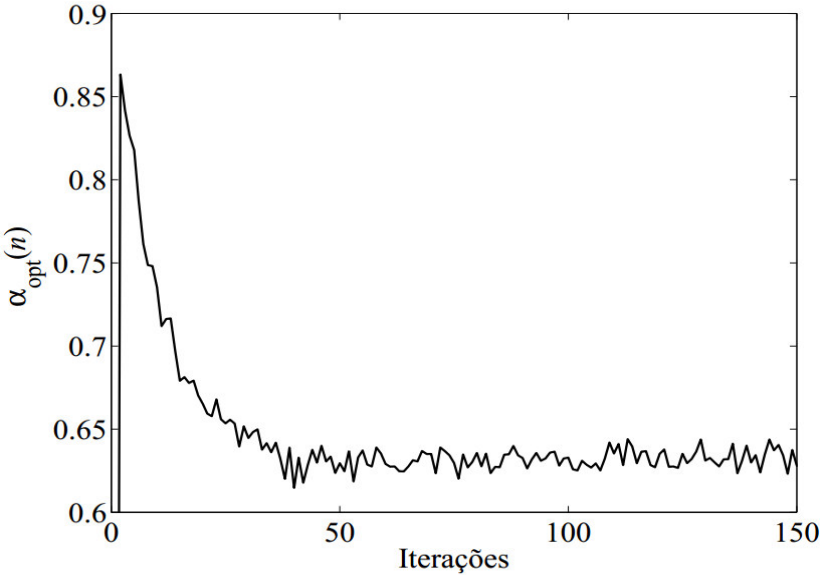
A Figura 6.25 mostra o desalinhamento normalizado em dB dos algoritmos considerados, o comportamento do parâmetro de proporcionalidade ótimo, $\alpha_{\text{opt}}(n)$, para o algoritmo TS-IPNLMS, a variação na esparsidade de $\mathbf{p}(n)$, e a quantidade de ciclos necessária para o método da busca tabu atingir a convergência em cada iteração do processo de adaptação do algoritmo TS-IPNLMS. Já a Figura 6.26 mostra, a partir do instante $n = 500$, o comportamento dos coeficientes $w_{30}(n)$, $w_{42}(n)$, $w_{85}(n)$ e $w_{97}(n)$, para cada um dos algoritmos considerados em relação aos valores correspondentes de $\mathbf{p}(n)$. Os parâmetros de passo adotados para os algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS são $\mu_{\text{IPN}} = 1{,}5$,

$\mu_{\mathrm{SC-IPN}} = 0{,}9$ e $\mu_{\mathrm{TS-IPN}} = 0{,}5$, respectivamente, de forma que todos apresentem o mesmo valor de desalinhamento em regime. Para os algoritmos IPNLMS e SC-IPNLMS também adota-se $\alpha_{\mathrm{IPN}} = \alpha_{\mathrm{SC-IPN}} = 0{,}0$.

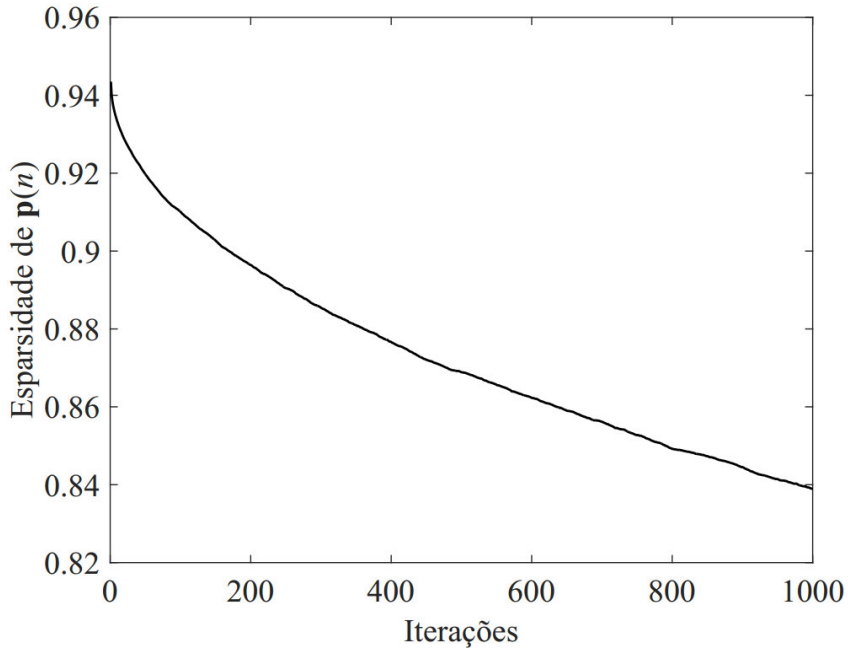
Note das curvas da Figura 6.25(a) que o algoritmo TS-IPNLMS alcança mais uma vez a maior velocidade de convergência, mesmo ocorrendo uma redução gradual na esparsidade da planta $\mathbf{p}(n)$, de $S[\mathbf{p}(n)] = 0{,}9435$ em $n = 0$ para $S[\mathbf{p}(n)] = 0{,}8389$ em $n = 1000$, como mostra a Figura 6.20(c). Note também, da Figura 6.20(b), que $\alpha_{\mathrm{opt}}(n)$ inicia com o valor de $-0{,}215$ em $n = 1$ e, logo na iteração seguinte, muda para $0{,}846$, estabilizando-se em $0{,}520$ após 40 iterações do processo de adaptação do algoritmo TS-IPNLMS. Observe da Figura 6.20(d) que o método da busca tabu alcança novamente, em menos de 20 iterações, uma média inferior a 7 ciclos para atingir a convergência em cada iteração do processo de adaptação do algoritmo TS-IPNLMS.



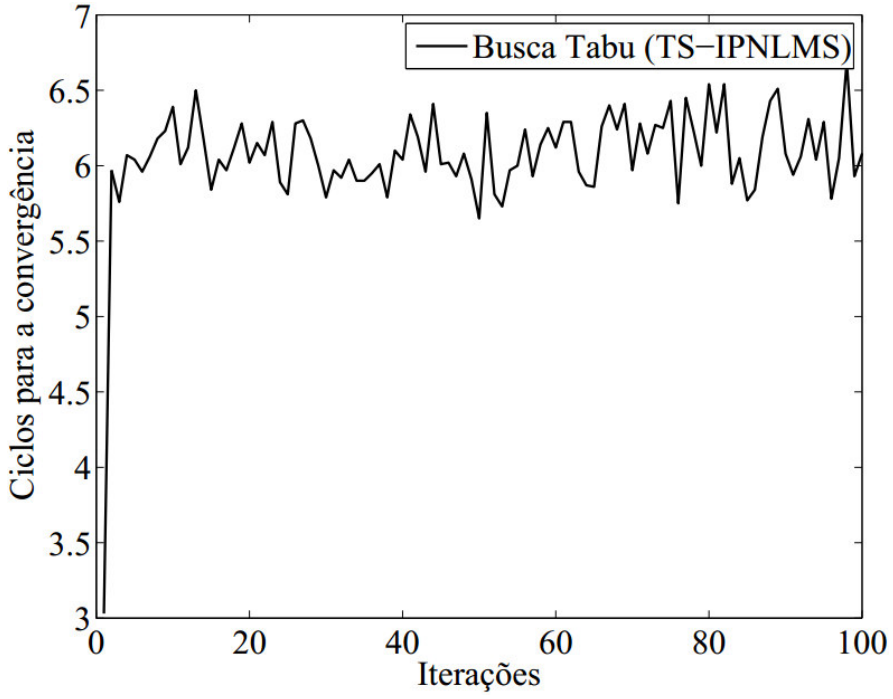

(a)




(b)

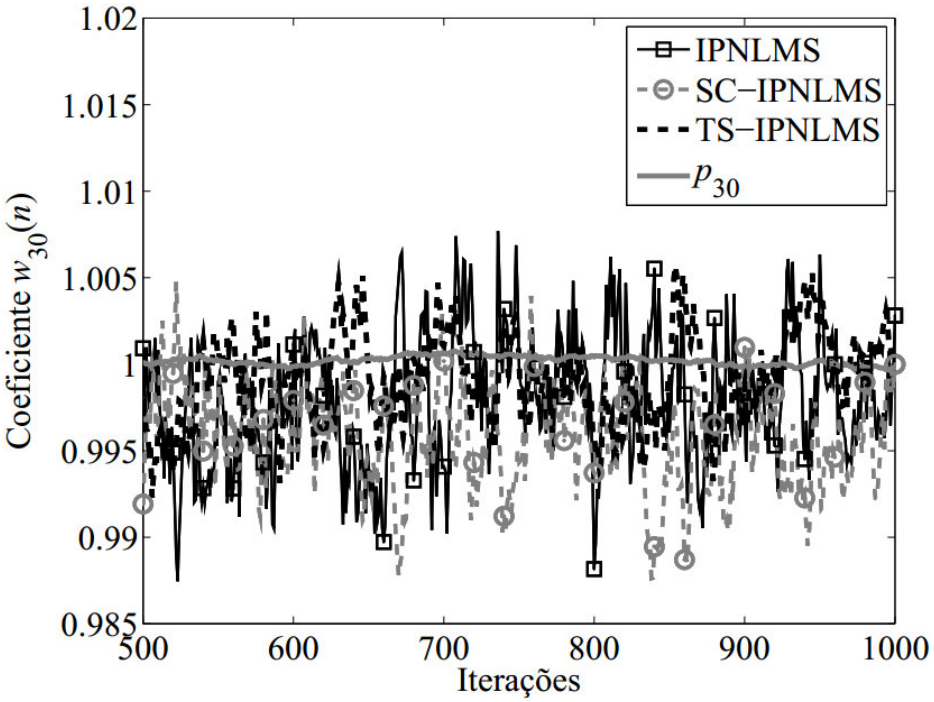



(c)

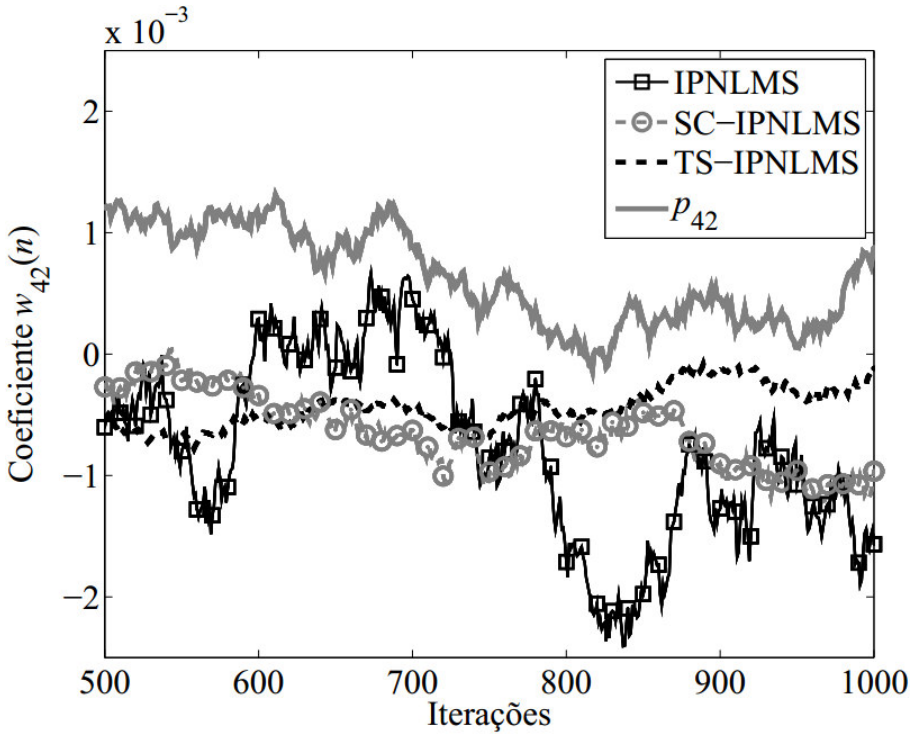



(d)

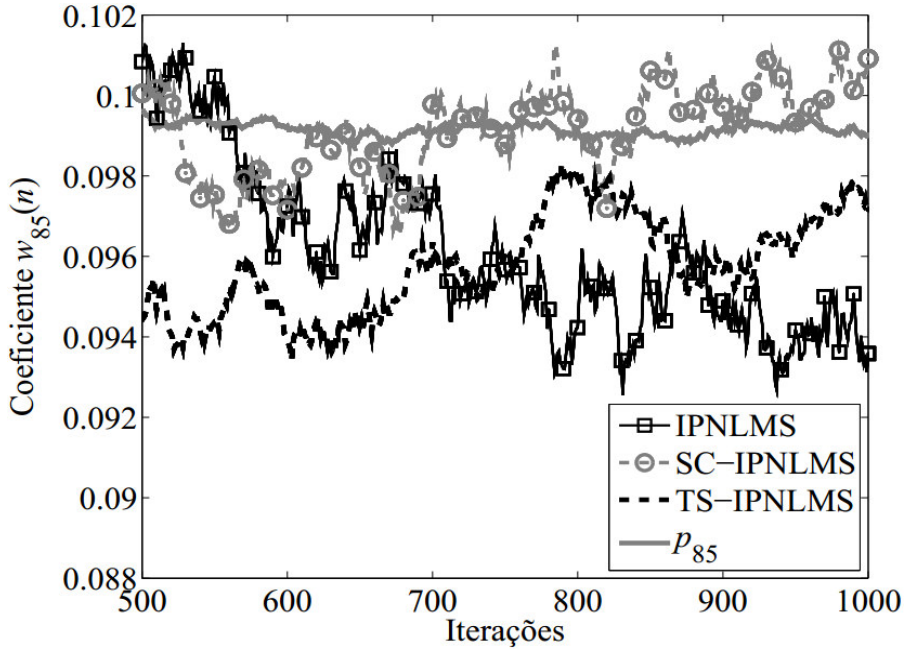
Figura 6.25. Comportamento dos algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS para a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Desalinhamento normalizado dos algoritmos IPNLMS (com $\mu_{\mathrm{IPN}} = 1{,}5$ e $\alpha_{\mathrm{IPN}} = 0{,}0$), SC-IPNLMS (com $\mu_{\mathrm{SC-IPN}} = 0{,}9$ e $\alpha_{\mathrm{SC-IPN}} = 0{,}0$) e TS-IPNLMS (com $\mu_{\mathrm{TS-IPN}} = 0{,}3$ $L_\alpha = -2{,}0$, $U_\alpha = +2{,}0$, $M = 5$ e PT $= 3$). (b) Parâmetro de proporcionalidade ótimo, $\alpha_{\mathrm{opt}}(n)$, para o algoritmo TS-IPNLMS. (c) Variação no grau de esparsidade de $\mathbf{p}(n)$. (d) Quantidade de ciclos para o método da razão áurea atingir a convergência.


1.02
1.015
1.01
1.005
1
0.995
0.99
0.985
500
600
700
800
900
1000
Iterações
Coeficiente $w_{30}(n)$
IPNLMS
SC−IPNLMS
TS−IPNLMS
$p_{30}$

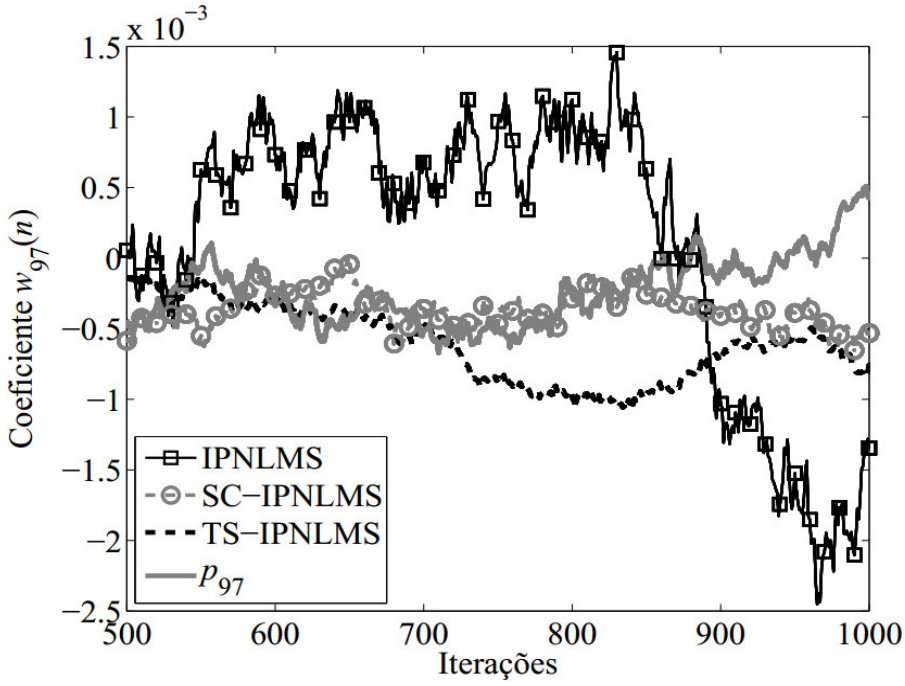
(a)


x $10^{-3}$
2
1
0
−1
−2
500
600
700
800
900
1000
Iterações
Coeficiente $w_{42}(n)$
IPNLMS
SC−IPNLMS
TS−IPNLMS
$p_{42}$

(b)




(c)




(d)

Figura 6.26. Comportamento dos coeficientes $w_{30}(n)$, $w_{42}(n)$, $w_{85}(n)$ e $w_{97}(n)$, a partir do instante $n = 500$, para os algoritmos IPNLMS, SC-IPNLMS e TS-IPNLMS, considerando a identificação de uma planta de esparsidade variável, $\mathbf{p}(n)$. (a) Coeficiente $w_{30}(n)$. (b) Coeficiente $w_{42}(n)$. (c) Coeficiente $w_{85}(n)$. (d) Coeficiente $w_{97}(n)$.

Note das curvas da Figura 6.26 que, apesar da variação aleatória a cada iteração nos valores dos coeficientes da planta de esparsidade variável, o algoritmo TS-IPNLMS consegue manter os valores de seus coeficientes, tanto de maior como de menor magnitude, próximos dos valores correspondentes de $\mathbf{p}(n)$.

# Capítulo 7

# Conclusão

Filtros adaptativos possuem papel fundamental na resolução de vários problemas envolvendo processamento de sinais, tais como identificação de sistemas, controle ativo de ruído, a equalização de canais de comunicação. O estudo destes dispositivos consiste basicamente na análise e desenvolvimento de algoritmos adaptativos, os quais são implementados em estruturas de filtragem.

Diversas aplicações de filtros adaptativos, tais como cancelamento de eco em telecomunicações, sistemas de transmissão de televisão digital e estimação de harmônicas em sistemas elétricos de potência, envolvem plantas cuja resposta impulsiva possui característica esparsa. Porém, algoritmos adaptativos clássicos, como o LMS e o NLMS, usam o mesmo passo de adaptação para todos os coeficientes do filtro e, portanto, não levam em consideração o grau de esparsidade das plantas a serem identificadas. Tais algoritmos apresentam desempenho pobre, em termos de velocidade de convergência, para a identificação de plantas esparsas. Uma alternativa para lidar com este tipo de planta são os algoritmos adaptativos LMS normalizados proporcionais, tais como os algoritmos PNLMS e IPNLMS.

Análises realizadas para avaliar o efeito dos parâmetros dos algoritmos PNLMS e IPNLMS no comportamento dos mesmos indicam que o desempenho destes algoritmos pode ser melhorado, em termos de velocidade de convergência, a partir o uso de metodologias para otimizar a escolha de tais parâmetros. Dessa forma, neste trabalho de pesquisa são propostas novas metodologias para otimizar a escolha dos parâmetros dos algoritmos adaptativos PNLMS e IPNLMS. Para tal, utiliza-se procedimentos baseados em dois métodos de otimização, a saber, os métodos da razão áurea e da busca tabu.

Resultados de simulações numéricas demonstram que as abordagens propostas apresentam desempenho superior, em termos de velocidade de convergência e resposta a perturbações, em relação aos algoritmos PNLMS e IPNLMS originais para a identificação de plantas esparsas. Os resultados obtidos mostram também que os valores escolhidos para os parâmetros dos algoritmos PNLMS e IPNLMS afetam diretamente a velocidade de convergência destes algoritmos durante todo o processo de adaptação.

Como proposta para trabalhos futuros, visando aumentar a velocidade de convergência e o desempenho em regime permanente, sugere-se o desenvolvimento de novos algoritmos da classe PNLMS empregando um parâmetro de passo variável no tempo. Outras alternativas incluem algoritmos PNLMS e IPNLMS com parâmetros de passo ou fatores de ativação individuais para cada coeficiente do filtro adaptativo e escolhidos de forma que a velocidade de convergência e o erro de regime permanente sejam aprimorados. A escolha dos valores ótimos

para os parâmetros dos algoritmos também pode ser baseada em outras funções objetivo que levem em consideração elementos tais como a estimativa do grau de esparsidade da planta ou a memória relacionada com valores do erro de estimação de iterações anteriores do processo de adaptação.

# REFERÊNCIAS


Abdohalli, A., Zhang., P., Xue, H. and Li, S. (2013) Enhanced Subspace-Least Mean Square for Fast and Accurate Power System Measurement. IEEE Transactions on Power Delivery, Vol 28, No 1, pp. 383-393.

Adali, T. and Haykin, S. (2010) Adaptive Signal Processing: Next Generation Solutions. Wiley & Sons, New Jersey, USA.

Albert, A. E. and Gardner Jr., L. A. (1967) Stochastic Approximation and Non-Linear Regression. Cambridge, MA: MIT Press.

Al-Shabilli, A., Taha, B., Elayan, H., Al-Ogaili, F., Alhalabi, L., Weruaga, L. and Jimaa, S. (2015) Sparse NLMS Adaptive Algorithms for Multipath Wireless Channel Estimation. IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). December 2015.

Arenas-García, J. and Figueiras-Vidal, A. (2009) Adaptive Combination of Proportionate Filters for Sparse Echo Cancellation. IEEE Transactions on Audio, Speech and Language Processing, Vol. 17, No. 6, pp. 1087-1098.

Bajwa, W. U., Haupt, J., Sayeed, A. M. and Nowak, R. (2010) Compressed Channel Sensing: A New Approach to Estimating Sparse Multipath Channels (invited paper). Proceedings of IEEE, Vol. 98, No. 6, pp. 1058-1076.

Bazaraa, M. S., Sherali H. D. and Shetty, C. M. (2006). Nonlinear Programming: Theory and Algorithms. 3rd ed. John Wiley & Sons.

Benesty, J. and Gay, S. L. (2002). An improved PNLMS algorithm. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2002, Vol 2, pp. 1881-1884.

Chen, Y., Gu, Y. and Hero III, A. O. (2009). Sparse LMS for System Identification. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2010, Taipei, Taiwan, pp. 3125-3128.

Chong, E. K. and Zak, S. H. (2013) An Introduction to optimization 4th ed. Wiley and Sons.

Clerc, M. (1999) The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. Congress of Evolutionary Computation, Washington, DC, July 1999, pp. 1951-1957.

Das J. C. (2001). Power System Analysis – Short-Circuit Load Flow and Harmonics. Marcel Dekker Inc., New York, USA.

Das, B. K. and Chakraborty, M. (2014) Sparse Adaptive Filtering by an Adaptive Convex Combination of the LMS and the ZA-LMS Algorithms. IEEE Transactions on Circuits and Systems–I: Regular Papers. Vol. 61, No. 5, pp. 1499-1507.

Dash, P. K., Krishnanand, K. R. and Padhee, M. (2001). Fast recursive Gauss–Newton adaptive filter for the estimation of power system frequency and harmonics in a noisy environment. IET Generation, Transmission & Distribution, Vol. 5, Issue 12, pp. 1277–1289.

Deng, H. and Doroslovacky, M. (2005) Improving Convergence of the PNLMS Algorithm for Sparse Impulse Response Identification. IEEE Signal Processing Letters, Vol. 12, No. 3, pp. 181-184.

Deng, H., Doroslovacky, M. (2006). Proportionate Adaptive Algorithms for Network Echo Cancellation. IEEE Transactions on Signal Processing, Vol 54, No 5, pp. 1794-1803.

Diniz, P. S. R. (2012) Adaptive Filtering: Algorithms and Practical Implementation 4th ed. Springer.

Duttweiler, D. (2000). Proportionate normalized least-mean-squares adaptation in echo cancelers. IEEE Transactions on Speech and Audio Processing, Vol 8, No 5, pp.508-518.

Fan, L., He, C., Wang, D. and Jiang, L. (2005) Efficient Robust Adaptive Decision Feedback Equalizer for Large Delay Sparse Channel. IEEE Transactions on Consumer Electronics, Vol. 51, Issue 2, pp. 449-456.

Farhang-Boroujeny, B. (2013). Adaptive Filters: theory and Applications 2nd ed. Wiley & Sons.

Filipski, P. S. and Labaj P. W. (1992). Evaluation of reactive power meters in the presence of high harmonic distortion. IEEE Transactions on Power Delivery, vol. 7, no. 4, pp. 1793–1799.

Fishman, G. S. (1996) Monte Carlo: Concepts, Algorithms and Applications. Springer.

Fogel, D. B. (1994) An Introduction to Simulated Evolutionary Optimization. IEEE Transactions on Neural Networks, Vol. 5, No. 1, pp. 3-14.

Fogel, L. J. (1962) Autonomous automata. Industrial Research, Vol. 4, pp. 14-19.

Fogel, L. J. (1964) On the organization of intellect. Doctoral Dissertation. ULCA.

Gabarda, S. and Cristóbal, G. (2009) Detection of events in seismic time series by time-frequency methods. IET Signal Processing, Vol. 4, Issue 4, pp. 413-420.

Gao, X., Dai, L., Chau, Y. and Wang, Z. (2016) Turbo-Like Beamforming based on Tabu Search Algorithm for Millimeter-Wave Massive MIMO Systems. IEEE Transactions on Vehicular Technology, Vol. 65, No. 7, pp. 5731-5737.

Gay, S. L. (1998). Na efficient, fast converging adaptive filter for network echo cancellation. XXXII Asilomar Conference on Signals, Systems and Computers, Monterey, USA, Vol 1, pp. 394-398.

Glover, F. (1989). Tabu Search Part I. ORSA Journal on Computing, Vol 1, No 3, pp. 190-206

Glover, F. (1990). Tabu Search Part II. ORSA Journal on Computing, Vol 2, No 1, pp. 4-32.

Glover, F. and Laguna, M. (1997) Tabu Search. Springer, USA.

Goodwin, G. C. and Sin, K. S. (1984) Adaptive Filtering, Prediction and Control. Prentice-Hall, New Jersey, USA.

Gu, Y., Jin, J. and Mei, S. (2009) $l_0$ norm constraint LMS algorithm for sparse system estimation. IEEE Signal Processing Letters, Vol. 16, No. 9, pp. 774-777.

Haykin, S. (2002) Adaptive Filter Theory 4th ed. Prentice Hall, New Jersey, EUA.

Holland, J. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press.

Homer, J., Mareels, I. and Hoang, C. (2006) Enhanced Detection Guided NLMS Estimation of Sparse FIR-Modeled Signal Channels. IEEE transactions on Circuits and Systems–I: Regular Papers, Vol. 53, No. 8, pp. 1783-1791.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research, vol. 5, pp. 1457–1469.

Huang, Y., Benesty, J. and Chen, J. (2006) Acoustic MIMO Signal Processing. Springer, New York, EUA.

Jiang, S. and Gu, Y. (2015) Block-sparsity-inducted adaptive filter for multiclustering system identification. IEEE Transactions on Signal Processing, Vol. 63, No. 20, pp. 5318-5330.

Junior, B. R. P., Cossi, A. M., Contreras, J. and Mantovani, J. R. S. (2013). Multiobjective multistage distribution system planning using tabu search. IET Generation, Transmission & Distribution, Vol 8, Issue 1, pp. 35-45.

Kelly, J. L. and Logan, B. F. (1970) Self-adaptive echo canceler. U.S. Patent 3 500 000.

Kennedy, J. and Eberhart, R. C. (1995) Particle swarm optimization. IEEE International Conference on Neural Networks, Vol. 4, pp. 1942-1948.

Khong, W. H. and Naylor, P. A. (2006) Efficient Use of Sparse Adaptive Filters. Asilomar Conference of Signals and Systems Computation, October 2006, Pacific Grove, CA, pp. 1375-1379.

Kuslevic, M. D., Tomic, J. J. and Marcetic, D. P. (2009). Active Power Measurement Algorithm for Power System Signals Under Non-Sinusoidal Conditions and Wide-Range Frequency Deviations. IET Generation, Transmition & Distribution, Vol. 3, No. 1, pp. 57–65.

Kusljevic, M. D. and Poljak, P. D. (2011). Simultaneous Reactive-Power and Frequency Estimations Using Simple Recursive WLS Algorithm and Adaptive Filtering. IEEE Transactions on Instrumentation and Measurement, Vol. 60, No. 12.

Laguna, M. (1994) A Guide to Implementing Tabu Search. Investigación Operativa, Vol. 4, No. 1, pp. 5-25.

Li, Y., Gu, Y. and Tang, K. (2006). Parallel NLMS Filters with Stocastic Active Taps and Step-Sizes for Sparse System Identification. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2006, Tolouse, France, Vol 3, pp. 109-112.

Loganatham, P., Khong, A. W. H. and Naylor, P. A. (2009) A Class of Sparseness-Controlled Algorithms for Echo Cancellation. IEEE Transactions on Audio, Speech and Language Processing, Vol. 17, No. 8, pp. 1591-1601.

Loganathan, P., Habets, E. A. P. and Naylor, P. A. (2010). Performance Analysis of IPNLMS for Identification of Time-Varying Systems. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, Texas, USA, March 2010, pp. 317-320.

Loganathan, P., Khong, A. W. H. and Naylor, P. A. (2008). A Sparesness Controlled Proportionate Algorithm for Acoustic Echo Cancellation. European Signal Processing Conference (EUSIPCO) August 2008, Lausanne, Switzerland.

Luenberger, D. G. and Ye Y. (2008) Linear and Nonlinear Programming. 3rd ed. Springer.

Machowski, J., Bialek, J. W. and Bumby, J. R. (2008). Power System Dynamics – Stability and Control 2nd ed. Wiley & Sons.

Manolakis, D.G., Ingle, V. K. and Kogon, S. M. (2005) Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing. Artech House.

Martin, R. K., Seathers, W. A., Williamson, R. C. and Johnson, C. R. (2002) Exploiting Sparsity in Adaptive Filters. IEEE Transactions on Signal Processing, Vol. 50, No. 8, pp. 1883-1894.

Morgan, D. R. (1980) An analysis of multiple correlation cancellation loops with a filter in the auxiliary path. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-28, pp. 454-467.

Morgan, D. R. (2013) History, Applications, and Subsequent Development of the FXLMS Algorithm [DSP History]. IEEE Signal Processing Magazine, Vol. 30, Issue 3, pp. 172-176.

Nagumo, J. and Noda, A. (1967) A learning method for system identification. IEEE Transactions on Automatic Control, Vol. 4, No. 3, pp. 282-287.

Nascimento, V. H. and Silva, M. T. M. (2014). Chapter 12 – Adaptive Filters. Academic Press Library in Signal Processing, Vol 1, pp. 619-671.

Paleologu, C., Benesty, J. and Ciochina, S. (2010) Sparse Adaptive Filters for Echo Cancellation. Morgan & Claypool.

Patra, J. P. and Dash, P. K. (2012). Fast Frequency and Harmonic Estimation in Power Systems Using a new Optimized Adaptive Filter. Springer-Velag Electrical Engineering 95, pp. 171-184.

Pelekanakis, K. and Chitre, M. (2013) New Sparse Adaptive Algorithms Based on the Natural Gradient and the $L_0$-Norm. IEEE Journal of Oceanic Engineering, Vol. 38, No. 02, pp. 323-332.

Pelekanakis, K. and Chitre, M. (2013) new Sparse Adaptive Algorithms Based on the Natural Gradient and the $L_0$-Norm. IEEE Journal of Oceanic Engineering, Vol. 38, No. 2, pp. 323-332.

Phadke, A., Thorp, J. and Adamiak, M. (1983). A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency. IEEE Transactions on Power Application Systems, vol. PAS-102, no. 5, pp. 1025–1038.

Pham, D. T. and Karaboga, D. (2000) Intelligent Optimization Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks. Springer.

Rechenberg, I. (1973) Evolutionsstrategie: Optimerung technischer systeme nach prinzipien der biolgischen evolution. Stuttgart: Frommann-Holzboog Verlag.

Reynolds, C. W. (1987) Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Conference, pp. 25-34.

Routray, A., Pradhan, A. K. and Rao K. P. (2002). A novel Kalman filter for frequency estimation of distorted signals in power system. IEEE Transactions on Instrumenation Measurement, Vol. 51, pp. 469–479.

Rubstein, R. Y. (2008) Simulation and the Monte Carlo Method 2nd ed. Wiley.

Sadiku, M. N. O. (2009). Monte Carlo Methods for Electromagnetics. CRC Press.

Sayed, H. A. (2003). Fundamentals of Adaptive Filtering. Wiley & Sons.

Schilling, R. J. and Harris, S. L. (2012) Fundamentals of Digital Signal Processing Using MATLAB® 2nd ed. Cengage Learning, Stamford, USA.

Schwefel, H.-P. (1965) Kybernetische evolution als strategie der experimentellen forchung in der strmungstechnik. Diploma thesis, Thecnical University of Berlim.

Shi, K. and Shi, P. (2010) Convergence analysis of sparse LMS algorithms with $l_1$-norm penality based on white input signal. Signal Processing, Vol. 90, No. 12, pp. 3289-3293.

Sondhi, M. M. (1967) An adaptive adaptive echo canceler. Bell Systems and Technology Journal, Vol. 46, No. 3, pp. 497-511.

Sondhi, M. M. and Presti, A. J. (1966) A self-adaptive echo canceler. Bell Systems and Technology Journal, Vol. 45, No. 12, pp. 1851-1854.

Souza, F. C. (2012). Algoritmos Adaptativos LMS Normalizados Proporcionais: Proposta de um Novo Algoritmo e sua Modelagem Estocástica. Tese: Doutorado. Universidade Federal de Santa Catarina (UFSC).

Souza, F. C., Tobias, O. J., Seara, R. and Morgan, D. R. (2010). A PNLMS Algorithm With Individual Activation Factors. IEEE Transactions on Signal Processing, Vol 58, No 4, pp. 2036-2047.

Starck, J.-L., Murtagh, F. and Fadili, J. M. (2010) Sparse Image and Signal Processing – Wavelets, Curvelets, Morphological Diversity. Cambridge University Press, New York, USA.

Su, G., Jin, J. and Gu, Y. (2012) Performance Analysis of $l_0$ Norm Constraint Least Mean Square Algorithm. IEEE Transactions on Signal Processing, Vol. 60, No. 5, pp. 2223-2235.

Sun, W. and Yuan, Y-X. (2006) Optimization Theory and Methods: Nonlinear Programming. Springer.

Therrien, C. W. (1992). Discrete Random Signals and Statistical Signal Processing. Englewood Cliffs, NJ: Prentice-Hall.

Wagner, K. and Doroslovacky, M. (2011) Proportionate-type Normalized Least Mean Square Algorithms With Gain Allocation Motivated by Mean-Square-Error Minimization for White Input. IEEE Transactions on Signal Processing, Vol. 59, No. 5, pp. 2410-2415.

Wagner, K. and Doroslovacky, M. (2013) Proportionate-type Normalized Least Mean Square Algorithms. ISTE and Wiley & Sons.

Widrow B. and Walach, E. (2007) Adaptive Inverse Control: A Signal Processing Approach. IEEE Press.

Widrow, B. and Stearns, P. N. (1985) Adaptive Signal Processing. Prentice Hall, New Jersey, EUA.

Zhao, H. and Zhang, J. (2009). Aptively Combined FIR and Functional Link Artificial Neural Network Equalizer for Nonlinear Communication Channel. IEEE Transactions on Neural Networks, Vol 20, No 4, pp. 665-674.

Zuo, X., Murray, C. C. and Smith, A. E. (2014). Solving an Extended Double Row Layout Problem Using Multiobjective Tabu Search and Linear Programing. IEEE Transactions on Automation Science and Engineering, Vol. 11, No 4, pp. 1122-1132.